Lev Manovich

SOFTWARE TAKES COMMAND

THIS VERSION:
November 20, 2008.
Please note that this version has not been proofread yet, and it is also missing illustrations.
Length: 82,071 Words (including footnotes).

CREATIVE COMMONS LICENSE:

ABOUT THE VERSIONS:
One of the advantages of online distribution which I can control is that I don't have to permanently fix the book's contents. Like contemporary software and web services, the book can change as often as I like, with new "features" and "big fixes" added periodically. I plan to take advantage of these possibilities. From time to time, I will be adding new material and making changes and corrections to the text.

LATEST VERSION:
Check www.softwarestudies.com/softbook for the latest version of the book.

SUGGESTIONS, CORRECTIONS AND COMMENTS:
send to manovich@ucsd.edu
with the word "softbook" in the email header.

Introduction: Software Studies for Beginners

Software, or the Engine of Contemporary Societies

In the beginning of the 1990s, the most famous global brands were the companies that were in the business of producing materials goods or processing physical matter. Today, however, the lists of best-recognized global brands are topped with the names such as Google, Yahoo, and Microsoft. (In fact, Google was number one in the world in 2007 in terms of brand recognition.) And, at least in the U.S., the most widely read newspapers and magazines - New York Times, USA Today, Business Week, etc. - daily feature news and stories about YouTube, MySpace, Facebook, Apple, Google, and other IT companies.

What about other media? If you access CNN web site and navigate to the business section, you will see a market data for just ten companies and indexes displayed right on the home page.[1] Although the list changes daily, it is always likely to include some of the same IT brands. Lets take January 21, 2008 as an example. On that day CNN list consisted from the following companies and indexes: Google, Apple, S&P 500 Index, Nasdaq Composite Index, Dow Jones Industrial Average, Cisco Systems, General Electric, General Motors, Ford, Intel.[2]

This list is very telling. The companies that deal with physical goods and energy appear in the second part of the list: General Electric, General Motors, Ford. Next we have two IT companies that provide hardware: Intel makes computer chips, while Cisco makes network equipment. What about the two companies which are on top: Google and Apple? The first appears to be in the business of information, while the second is making consumer electronics: laptops, monitors, music players, etc. But actually, they are both really making something else. And apparently, this something else is so crucial to the workings of US economy—and consequently, global world as well—that these companies almost daily appear in business news. And the major Internet companies that also daily appear in news - Yahoo, Facebook, Amazon, eBay – are in the same business.

This “something else” is *software*. Search engines, recommendation systems, mapping applications, blog tools, auction tools, instant messaging clients, and, of course, platforms which allow others to write new software – Facebook, Windows, Unix, Android – are in the center of the global economy, culture, social life, and, increasingly, politics. And this “cultural software” – cultural in a sense that it is directly used by hundreds of millions of people and that it carries “atoms” of culture (media and information, as well as human interactions around these media and information) – is only the visible part of a much larger software universe.

Software controls the flight of a smart missile toward its target during war, adjusting its course throughout the flight. Software runs the warehouses and production lines of Amazon, Gap, Dell, and numerous other companies allowing them to assemble and dispatch material objects around the world, almost in no time. Software allows shops and supermarkets to automatically restock their shelves, as well as automatically determine which items should go on sale, for how much, and when and where in the store. Software, of course, is what organizes the Internet, routing email messages, delivering Web pages from a server, switching network traffic, assigning IP addresses, and rendering Web pages in a browser. The school and the hospital, the military base and the scientific laboratory, the airport and the city—all social, economic, and cultural systems of modern society—run on software. Software is the invisible glue that ties it all together. While various systems of modern society speak in different languages and have different goals, they all share the syntaxes of software: control statements “if/then” and “while/do”, operators and data types including characters and floating point numbers, data structures such as lists, and interface conventions encompassing menus and dialog boxes.

If electricity and the combustion engine made industrial society possible, software similarly enables gllobal information society. The “knowledge workers”, the “symbol analysts”, the “creative industries”, and the “service industries” - all these key economic players of information society can’t exist without software. Data visualization software used by a scientist, spreadsheet software used a financial analyst, Web design software used by a designer working for a transnational advertising energy, reservation software used by an airline. Software is what also drives the process of globalization, allowing companies to distribute management nodes, production facilities, and storage and consumption outputs around the world. Regardless of which new dimension of contemporary existence a particular social theory of the last few decades has focused on—information society, knowledge society, or network society—all these new dimensions are enabled by software.

Paradoxically, while social scientists, philosophers, cultural critics, and media and new media theorists have by now seem to cover all aspects of IT revolution, creating a number of new disciplines such as cyber culture, Internet studies, new media theory, and digital culture, the underlying engine which drives most of these subjects—software—has received little or not direct attention. Software is still invisible to most academics, artists, and cultural professionals interested in IT and its cultural and social effects. (One important exception is Open Source movement and related issues around copyright and IP that has been extensively discussed in many academic disciplines). But if we limit critical discussions to the notions of “cyber”, “digital”, “Internet,” “networks,” “new media”, or “social media,” we will never be able to get to what is behind new representational and communication media and to understand what it really is and what it does. If we don’t address software itself, we are in danger of always dealing only with its effects rather than the causes: the output that appears on a computer screen rather than the programs and social cultures that produce these outputs.

"Information society," "knowledge society," "network society," "social media" – regardless of which new feature of contemporary existence a particular social theory has focused on, all these new features are enabled by software. It is time we focus on software itself.

What is "software studies"?

This book aims to contribute to the developing intellectual paradigm of "software studies." What is software studies? Here are a few definitions. The first comes from my own book *The Language of New Media* (completed in 1999; published by MIT Press in 2001), where, as far as I know, the terms "software studies" and "software theory" appeared for the first time. I wrote: "New media calls for a new stage in media theory whose beginnings can be traced back to the revolutionary works of Robert Innis and Marshall McLuhan of the 1950s. To understand the logic of new media we need to turn to computer science. It is there that we may expect to find the new terms, categories and operations that characterize media that became programmable. From media studies, we move to something which can be called software studies; from media theory — to software theory."

Reading this statement today, I feel some adjustments are in order. It positions computer science as a kind of absolute truth, a given which can explain to us how culture works in software society. But computer science is itself part of culture. Therefore, I think that Software Studies has to investigate both the role of software in forming contemporary culture, and cultural, social, and economic forces that are shaping development of software itself.

The book that first comprehensively demonstrated the necessity of the second approach was *New Media Reader* edited by Noah Wardrip-Fruin and Nick Montfort (The MIT Press, 2003). The publication of this groundbreaking anthology laid the framework for the historical study of software as it relates to the history of culture. Although *Reader* did not explicitly use the term "software studies," it did propose a new model for how to think about software. By systematically juxtaposing important texts by pioneers of cultural computing and key artists active in the same historical periods, the Reader demonstrated that both belonged to the same larger epistemes. That is, often the same idea was simultaneously articulated in thinking of both artists and scientists who were inventing cultural computing. For instance, the anthology opens with the story by Jorge Borges (1941) and the article by Vannevar Bush (1945) which both contain the idea of a massive branching structure as a better way to organize data and to represent human experience.

In February 2006 Mathew Fuller who already published a pioneering book on software as culture (*Behind the Blip, essays on the culture of software*, 2003) organized the very first *Software Studies Workshop* at Piet Zwart Institute in Rotterdam. Introducing the workshop, Fuller wrote: "Software is often a blind spot in the theorization and study of computational and networked digital media. It is the very grounds and 'stuff' of media design. In a sense, all intellectual work is now 'software study', in that software provides its media and its context, but there are very few places where the specific nature, the materiality, of software is studied except as a matter of engineering."[3]

I completely agree with Fuller that "all intellectual work is now 'software study." Yet it will take some time before the intellectuals will realize it. At the moment of this writing (Spring 2008), software studies is a new paradigm for intellectual inquiry that is now just beginning to emerge. The MIT Press is publishing the very first book that has this term in its title later this year (*Software Studies: A Lexicon*, edited by Matthew Fuller.) At the same time, a number of already published works by the leading media theorists of our times - Katherine Hayles, Friedrich A. Kittler, Lawrence Lessig, Manual Castells, Alex Galloway, and others - can be retroactively identified as belonging to "software studies.[4] Therefore, I strongly believe that this paradigm has already existed for a number of years but it has not been explicitly named so far. (In other words, the state of "software studies" is similar to where "new media" was in the early 1990s.)

In his introduction to 2006 Rotterdam workshop Fuller writes that "software can be seen as an object of study and an area of practice for art and design theory and the humanities, for cultural studies and science and technology studies and for an emerging reflexive strand of computer science." Given that a new academic discipline can be defined either through a unique object of study, a new research method, or a combination of the two, how shall we think of software studies? Fuller's statement implies that "software" is a new object of study which should be put on the agenda of existing disciplines and which can be studied using already existing methods – for instance, object-network theory, social semiotics, or media archeology.

I think there are good reasons for supporting this perspective. I think of software as *a layer that permeates all areas of contemporary societies*. Therefore, if we want to understand contemporary techniques of control, communication, representation, simulation, analysis, decision-making, memory, vision, writing, and interaction, our analysis can't be complete until we consider this software layer. Which means that all disciplines which deal with contemporary society and culture – architecture, design, art criticism, sociology, political science, humanities, science and technology studies, and so on – need to account for the role of software and its effects in whatever subjects they investigate.

At the same time, the existing work in software studies already demonstrates that if we are to focus on software itself, we need a new methodology. That is, it helps to practice what one writes about. It is not accidental that the intellectuals who have most systematically written about software's roles in society and culture so far all either have programmed themselves or have been systematically involved in cultural projects which centrally involve writing of new software: Katherine Hales, Mathew Fuller, Alexander Galloway, Ian Bogust, Geet Lovink, Paul D. Miller, Peter Lunenfeld, Katie Salen, Eric Zimmerman, Matthew Kirschenbaum, William J. Mitchell, Bruce Sterling, etc. In contrast, the scholars without this experience such as Jay Bolter, Siegfried Zielinski, Manual Castells, and Bruno Latour as have not included considerations of software in their overwise highly influential accounts of modern media and technology.

In the present decade, the number of students in media art, design, architecture, and humanities who use programming or scripting in their work has grown substantially – at least in comparison to 1999 when I first mentioned "software studies" in T*he Language of New Media*. Outside of culture and academic industries, many more people today are writing software as well. To a significant extent, this is the result of new programming and scripting languages such as JavaScript, ActionScript, PHP, Processing, and others. Another important factor is the publication of their APIs by all major Web 2.0 companies in the middle of 2000s. (API, or Application Programming Interface, is a code that allows other computer programs to access services offered by an application. For instance, people can use Google Maps API to embed full Google Maps on their own web sites.) These programming and scripting languages and APIs did not necessary made programming itself any easier. Rather, they made it much more efficient. For instance, when a young designer can create an interesting design with only couple of dozens of code written in Processing versus writing a really long Java program, s/he is much more likely to take up programming. Similarly, if only a few lines in JavaScript allows you to integrate all the functionality offered by Google Maps into your site, this is a great motivation for beginning to work with JavaScript.

In a 2006 article that reviewed other examples of new technologies that allow people with very little or no programming experience to create new custom software (such as Ning and Coghead), Martin LaMonica wrote about a future possibility of "a long tail for apps."[5] Clearly, today the consumer technologies for capturing and editing media are much easier to use than even most high-level programming and scripting languages. But it does not necessary have to stay this way. Think, for instance, of what it took to set up a photo studio and take photographs in 1850s versus simply pressing a single button on a digital camera or a mobile phone in 2000s. Clearly, we are very far from such simplicity in programming. But I don't see any logical reasons why programming can't one day become as easy.

For now, the number of people who can script and program keeps increasing. Although we are far from a true "long tail" for software, software development is gradually getting more democratized. It is, therefore, the right moment, to start thinking theoretically about how software is shaping our culture, and how it is shaped by culture in its turn. The time for "software studies" has arrived.

Cultural Software

German media and literary theorist Friedrich Kittler wrote that the students today should know at least two software languages; only "then they'll be able to say something about what 'culture' is at the moment."[6] Kittler himself programs in an assembler language - which probably determined his distrust of Graphical User Interfaces and modern software applications, which use these interfaces. In a classical modernist move, Kittler argued that we need to focus on the "essence" of computer - which for Kittler meant mathematical and logical foundations of modern computer and its early history characterized by tools such as assembler languages.

This book is determined by my own history of engagement with computers as a programmer, computer animator and designer, media artist, and a teacher. This practical engagement begins in the early 1980s, which was the decade of procedural programming (Pascal), rather than assembly programming. It was also the decade that saw introduction of PCs and first major cultural impact of computing as desktop publishing become popular and hypertext started to be discussed by some literary scholars. In fact, I came to NYC from Moscow in 1981, which was the year IBM introduced their first PC. My first experience with computer graphics was in 1983-1984 on Apple IIE. In 1984 I saw Graphical User Interface in its first successful commercial implementation on Apple Macintosh. The same year I got the job at one of the first computer animation companies (Digital Effects) where I learned how to program 3D computer models and animations. In 1986 I was writing computer programs, which would automatically process photographs to make them look like paintings. In January 1987 Adobe Systems shipped illustrator, followed by Photoshop in 1989. The same year saw the release by *The Abyss* directed by James Cameron. This movie used pioneering CGI to create the first complex virtual character. And, by Christmas of 1990s, Tim Berners-Lee already created all the components of World Wide Web as it exists today: a web server, web pages, and a web browser.

In short, during one decade a computer moved from being a culturally invisible technology to being the new engine of culture. While the progress in hardware and Moore's Law of course played crucial roles in this, even more crucial was the release of software aimed at non-technical users: new graphical user interface, word processing, drawing, painting, 3D modeling, animation, music composing and editing, information management, hypermedia and multimedia authoring (HyperCard, Director), and network information environments (World Wide Web.) With easy-to-use software in place, the stage was set for the next decade of the 1990s when most culture industries gradually shifted to software environments: graphic design, architecture, product design, space design, filmmaking, animation, media design, music, higher education, and culture management.

Although I first learned to program in 1975 when I was in high school in Moscow, my take on software studies has been shaped by watching how beginning in the middle of the 1980s, GUI-based software quickly put computer in the center of culture. Theoretically, I think we should think of the subject of software in the most expanded way possible. That is, we need to consider not only "visible" software used by consumers but also "grey" software, which runs all systems and processes in contemporary society. Yet, since I don't have personal experience writing logistics software or industrial automation software, I will be not be writing about such topics. My concern is with a particular subset of software which I used and taught in my professional life and which I would call *cultural software*. While this term has previously used metaphorically (see J.M. Balkin, *Cultural Software: A Theory of Ideology*, 2003), in this book I am using this term literally to refer to *software programs which are used to create and access media objects and environments*. The examples are programs such as Word, PowerPoint, Photoshop, Illustrator, Final Cut, After Effects, Flash, Firefox, Internet Explorer, etc. Cultural software, in other words, is *a subset of application software* which enables creation, publishing, accessing, sharing, and remixing images, moving image sequences, 3D designs, texts, maps, interactive elements, as well as various combinations of these elements such as web sites, 2D designs, motion graphics, video games, commercial and artistic interactive installations, etc. (While originally such application software was designed to run on the desktop, today some of the media creation and editing tools are also available as webware, i.e., applications which are accessed via Web such as Google Docs.)

Given that today the multi-billion global culture industry is enabled by these software programs, it is interesting that there is no a single accepted way to classify them. Wikipedia article on "application software" includes the categories of "media development software" and "content access software." This is generally useful but not completely accurate – since today most "content access software" also includes at least some media editing functions. QuickTime Player can be used to cut and paste parts of video; iPhoto allows a number of photo editing operations, and so on. Conversely, in most cases "media development" (or "content creation") software such as Word or PowerPoint is the same software commonly used to both develop and access content. (This co-existence of authoring and access functions is itself an important distinguishing feature of software culture). If we are visit web sites of popular makes of these software applications such as Adobe and Autodesk, we will find that these companies may break their products by market (web, broadcast, architecture, and so on) or use sub-categories such as "consumer" and "pro." This is as good as it commonly gets – another reason why we should focus our theoretical tools on interrogating cultural software.

In this book my focus will be on these applications for media development (or "content creation") – but cultural software also includes other types of programs and IT elements. One important category is the *tools for social communication and sharing of media, information, and knowledge* such as web browsers, email clients, instant messaging clients, wikis, social bookmarking, social citation tools, virtual worlds, and so on- in short, social software[7] (Note that such use of the term "social software" partly overlaps with but is not equivalent with the way this term started to be used during 200s to refer to Web 2.0 platforms such as Wikipedia, Flickr, YouTube, and so on.) Another category is the *tools for personal information management* such as address books, project management applications, and desktop search engines. (These categories shift over time: for instance, during 2000s the boundary between "personal information" and "public information" has started to dissolve disappeared as people started to routinely place their media on social networking sites and their calendars online. Similarly, Google's search engine shows you the results both on your local machine and the web – thus conceptually and practically erasing the boundary between "self" and the "world.") Since creation of interactive media often involves at least some original programming and scripting besides what is possible within media development applications such as Dreamweaver or Flash, the *programming environments* also can be considered under cultural software. Moreover, *the media interfaces* themselves – icons, folders, sounds, animations, and user interactions - are also cultural software, since these interface mediate people's interactions with media and other people. (While the older term Graphical User Interface, or GUI, continues to be widely used, the newer term "media interface" is usually more appropriate since many interfaces today – including interfaces of Windows, MAC OS, game consoles, mobile phones and interactive store or museums displays such as Nanika projects for Nokia and Diesel or installations at Nobel Peace Center in Oslo – use all types of media besides graphics to communicate with the users.[8]) I will stop here but this list can easily be extended to include additional categories of software as well.

Any definition is likely to delight some people and to annoy others. Therefore, before going forward I would like to meet one likely objection to the way I defined "cultural software." Of course, the term "culture" is not reducible to separate media and design "objects" which may exist as files on a computer and/or as executable software programs or scripts. It includes symbols, meanings, values, language, habits, beliefs, ideologies, rituals, religion, dress and behavior codes, and many other material and immaterial elements and dimensions. Consequently, cultural anthropologists, linguists, sociologists, and many humanists may be annoyed at what may appear as an uncritical reduction of all these dimensions to a set of media-creating tools. Am I saying that today "culture" is equated with particular subset of application software and the cultural objects can be created with their help? Of course not. However, what I am saying - and what I hope this book explicates in more detail – is that in the end of the 20th century humans have added a fundamentally new dimension to their culture. This dimension is software in general, and application software for creating and accessing content in particular.

I feel that the metaphor of a new dimension added to a space is quite appropriate here. That is, "cultural software" is not simply a new object – no matter how large and important – which has been dropped into the space which we call "culture." In other words, it would be imprecise to think of software as simply another term which we can add to the set which includes music, visual design, built spaces, dress codes, languages, food, club cultures, corporate norms, and so on. So while we can certainly study "the culture of software" – look at things such as programming practices, values and ideologies of programmers and software companies, the cultures of Silicon Valley and Bangalore, etc.- if we only do this, we will miss the real importance of software. Like alphabet, mathematics, printing press, combustion engine, electricity, and integrated circuits, software re-adjusts and re-shapes everything it is applied to – or at least, it has a potential to do this. In other word, just as adding a new dimension of space adds a new coordinate to every element in this space, "adding" software to culture changes the identity of everything which a culture is made from.

In other words, our contemporary society can be characterized as a *software society* and our culture can be justifiably called a *software culture* – because today software plays a central role in shaping both the material elements and many of the immaterial structures which together make up "culture."

As just one example of how the use of software reshapes even most basic social and cultural practices and makes us rethink the concepts and theories we developed to describe them, consider the "atom" of cultural creation, transmission, and memory: a "document" (or a "work"), i.e. some content stored in some media. In a software culture, we no longer deal with "documents," "works," "messages" or "media" in a 20th century terms. Instead of fixed documents whose contents and meaning could be full determined by examining their structure (which is what the majority of twentieth century theories of culture were doing) we now interact with dynamic "software performances." I use the word "performance" because what we are experiencing is constructed by software in real time. So whether we are browsing a web site, use Gmail, play a video game, or use a GPS-enabled mobile phone to locate particular places or friends nearby, we are engaging not with pre-defined static documents but with the dynamic outputs of a real-time computation. Computer programs can use a variety of components to create these "outputs": design templates, files stored on a local machine, media pulled out from the databases on the network server, the input from a mouse, touch screen, or another interface component, and other sources. Thus, although some static documents may be involved, the final media experience constructed by software can't be reduced to any single document stored in some media. In other words, in contrast to paintings, works of literature, music scores, films, or buildings, a critic can't simply consult a single "file" containing all of work's content.

"Reading the code" – i.e., examining the listing of a computer program – also would not help us. First, in the case of any real-life interactive media project, the program code will simply be too long and complex to allow a meaningful reading - plus you will have to examine all the code libraries it may use. And if we are dealing with a web application (referred to as "webware") or a dynamic web site, they often use multitier software architecture where a number of separate software modules interact together (for example, a web client, application server, and a database.[9]) (In the case of large-scale commercial dynamic web site such as amazon.com, what the user experiences as a single web page may involve interactions between more than sixty separate software processes.)

Second, even if a program is relatively short and a critic understands exactly what the program is supposed to do by examining the code, this understanding of the logical structure of the program can't be translated into envisioning the actual user experience. (If it could, the process of extensive testing with the actual users which all software or media company goes through before they release new products – anything from a new software application to a new game – would not be required.) In short, I am suggesting "software studies" should not be confused with "code studies." And while another approach - comparing computer code to a music score which gets interpreted during the performance (which suggests that music theory can be used to understand software culture) – appears more promising, is also very limited since it can't address the most fundamental dimension of software-driven media experience – interactivity.

Even in such seemingly simple cases such as viewing a single PDF document or opening an photo in a media player, we are still dealing with "software performances" - since it is software which defines the options for navigating, editing and sharing the document, rather than the document itself. Therefore examining the PDF file or a JPEG file the way twentieth century critics would examine a novel, a movie, or a TV show will only tell us some things about the experience that we would get when we interact with this document via software. While the content's of the file obviously forms a part of this experience, it is also shaped b the interface and the tools provided by software. This is why the examination of the assumptions, concepts, and the history of culture software – including the theories of its designers - is essential if we are to make sense of "contemporary culture."

The shift in the nature of what constitutes a cultural "object" also calls into questions even most well established cultural theories. Consider what has probably been one of the most popular paradigms since the 1950s – "transmission" view of culture developed in Communication Studies. This paradigm describes mass communication (and sometimes culture in general) as a communication process between the authors who create "messages" and audiences that "receive" them. These messages are not always fully decoded by the audiences for technical reasons (noise in transmission) or semantic reasons (they misunderstood the intended meanings.) Classical communication theory and media industries consider such partial reception a problem; in contrast, from the 1970s Stuart Hall, Dick Hebdige and other critics which later came to be associated with Cultural Studies argued that the same phenomenon is positive – the audiences construct their own meanings from the information they receive. But in both cases theorists implicitly assumed that the message was something complete and definite – regardless of whether it was stored in some media or constructed in "real time" (like in live TV programs). Thus, the audience member would read all of advertising copy, see a whole movie, or listen to the whole song and only after that s/he would interpret it, misinterpret it, assign her own meanings, remix it, and so on.

While this assumption has already been challenged by the introduction of timeshifting technologies and

DVR (digital video recorders), it is just does not apply to "born digital" interactive software media. When a user interacts with a software application that presents cultural content, this content often does not have definite finite boundaries. For instance, a user of Google Earth is likely to find somewhat different information every time she is accessing the application. Google could have updated some of the satellite photographs or added new Street Views; new 3D building models were developed; new layers and new information on already existing layers could have become available. Moreover, at any time a user can load more geospatial data created by others users and companies by either clicking on *Add Content* in the Places panel, or directly opening a KLM file. Google Earth is an example of a new interactive "document" which does not have its content all predefined. Its content changes and grows over time.

But even in the case of a document that does correspond to a single computer file, which is fully predefined and which does not allow changes (for instance, a read-only PDF file), the user's experience is still only partly defined by the file's content. The user is free to navigate the document, choosing both what information to see and the sequence in which she is seeing it. In other words, the "message" which the user "receives" is not just actively "constructed" by her (through a cognitive interpretation) but also actively "managed" (defining what information she is receiving and how.)

Why the History of Cultural Software Does not Exist

"Всякое описание мира сильно отстает от его развития."
(Translation from Russian: "Every description of the world seriously lags behind its actual development.")
Тая Катюша, VJ on MTV.ru.[10]

We live in a software culture - that is, a culture where the production, distribution, and reception of most content - and increasingly, experiences - is mediated by software. And yet, most creative professionals do not know anything about the intellectual history of software they use daily - be it Photoshop, GIMP, Final Cut, After Effects, Blender, Flash, Maya, or MAX.

Where does contemporary cultural software came from? How did its metaphors and techniques were arrived yet? And why was it developed in the first place? We don't really know. Despite the common statements that digital revolution is at least as important as the invention of a printing press, we are largely ignorant of how the key part of this revolution - i.e., cultural software - was invented. Then you think about this, it is unbelievable. Everybody in the business of culture knows about Guttenberg (printing press), Brunelleschi (perspective), The Lumiere Brothers, Griffith and Eisenstein (cinema), Le Corbusier (modern architecture), Isadora Duncan (modern dance), and Saul Bass (motion graphics). (Well, if you happen not to know one of these names, I am sure that you have other cultural friends who do). And yet, a few people heard about J.C. Liicklider, Ivan Sutherland, Ted Nelson, Douglas Engelbart, Alan Kay, Nicholas Negroponte and their colloborators who, between approximately 1960 and 1978, have gradually turned computer into a cultural machine it is today.

Remarkably, history of cultural software does not yet exist. What we have are a few largely biographical books about some of the key individual figures and research labs such as Xerox PARC or Media Lab - but no comprehensive synthesis that would trace the genealogical tree of cultural software.[11] And we also don't have any detailed studies that would relate the history of cultural software to history of media, media theory, or history of visual culture.

Modern art institutions - museums such as MOMA and Tate, art book publishers such as Phaidon and Rizzoli, etc. – promote the history of modern art. Hollywood is similarly proud of its own history – the stars, the directors, the cinematographers, and the classical films. So how can we understand the neglect of the history of cultural computing by our cultural institutions and computer industry itself? Why, for instance, Silicon Valley does not a museum for cultural software? (The Computer History museum in Mountain View, California has an extensive permanent exhibition, which is focused on hardware, operating systems and programming languages – but not on the history of cultural software[12]).

I believe that the major reason has to do with economics. Originally misunderstood and ridiculed, modern art has eventually became a legitimate investment category – in fact, by middle of 2000s, the paintings of a number of twentieth century artists were selling for more than the most famous classical artists. Similarly, Hollywood continues to rip profits from old movies as these continue to be reissued in new formats. What about IT industry? It does not derive any profits from the old software – and therefore, it does nothing to promote its history. Of course, contemporary versions of Microsoft Word, Adobe Photoshop, AutoDesk's AutoCAD, and many other popular cultural applications build up on the first versions which often date from the 1980s, and the companies continue to benefit from the patents they filed for new technologies used in these original versions – but, in contrast to the video games from the 1980s, these early software versions are not treated as a separate products which can be re-issued today. (In principle, I can imagine software industry creating a whole new market for old software versions or applications which at some point were quite important but no longer exist today – for instance, Aldus PageMaker. In fact, given that consumer culture systematically exploits nostalgia of adults for the cultural experiences of their teenage years and youth by making these experiences into new products, it is actually surprising that early software versions were not turned into a market yet. If I used daily MacWrite and MacPaint in the middle of the 1980s, or Photoshop 1.0 and 2.0 in 1990-1993, I think these experiences were as much part of my "cultural genealogy" as the movies and art I saw at the same time. Although I am not necessary advocating creating yet another category of commercial products, if early software was widely available in simulation, it would catalyze cultural interest in software similar to the way in which wide availability of early computer games fuels the field of video game studies.)

Since most theorists so far have not considered cultural software as a subject of its own, distinct from "new media," media art," "internet," "cyberspace," "cyberculture" and "code," we lack not only a conceptual history of media editing software but also systematic investigations of its roles in cultural production. For instance, how did the use of the popular animation and compositing application After Effects has reshaped the language of moving images? How did the adoption of Alias, Maya and other 3D packages by architectural students and young architects in the 1990s has similarly influenced the language of architecture? What about the co-evolution of Web design tools and the aesthetics of web sites – from the bare-bones HTML in 1994 to visually rich Flash-driven sites five years later? You will find frequent mentions and short discussions of these and similar questions in articles and conference discussions, but as far as I know, there have been no book-length study about any of these subjects. Often, books on architecture, motion graphics, graphic design and other design fields will briefly discuss the importance of software tools in facilitating new possibilities and opportunities, but these discussions usually are not further developed.

Summary of the book's argument and chapters

Between early 1990s and middle of the 2000s, cultural software has replaced most other media technologies that emerged in the 19th and 20th century. Most of today's culture is created and accessed via cultural software - and yet, surprisingly, few people know about its history. What was the thinking and motivations of people who between 1960 and late 1970s created concepts and practical techniques which underlie today's cultural software? How does the shift to software-based production methods in the 1990s change our concepts of "media"? How do interfaces and the tools of content development software have reshaped and continue to shape the aesthetics and visual languages we see employed in contemporary design and media? Finally, how does a new category cultural software that emerged in the 2000s – "social software" (or "social media") – redefined the functioning of media and its identity once again? These are the questions that I take up in this book.

My aim is not provide a comprehensive history of cultural software in general, or media authoring software in particular. Nor do I aim to discuss all new creative techniques it enables across different cultural fields. Instead, I will trace a particular path through this history that will take us from 1960 to today and which will pass through some of its most crucial points.

While new media theorists have spend considerable efforts in trying to understand the relationships between digital media and older physical and electronic media, the important sources – the writing and projects by Ivan Sutherland, Douglas Englebardt, Ted Nelson, Alan Kay, and other pioneers of cultural software working in the 1960s and 1970s – still remain largely unexamined. What were their reasons for inventing the concepts and techniques that today make it possible for computers to represent, or "remediate" other media? Why did these people and their colleagues have worked to systematically turn a computer into a machine for media creation and manipulation? These are the questions that I take in part 1, which explores them by focusing on the ideas and work of the key protagonist of “cultural software movement” – Alan Kay.

I suggest that Kay and others aimed to create a particular kind of new media – rather than merely simulating the appearances of old ones. These new media use already existing representational formats as their building blocks, while adding many new previously nonexistent properties. At the same time, as envisioned by Kay, these media are expandable – that is, users themselves should be able to easily add new properties, as well as to invent new media. Accordingly, Kay calls computers the first “metamedium” whose content is “a wide range of already-existing and not-yet-invented media.”

The foundations necessary for the existence of such metamedium were established between 1960s and late 1980s. During this period, most previously available physical and electronic media were systematically simulated in software, and a number of new media were also invented. This development takes us from the very interactive design program – Ivan Sutherland’s Sketchpad (1962) - to the commercial desktop applications that made software-based media authoring and design widely available to members of different creative professions and, eventually, media consumers as well – Word (1984), PageMaker (1985), Illustrator (1987), Photoshop (1989), After Effects (1993), and others.

So what happens next? Do Kay’s theoretical formulations as articulated in 1977 accurately predict the developments of the next thirty years, or have there been new developments which his concept of “metamedium” did not account for? Today we indeed use variety of previously existing media simulated in software as well as new previously non-existent media. Both are been continuously extended with new properties. Do these processes of invention and amplification take place at random, or do they follow particular paths? In other words, what are the key mechanisms responsible for the extension of the computer metamedium?

In part 2 I look at the next stage in the development of media authoring software which historically can be centered on the 1990s. While I don't discuss all the different mechanisms responsible for the continuous development and expansion of computer metamedium, I do analyze in detail a number of them. What are they? At the first approximation, we can think of these mechanisms as forms of remix. This should not be surprising. In the 1990s, remix has gradually emerged as the dominant aesthetics of the era of globalization, affecting and re-shaping everything from music and cinema to food and fashion. (If Fredric Jameson once referred to post-modernism as "the cultural logic of late capitalism," we can perhaps call remix the cultural logic of global capitalism.) Given remix's cultural dominance, we may also expect to find remix logics in cultural software. But if we state this, we are not yet finished. There is still plenty of work that remains to be done. Since we don't have any detailed theories of remix culture (with the possible exception of the history and uses of remix in music), calling something a "remix" simultaneously requires development of this theory. In other words, if we simply labell some cultural phenomenon a remix, this is not by itself an explanation. So what are remix operations that are at work in cultural software? Are they different from remix operations in other cultural areas?

My arguments which are developed in part 2 in the book can be summarized as follows. In the process of the translation from physical and electronic media technologies to software, all individual techniques and tools that were previously unique to different media "met" within the same software environment. This meeting had most fundamental consequences for human cultural development and for the media evolution. It disrupted and transformed the whole landscape of media technologies, the creative professions that use them, and the very concept of "media" itself.

To describe how previously separate media work together in a common software-based environment, I coin a new term "deep remixability." Although "deep remixability" has a connection with "remix" as it is usually understood, it has its own distinct mechanisms. Software production environment allows designers to remix not only the content of different media, but also their fundamental techniques, working methods, and ways of representation and expression.

Once they were simulated in a computer, previously non-compatible techniques of different media begin to be combined in endless new ways, leading to new media hybrids, or, to use a biological metaphor, new "media species." As just one example among countless others think, for instance, of popular Google Earth application that combines techniques of traditional mapping, the field of Geographical Information Systems (GIS), 3D computer graphics and animation, social software, search, and other elements and functions. In my view, this ability to combine previously separate media techniques represents a fundamentally new stage in the history of human media, human semiosis, and human communication, enabled by its "softwarization."

While today "deep remixability" can be found at work in all areas of culture where software is used, I focus on particular areas to demonstrate how it functions in detail. The first area is *motion graphics* – a dynamic part of cotemporary culture, which, as far as I know, has not yet been theoretically analyzed in detail anywhere. Although selected precedents for contemporary motion graphics can already be found in the 1950s and 1960s in the works by Saul Bass and Pablo Ferro, its exponential growth from the middle of 1990s is directly related to adoption of software for moving image design – specifically, After Effects software released by Adobe in 1993. Deep remixability is central to the aesthetics of motion graphics. That is, the larger proportion of motion graphics projects done today around the world derive their aesthetic effects from combining different techniques and media traditions – animation, drawing, typography photography, 3D graphics, video, etc – in new ways. As a part of my analysis, I look at how the typical software-based production workflow in a contemporary design studio – the ways in which a project moves from one software application to another – shapes the aesthetics of motion graphics, and visual design in general.

Why did I select motion graphics as my central case study, as opposed to any other area of contemporary culture which has either been similarly affected by the switch to a software-based production processes, or is native to computers? The examples of the former area sometimes called "going digital" are architecture, graphic design, product design, information design, and music; the examples of the later area (refered to as "born digital") are game design, interaction design, user experience design, user interface design, web design, and interactive information visualization. Certainly, most of the new design areas which have a word "interaction" or "information" as part of their titles and which emerged since middle of the 1990s have been as ignored by cultural critics as motion graphics, and therefore they demand as much attention.

My reason has to do with the richness of new forms – visual, spatial, and temporal - that developed in motion graphics field since it started to rapidly grow after the introduction of After Effects (1993-). If we approach motion graphics in terms of these forms and techniques (rather than only their content), we will realize that they represent a significant turning point in the history of human communication techniques. Maps, pictograms, hieroglyphs, ideographs, various scripts, alphabet, graphs, projection systems, information graphics, photography, modern language of abstract forms (developed first in European painting and since 1920 adopted in graphic design, product design and architecture), the techniques of 20th century cinematography, 3D computer graphics, and of course, variety of "born digital" visual effects – practically all communication techniques developed by humans until now are routinely get combined in motion graphics projects. Although we may still need to figure out how to fully use this new semiotic metalanguage, the importance of its emergence is hard to overestimate.

I continue discussion of "deep remixability" by looking at another area of media design - visual effects in feature films. Films such as Larry and Andy Wachowski's *Matrix* series (1999–2003), Robert Rodriguez's *Sin City* (2005), and Zack Snyder's *300* (2007) are a part of a growing trend to shoot a large portion or the whole film using a "digital backlot" (green screen).[13] These films combine multiple media techniques to create various stylized aesthetics that cannot be reduced to the look of twentieth century live-action cinematography or 3D computer animation. As a case study, I analyze in detail the production methods called Total Capture and Virtual Cinematography. They were originally developed for *Matrix* films and since then has used in other feature films and video games such as EA SPORT Tiger Woods 2007. These methods combine multiple media techniques in a particularly intricate way, thus providing us one of the most extreme examples of "deep remixability."

If the development of media authoring software in the 1990s has transformed most professional media and design fields, the developments of 2000s – the move from desktop applications to webware (applications running on the web), social media sites, easy-to-use blogging and media editing tools such as Blogger, iPhoto and iMovie, combined with the continuously increasing speed of processors, the decreasing cost of noteboos, netbooks, and storage, and the addition of full media capabilities to mobile phones – have transformed how ordinary people use media. The exponential explosion of the number of people who are creating and sharing media content, the mind-boggling numbers of photos and videos they upload, the ease with which these photos and videos move between people, devices, web sites, and blogs, the wider availability of faster networks – all these factors contribute to a whole new "media ecology." And while its technical, economic, and social dimensions have already been analyzed in substantial details – I am thinking, for instance, of detailed studies of the economics of "long tail" phenomena, discussions of fan cultures[14], work on web-based social production and collaboration[15], or the research within a new paradigm of "web science" – its media theoretical and media aesthetics dimensions have not been yet discussed much at the time I am writing this.

Accordingly, Part 3 focuses on the new stage in the history of cultural software - shifting the focus from professional media authoring to the social web and consumer media. The new software categories include
social networking websites (MySpace, Facebook, etc.), media sharing web sites (Flickr, Photobucket, YouTube, Vimeo, etc.); consumer-level software for media organization and light editing (for example, iPhoto); blog editors (Blogger, Wordpress); RSS Readers and personalized home pages (Google Reader, iGoogle, netvibes, etc). (Keep in mind that software – especially webware designed for consumers – continuously evolves, so some of the categories above, their popularity, and the identity of particular applications and web sites may change may change by the time your are reading this. One graphic example is the shift in the identity of Facebook. Suring 2007, it moved from being yet another social media application competing with MySpace to becoming "social OS" aimed to combine the functionality of previously different applications in one place – replacing, for instance, stand-alone email software for many users.)

This part of the book also offers additional perspective on how to study cultural software in society. None of the software programs and web sites mentioned in the previous paragraph function in isolation. Instead, they participate in larger ecology which includes search engines, RSS feeds, and other web technologies; inexpensive consumer electronic devices for capturing and accessing media (digital cameras, mobile phones, music players, video players, digital photo frames); and the technologies which enable transfer of media between devices, people, and the web (storage devices, wireless technologies such as Wi-Fi and WiMax, communication standards such as Firewire, USB and 3G). Without this ecology social software would not be possible. Therefore, this whole ecology needs to be taken into account in any discussion of social software, as well as consumer-level content access / media development software designed to work with web-based media sharing sites. And while the particular elements and their relationship in this ecology are likely to change over time – for instance, most media content may eventually be available on the network; communication between devices may similarly become fully transparent; and the very rigid physical separation between people, devices they control, and "non-smart" passive space may become blurred – the very idea of a technological ecology consisting of many interacting parts which include software is not unlikely to go away anytime soon. One example of how the 3rd part of this book begins to use this new perspective is the discussion of "media mobility" – an example of a new concept which can allow to us to talk about the new techno-social ecology as a whole, as opposed to its elements in separation.

PART 1: Inventing Cultural Software

Chapter 1. Alan Kay's Universal Media Machine

Medium:
8.
a. A specific kind of artistic technique or means of expression as determined by the materials used or the creative methods involved: the medium of lithography.
b. The materials used in a specific artistic technique: oils as a medium.
American Heritage Dictionary, 4th edition (Houghton Mifflin, 2000).

"The best way to predict the future is to invent it."
Alan Kay

Appearance versus Function

Between its invention in mid 1940s and arrival of PC in middle of 1980s, a digital computer was mostly used for military, scientific and business calculations and data processing. It was not interactive. It was not designed to be used by a single person. In short, it was hardly suited for cultural creation.

As a result of a number of developments of the 1980s and 1990s – the rise of personal computer industry, adoption of Graphical User Interfaces (GUI), the expansion of computer networks and World Wide Web – computers moved into the cultural mainstream. Software replaced many other tools and technologies for the creative professionals. It has also given hundreds of millions of people the abilities to create, manipulate, sequence and share media – but has this lead to the invention of fundamentally *new* forms of culture? Today media companies are busy inventing *e-books* and interactive *television*; the consumers are happily purchasing music *albums* and *feature films* distributed in digital form, as well making *photographs* and *video* with their digital cameras and cell phones; office workers are reading PDF *documents which imitate paper*. (And even at the futuristic edge of digital culture - smart objects/ambient intelligence – traditional forms persist: Philips showcases "smart" *household mirror* which can hold electronic notes and videos, while its director of research dreams about a normal looking *vase* which can hold digital photographs.[16])

In short, it appears that the revolution in means of production, distribution, and access of media has not been accompanied by a similar revolution in syntax and semantics of media. Who shall we blame for this? Shall we put the blame on the pioneers of cultural computing – J.C. Licklider, Ivan Sutherland, Ted Nelson, Douglas Engelbart, Seymour Paper, Nicholas Negroponte, Alan Kay, and others? Or, as Nelson and Kay themselves are eager to point out, the problem lies with the way the industry implemented their ideas?

Before we blame the industry for bad implementation – we can always pursue this argument later if necessary – let us look into the thinking of the inventors of cultural computing themselves. For instance, what about the person who guided the development of a prototype of a modern person computer - Alan Kay?

Between 1970 and 1981 Alan Kay was working at Xerox PARC – a research center established by Xerox in Palo Alto. Building on the previous work of Sutherland, Nelson, Englebart, Licklider, Seymour Papert, and others, the Learning Research Group at Xerox PARC headed by Kay systematically articulated the paradigm and the technologies of *vernacular media computing,* as it exists today.[17]

Although selected artists, filmmakers, musicians, and architects were already using computers since the

1950s, often developing their software in collaboration with computer scientists working in research labs (Bell Labs, IBM Watson Research Center, etc.) most of this software was aimed at producing only particular kind of images, animations or music congruent with the ideas of their authors. In addition, each program was designed to run on a particular machine. Therefore, these software programs could not function as general-purpose tools easily usable by others.

It is well known most of the key ingredients of personal computers as they exist today came out from Xerox PARC: Graphical User Interface with overlapping windows and icons, bitmapped display, color graphics, networking via Ethernet, mouse, laser printer, and WYIWYG ("what you see is what you get") printing. But what is equally important is that Kay and his colleagues also developed a range of applications for media manipulation and creation which all used a graphical interface. They included a word processor, a file system, a drawing and painting program, an animation program, a music editing program, etc. Both the general user interface and the media manipulation programs were written in the same programming language Smalltalk. While some the applications were programmed by members of Kay's group, others were programmed by the users that included seventh-grade high-school students.[18] (This was consistent with the essence of Kay's vision: to provide users with a programming environment, examples of programs, and already-written general tools so the users will be able to make their own creative tools.)

When Apple introduced first Macintosh computer in 1984, it brought the vision developed at Xerox PARC to consumers (the new computer was priced at US$2,495). The original Macintosh 128K included a word processing and a drawing application (MacWrite and MacDraw, respectively). Within a few years they were joined by other software for creating and editing different media: Word, PageMaker and VideoWorks (1985)[19], SoundEdit (1986), Freehand

and Illustrator (1987), Photoshop (1990), Premiere (1991), After Effects (1993), and so on. In the early 1990s, similar functionality became available on PCs running Microsoft Windows.[20] And while MACs and PCs were at first not fast enough to offer a true competition for traditional media tools and technologies (with the exception of word processing), other computer systems specifically optimized for media processing started to replace these technologies already in the 1980s. (The examples are Next workstation, produced between 1989 and 1996; Amiga, produced between 1985 and 1994; and Paintbox, was first released in 1981.)

By around 1991, the new identity of a computer as a personal media editor was firmly established. (This year Apple released QuickTime that brought video to the desktop; the same year saw the release of James Cameron's *Terminator II*, which featured pioneering computer-generated special effects). The vision developed at Xerox PARC became a reality – or rather, one important part of this vision in which computer was turned into a personal machine for display, authoring and editing content in different media. And while in most cases Alan Kay and his collaborators were not the first to develop particular kinds of media applications – for instance, paint programs and animation programs were already written in the second part of the 1960s[21] - by implementing all of them on a single machine and giving them consistent appearance and behavior, Xerox PARC researchers established a new paradigm of media computing.

I think that I have made my case. The evidence is overwhelming. It is Alan Kay and his collaborators at PARC that we must call to task for making digital computers imitate older media. By developing easy to use GUI-based software to create and edit familiar media types, Kay and others appear to have locked the computer into being a simulation machine for "old media." Or, to put this in terms of Jay Bolter and Richard Grusin's influential book *Remediation: Understanding New Media* (2000), we can say that

GUI-based software turned a digital computer into a "remediation machine:" a machine that expertly represents a range of earlier media. (Other technologies developed at PARC such as bitmapped color display used as the main computer screen, laser printing, and the first Page Description Language which eventually lead to Postscript were similarly conceived to support computer's new role as a machine for simulation of physical media.)

Bolter and Grusin define remediation as "the representation of one medium in another."[22] According to their argument, new media always remediate the old ones and therefore we should not expect that computers would function any differently. This perspective emphasizes the continuity between computational media and earlier media. Rather than being separated by different logics, all media including computers follow the same logic of remediation. The only difference between computers and other media lies in how and what they remediate. As Bolter and Grusin put this in the first chapter of their book, "What is new about digital media lies in their particular strategies for remediating television, film, photography, and painting." In another place in the same chapter they make an equally strong statement that leaves no ambiguity about their position: "We will argue that remediation is a defining characteristic of the new digital media."

It we consider today all the digital media created by both consumers and by professionals – digital photography and video shot with inexpensive cameras and cell phones, the contents of personal blogs and online journals, illustrations created in Photoshop, feature films cut on AVID, etc. – in terms of its appearance digital media indeed often looks exactly the same way as it did before it became digital. Thus, if we limit ourselves at looking at the media surfaces, remediation argument accurately describes much of computational media. But rather than accepting this condition as an inevitable consequence of the universal logic of remediation, we should ask why this is the case. In other words, if contemporary computational media imitates other media, how did this become possible? There was definitely nothing in the original theoretical formulations of digital computers by Turing or Von Neumann about computers imitating other media such as books, photography, or film.

The conceptual and technical gap which separates first room size computers used by military to calculate the shooting tables for anti-aircraft guns and crack German communication codes and contemporary small desktops and laptops used by ordinary people to hold, edit and share media is vast. The contemporary identity of a computer as a media processor took about forty years to emerge – if we count from 1949 when MIT's Lincoln Laboratory started to work on first interactive computers to 1989 when first commercial version of Photoshop was released. *It took generations of brilliant and creative thinkers to invent the multitude of concepts and techniques that today make possible for computers to "remediate" other media so well*. *What were their reasons for doing this? What was their thinking?* In short, why did these people dedicate their careers to inventing the ultimate "remediation machine"?

While media theorists have spend considerable efforts in trying to understand the relationships between digital media and older physical and electronic media, the important sources – the writing and projects by Ivan Sutherland, Douglas Englebardt, Ted Nelson, Alan Kay, and other pioneers working in the 1960s and 1970s – remained largely unexamined. This book does not aim to provide a comprehensive intellectual history of the invention of media computing. Thus, I am not going to consider the thinking of all the key figures in the history of media computing (to do this right would require more than one book.) Rather, my concern is with the present and the future. Specifically, I want to understand some of the dramatic transformations in what media is, what it can do, and how we use –

the transformations that are clearly connected to the shift from previous media technologies to software. Some of these transformations have already taken place in the 1990s but were not much discussed at the time (for instance, the emergence of a new language of moving images and visual design in general). Others have not even been named yet. Still others – such as remix and mash-up culture – are being referred to all the time, and yet the analysis of how they were made possible by the evolution of media software so far was not attempted.

In short, I want to understand what is *"media after software"* – that is, what happened to the techniques, languages, and the concepts of twentieth century media as a result of their computerization. Or, more precisely, what has happened to media after they have been *software-ized*. (And since in the space of a single book I can only consider some of these techniques, languages and concepts, I will focus on those that, in my opinion, have not been yet discussed by others). To do this, I will trace a particular path through the conceptual history of media computing from the early 1960s until today.

To do this most efficiently, in this chapter we will take a closer look at one place where the identity of a computer as a "remediation machine" was largely put in place – Alan Kay's Learning Research Group at Xerox PARC that was in operation during the 1970s. We can ask two questions: first, what exactly Kay wanted to do, and second, how he and his colleagues went about achieving it. The brief answer – which will be expanded below - is that Kay wanted to turn computers into a "personal dynamic media" which can be used for learning, discovery, and artistic creation. His group achieved this by systematically simulating most existing media within a computer while simultaneously adding many new properties to these media. Kay and his collaborators also developed a new type of programming language that, at least in theory, would allow the users to quickly invent new types of media using the set of general tools already provided for them. All these tools and simulations of already existing media were given a unified user interface designed to activate multiple mentalities and ways of learning - kinesthetic, iconic, and symbolic.

Kay conceived of "personal dynamic media" as a fundamentally new kind of media with a number of historically unprecedented properties such as the ability to hold all of user's information, simulate all types media within a single machine, and "involve learner in a two-way conversation."[23] These properties enable new relationships between the user and the media she may be creating, editing, or viewing on a computer. And this is essential if we want to understand the relationships between computers and earlier media. Briefly put, while visually computational media may closely mimic other media, these media now function in different ways.

For instance, consider digital photography that often does imitate in appearance traditional photography. For Bolter and Grusin, this is example of how digital media 'remediates" its predecessors. But rather than only paying attention to their appearance, let us think about how digital photographs can function. If a digital photograph is turned into a physical object in the world – an illustration in a magazine, a poster on the wall, a print on a t-shirt – it functions in the same ways as its predecessor.[24] But if we leave the same photograph inside its native computer environment – which may be a laptop, a network storage system, or any computer-enabled media device such as a cell phone which allows its user to edit this photograph and move it to other devices and the Internet – it can function in ways which, in my view, make it radically different from its traditional equivalent. To use a different term, we can say that a digital photograph offers its users many affordances that its non-digital predecessor did not. For example, a digital photograph can be quickly modified in numerous ways and equally quickly combined with other images; instantly moved around the world and shared with other people; and inserted into a text document, or an architectural design. Furthermore, we can automatically (i.e., by running the appropriate algorithms) improve its contrast, make it sharper, and even in some situations remove blur.

Note that only some of these new properties are specific to a particular media – in our example, a digital photograph, i.e. an array of pixels represented as numbers. Other properties are shared by a larger class of media species – for instance, at the current stage of digital culture all types of media files can be attached to an email message. Still others are even more general features of a computer environment within the current GUI paradigm as developed thirty years ago at PARC: for instance, the fast response of computer to user's actions which assures "no discernable pause between cause and effect."[25] Still others are enabled by network protocols such as TCP-IP that allows all kinds of computers and other devices to be connected to the same network. In summary, we can say that only some of the "new DNAs" of a digital photograph are due its particular place of birth, i.e., inside a digital camera. Many others are the result of current paradigm of network computing in general.

Before diving further into Kay's ideas, I should more fully disclose my reasons why I chose to focus on him as opposed to somebody else. The story I will present could also be told differently. It is possible to put Sutherland' work on Sketchpad in the center of computational media history; or Englebart and his Research Center for Augmenting Human Intellect which throughout the 1960s developed hypertext (independently of Nelson), the mouse, the window, the word processor, mixed text/graphics displays, and a number of other "firsts." Or we can shift focus to the work of Architecture Machine Group at The MIT, which since 1967 was headed by Nicholas Negroponte (In 1985 this group became The Media Lab). We also need to recall that by the time Kay's Learning Research Group at PARC flashed out the details of GUI and programmed various media editors in Smalltalk (a paint program, an illustration program, an animation program, etc.), artists, filmmakers and architects were already using computers for more than a decade and a number of large-scale exhibitions of computer art were put in major museums around the world such as the Institute of Contemporary Art, London, The Jewish Museum, New York, and Los Angeles County Museum of Art. And certainly, in terms of advancing computer techniques for visual representation enabled by computers, other groups of computer scientists were already ahead. For instance, at University of Utah, which became the main place for computer graphics research during the first part of the 1970s, scientists were producing 3D computer graphics much superior to the simple images that could be created on computers being build at PARC. Next to University of Utah a company called Evans and Sutherland (headed by the same Ivan Sutherland who was also teaching at University of Utah) was already using 3D graphics for flight simulators – essentially pioneering the type of new media that can be called "navigable 3D virtual space."[26]

While the practical work accomplished at Xerox PARC to establish a computer as a comprehensive media machine is one of my reasons, it is not the only one. The key reason I decided to focus on Kay is his theoretical formulations that place computers in relation to other media and media history. While Vannevar Bush, J.C. Lindlicker and Douglas Englebart were primary concerned with augmentation of intellectual and in particular scientific work, Kay was equally interested in computers as "a medium of expression through drawing, painting, animating pictures, and composing and generating music."[27] Therefore if we really want to understand how and why computers were redefined as a cultural media, and how the new computational media is different from earlier physical and electronic media, I think that Kay provides us with the best theoretical perspective.

"Simulation is the central notion of the Dynabook"

While Alan Kay articulated his ideas in a number of articles and talks, his 1977 article co-authored with one of his main PARC collaborators, computer scientist Adele Goldberg, is particularly useful resource if we want to understand contemporary computational media. In this article Kay and Goldberg describes the vision of the Learning Research Group at PARC in the following way: to create "*a personal dynamic medium* the size of a notebook (the Dynabook) which could be owned by everyone and could have the power to handle virtually all of its owner's information-related needs."[28] Kay and Goldberg ask the readers to imagine that this device "had enough power to outrace your senses of sight and hearing, enough capacity to store for later retrieval thousands of page-equivalents of reference materials, poems, letters, recipes, records, drawings, animations, musical scores, waveforms, dynamic simulations and anything else you would like to remember and change."[29]

In my view, "all" in the first statement is important: it means that the Dynabook – or computational media environment in general, regardless of the size of a form of device in which it is implemented – should support viewing, creating and editing all possible media which have traditionally were used for human expression and communication. Accordingly, while separate programs to create works in different media were already in existence, Kay's group for the first time implemented them all together within a single machine. In other words, Kay's paradigm was not to simply create a new type of computer-based media which would co-exist with other physical media. Rather, the goal was to establish a computer as an umbrella, a platform for *all* already existing expressive artistic media. (At the end of the article Kay and Goldberg give a name for this platform – "metamedium.") This paradigm changes our understanding of what media is. From Lessing's *Laocoon; or, On the Limits of Painting and*

Poetry (1766) to Nelson Goodman's *Languages of Art* (1968), the modern discourse about media depends on the assumption that different mediums have distinct properties and in fact should be understood in opposition to each other. Putting all mediums within a single computer environment does not necessary erases all differences in what various mediums can represent and how they are perceived – but it does bring them closer to each other in a number of ways. Some of these new connections were already apparent to Kay and his colleagues; others became visible only decades later when the new logic of media set in place at PARC unfolded more fully; some maybe still not visible to us today because they have not been given practical realization. One obvious example such connections is the emergence of multimedia as a standard form of communication: web pages, PowerPoint presentations, multimedia artworks, mobile multimedia messages, media blogs, and other communication forms which combine few mediums. Another is the rise of common interface conventions and tools which we use in working with different types of media regardless of their origin: for instance, a virtual camera, a magnifying lens, and of course the omnipresent copy, cut and paste commands.[30] Yet another is the ability to map one media into another using appropriate software – images into sound, sound into images, quantitative data into a 3D shape or sound, etc. – used widely today in such areas as DJ/VJ/live cinema performances and information visualization. All in all, it is as though different media are actively trying to reach towards each other, exchanging properties and letting each other borrow their unique features. (This situation is the direct opposite of modernist media paradigm of the early twentieth century which was focused on discovering a unique language of each artistic medium.)

Alan Turing theoretically defined a computer as a machine that can simulate a very large class of other machines, and it is this simulation ability that is largely responsible for the proliferation of computers in modern society. But as I already mentioned, neither he nor other theorists and inventors of digital computers explicitly considered that this simulation could also include media. It was only Kay and his generation that extended the idea of simulation to media – thus turning Universal Turing Machine into a Universal Media Machine, so to speak.

Accordingly, Kay and Goldberg write: "In a very real sense, simulation is the central notion of the Dynabook."[31] When we use computers to simulate some process in the real world – the behavior of a weather system, the processing of information in the brain, the deformation of a car in a crash – our concern is to correctly model the necessary features of this process or system. We want to be able to test how our model would behave in different conditions with different data, and the last thing we want to do is for computer to introduce some new properties into the model that we ourselves did not specify. In short, when we use computers as a general-purpose medium for simulation, we want this medium to be completely "transparent."

But what happens when we simulate different media in a computer? In this case, the appearance of new properties may be welcome as they can extend the expressive and communication potential of these media. Appropriately, when Kay and his colleagues created computer simulations of existing physical media – i.e. the tools for representing, creating, editing, and viewing these media – they "added" many new properties. For instance, in the case of a book, Kay and Goldberg point out "It need not be treated as a simulated paper book since this is *a new medium with new properties.* A dynamic search may be made for a particular context. The non-sequential nature of the file medium and the use of dynamic manipulation allows a story to have many accessible points of view."[32] Kay and his colleagues also added various other properties to the computer simulation of paper documents. As Kay has referred to this in another article, his idea was not to simply imitate paper but rather to create "magical paper."[33] For instance, PARC team gave users the ability to modify the fonts in a document and create new fonts. They also implemented another important idea that was already developed by Douglas Englebardt's team in the 1960s: the ability to create different views of the same structure (I will discuss this in more detail below). And both Englebart and Ted Nelson also already "added" something else: the ability to connect different documents or different parts of the same document through hyperlinking – i.e. what we now know as hypertext and hypermedia. Englebart's group also developed the ability for multiple users to collaborate on the same document. This list goes on and on: e-mail in 1965, newsgroups in 1979, World Wide Web in 1991, etc.

Each of these new properties has far-reaching consequences. Take search, for instance. Although the ability to search through a page-long text document does not sound like a very radical innovation, as the document gets longer this ability becomes more and more important. It becomes absolutely crucial if we have a very large collection of documents – such as all the web pages on the Web. Although current search engines are far from being perfect and new technologies will continue to evolve, imagine how different the culture of the Web would be without them.

Or take the capacity to collaborate on the same document(s) by a number of users connected to the same network. While it was already widely used by companies in the 1980s and 1990s, it was not until early 2000s that the larger public saw the real cultural potential of this "addition" to print media. By harvesting the small amounts of labor and expertise contributed by a large number of volunteers, social software projects – most famously, Wikipedia – created vast and dynamically updatable pools of knowledge which would be impossible to create in traditional ways. (In a less visible way, every time we do a search on the Web and then click on some of the results, we also contribute to a knowledge set used by everybody else. In deciding in which sequence to present the results of a particular search, Google's algorithms take into account which among the results of previous searches for the same words people found most useful.)

Studying the writings and public presentations of the people who invented interactive media computing – Sutherland, Englebart, Nelson, Negroponte, Kay, and others – makes it clear that they did not come with new properties of computational media as an after-thought. On the contrary, they knew that were turning physical media into new media. In 1968 Englebart gave his famous demo at the Fall Joint Computer Conference in San Francisco before few thousand people that included computer scientists, IBM engineers, people from other companies involved in computers, and funding officers from various government agencies.[34] Although Englebart had whole ninety minutes, he had a lot to show. Over the few previous years, his team at The Research Center for Augmenting Human Intellect had essentially developed modern *office* environment as it exists today (not be confused with modern *media design* environment which was developed later at PARC). Their computer system included word processing with outlining features, documents connected through hypertext, online collaboration (two people at remote locations working on the same document in real-time), online user manuals, online project planning system, and other elements of what is now called "computer-supported collaborative work." The team also developed the key elements of modern user interface that were later refined at PARC: a mouse and multiple windows.

Paying attention to the sequence of the demo reveals that while Englebart had to make sure that his audience would be able to relate the new computer system to what they already know and use, his focus was on new features of simulated media never before available previously. Englebart devotes the first segment of the demo to word processing, but as soon as he briefly demonstrated text entry, cut, paste, insert, naming and saving files – in other words, the set of tools which make a computer into a more versatile typewriter – he then goes on to show in more length the features of his system which no writing medium had before: "view control."[35] As Englebart points out, the new writing medium could switch at user's wish between *many different views of the same information*. A text file could be sorted in different ways. It could also be organized as a hierarchy with a number of levels, like in outline processors or outlining mode of contemporary word processors such as Microsoft Word. For example, a list of items can be organized by categories and individual categories can be collapsed and expanded.

Englebart next shows another example of view control, which today, forty years after his demo, is still not available in popular document management software. He makes a long "to do" list and organizes it by locations. He then instructs the computer to displays these locations as a visual graph (a set of points connected by lines.) In front of our eyes, representation in one medium changes into another medium – text becomes a graph. But this is not all. The user can control this graph to display different amounts of information – something that no image in physical media can do. As Englebart clicks on different points in a graph corresponding to particular locations, the graph shows the appropriate part of his "to do" list. (This ability to interactively change how much and what information an image shows is particularly important in today's information visualization applications.)

Next Englebart presents "a chain of views" which he prepared beforehand. He switches between these views using "links" which may look like hyperlinks the way they exist on the Web today – but they actually have a different function. Instead of creating a path between many different documents a la Vannevar Bush's Memex (often seen as the precursor to modern hypertext), Englebart is using links as a method for switching between different views of a single document organized hierarchically. He brings a line of words displayed in the upper part of the screen; when he clicks on these words, more detailed information is displayed in the lower part of the screen. This information can in its turn contain links to other views that show even more detail.

Rather than using links to drift through the textual universe associatively and "horizontally," we move "vertically" between more general and more detailed information. Appropriately, in Englebart's paradigm, we are not "navigating" – we are "switching views." We can create many different views of the same information and switch between these views in different ways. And this is what Englebart systematically explains in this first part of his demo. He demonstrates that you can change views by issuing commands, by typing numbers that correspond to different parts of a hierarchy, by clicking on parts of a picture, or on links in the text. (In 1967 Ted Nelson articulates a similar idea of a type of hypertext which would allow a reader to "obtain a greater detail on a specific subject" which he calls "stretchtext."[36]

Since new media theory and criticism emerged in the early 1990s, endless texts have been written about interactivity, hypertext, virtual reality, cyberspace, cyborgs, and so on. But I have never seen anybody discuss "view control." And yet this is one of the most fundamental and radical new techniques for working with information and media available to us today. It is used daily by each of us numerous times. "View control," i.e. the abilities to switch between many different views and kinds of views of the same information is now implemented in multiple ways not only in word processors and email clients, but also in all "media processors" (i.e. media editing software): AutoCAD, Maya, After Effects, Final Cut, Photoshop, inDesign, and so on. For instance, in the case of 3D software, it can usually display the model in at least half a dozen different ways: in wireframe, fully rendered, etc. In the case of animation and visual effects software, since a typical project may contain dozens of separate objects each having dozens of parameters, it is often displayed in a way similar to how outline processors can show text. In other words, the user can switch between more and less information. You can choose to see only those parameters which you are working on right now. You can also zoom in and out of the composition. When you do this, parts of the composition do not simply get smaller or bigger – they show less or more information automatically. For instance, at a certain scale you may only see the names of different parameters; but when you zoom into the display, the program may also display the graphs which indicate how these parameters change over time.

Let us look at another example – Ted Nelson's concept of hypertext that he articulated in the early 1960s (independently but parallel to Engelbart).[37] In his 1965 article *A File Structure for the Complex, the Changing, and the Indeterminate*, Nelson discusses the limitations of books and other paper-based systems for organizing information and then introduces his new concept:

However, with the computer-driven display and mass memory, it has become possible to create *a new, readable medium*, for education and enjoyment, that will let the reader find his level, suit his taste, and find the parts that take on special meaning for him, as instruction and enjoyment.
Let me introduce the word "hypertext" to mean a body of written or pictorial material interconnected in such a complex way that it could not be conveniently be presented or represented on paper.[38] (Emphasis mine – L.M.)

"A new, readable medium" – these words make it clear that Nelson was not simply interested in "patching up" books and other paper documents. Instead, he wanted to create something distinctively new. But was not hypertext proposed by Nelson simply an extension of older textual practices such as exegesis (extensive interpretations of holy scriptures such as Bible, Talmud, Qur'ān), annotations, or footnotes? While such historical precedents for hypertext are often proposed, they mistakenly equate Nelson's proposal with a very limited form in which hypertext is experienced by most people today – i.e., World Wide Web. As Noah Wardrip-Fruin pointed out, The Web implemented only one of many types of structures proposed by Nelson already in 1965 – "chunk style" hypertext – static links that allow the user to jump from page to page."[39]

Following the Web implementation, most people today think of hypertext is a body of text connected through one-directional links. However, the terms "links" does not even appear in Nelson's original definition of hypertext. Instead, Nelson talks about new complex interconnectivity without specifying any particular mechanisms that can be employed to achieve it. A particular system proposed in Nelson's 1965 article is one way to implement such vision, but as his definition implicitly suggests, many others are also possible.

"What kind of structure are possible in hypertext?" asks Nelson in a research note from 1967. He answers his own question in a short but very suggestive answer: "Any."40 Nelson goes on to explain: "Ordinary text may be regarded as a special case – the simple and familiar case – of hypertext, just as three-dimensional space and the ordinary cube are the simple and familiar special cases of hyperspace and hypercube."41 (In 2007 Nelson has re-stated this idea in the following way: "'Hypertext'- - a word I coined long ago -- is not technology but potentially the fullest generalization of documents and literature."42)

If hypertex" does not simply means "links," it also does not only mean "text." Although in its later popular use the word "hypertext" came to refer to refer to linked text, as can see from the quote above, Nelson included "pictures" in his definition of hypertext.[43] And In the following paragraph, he introduces the terms *hyperfilm* and *hypermedia*:

Films, sound recordings, and video recordings are also linear strings, basically for mechanical reasons. But these, too, can now be arranged as non-linear systems – for instance, lattices – for educational purposes, or for display with different emphasis…The hyperfilm – a browsable or vari-sequenced movie – is only one of the possible hypermedia that require our attention."[44]

Where is hyperfim today, almost fifty years after Nelson has articulated this concept? If we understand hyperfilm in the same limited sense as hypertext is understood today – shots connected through links which a user can click on – it would seems that hyperfilm never fully took off. A number of early pioneering projects – *Aspen Movie Map* (Architecture Machine Group, 1978-79), *Earl King* and *Sonata* (Grahame Weinbren, 1983-85; 1991-1993), CD-ROMs by Bob Stein's Voyager Company, and *Wax: Or the Discovery of Television Among the Bees* (David Blair, 1993) – have not been followed up. Similarly, interactive movies and FMV-games created by video game industry in the first part of the 1990s soon feel out of favor, to be replaced by 3D games which offered more interactivity.[45] But if instead we think of hyperfilm in a broader sense as it was conceived by Nelson – any interactive structure for connecting video or film elements, with a traditional film being a special case – we realize that hyperfilm is much more common today than it may appear. Numerous Interactive Flash sites which use video, video clips with markers which allow a user jump to a particular point in a video (for instance, see videos on ted.com[46]), and database cinema[47] are just some of the examples of hyperfilm today.

Decades before hypertext and hypermedia became the common ways for interacting with information, Nelson understood well what these ideas meant for our well-established cultural practices and concepts. The announcement for his January 5, 1965 lecture at Vassar College talks about this in terms that are even more relevant today than they were then: "The philosophical consequences of all this are very grave.

Our concepts of 'reading', 'writing', and 'book' fall apart, and we are challenged to design 'hyperfiles' and write 'hypertext' that may have more teaching power than anything that could ever be printed on paper.[48]

These statements align Nelson's thinking and work with artists and theorists who similarly wanted to destabilize the conventions of cultural communication. Digital media scholars extensively discussed similar parallels between Nelson and French theorists writing the 1960s - Roland Barthes, Michel Foucault and Jacque Derrida.[49] Others have already pointed our close parallels between the thinking of Nelson and literary experiments taken place around the same time, such as works by Oulipo.[50] (We can also note the connection between Nelson's hypertext and the non-linear structure of the films of French filmmakers who set up to question the classical narrative style: *Hiroshima Mon Amou*r, *Last Year at Marienbad*, *Breathless* and others).

How far shall we take these parallels? In 1987 Jay Bolter and Michael Joyce wrote that hypertext could be seen as "a continuation of the modern 'tradition' of experimental literature in print" which includes "modernism, futurism, Dada surrealism, letterism, the nouveau roman, concrete poetry."[51] Refuting their claim, Espen J. Aarseth has argued that hyperext is not a modernist structure per ce, although it can support modernist poetics if the author desires this.[52] Who is right? Since this book argues that cultural software turned media into metamedia – a fundamentally new semiotic and technological system which includes most previous media techniques and aesthetics as its elements – I also think that hypertext is actually quite different from modernist literary tradition. I agree with Aarseth that hypertext is indeed much more general than any particular poetics such as modernist ones. Indeed, already in 1967 Nelson said that hypertext could support any structure of information including that of traditional texts – and presumably, this also includes different modernist poetics. (Importantly, this statement is echoed in Kay and Godberg's definition of computer as a "metamedium" whose content is "a wide range of already-existing and not-yet-invented media.")

What about the scholars who see the strong connections between the thinking of Nelson and modernism? Although Nelson says that hypertext can support any information structure and that that this information does not need to be limited to text, his examples and his style of writing show an unmistakable aesthetic sensibility – that of literary modernism. He clearly dislikes "ordinary text." The emphasis on *complexity and interconnectivity* and on *breaking up conventional units for organizing information* such as a page clearly aligns Nelson's proposal for hypertext with the early 20th century experimental literature – the inventions of Virginia Wolf, James Joyce, Surrealists, etc. This connection to literature is not accidental since Nelson's original motivation for his research which led to hypertext was to create a system for handling notes for literary manuscripts and manuscripts themselves. Nelson also already knew about the writings of William Burroughs. The very title of the article - *A File Structure for the Complex, the Changing, and the Indeterminate* – would make the perfect title for an early twentieth century avant-garde manifesto, as long as we substitute "file structure" with some "ism."

Nelson's modernist sensibility also shows itself in his thinking about new mediums that can be established with the help of a computer. However, his work should not be seen as a simple continuation of modernist tradition. Rather, both his and Kay's research represent the next stage of the avant-garde project. The early twentieth century avant-garde artists were primarily interested in questioning conventions of already established media such as photography, print, graphic design, cinema, and architecture. Thus, no matter how unconventional were the paintings that came out from Futurists, Orphism, Suprematism or De Stijl, their manifestos were still talking about them as paintings - rather than as a new media. In contrast, Nelson and Kay explicitly write about creating new media and not only changing the existing ones. Nelson: "With the computer-driven display and mass memory, it has become possible to create a new, readable medium." Kay and Goldberg: "It [computer text] need not be treated as a simulated paper book since this is a new medium with new properties."

Another key difference between how modernist artists and pioneers of cultural software approached the job of inventing new media and extending existing ones is captured by the title of Nelson's article I have been already quoted from above: "*A File Structure for the Complex, the Changing, and the Indeterminate*." Instead of a particular modernist "ism," we get a file structure. Cubism, Expressionism, Futurism, Orphism, Suprematism, Surrealism proposed new distinct systems for organizing information, with each systems fighting all others for the dominance in the cultural memesphere. In contrast, Bush, Licklider, Nelson, Engelbart, Kay, Negroponte, and their colleagues created meta-systems that can support many kinds of information structures. Kay called such a system "a first metamedium," Nelson referred to it as hypertext and hypermedia, Engelbart wrote about "automated external symbol manipulation" and "bootstraping," – but behind the differences in their visions lied the similar understanding on the radically new potential offered by computers for information manipulation. The hyphens "meta" and "hyper" used by Kay and Nelson were the appropriate characterizations for a system which was more than another new medium which could remediate other media in its particular ways. Instead, the new system would be capable of simulating all these media with all their remediation strategies – as well as supporting development of what Kay and Goldberg referred to as new "not-yet-invented media." And of course, this was not all. Equally important was the role of the interactivity. The new meta-systems proposed by Nelson, Kay and others were to be used interactively to support the processes of thinking, discovery, decision making, and creative expression. In contrast, the aesthetics created by modernist movements could be understood as "information formatting" systems – to be used for selecting and organizing information into fixed presentations that are then distributed to the users, not unlike PowerPoint slides. Finally, at least in Kay's and Nelson's vision, the task of defining of new information structures and media manipulation techniques – and, in fact, whole new media – was given to the user, rather than being the sole province of the designers. (As I will discuss below, this decision had far-reaching consequences for shaping contemporary culture. Once computers and programming were democratized enough, more cultural and creativity started to go into creating these new structures and techniques energy rather than using them to make "content.")

Today a typical article in computer science or information science will not be talking about inventing a "new medium" as a justification for research. Instead, it is likely to refer to previous work in some field or sub-field of computer science such as "knowledge discovery," "data mining," "semantic web," etc. It can also refer to existing social and and cultural practices and industries – for instance, "e-learning," "video game development," "collaborative tagging," or "massively distributed collaboration." In either case, the need for new research is justified by a reference to already established or, at least, popular practices – academic paradigms which have been funded, large-scale industries, and mainstream social routines which do threaten or question existing social order. This means that practically all of computer science research which deals with media – web technologies, media computing, hypermedia, human-computer interfaces, computer graphics, and so on – is oriented towards "mainstream" media usage.

In other words, either computer scientists are trying to make more efficient the technologies already used in media industries (video games, web search engines, film production, etc.) or they are inventing new technologies that are likely to be used by these industries in the future. The invention of new mediums for its own sake is not something which anybody is likely to pursue, or get funded. From this perspective, software industry and business in general is often more innovative than academic computer science. For instance, social media applications (Wikipedia, Flickr, YouTube, Facebook, del.is.ous, Digg, etc.) were not invented in the academy; nor were Hypercard, QuickTime, HTML, Photoshop, After Effects, Flash, or Google Earth. This was no different in previous decades. It is, therefore, not accidental that the careers of both Ted Nelson and Alan Kay were spend in the industry and not the academy: Kay worked for or was a fellow at Xerox PARC, Atari, Apple and Hewlett-Packard; Nelson was a consultant or a fellow at Bell Laboratories, Datapoint Corporation, Autodesk; both also were associated with Disney.

Why did Nelson and Kay found more support in industry than in academy for their quest to invent new computer media? And why does the industry (by which I simply mean any entity which creates the products which can be sold in large quantities, or monetized in other ways, regardless of whether this entity is a large multinational company or a small start-up) – is more interested in innovative media technologies, applications, and content than computer science? The systematic answer to this question will require its own investigation. Also, what kinds of innovations each modern institution can support changes over with time. But here is one brief answer. Modern business thrives on creating new markets, new products, and new product categories. Although the actual creating of such new markets and products is always risky, it is also very profitable. This was already the case in the previous decades when Nelson and Kay were supported by Xerox, Atari, Apple, Bell Labs, Disney, etc. In 2000s, following the globalization of the 1990s, all areas of business have embraced innovation to an unprecedented degree; this pace quickened around 2005 as the companies fully focused on competing for new consumers in China, India, and other formerly "emerging" economies. Around the same time, we see a similar increase in the number of innovative products in IT industry: open APIs of leading Web 2.0 sites, daily announcements of new webware services[53], locative media applications, new innovative products such as iPhone and Microsoft Surface, new paradigms in imaging such as HDR and non-destructive editing, the beginnings of a "long tail" for hardware, and so on.

As we can see from the examples we have analyzed, the aim of the inventors of computational media – Englebart, Nelson, Kay and people who worked with them – was not to simply create accurate simulations of physical media. Instead, in every case the goal was to create "a new medium with new properties" which would allow people to communicate, learn, and create in new ways. So while today the content of these new media may often look the same as with its predecessors, we should not be fooled by this similarity. The newness lies not in the content but in software tools used to create, edit, view, distribute and share this content. Therefore, rather than only looking at the "output" of software-based cultural practices, we need to consider software itself – since it allows people to work with media in of a number of historically unprecedented ways. So while on the level of appearance computational media indeed often remediate (i.e. represents) previous media, the software environment in which this media "lives" is very different.

Let me add to the examples above two more. One is Ivan Sutherland's *Sketchpad* (1962). Created by Sutherland as a part of his PhD thesis at MIT, Sketchpad deeply influenced all subsequent work in computational media (including that of Kay) not only because it was the first interactive media authoring program but also because it made it clear that computer simulations of physical media can add many exiting new properties to media being simulated. Sketchpad was the first software that allowed its users to interactively create and modify line drawings. As Noah Wardrip-Fruin points out, it "moved beyond paper by allowing the user to work at any of 2000 levels of magnification – enabling the creation of projects that, in physical media, would either be unwieldy large or require detail work at an impractically small size."[54] Sketchpad similarly redefined graphical elements of a design as objects which "can be manipulated, constrained, instantiated, represented ironically, copied, and recursively operated upon, even recursively merged.'[55] For instance, if the designer defined new graphical elements as instances of a master element and later made a change to the master, all these instances would also change automatically.

Another new property, which perhaps demonstrated most dramatically how computer-aided drafting and drawing were different from their physical counterparts, was Sketchpad's use of constraints. In Sutherland's own words, "The major feature which distinguishes a Sketchpad drawing from a paper and pencil drawing is the user's ability to specify to Sketchpad mathematical conditions on already drawn parts of his drawing which will be automatically satisfied by the computer to make the drawing take the exact shape desired."[56] For instance, if a user drew a few lines, and then gave the appropriate command, Sketchpad automatically moved these lines until they were parallel to each other. If a user gave a different command and selected a particular line, Sketchpad moved the lines in such a way so they would parallel to each other and perpendicular to the selected line.

Although we have not exhausted the list of new properties that Sutherland built into Sketchpad, it should be clear that this first interactive graphical editor was not only simulating existing media.

Appropriately, Sutherland's 1963 paper on Sketchpad repeatedly emphasizes the new graphical capacities of his system, marveling how it opens new fields of "graphical manipulation that has never been available before."[57] The very title given by Sutherland to his PhD thesis foregrounds the novelty of his work: *Sketchpad: A man-machine graphical communication system*. Rather than conceiving of Sketchpad as simply another media, Sutherland presents it as something else - a communication system between two entities: a human and an intelligent machine. Kay and Goldberg will later also foreground this communication dimension referring to it as "a two-way conversation" and calling the new "metamedium" "active."[58] (We can also think of Sketchpad as a practical demonstration of the idea of "man-machine symbiosis" by J.C. Licklider applied to image making and design.[59]

My last example comes from the software development that at first sight may appear to contradict my argument: paint software. Surely, the applications which simulate in detail the range of effects made possible with various physical brushes, paint knifes, canvases, and papers are driven by the desire to recreate the experience of working with in a existing medium rather than the desire to create a new one? Wrong. In 1997 an important computer graphics pioneer Alvy Ray Smith wrote a memo *Digital Paint Systems: Historical Overview*.[60] In this text Smith (who himself had background in art) makes an important distinction between *digital paint programs* and *digital paint systems*. In his definition, "A digital paint *program* does essentially no more than implement a digital simulation of classic painting with a brush on a canvas. A digital paint *system* will take the notion much farther, using the "simulation of painting" as a familiar metaphor to seduce the artist into the new digital, and perhaps forbidding, domain." (Emphasis in the original). According to Smith's history, most commercial painting applications, including Photoshop, fall into paint system category. His genealogy of paint systems begins with Richard Shoup's SuperPaint developed at Xerox PARC in 1972-1973.[61] While SuperPaint allowed the user to paint with a variety of brushes in different colors, it also included many techniques not possible with traditional painting or drawing tools. For instance, as described by Shoup in one of his articles on SuperPaint, "Objects or areas in the picture may be scaled up or down in size, moved, copied, overlaid, combined or changed in color, and saved on disk for future use or erased."[62]

Most important, however, was the ability to grab frames from video. Once loaded into the system, such a frame could be treated as any other images – that is, an artist could use all of SuperPaint drawing and manipulation tools, add text, combine it with other images etc. The system could also translate what it appeared on its canvas back into a video signal. Accordingly, Shoup is clear that his system was much more than a way to draw and paint with a computer. In a 1979 article, he refers to SuperPaint as a new "videographic medium."[63] In another article published a year later, he refines this claim: "From a larger perspective, we realized that the development of SuperPaint signaled the beginning of the synergy of two of the most powerful and pervasive technologies ever invented: digital computing and video or television."[64]

This statement is amazing perceptive. When Shoup was writing this in 1980, computer graphics were used in TV just a hand-full of times. And while in the next decade their use became more common, only in the middle of the 1990s the synergy Shoup predicted truly became visible. As we will see in the chapter on After Effects below, the result was a dramatic reconfiguration not just of visual languages of television but of all visual techniques invented by humans up to that point. In other words, what begun as a new "videographic medium" in 1973 had eventually changed all visual media.

But even if we forget about SuperPaint's revolutionary ability to combine graphics and video, and discount its new tools such resizing, moving, copying, etc., we are still dealing with *a new creative medium* (Smith's term). As Smith pointed out, this medium is the *digital frame buffer*,[65] a special kind of computer memory designed to hold images represented as an array of pixels (today a more common name is *graphics card*). An artist using a paint system is actually modifying pixel values in a frame buffer – regardless of what particular operation or tool she is employing at the moment. This opens up a door to all kinds of new image creation and modification operations, which follow different logic than physical painting. The telling examples of this can be found in paint system called simply Paint developed by Smith in 1975-1976. In Smith's own words, "Instead of just simulating painting a stroke of constant color, I extended the notion to mean 'perform any image manipulation you want under the pixels of the paintbrush."[66] Beginning with this conceptual generalization, Smith added a number of effects which sill used a paintbrush tool but actually no longer referred to painting in a physical world. For instance, in Paint "any image of any shape could be used as a brush." In another example, Smith added " 'not paint' that reversed the color of every pixel under the paintbrush to its color complement." He also defined 'smear paint' that averaged the colors in the neighborhood of each pixel under the brush and wrote the result back into the pixel." And so on. Thus, the instances where the paintbrush tool behaved more like a real physical paintbrush were just particular cases of a much larger universe of new behaviors made possible in a new medium.

The Permanent Extendibility

As we saw, Sutherland, Nelson, Englebart, Kay and other pioneers of computational media have added many previously non-existent properties to media they have simulated in a computer. The subsequent generations of computer scientists, hackers, and designers added many more properties – but this process is far from finished. And there is no logical or material reason why it will ever be finished. It is the "nature" of computational media that it is open-ended and new techniques are continuously being invented.

To add new properties to physical media requires modifying its physical substance. But since computational media exists as software, we can add new properties or even invent new types of media by simply changing existing or writing new software. Or by adding plug-ins and extensions, as programmers have been doing it with Photoshop and Firefox, respectively. Or by putting existing software together. (For instance, at the moment of this writing – 2006 - people are daily extending capacities of mapping media by creating software mashups which combining the services and data provided by Goggle Maps, Flickr, Amazon, other sites, and media uploaded by users.)

In short, *"new media" is "new" because new properties (i.e., new software techniques) can always be easily added to it.* Put differently, in industrial, i.e. mass-produced media technologies, "hardware" and "software" were one and the same thing. For example, the book pages were bound in a particular way that fixed the order of pages. The reader could not change nether this order nor the level of detail being displayed a la Englebart's "view control." Similarly, the film projector combined hardware and what we now call a "media player" software into a single machine. In the same way, the controls built into a twentieth-century mass-produced camera could not be modified at user's will. And although today the user of a digital camera similarly cannot easily modify the hardware of her camera, as soon as transfers the pictures into a computer she has access to endless number of controls and options for modifying her pictures via software.

In the nineteenth and twentieth century there were two types of situations when a normally fixed industrial media was more fluid. The first type of situation is when a new media was being first developed: for instance, the invention of photography in the 1820s-1840s. The second type of situation is when artists would systematically experiment with and "open up" already industrialized media – such as the experiments with film and video during the 1960s, which came to be called "Expanded Cinema."

What used to be separate moments of experimentations with media during the industrial era became the norm in a software society. In other words, computer legitimizes experimentation with media. Why this is so? What differentiates a modern digital computer from any other machine – including industrial media machines for capturing and playing media – is separation of hardware and software. It is because endless number of different programs performing different tasks can be written to run on the same type of machine, this machine – i.e. a digital computer - is used so widely today. Consequently, the constant invention of new and modification of existing media software is simply one example of this general principle. In other words, experimentation is a default feature of computational media. In its very structure it is "avant-garde" since it is constantly being extended and thus redefined.

If in modern culture "experimental" and "avant-garde" were opposed to normalized and stable, this opposition largely disappears in software culture. And the role of the media avant-garde is performed no longer by individual artists in their studios but by a variety of players, from very big to very small - from companies such as Microsoft, Adobe, and Apple to independent programmers, hackers, and designers.

But this process of continual invention of new algorithms does not just move in any direction. If we look at contemporary media software – CAD, computer drawing and painting, image editing, word processors – we will see that most of their fundamental principles were already developed by the generation of Sutherland and Kay. In fact the very first interactive graphical editor – Sketchpad – already contains most of the genes, so to speak, of contemporary graphics applications. As new techniques continue to be invented they are layered over the foundations that were gradually put in place by Sutherland, Englebart, Kay and others in the 1960s and 1970s.

Of course we not dealing here only with the history of ideas. Various social and economic factors – such as the dominance of the media software market by a handful of companies or the wide adoption of particular file formats –– also constrain possible directions of software evolution. Put differently, today software development is an industry and as such it is constantly balances between stability and innovation, standardization and exploration of new possibilities. But it is not just any industry. New programs can be written and existing programs can be extended and modified (if the source code is available) by anybody who has programming skills and access to a computer, a programming language and a compiler. In other words, today software is fundamentally "fabbable" in a way that physical industrially produced objects usually are not.

Although Turing and Von Neumann already formulated this fundamental extendibility of software in theory, its contemporary practice – hundreds of thousands of people daily involved in extending the capabilities of computational media - is a result of a long historical development. This development took us from the few early room-size computers, which were not easy to reprogram to a wide availability of cheap computers and programming tools decades later. This democratization of software development was at the core of Kay's vision. Kay was particularly concerned with how to structure programming tools in such a way that would make development of media software possible for ordinary users. For instance, at the end of the 1977 article I have been already extensively quoting, he and Goldberg write: "We must also provide enough already-written general tools so that a user need not start from scratch for most things she or he may wish to do."

Comparing the process of continuous media innovation via new software to history of earlier, pre-computational media reveals a new logic at work. According to a commonplace idea, when a new medium is invented, it first closely imitates already existing media before discovering its own language and aesthetics. Indeed, first printed bibles by Guttenberg closely imitated the look of the handwritten manuscripts; early films produced in the 1890s and 1900s mimicked the presentational format of theatre by positioning the actors on the invisible shallow stage and having them face the audience. Slowly printed books developed a different way of presenting information; similarly cinema also developed its own original concept of narrative space. Through repetitive shifts in points of view presented in subsequent shots, the viewers were placed inside this space – thus literally finding themselves inside the story.

Can this logic apply to the history of computer media? As theorized by Turing and Von Neuman, computer is a general-purpose simulation machine. This is its uniqueness and its difference from all other machines and previous media. This means that the idea that a new medium gradually finds its own language cannot apply to computer media. If this was true it would go against the very definition of a modern digital computer. This theoretical argument is supported by practice. The history of computer media so far has been not about arriving at some standardized language – the way this, for instance, happened with cinema – but rather about the gradual expansion of uses, techniques, and possibilities. Rather than arriving at a particular language, we are gradually discovering that the computer can speak more and more languages.

If we are to look more closely at the early history of computer media – for instance, the way we have been looking at Kay's ideas and work in this text –

we will discover another reason why the idea of a new medium gradually discovering its own language does not apply to computer media. The systematic practical work on making a computer simulate and extend existing media (Sutherland's Sketchpad, first interactive word processor developed by Englebart's group, etc.) came after computers were already put to multiple uses – performing different types of calculations, solving mathematical problems, controlling other machines in real time, running mathematical simulations, simulating some aspects of human intelligence, and so on. (We should also mention the work on SAGE by MIT Lincoln Laboratory which by the middle of the 1950s already established the idea of interactive communication between a human and a computer via a screen with a graphical display and a pointing device. In fact, Sutherland developed Sketchpad on TX-2 that was the new version of a larger computer MIT constructed for SAGE.) Therefore, when the generation of Sutherland, Nelson and Kay started to create "new media," they built it on top, so to speak, of what computers were already known to be capable off. Consequently they added new properties into physical media they were simulating right away. This can be very clearly seen in the case of Sketchpad. Understanding that one of the roles a computer can play is that of a problem solver, Sutherland built in a powerful new feature that never before existed in a graphical medium – satisfaction of constraints. To rephrase this example in more general terms, we can say that rather than moving from an imitation of older media to finding its own language, computational media was from the very beginning speaking a new language.

In other words, the pioneers of computational media did not have the goal of making the computer into a 'remediation machine" which would simply represent older media in new ways. Instead, knowing well new capabilities provided by digital computers, they set out to create fundamentally new kinds of media for expression and communication. These new media would use as their raw "content" the older media which already served humans well for hundreds and thousands of years – written language, sound, line drawings and design plans, and continuous tone images, i.e. paintings and photographs. But this does not compromise the newness of new media. For computational media uses these traditional human media simply as building blocks to create previously unimaginable representational and information structures, creative and thinking tools, and communication options.

Although Sutherland, Engelbart, Nelson, Kay, and others developed computational media on top of already existing developments in computational theory, programming languages, and computer engineering, it will be incorrect to conceive the history of such influences as only going in one direction – from already existing and more general computing principles to particular techniques of computational media. The inventors of computational media had to question many, if not most, already established ideas about computing. They have defined many new fundamental concepts and techniques of how both software and hardware thus making important contributions to hardware and software engineering. A good example is Kay's development of Smalltalk, which for the first time systematically established a paradigm of object-oriented programming. Kay's rationale to develop this new programming language was to give a unified appearance to all applications and the interface of PARC system and, even more importantly, to enable its users to quickly program their own media tools. (According to Kay, an object-oriented illustration program written in Smalltalk by a particularly talented 12-year old girl was only a page long.[67]) Subsequently object-oriented programming paradigm became very popular and object-oriented features have been added to most popular languages such as C.

Looking at the history of computer media and examining the thinking of its inventors makes it clear that we are dealing with the opposite of technological determinism. When Sutherland designed Sketchpad, Nelson conceived hypertext, Kay programmed a paint program, and so on, each new property of computer media had to be imagined, implemented, tested, and refined. In other words, these characteristics did not simply come as an inevitable result of a meeting between digital computers and modern media. Computational media had to be invented, step-by-step. And it was invented by people who were looking for inspiration in modern art, literature, cognitive and education psychology, and theory of media as much as technology. For example, Kay recalls that reading McLuhan's *Understanding Media* led him to a realization that computer can be a medium rather than only a tool.[68] Accordingly, the opening section of Kay and Goldberg' article is called "Humans and Media," and it does read like media theory. But this is not a typical theory which only describes the word as it currently exists. Like in Marxism, the analysis is used to create a plan for action for building a new world - in this case, enabling people to create new media.)

So far I have talked about the history of computational media as series of consecutive "additions." However this history is not only a process of accumulation of more and more options. Although in general we have more techniques at our disposal today when twenty of thirty years ago, it is also important to remember that many fundamentally new techniques which were conceived were never given commercial implementation. Or they were poorly implemented and did not become popular. Or they were not marketed properly. Sometimes the company making the software would go out of business. At other times the company that created the software was purchased by another company that "shelved" the software so it would not compete with its own products. And so on. In short, the reasons why many of new techniques did not become commonplace are multiple, and are not reducible to a single principle such as "the most easy to use techniques become most popular."

For instance, one of the ideas developed at PARC was "project views." Each view "holds all the tools and materials for a particular project and is automatically suspended when you leave it."[69] Thirty years later none of the popular operating system had ths feature.[70] The same holds true for the contemporary World Wide Web implementation of hyperlinks. The links on the Web are static and one-directional. Ted Nelson who is credited with inventing hypertext around 1964 conceived it from the beginning to have a variety of other link types. In fact, when Tim Berners-Lee submitted his paper about the Web to ACM Hypertext 1991 conference, his paper was only accepted for a poster session rather than the main conference program. The reviewers saw his system as being inferior to many other hypertext systems that were already developed in academic world over previous two decades.[71]

Computer as a Metamedium

As we have established, the development of computational media runs contrary to previous media history. But in a certain sense, the idea of a new media gradually discovering its own language actually does apply to the history of computational media after all. And just as it was the case with printed books and cinema, this process took a few decades. When first computers were built in the middle of the 1940s, they could not be used as media for cultural representation, expression and communication. Slowly, through the work of Sutherland, Englebart, Nelson, Papert and others in the 1960s, the ideas and techniques were developed which made computers into a "cultural machine." One could create and edit text, made drawings, move around a virtual object, etc. And finally, when Kay and his colleagues at PARC systematized and refined these techniques and put them under the umbrella of GUI that made computers accessible to multitudes, a digital computer finally was given its own language - in cultural terms. In short, only when a computer became a cultural medium – rather than only a versatile machine.

Or rather, it became something that no other media has been before. For what has emerged was not yet another media, but, as Kay and Goldberg insist in their article, something qualitatively different and historically unprecedented. To mark this difference, they introduce a new term – "metamedium."

This metamedium is unique in a number of different ways. One of them we already discussed in detail – it could represent most other media while augmenting them with many new properties. Kay and Goldberg also name other properties that are equally crucial. The new metamedium is "active – it can respond to queries and experiments – so that the messages may involve the learner in a two way conversation." For Kay who was strongly interested in children and learning, this property was particularly important since, as he puts it, it "has never been available before except through the medium of an individual teacher."[72] Further, the new metamedium can handle "virtually all of its owner's information-related needs." (I have already discussed the consequence of this property above.) It can also "serve as "a programming and problem solving tool," and "an interactive memory for the storage and manipulation of data."[73] But the property that is the most important from the point of view of media history is that *computer metamedium is simultaneously a set of different media and a system for generating new media tools and new types of media.* In other words, a computer can be used to create *new tools for working in the media it already provides as well as to develop new not-yet-invented media*.

Using the analogy with print literacy, Kay's motivates this property in this way: "The ability to 'read' a medium means you can *access* materials and tools generated by others. The ability to write in a medium means you can *generate* materials and tools for others. You must have both to be literate."[74]

Accordingly, Kay's key effort at PARC was the development of Smalltalk programming language. All media editing applications and GUI itself were written in Smalltalk. This made all the interfaces of all applications consistent facilitating quick learning of new programs. Even more importantly, according to Kay's vision, Smalltalk language would allow even the beginning users write their own tools and define their own media. In other words, all media editing applications, which would be provided with a computer, were to serve also as examples inspiring users to modify them and to write their own applications.

Accordingly, the large part of Kay and Goldberg's paper is devoted to description of software developed by the users of their system: "an animation system programmed by animators"; "a drawing and painting system programmed by a child," "a hospital simulation programmed by a decision-theorist," "an audio animation system programmed by musicians"; "a musical score capture system programmed by a musician"; "electronic circuit design by a high school student." As can be seen from this list that corresponds to the sequence of examples in the article, Kay and Goldberg deliberately juxtapose different types of users - professionals, high school students, and children – in order to show that everybody can develop new tools using Smalltalk programming environment.

The sequence of examples also strategically juxtaposes media simulations with other kinds of simulations in order to emphasize that simulation of media is only a particular case of computer's general ability to simulate all kinds of processes and systems. This juxtaposition of examples gives us an interesting way to think about computational media. Just as a scientist may use simulation to test different conditions and play different what/if scenarios, a designer, writer, a musician, a filmmaker, or an architect working with computer media can quickly "test" different creative directions in which the project can be developed as well as see how modifications of various "parameters" affect the project. The later is particularly easy today since the interfaces of most media editing software not only explicitly present these parameters but also simultaneously give the user the controls for their modification. For instance, when the Formatting Palette in Microsoft Word shows the font used by the currently selected text, it is displayed in column next to all other fonts available. Trying different font is as easy as scrolling down and selecting the name of a new font.

To give users the ability to write their own programs was a crucial part of Kay's vision for the new "metamedium" he was inventing at PARC. According to Noah Wardrip-Fruin, Englebart research program was focused on a similar goal: "Englebart envisioned users creating tools, sharing tools, and altering the tools of others."[75] Unfortunately, when in 1984 Apple shipped Macintosh, which was to become the first commercially successful personal computer modeled after PARC system, it did not have easy-to-use programming environment. HyperCard written for Macintosh in 1987 by Bill Atkinson (who was one of PARC alumni) gave users the ability to quickly create certain kinds of applications – but it did not have the versatility and breadth envisioned by Kay. Only recently, as the general computer literacy has widen and many scripting languages became available – Perl, PHP, Python, ActionScript, Vbscript, JavaScript, etc. – more people started to create their own tools by writing software. A good example of a contemporary programming environment, which is currently very popular among artists and designers and which, in my view, is close to Kay's vision is Processing.[76] Build on top of Java programming language, Processing features a simplified programming style and an extensive library of graphical and media functions. It can be used to develop complex media programs and also to quickly test ideas. Appropriately, the official name for Processing projects is sketches.[77] In the words of Processing initiators and main developers Ben Fry and Casey Reas, the language's focus "on the 'process' of creation rather than end results."[78] Another popular programming environment that similarly enables quick development of media projects is MAX/MSP and its successor PD developed by Miller Puckette.

Conclusion

At the end of the 1977 article that served as the basis for our discussion, he and Goldberg summarize their arguments in the phrase, which in my view is a best formulation we have so far of what computational media is artistically and culturally. They call computer *"a metamedium" whose content is "a wide range of already-existing and not-yet-invented media."* In another article published in 1984 Kay unfolds this definition. As a way of conclusion, I would like to quote this longer definition which is as accurate and inspiring today as it was when Kay wrote it:

> It [a computer] is a medium that can dynamically simulate the details of any other medium, including media that cannot exist physically. It is not a tool, though it can act like many tools. It is the first *metamedium*, and as such it has degrees of freedom for representation and expression never before encountered and as yet barely investigated.[79]

Chapter 2. Understanding Metamedia

Metamedia vs. Multimedia

“The first metamedium” envisioned by Kay in 1977 has gradually become a reality. Most already existing physical and electronic media were simulated as algorithms and a variety of new properties were added to them. A number of brand new media types were invented (for instance, navigable virtual space and hypermedia, pioneered by Ivan Sutherland and Ted Nelson, accordingly). New media-specific and general (i.e., media-agnostic) data management techniques were introduced; and, most importantly, by the middle of the 1990s computers became fast enough to “run” all these media. So what happens next? What is the next stage in the metamedium evolution? (I am using the word “stage” in a logical rather than a historical sense - although it is also true that the developments I will be now describing manifests themselves now more prominently than thirty years). This is something that, as far as I can see, the inventors of computational media – Sutherland, Nelson, Engelbradt, Kay and all the people who worked with them – did not write about. However, since they setup all the conditions for it, they are indirectly responsible for it.

I believe that we are now living through a second stage in the evolution of a computer metamedium, which follows the first stage of its invention and implementation. This new stage is about *media hybridization*. Once computer became a comfortable home for a large number of simulated and new media, it is only logical to expect that they would start creating hybrids. And this is exactly what is taking place at this new stage in media evolution. Both the simulated and new media types - text, hypertext, still photographs, digital video, 2D animation, 3D animation, navigable 3D spaces, maps, location information – now function as building blocks for new mediums. For instance, Google Earth combines aerial photography, satellite imagery, 3D computer graphics, still photography and other media to create a new hybrid representation which Google engineers called “3D interface to the planet.” A motion graphics sequence may combine content and techniques from different media such as live action video, 3D computer animation, 2D animation, painting and drawing. (Motion graphics are animated visuals that surround us every day; the examples are film and television titles, TV graphics, the graphics for mobile media content, and non-figurative parts of commercials and music videos.) A web site design may blends photos, typography, vector graphics, interactive elements, and Flash animation. Physical installations integrated into cultural and commercial spaces – such as Nobel Field at Nobel Peace Center in Oslo by Small Design, interactive store displays for Nokia and Diesel by Nanika, or the lobby at the 8th floor of Puerta America hotel in Madrid by Karen Finlay and Jason Bruges – combine animations, video, computer control, and various interfaces from sensors to touch to create interactive spatial media environments.[80]

It is important to make it clear that I am not talking about something that already has a name - “computer multimedia,” or simply “multimedia.” This term became popular in the 1990s to describe applications and electronic documents in which different media exist next to each other. Often these media types - which may include text, graphics, photographs, video, 3D scenes, and sound - are situated within what looks visually as a two-dimensional space. Thus a typical Web page is an example of multimedia; so is a typical PowerPoint presentation. Today, at least, this is the most common way of structuring multimedia documents. In fact, it is built-in into the workings of most multimedia authoring application such as presentation software or web design software. When a user of Word, PowerPoint or Dreamweaver creates a “new document,” she is presented with a white page ready to be typed into; other media types have to be “inserted” into this page via special commands.

But interfaces for creating multimedia do not necessary have to follow this convention. Another common paradigm for adding media together used in email and mobile devices is “attachments.” Thus, a user of a mobile phone which supports MMS (“Multimedia Media Surface”) can send text messages with attachments that can include picture, sound, and video files. Yet another paradigm persistent in digital culture – from Aspen Movie Map (1978) to VRML (1994-) to Second Life (2003-) – uses 3D space as the default platform with other media such as video attached to or directly inserted into this space.

“Multimedia” was an important term when interactive cultural applications, which featured a few media types, started to appear in numbers in the early 1990s. The development of these applications was facilitated by the introduction of the appropriate storage media, i.e. recordable CD-ROMs (i.e. CD-R) in 1991, computer architectures and filer formats designed to support multiple media file formats (QuickTime, 1991-) and multimedia authoring software (a version of Macromedia Director with Lingo scripting language was introduced in 1987). By the middle of the 1990s digital art exhibitions featured a variety of multimedia projects; digital art curricula begun to feature courses in “multimedia narrative”; and art museums started to publish multimedia CD-ROMs offering tours of their collections. In the second part of the decade multimedia took over the Web as more and more web sites begun to incorporate different types of media. By the end of the decade, “multimedia” became the default in interactive computer applications. Multimedia CD-ROMs, multimedia Web sites, interactive kisks, and multimedia communication via mobile devices became so commonplace and taken for granted that the term lost its relevance. So while today we daily encounter and use computer multimedia, we no longer wonder at the amazing ability of computers and computer-enabled consumer electronics devices to show multiple media at once.

Seen from the point of view of media history, “computer multimedia” is certainly a development of fundamental importance. Previously “multimedia documents” combining multiple media were static and/or not interactive: for instance, medieval illustrated manuscripts, sacred architecture, or twentieth century cinema, which combined live action, music, voice and titles. But co-existence of multiple media types within a single document or an application is only one of the new developments enabled by simulation of all these media in a computer. In putting forward the term *hybrid media* I want to draw attention to another, equally fundamental development that, in contrast to “multimedia,” so far did not receive a name.

It is possible to conceive of “multimedia” as a particular case of “hybrid media.” However, I prefer to think of them as overlapping but ultimately two different phenomena. While some of classical multimedia applications of the 1990s would qualify as media hybrids, most will not. Conversely, although media hybrids often feature content in different media, this is only one aspect of their make-up. So what is the difference between the two? In multimedia documents and interactive applications, content in multiple media appears *next to each other*. In a web page, images and video appear next to text; a blog post may similarly show text, followed by images and more text, a 3D world may contain a flat screen object used to display video. In contrast, in the case of media hybrids, interfaces, techniques, and ultimately the most fundamental assumptions of different media forms and traditions are brought together resulting in new species of media. To use a biological metaphor, we can say that media hybridity involves the coming together of the DNAs of different media to form new offsprings and species.

Put differently, media hybridity is a more fundamental reconfiguration of media universe than multimedia. In both cases we see “coming together” of multiple media. But, as I see it, *multimedia does not threaten the autonomy of different media*. *They retain their own languages, i.e. ways of organizing media data and accessing this data*. The typical use of multiple media on the Web or in PowerPoint presentations illustrates this well. Imagine a typical HTML page which consists from text and a video clip inserted somewhere on the page. Both text and video remain separate on every level. Their media languages do not spill into each other. Each media type continues to offer us its own interface. With text, we can scroll up and down; we can change its font, color and size, or number of columns, and so on. With video, we can play it, pause or rewind it, loop a part, and change sound volume. In this example, different media are positioned next to each other but their interfaces and techniques do not interact. This, for me, is a typical multimedia.

In contrast, in hybrid media the languages of previously distinct media come together. They exchange properties, create new structures, and interact on the deepest level. For instance, in motion graphics text takes on many properties which were previously unique to cinema, animation, or graphic design. To put this differently, while retaining its old typographic dimensions such as font size or line spacing, text also acquires cinematographic and computer animation dimensions. It can now move in a virtual space as any other 3D computer graphics object. Its proportions will change depending on what virtual lens the designer has selected. The individual letters, which make up a text string can be exploded into many small particles. As a word moves closer to us, it can appear out of focus; and so on. In short, in the process of hybridization, the language of typography does not stay "as is." Instead we end up with a new metalanguage that combines the techniques of all previously distinct languages, including that of typography.

Another way to distinguish between "multimedia" and "hybrid media" is by noting whether the conventional structure of media data is affected or not when different media types are combined. For example, when video appears in multimedia documents such as MMS messages, emails in HTML format, web pages or PowerPoint presentations, the structure of video data does not change in any way. Just as with twentieth century film and video technology, a digital video file is a sequence of individual frames, which have the same size, proportions, and color depth. Accordingly, the standards methods for interacting with this data also do not challenge our idea of what video is. Like with VCR media players of the twentieth century, when the user selects "play," the frames quickly replace each other producing the effect of motion. Video, in short, remains video.

This is typical of multimedia. An example of how same media structure can be reconfigured – the capacity that I take as one of the identifying features of media hybrids – is provided by *Invisible Shape of Things Pas*t, a digital "cultural heritage" project created by Berlin-based media design company Art+Com (1995-2007).[81] In this project a film clip becomes a solid object positioned in a virtual space. This object is made from individual frames situated behind each other in space. The angles between frames and the sizes of individual frames are determined by the parameters of the camera that originally shot the film. While we now interact with this film object as any other object in a 3d space, it is still possible to "see the movie," that is, access the film data in a conventional way. But even this operation of access has been rethought. When a user clicks on the front most frame, the subsequent frames positioned behind one another are quickly deleted. You simultaneously see the illusion of movement and the virtual object shrinking at the same time.

In summary, in this example of media restructuring the elements which make up the original film structure – individual frames – have been placed in a new configuration. The old structure has been mapped into a new structure. This new structure retains original data and their relationship – film frames organized into a sequence. But it also has new dimensions – size of frames and their angles.

I hope that this discussion makes it clear why hybrid media is not multimedia, and why we need this new term. The term "multimedia" captured the phenomenon of coming together of content of different media coming together – but not of their languages. Similarly, we cannot use another term that has been frequently used in discussions of computational media – "convergence." The dictionary meanings of "convergence" include "to reach the same point" and "to become gradually less different and eventually the same." But this is not what happens with media languages as they hybridize. Instead, they acquire new properties - becoming richer as a result. For instance, in motion graphics, text acquires the properties of computer animation and cinematography. In 3D computer graphics, rendering of 3D objects can take on all the techniques of painting. In virtual globes such as Google Earth and Microsoft Virtual Earth, representational possibilities and interfaces for working with maps, satellite imagery, 3D building and photographs are combined to create new richer hybrid representations and new richer interfaces.

In short, "softwarization" of previous media did not led to their convergence. Instead, after representational formats of older media types, the techniques for creating content in these media and the interfaces for accessing them were unbundled from their physical bases and translated into software, these elements start interacting producing new hybrids.

This, for me, is the essence of the new stage of a computer metamedium in which we are living today. *The previously unique properties and techniques of different media became the elements that can be combined together in previously impossible ways*.

Consequently, if in 1977 Kay and Goldberg speculated that the new computer metamedium would contain “a wide range of already existing and not-yet-invented media,” we now describe one of the key mechanisms responsible for the invention of these new media. This mechanism is hybridization. The techniques and representational formats of previous physical and electronic media forms, and the new information manipulation techniques and data formats unique to a computer are brought together in new combinations.

The Evolution of a Computer Metamedium

To continue with the biological metaphor I already invoked, imagine that the process of the computer metamedium development is like a biological evolution, and the new combinations of media elements are like new biological species.[82] Some of these combinations may appear only once or twice. For instance, a computer science paper may propose a new interface design; a designer may create a unique media hybrid for a particular design project; a film may combine media techniques in a novel way. Imagine that in each case, a new hybrid is never replicated. This happens quite often.

Thus, some hybrids that emerge in the course of media evolution will not be “selected” and will not “replicate.” Other hybrids, on the other hand, may “survive” and successfully “replicate.” (I am using quote marks to remind that for now I am using biological model only as a metaphor, and I am not making any claims that the actual mechanisms of media evolution are indeed like the mechanisms of biological evolution.) Eventually such successful hybrids may become the common conventions in media design; built-in features of media development/access applications; commonly used features in social media sites; widely used design patterns; and so on. In other words, they become new basic building blocks of the computer metamedium that can now be combined with other blocks.

An example of such a successful combination of media “genes” is an “image map” technique. This technique emerged in the middle of the 1990s and quickly become commonly used in numerous interactive media projects, games, and web sites. How does it work? A continuous raster image– photograph, a drawing, a white background, or any other part of a screen - is divided into a few invisible parts. When a user clicks inside one of the parts, this activates a hyperlink connected to this part.

As a hybrid, “image map” combines the technique of hyperlinking with all the techniques for creating and editing still images. Previously, hyperlinks were only attached to a word or a phrase of text and they were usually explicitly marked in some way to make them visible – for instance, by underlying them. When designers start attaching hyperlinks to parts of continuous images or whole surfaces and hiding them, a new “species” of media is born. As a new species, it defines new types of user behavior and it generates a new experience of media. Rather than immediately being presented with clearly marked, ready to be acted upon hyperlinks, a user now has to explore the screen, mousing over and clicking until she comes across a hyperlinked part. Rather than thinking of hyperlinks as discrete locations inside a “dead” screen, a user comes to think of the whole screen as a “live” interactive surface. Rather than imagining a hyperlink as something which is either present or absent, a user may now experience it as a continuous dimension, with some parts of a surface being “more” strongly hyperlinked than others.

As we will see in detail in the next chapter, the new language of visual design (graphic design, web design, motion graphics, design cinema and so on) that emerged in the second part of the 1990s offers a particularly striking example of media hybridization that follows its “softwarization.” Working in a software environment, a designer has access to any of the techniques of graphic design, typography, painting, cinematography, animation, computer animation, vector drawing, and 3D modeling. She also can use many new algorithmic techniques for generating new visuals (such as particle systems or procedural modeling) and transforming them (for instance, image processing), which do not have direct equivalent in physical or electronic media. All these techniques are easily available within a small number of media authoring programs (Photoshop, Illustrator, Flash, Maya, Final Cut, After Effects, etc.) and they can be easily combined within a single design. This new "media condition" is directly reflected in the new design language used today around the world. The new "global aesthetics" celebrates media hybridity and uses it to create emotional impacts, drive narratives, and create user experiences. In other words, it is all about hybridity. To put this differently, it is the ability to combine previously non-compatible techniques of different media which is the single common feature of millions of designs being created yearly by professionals and students alike and seen on the web, in print, on big and small screens, in built environments, and so on.

Like post-modernism of the 1980s and the web of the 1990s, the process of transfer from physical media to software has flattened history – in this case, the history of modern media. That is, while the historical origins of all building blocks that make up a computer metamedium – or a particular hybrid - maybe still important in some cases, they play no role in other cases. Clearly, for a media historian the historical origins of all techniques now available in media authoring software are important. They also may be made important for the media users - if a designer chooses to do this. For instance, in the logo sequence for DC Comics created by Imaginary Forces (2005), designers used exaggerated artifacts of print and film to evoke particular historical periods in the 1920s century. But when we consider the actual process of design – the ways in which designers work to go from a sketch or a storyboard or an idea in their head to a finished product – these historicals origins no longer matter. When a designer opens her computer and starts working, all this is incosequential. It does not matter if the technique was originally developed as a part of the simulation of physical or eletronic media, or not. Thus, a camera pan, an aerial perspective, splines and polygonal meshes, blur and sharpen filters, particle systems – all of these have equal status as the building blocks for new hybrids.

To summarize: thirty years after Kay and Goldberg predicted that the new computer metamedium would contain "a wide range of already existing and not-yet-invented media," we can see clearly that their prediction was correct. A computer metamedium has indeed been systematically expanding. However, this expansion should not be understood as simple addition of more and more new media types.

Following the first stage where most already existing media were simulated in software and a number of new computer techniques for generating and editing of media were invented – the stage that conceptually and practically has been largely completed by the late 1980s – we enter a new period governed by hybridization. The already simulated media start exchanging properties and techniques. As a result, the computer metamedium is becoming to be filled with endless new hybrids. In parallel, we do indeed see a continuous process of the invention of the *new* – but what is being invented are not whole new media types but rather new elements and constellations of elements which. As soon as they are invented, these new elements and constellations start interact with other already existing elements and constellations. Thus, the processes of invention and hybridization are closely linked and work together.

This, in my view, is the key mechanism responsible for evolution and expansion of the computer metamedium from the late 1980s until now – and right now I don't see any reason why it would change in the future. And while at the time when Kay and Goldberg were writing their article the process of hybridization just barely started – the first truly significant media hybrid being Aspen Movie Map created at MIT's Architecture Machine Group in 1978-1979 – today it is what media design is all about. Thus, from the point of view of today, *the computer metamedium is indeed an umbrella for many things – but rather than containing a set of separate media, it instead contains of a larger set of smaller building blocks.* These building blocks include algorithms for media creation and editing, interface metaphors, navigation techniques, physical interaction techniques, data formats, and so on. Over time, new elements are being invented and placed inside the computer metamedium's umbrella, so to speak. Periodically people figure out new ways in which some of the elements available can work together, producing new hybrids. Some of these hybrids may survive. Some may become new conventions which are so omnipresent that they are not perceived anymore as combinations of elements which can be taken apart. Still others are forgotten - only to be sometimes reinvented again later.

Clearly, the building blocks, which together form a computer metamedium, do not all have equal importance and equal "linking" possibilities. Some are used more frequently than others, entering in many more combinations. (For example, currently a virtual 3D camera is used much more widely than a "tag cloud.") In fact, some of the new elements may become so important and influential that it seems no longer appropriate to think of them as normal elements. Instead, they may be more appropriately called new "media dimensions" or "media platforms." 3D virtual space, World Wide Web and geo media (media which includes GPS coordinates) are three examples of such new media dimensions or platforms (popularized in the 1980s, 1990s, and 2000s, respectively). These media platforms do no simply mix with other elements enabling new hybrids – although they do it also. They fundamentally reconfigure how all media is understood and how it can be used. Thus, when we add spatial coordinates to media objects (geo media), place these objects within a networked environment (the web), or when we start using 3D space as a new platform to design these objects, the identity of what we think of as "media" changes in very fundamental ways. In fact, some would say that these changes have been as fundamental as the effects of media "softwarization" in the first place.

But is it? There is no easy way to resolve this question. Ultimately, it is a matter of perspective. If we look at contemporary visual and spatial aesthetics, in my view simulation of existing media in software and the subsequent period of media hybridization so far had much more substantial effects on these aesthetics than the web. Similarly, if we think about the histories of representation, human semiosis, and visual communication, I do think that the universal adoption of *software* throughout global culture industries is at least as importance as the invention of print, photography or cinema. But if we are to focus on social and political aspects of contemporary media culture and ignore the questions of how media looks and what it can represent – asking instead about who gets to create and distribute media, how people understand themselves and the world through media, etc. – we may want to put *networks* (be it web of the 1990s, social media of the 2000s, or whatever will come in the future) in the center of discussions.

And yet, it is important to remember that without software contemporary networks would not exist. Logically and practically, software lies underneath everything that comes later. If I disconnect my laptop from Wi-Fi right now, I can still continue using all applications on my laptop, including Word to write this sentence. I can also edit images and video, create a computer animation, design a fully functional web site, and compose blog posts. But if somebody disables software running the network, it will go dead.[83]

In other words, without the underlying software layers *The Internet Galaxy* (to quote the title of 2001 book by Manual Castells) would not exist. Software is what allows for media to exist on the web in the first place: images and video embedded in web pages and blogs, Flickr and YouTube, aerial photography and 3D buildings in Google Earth, etc. Similarly, the use of 3D virtual space as a platform for media design (which will be discussed in the next chapter) really means using a number of algorithms which control virtual camera, position the objects in space and calculate how they look in perspective, simulate the spatial diffusion of light, and so on.

Hybrids Everywhere

The examples of media hybrids are all around us: they can be found in user interfaces, web applications, visual design, interactive design, visual effects, locative media, digital art, interactive environments, and other areas of digital culture. Here are a few more examples that I have deliberately drawn from different areas. Created in 2005 by Stamen Design, Mappr! was one the first popular web mashups. It combined a geographic map and photos from the popular photo sharing site Flickr.[84] Using information enterted by Flickr uses, the application guessed geographical locations where photos where taken and displays them on the map. Since May 2007, Google Maps has offered Street Views that add panoramic photo-based views of city streets to other media types already used in Google Maps.[85] An interesting hybrid between photography and interfaces for space navigation, Street Views allow user can navigate though a space on a street level using the arrows that appear in the views.[86]

Japanese media artist Masaki Fujihata created a series of projects called *Field Studies*.[87] These projects place video recordings made in particular places within highly abstracted 3D virtual spaces representing these places. For instance, in *Alsase* (2000) Fujihata recorded a number of video interviews with the people living in and passing through the area around the border between France and Germany. Fujihata started to work on *Field*

Studies already in 1990s - a decade before the term "locative media" made its appearance. As cameras with built in GPS did not yet commercially exist at that time, the artist made a special video camera which captured geographical coordinates of each interview location along with the camera direction and angle while he was video taping the interview. In *Alsase* rectangles corresponding to video interviews were placed within an empty 3D space that contained only a handful of white lines corresponding to artist's movement through the geographical area of the project. The user of the installation could navigate through this space and when she would click on one of the rectangles, it would play a video interview. Each rectangle was positioned at a unique angle that corresponded to the angle of the hand-held video camera during the interview.

In my view, *Alsase* represents a particularly interesting media hybrid. It fuses photography (still images which appear inside rectangles), video documentary (video playing once a user clicks inside a rectangle), the locative media (the movement trajectories recorded by GPS) and 3D virtual space. In addition, *Alsace* uses a new media technique developed by Fujihata – the recording not just of the 2D location but also of the 3D orientation of the camera.

The result is a new way to represent collective experiences using 3D space as an overall coordinate system - rather than, for instance, a narrative or a database. At the same time, Fujihata found a simple and elegant way to render the subjective and unique nature of each video interview – situating each rectangle at a particular angle that actually reflects where camera was during the interview. Additionally, by defining 3D space as an empty void containing only trajectories of Fujihata's movement through the region, the artist introduced additional dimension of subjectivity. Even today after Google Earth has made 3D navigation of space containing photos and video a common experience, *Alsace* and other projects by Fujihata continue to stand out. They show that to create a new kind of representation it is not enough to simply “add” different media formats and techniques together. Rather, it may be necessary to systematically question the conventions of different media types that make up a hybrid, changing their structure in the process.

A well-known project I already mentioned - *Invisible Shape of Things Past* by Joachim Sauter and his company Art+Com - also uses 3D space as an umbrella that contains other media types. As I already discussed, the project maps historical film clips of Berlin recorded throughout the 20th century into new spatial forms that are integrated into a 3D navigable reconstruction of the city.[88] The forms are constructed by placing subsequent film frames one behind another. In addition to being able to move around the space and play the films, the user can mix and match parts of Berlin by choosing from a number of maps of Berlin, which represent city development in different periods of the twentieth century. Like Alsace, *Invisible Shape* combines a number of common media types while changing their structure. A video clip becomes a 3D object with a unique shape. Rather than representing a territory as it existed in a particular time, a map can mix parts of the city as they existed in different times.

Another pioneering media hybrid created by Sauter and Art+Com is *Interactive Generative Stage* (2002) – a virtual set whose parameters are interactively controlled by actors during the opera.[89] During the opera performance, computer reads the body movements and gestures of the actors and uses this information to control the generation of a virtual set projected on a screen behind the stage. The positions of a human body are mapped into various parameters of a virtual architecture such as the layout, texture, color, and light.

Sauter felt that it was important to preserve the constraints of the traditional opera format – actors foregrounded by lighting with the set behind them – while carefully adding new dimensions to it.[90]

Therefore, following the conventions of traditional opera the virtual set appears as a backdrop behind the actors – except now it not a static picture but a dynamic architectural construction that changes throughout the opera. As a result, the identity of a theatrical space changes from that of a backdrop to a main actor – and a very versatile actor at that since throughout the opera it adopts different personalities and continues to surprise the audience with new behaviors. This kind of fundamental redefinition of an element making a new hybrid is rare, but when a designer is able to achieve this, the result is very powerful.

Not every hybrid is necessary elegant, convincing, or forward-looking. Some of the interfaces of popular software applications for media creation and access look like the work of an aspiring DJ who mixes operations from the old interfaces of various media with new GUI principles in somewhat erratic and unpredictable ways. In my view, a striking example of such a problematic hybrid is the interface of Adobe Acrobat Reader. (Note that since the interfaces of all commercial software applications typically change from version to version, this example refers to the versions of Adobe Acrobat current at the time when this book was written.) Acrobat UI combines interfaces metaphors from variety of media traditions and technologies in a way that, at least to me, does not always seem to be logical. Within a single interface, we get 1) the interface elements from analog media recorders/players of the 20th century, i.e. VCR-style arrow buttons; 2) the interface element from image editing software, i.e. a zoom tool; 3) the interface elements which have strong association with print tradition - although they never existed in print (page icons also controlling the zoom factor); (4) the elements which have existed in books (the bookmarks window); (5) the standard elements of GUI such as search, filter, multiple windows. It seems that Acrobat designers wanted to give users variety of ways to navigate though documents. However, I personally find the co-presence of navigation techniques, which are normally used with media other than print confusing. For instance, given that Acrobat was designed to closely simulate the experience with print documents, it is not clear to me why I am asked to move through the pages by clicking on forward and backward arrow – an interface convention which is normally used for moving image media.

The hybrids also do not necessary have to involve a "deep" reconfiguration of previously separate media languages and/or the common structures of media objects – the way, for example, *The Invisible Shape* reconfigures the structure of a film object. Consider web mashups which "combine data elements from multiple sources, hiding this behind a simple unified graphical interface.[91] For example, a popular *flickrvision 3D* (David Troy, 2007) uses data provided by Flickr and the virtual globe from Poly 9 FreeEarth to create a mashup which continually shows the new photos uploaded to Flickr attached to the virtual globe in the places where photos' are taken. Another popular mashup LivePlazma (2005) uses Amazon services and data to offer a "discovery engine." When a user selects an actor, a movie director, a movie title, or a band name, LivePlazma generates an interactive map that shows related actors, movie directors, etc. related to the chosen item/name in terms of style, epoch, influences, popularity, and other dimensions.[92] Although LivePlazma suggests that the purpose of these maps is to lead you to discover the items that you are also likely to like (so you purchase them on amazon.com), these maps are valuable in themselves. They use newly available rich data about people's cultural preferences and behavior collected by Web 2.0 site such as Amazon to do something that was not possible until 2000s. That is, rather than mapping cultural relationships based on the ideas of a single person or a group of experts, they reveal how these relationships are understood by the actual cultural consumers.

Visually, many mashups may appear as typical multimedia documents – but they are more than that. As Wikipedia article on "mashup (web application hybrid)" explains, "A site that allows a user to embed a YouTube video for instance, is not a mashup site… the site should itself access 3rd party data using an API, and *process that data in some way to increase its value* to the site's users." (Emphasis mine – L.M.) Although the terms used by the authors - processing data to increase its value – may appear to be strictly and business like, they actually capture the difference between multimedia and hybrid media quite accurately. Paraphrasing the article's authors, we can say that in the case of a truly successful artistic hybrids such as *The invisible Shape* or *Alsase*, separate representational formats (video, photography, 2D map, 3D virtual globe) and media navigation techniques (playing a video, zooming into a 2D document, moving around a space using a virtual camera) are brought together in ways which increase the value offered by each of the media type used. However, in contrast to the web mashups which started to appear in mass in 2006 when Amazon, Flickr, Google and other major web companies offered public API (i.e., they made it possible for others to use their services and some of the data – for instance, using Google Maps as a part of a mashup), these projects also use their own data which the artists carefully selected or created themselves. As a result, the artists have much more control over the aesthetic experience and the "personality" projected by their works than an author of a mashup, which relies on both data and the interfaces provided by other companies. (I am not trying to criticize the web mashup phenomenon - I only want to suggest that if an artist goal is to come up with a really different representation model and a really different aesthetic experiences, choosing from the same set of web sources and data sources available to everybody else may be not the right solution. And the argument that web mashup author acts as a DJ who creates by mixing what already exists also does not work here – since a DJ has both more control over the parameters of the mix, and many more recordings to choose from.)

Representation and Interface

As we see, media hybrids can be structured in different ways and they can serve different functions. But behind this diversity we can find a smaller number of common goals shared by many if not most hybrids. Firstly, hybrids may combine and/or reconfigure familiar media formats and media interfaces to offer *new representations*. For instance, Google Earth and Microsoft Virtual Earth combine different media and interface techniques to provide more comprehensive information about places when either media can do by itself. The ambition behind *Alsase* and *Invisible Shape* is different – not to provide more information by combining existing media formats but rather to reconfigure these formats in order to create new representations of human collective and individual experiences which fuse objective and subjective dimensions. But in both cases, we can say that the overall goal is to represent something differently from the ways it was represented differently.

Secondly, the hybrids may aim to provide *new ways of navigation and working* with existing media formats – in other words, i.e. new interfaces and tools. For example, in UI of Acrobat Reader the interface techniques which previously belonged to specific physical, electronic, and digital media are combined to offer the user more ways to navigate and work with the electronic documents (i.e., PDF files). Mappr! exemplifies a different version of this strategy: using one media format as an interface to another. In this case, a map serves as an interface to a media collection, i.e. photos uploaded on Flickr. (It also exemplifies a new trend within metamedium evolution which has been becoming increasingly important from the early 2000s onwards: a joining between text, image, and video and spatial representations such as GPS coordinates, maps, and satellite photography – a trend which a German media historian and theorist Tristan Thielmann called "a spatial turn.") LivePlazma offers yet another version of this strategy: it uses techniques of interactive visualization to offer a new visual interface to the amazon.com's wealth of data.

You may notice that the distinction between a "representation" (or a "media format") and an "interface/tool" corresponds to the two fundamental components of all modern software: data structures and algorithms. This is not accidental. Each tool offered by a media authoring or media access application is essentially an algorithm that either processes in some way data in particular format or generates new data in this format. Thus, "working with media" using application software essentially means running different algorithms over the data.

However, the experience of users is actually different. Since today the majority of media application users don't know how to program, so they never encounter the data structures directly. Instead, they always work with data it a context of some application that comes with its interface and tools. Which means that *as experienced by a user of interactive application, "representation" consists from two interlinked parts: media structured in particular ways and the interfaces/tools provided to navigate and work with this media*. For example, a "3D virtual space" as it defined in 3D computer animation and CAD applications, computer games, and virtual globes is not only a set of coordinates that make up 3D objects and a perspective transformation but also a set of navigation methods – i.e. a virtual camera model. LivePlazma's interactive culture maps are not only relationships between the items on the map which we can see but also the tools provided to construct and navigate these maps. And so on.

PART 2: Software Takes Command

Chapter 3. After Effects, or How Cinema Became Design

First we shape our tools, thereafter they shape us - McLuhan, 1964.

Introduction

Having explored the logic of media hybridity using examples drawn from different areas of digital culture, I now want to test its true usefulness by looking at a single area in depth. This area is moving image design. A radically new visual language of moving images emerged during the period of 1993-1998 – which is the same period when filmmakers and designers started systematically using media authoring and editing software running on PCs. Today this language dominates our visual culture. We see it daily in commercials, music videos, motion graphics, TV graphics, design cinema, interactive interfaces of mobile phone and other devices, the web, etc. Below we will look at what I perceive to be some of its defining features: variable continuously changing forms, use of 3D space as a common platform for media design, and systematic integration of previously non-compatible media techniques.

How did this language come about? I believe that looking at software involved in the production of moving images goes a long way towards explaining why they now look the way they do. Without such analysis we will never be able to move beyond the commonplace generalities about contemporary culture – post-modern, global, remix, etc. – to actually describe the particular languages of different design areas, to understand the causes behind them and their evolution over time. In other words, I think that "software theory" which this book aims to define and put in practice is not a luxury but a necessity.

In this chapter I will analyze design and use of particular software application that played the key role in the emergence of this new visual language – After Effects. Introduced in 1993, After Effects was the first software designed to do animation, compositing, and special effects on MAC and PC. Its broad effect on moving image production can be compared to the effects of Photoshop and Illustrator on photography, illustration, and graphic design. As I will show, After Effects's UI and tools bring together fundamental techniques, working methods, and assumptions of previously separate fields of filmmaking, animation and graphic design. This hybrid production environment encapsulated in a single software application finds a direct reflection in the new visual language it enables - specifically, is focus on exploring aesthetic, narrative, and affective possibilities of hybridization.

The shift to software-based tools in the 1990s affected not only moving image culture but also all other areas of design. All of them adopted the same type of production workflow. (When the project is big and involves lots of people working on lots of files, the production workflow is called "pipeline"). A production process now typically involves either combining elements created in different software application, or moving the whole project from one application to the next to take advantage of their particular functions. And while each design field also uses its own specialized applications (for instance, web designers use Dreamweaver while architects use Revit), they also all use a number of common applications. They are Photoshop, Illustrator, Flash, Final Cut, After Effects, Maya, and a few others. (If you use open source software like Gimp and Cinepaint instead of these commercial applications, your list will be different but the principles would not change).

This adoption of this production environment that consists from a small number of compatible applications in all areas of creative industries had many fundamental effects. The professional boundaries between different design fields became less important. A single designer or a small studio may work on a music video today, a product design tomorrow, an architectural project or a web site design the day after, and so on. Another previously fundamental distinction - scale of a project – also now matters less, and sometimes not at al. Today we can expect to find exactly the same shapes and forms in very small objects (like jewelry), small and medium size objects (table ware, furniture), large buildings and even urban designs. (Zaha Hadid's lifestyle objects, furniure and architectual and urban design illustrate this well.)

While a comprehensive discussions of these and many other effects will take more than one book, in this chapter I wil analyze one them – the effect of which software-based workflow on contemporary visual aesthetics. As we will see, this workflow shapes contemporary visual culture in a number of ways. On the one hand, never before in the history of human visual communication have we witnessed such a variety of forms as today. On the other hand, exactly the same techniques, compositions and iconography can now appear in any media. To envoke the metaphor of biological evolution, we can say that despite seemingly infinite diversity of contemporary media, visual, and spatial "species," they all share some common DNAs. Besides these, many of these species also share a basic design principle: integration of previously non-compatible techniques of media design – a process which in the case of moving images I am going to name "deep remixability." Thus, a consideration of media authoring software and its usage in production would allow us to begin constructing a map of our current media/design universe, seeing how its species are related to each other and revealing the mechanisms behind their evolution.

The invisible revolution

During the heyday of post-modern debates, at least one critic in America noticed the connection between post-modern pastiche and computerization. In his book *After the Great Divide* (1986), Andreas Huyssen writes: "All modern and avantgardist techniques, forms and images are now stored for instant recall in the computerized memory banks of our culture. But the same memory also stores all of pre-modernist art as well as the genres, codes, and image worlds of popular cultures and modern mass culture." [93] His analysis is accurate – except that these "computerized memory banks" did not really became commonplace for another fifteen years. Only when the Web absorbed enough of the media archives it became this universal cultural memory bank accessible to all cultural producers. But even for the professionals, the ability to easily integrate multiple media sources within the same project – multiple layers of video, scanned still images, animation, graphics, and typography – only came towards the end of the 1990s.

In 1985 when Huyssen book was in preparation for publication I was working for one of the few computer animation companies in the world called *Digital Effects*.[94] Each computer animator had his own interactive graphics terminal that could show 3D models but only in wireframe and in monochrome; to see them fully rendered in color, we had to take turns as the company had only one color raster display which we all shared. The data was stored on bulky magnetic tapes about a feet in diameter; to find the data from an old job was a cumbersome process which involved locating the right tape in tape library, putting it on a tape drive and then searching for the right part of the tape. We did not had a color scanner, so getting "all modern and avantgardist techniques, forms and images" into the computer was far from trivial. And even if we had one, there was no way to store, recall and modify these images. The machine that could do that – Quantel Paintbox – cost over USD 160,000, which we could not afford.

And when in 1986 Quantel introduced Harry, the first commercial non-linear editing system which allowed for digital compositing of multiple layers of video and special effects, its cost similarly made it prohibitive for everybody expect network television stations and a few production houses. Harry could record only eighty seconds of broadcast quality video. In the realm of still images, things were not much better: for instance, digital still store Picturebox released by Quantel in 1990 could hold only 500 broadcast quality images and it cost was similarly very high.

In short, in the middle of the 1980s neither we nor other production companies had anything approachable "computerized memory banks" imagined by Huyssen. And of course, the same was true for the visual artists that were when associated with post-modernism and the ideas of pastiche, collage and appropriation. In 1986 BBC produced documentary *Painting with Light* for which half a dozen well-known painters including Richard Hamilton and David Hockney were invited to work with Quantel Paintbox. The resulting images were not so different from the normal paintings that these artists were producing without a computer. And while some artists were making references to "modern and avantgardist techniques, forms and images," these references were painted rather than being directly loaded from "computerized memory banks." Only about ten years later, when relatively inexpensive graphics workstations and personal computers running image editing, animation, compositing and illustration software became commonplace and affordable for freelance graphic designers, illustrators, and small post-production and animation studious, the situation described by Huyssen started to become a reality.

The results were dramatic. Within the space of less than five years, modern visual culture was fundamentally transformed. Visuals which previously were specific to differenly media - live action cinematography, graphics, still photography, animation, 3D computer animation, and typography

– started to be combined in numerous ways. By the end of the decade, the “pure” moving image media became an exception and hybrid media became the norm. However, in contrast to other computer revolutions such as the rise of World Wide Web around the same time, this revolution was not acknowledged by popular media or by cultural critics. What received attention were the developments that affected narrative filmmaking – the use of computer-produced special effects in Hollywood feature films or the inexpensive digital video and editing tools outside of it. But another process which happened on a larger scale - the transformation of the visual language used by all forms of moving images outside of narrative films – has not been critically analyzed. In fact, while the results of these transformations have become fully visible by about 1998, at the time of this writing (2008) I am not aware of a single theoretical article discussing them.

One of the reasons is that in this revolution no new media per se were created. Just as ten years ago, the designers were making still images and moving images. But the aesthetics of these images was now very different. In fact, it was so new that, in retrospect, the post-modern imagery of just ten years ago that at the time looked strikingly different now appears as a barely noticeable blip on the radar of cultural history.

Visual Hybridity

The new hybrid visual language of moving images emerged during the period of 1993-1998. Today it is everywhere. While narrative features still mostly use live-action footage, and videos shot by “consumers” and “prosumers” with commercial video cameras and cell phones are similarly usually left as is (at least, for now), almost everything else is hybrid. This includes commercials, music videos, TV graphics, film tites, dynamic menus, animated Flash web pages, graphics for mobile media content, and other types of animated, short non-narrative films and moving-image sequences being produced around the world today by media professionals, including companies, individual designers and artists, and students. I believe that at least 80 percent of such moving image sequences, animated interfaces and short films follow the aesthetics of hybridity.

Of course, I could have picked the different dates, for instance starting a few years earlier - but since After Effects software which will play the key role in my account was released in 1993, I decided to pick this year as my first date. And while my second date also could have been different, I believe that by 1998 the broad changes in the aesthetics of moving image became visible. If you want to quickly see this for yourself, simply compare demo reels from the same visual effects companies made in early 1990s and late 1990s (a number of them are available online – look for instance at the work of Pacific Data Images.[95]) In the work from the beginning of the decade, computer imagery in most cases appears by itself – that is, we see whole commercials and promotional videos done in 3D computer animation, and the novelty of this new media is foregrounded. By the end of the 1990s, computer animation becomes just one element integrated in the media mix that also includes live action, typography, and design.

Although these transformations happened only recently, the ubiquity of the new hybrid visual language today is such that it takes an effort to recall how different things looked before. Similarly, the changes in production processes and equipment that made this language possible also quickly fade from both the public and professional memory. As a way to quick evoke these changes as seen from the professional perspective, I am going to quote from 2004 interview with Mindi Lipschultz who has worked as an editor, producer and director in Los Angeles since 1979:

If you wanted to be more creative [in the 1980s], you couldn’t just add more software to your system. You had to spend hundreds of thousands of dollars and buy a paintbox. If you wanted to do something graphic – an open to a TV show with a lot of layers – you had to go to an editing house and spend over a thousand dollars an hour to do the exact same thing you do now by buying an inexpensive computer and several software programs. Now with Adobe After Effects and Photoshop, you can do everything in one sweep. You can edit, design, animate. You can do 3D or 2D all on your desktop computer at home or in a small office.[96]

In the 1989 former Soviet satellites of Central and Eastern Europe have peacefully liberated themselves from the Soviet Union. In the case of Czechoslovakia, this event came to be referred as Velvet Revolution – to contrast it to typical revolutions in modern history that were always accompanied by bloodshed. To emphasize the gradual, almost invisible pace of the transformations which occurred in moving image aesthetics between approximately 1993 and 1998, I am going to appropriate the term Velvet Revolution to refer to these transformations.

Although the Velvet Revolution I will be discussing involved many technological and social developments – hardware, software, production practices, new job titles and new professional fields – it is appropriate to highlight one software package as being in the center of the events. This software is After Effects. Introduced in 1993, After Effects was the first software designed to do animation, compositing, and special effects on the personal computer.[97] Its broad effect on moving image production can be compared to the effects of Photoshop and Illustrator on photography, illustration, and graphic design. Although today (2008) media design and post-production companies still continue to rely on more expensive "high-end" software such as Flame, Inferno or Paintbox that run on specialized graphics workstations, because of its affordability and length of time on the market After Effects is the most popular and well-known application in this area. Consequently, After Effects will be given a privileged role in this account as both the symbol and the key material foundation which made Velvet Revolution in moving image culture possible – even though today other programs in the similar price category such as Apple's Motion, Autodesk's Combustion, and Adobe's Flash have challenged After Effects dominance.

Finally, before proceeding I should explain my use of examples. The visual language I am analyzing is all around us today (this may explain why academics have remained blind to it). After globalization, this language is spoken by communication professionals in dozens of countries around the world. You can see for yourself all the examples of various aesthetics I will be mentioning below by simply watching television and paying attention to graphics, or going to a club to see a VJ performance, or visiting the web sites of motion graphics designers and visual effects companies, or opening any book on contemporary design. Nevertheless, below I have included titles of particular projects so the reader can see exactly what I am referring to.[98] But since my goal is to describe the new cultural language that by now has become practically universal, I want to emphasize that each of these examples can be substituted by numerous others.

Examples

The use of After Effects is closely identified with a particular type of moving images which became commonplace to a large part because of this software – "motion graphics." Concisely defined by 2003 Matt Frantz in his Master Thesis as "designed non-narrative, non-figurative based visuals that change over time,"[99] motion graphics include film and television titles, TV graphics, dynamic menus, the graphics for mobile media content, and other animated sequences. Typically motion graphics appear as parts of longer pieces: commercials, music videos, training videos, narrative and documentary films, interactive projects. Or at least, this is how it was in 1993; since that time the boundary between motion graphics and everything else has progressively become harder to define. Thus, in 2008 version of the Wikipedia article about motion graphics, the authors already wrote that "The term "motion graphics" has the potential for less ambiguity than the use of the term film to describe moving pictures in the 21st century."[100])

One of the key identifying features of motion graphics in the 1990s that used to clearly separate it from other forms of moving image was a central role played by dynamic typography. The term "motion graphics" has been used at least since 1960 when a pioneer of computer filmmaking John Whitney named his new company Motion Graphics. However until Velvet Revolution only a handful of people and companies have systematically explored the art of animated typography: Norman McLaren, Saul Blass, Pablo Ferro, R/Greenberg, and a few others.[101] But in the middle of the 1990s moving image sequences or short films dominated by moving animated type and abstract graphical elements rather than by live action started to be produced in large numbers. The material cause for motion graphics take off? After Effects and other related software running on PCs or relatively inexpensive graphics workstations became affordable to smaller design, visual effects, post-production houses, and soon individual designers. Almost overnight, the term "motion graphics" became well known. (As Wikipedia article about this term points out, "The term "Motion Graphics" was popularized by Trish and Chris Meyer's book about the use of Adobe After Effects titled "Creating Motion Graphics."[102]) The five hundred year old Guttenberg universe came into motion.

Along with typography, the whole language of twentieth graphical century design was "imported" into moving image design. This development did not receive a name of its own which would become as popular, but it is obviously at least as important. (Although the term "design cinema" has been used, it never achieved anything comparable to the popularity of "motion graphics.") So while motion graphics were for years limited to film titles and therefore focused on typography, today the term "motion graphics" is often used to moving image sequences that are dominated by typography and/or design. But we should recall that while in the twentieth century typography was indeed often used in combination with other design elements, for five hundred years it formed its own word. Therefore I think it is important to consider the two kinds of "import" operations that took place during Velvet Revolution – typography and twentieth century graphic design – as two distinct historical developments.

While motion graphics definitely exemplify the changes that took place during Velvet Revolution, these changes are more broad. Simply put, the result of Velvet Revolution is *a new hybrid visual language of moving images in general*. This language is not confined to particular media forms. And while today it manifests itself most clearly in non-narrative forms, it is also often present in narrative and figurative sequences and films.

Here are a few examples. A music video may use live action while also employing typography and a variety of transitions done with computer graphics (video for "Go" by Common, directed by Convert/MK12/Kanye West, 2005). Another music video may embed the singer within an animated painterly space (video for Sheryl Crow's "Good Is Good," directed by Psyop, 2005). A short film may mix typography, stylized 3D graphics, moving design elements, and video (*Itsu* for Plaid, directed by the Pleix collective, 2002[103]). (Sometimes, a term "design cinema" I already mentioned is used to differentiate such short independent films organized around design, typography and computer animation rather than live action from similar "motion graphics" works produced for commercial clients.)

In some cases, the juxtaposition of different media is clearly visible (video for "Don't Panic" by Coldplay, 2001; main title for the television show *The Inside* by Imaginary Forces, 2005). In other cases, a sequence may move between different media so quickly that the shifts are barely noticeable (GMC Denali "Holes" commercial by Imaginary Forces, 2005). Yet in other cases, a commercial or a movie title may feature continuous action shot on video or film, with the image periodically changing from a more natural to a highly stylized look.

Such media hybridity does not necessary manifest itself in a collage-like aesthetics that foregrounds the juxtaposition of different media and different media techniques. As a very different example of what media hybridity can result in, consider a more subtle aesthetics well captured by the name of the software that to a large extent made the hybrid visual language possible: After Effects. This name has anticipated the changes in visual effects which only took place a number of years later. in the 1990s computers were used to create highly spectacular special effects or "invisible effects,"[104] toward the end of that decade we see something else emerging: a new visual aesthetics that goes "beyond effects." In this aesthetics, the whole project—whether a music video, a TV commercial, a short film, or a large segment of a feature film—displays a hyper-real look in which the enhancement of live-action material is not completely invisible but at the same time it does not call attention to itself the way special effects usually tended to do (examples: Reebok I-Pump "Basketball Black" commercial and *The Legend of Zorro* main title, both by Imaginary Forces, 2005).

Although the particular aesthetic solutions vary from one video to the next and from one designer to another, they all share the same logic: the simultaneous appearance of multiple media within the same frame. Whether these media are openly juxtaposed or almost seamlessly blended together is less important than the fact of this co-presence itself. (Again, note that each of the examples above can be substituted by numerous others.)

Hybrid visual language is also now common to a large proportion of short “experimental” and “independent” (i.e., not commissioned by commercial clients) videos being produced for media festivals, the web, mobile media devices, and other distribution platforms.[105] Many visuals created by VJs and “live cinema” artists are also hybrid, combining video, layers of 2D imagery, animation, and abstract imagery generated in real time.[106] And as the animations of artists Jeremy Blake, Ann Lislegaard, and Takeshi Murata that I will discuss below demonstrate, at least some of the works created explicitly for art-world distribution similarly choose to use the same language of hybridity.

Today, narrative features rarely mix different graphical styles within the same frame. However, a gradually growing number of films do feature the kind of highly stylized aesthetics that would have previously been identified with illustration rather than filmmaking: Larry and Andy Wachowski’s *Matrix* series (1999–2003), Robert Rodriguez’s *Sin City* (2005), and Zack Snyder’s *300* (2007). These feature films are a part of a growing trend to shoot a large portion of the film using a “digital backlot” (green screen).[107] Consequently, most or all shots in such films are created by composing the footage of actors with computer-generated sets and other visuals.

These films do not juxtapose their different media in as dramatic a way as what we commonly see in motion graphics. Nor do they strive for the seamless integration of CGI (computer-generated imagery) visuals and live action that characterized the earlier special-effects features of the 1990s, such as *Terminator 2* (1991) and *Titanic* (1997) (both by James Cameron). Instead, they explore the space in between juxtaposition and complete integration.

Matrix, *Sin City*, *300*, and other films shot on a digital backlot combine multiple media to create a new stylized aesthetics that cannot be reduced to the already familiar look of live-action cinematography or

3D computer animation. Such films display exactly the same logic as short motion graphics works, which at first sight might appear to be very different. This logic is also the same one we observe in the creation of new hybrids in biology. That is, the result of the hybridization process is not simply a mechanical sum of the previously existing parts but a new "species"—a new kind of visual aesthetics that did not exist previously.

Media Hybridity in Sodium Fox and Untitled (Pink Dot)

Blake's *Sodium Fox* and Murata's *Untitled (Pink Dot)* (both 2005) offer excellent examples of the new hybrid visual language that currently dominates moving-image culture. Among the many well-known artists working with moving images today, Blake was the earliest and most successful in developing his own style of hybrid media. His video *Sodium Fox* is a sophisticated blend of drawings, paintings, 2D animation, photography, and effects available in software. Using a strategy commonly employed by artists in relation to commercial media in the twentieth century, Blake slows down the fast-paced rhythm of motion graphics as they are usually practiced today. However, despite the seemingly slow pace of his film, it is as informationally dense as the most frantically changing motion graphics such as one may find in clubs, music videos, television station IDs, and so on. *Sodium Fox* creates this density by exploring in an original way the basic feature of the software-based production environment in general and programs such as After Effects in particular, namely, the construction of an image from potentially numerous layers. Of course, traditional cel animation as practiced in the twentieth century also involved building up an image from a number of superimposed transparent cells, with each one containing some of the elements that together make up the whole image. For instance, one cel could contain a face, another lips, a third hair, yet another a car, and so on.

With computer software, however, designers can precisely control the transparency of each layer; they can also add different visual effects, such as blur, between layers. As a result, rather than creating a visual narrative based on the motion of visual elements through space (as was common in twentieth-century animation, both commercial and experimental), designers now have many new ways to create visual changes. Exploring these possibilities, Blake crafts his own visual language in which visual elements positioned on different layers are continuously and gradually "written over" each other. If we connect this new language to twentieth-century cinema rather than to cel animation, we can say that rather than fading in a new frame as a whole, Blake continuously fades in separate parts of an image. The result is an aesthetics that balances visual continuity with a constant rhythm of visual rewriting, erasing, and gradual superimposition.

Like *Sodium Fox,* Murata's *Untitled (Pink Dot)* also develops its own language within the general paradigm of media hybridity. Murata creates a pulsating and breathing image that has a distinctly biological feel to it. In the last decade, many designers and artists have used biologically inspired algorithms and techniques to create animal-like movements in their generative animations and interactives. However, in the case of *Untitled (Pink Dot)*, the image as a whole seems to come to life.

To create this pulsating, breathing-like rhythm, Murata transforms live-action footage (scenes from one of the *Rambo* films) into a flow of abstract color patches (sometimes they look like oversize pixels, and at other times they may be taken for artifacts of heavy image compression). But this transformation never settles into a final state. Instead, Murata constantly adjusts its degree. (In terms of the interfaces of media software, this would correspond to animating a setting of a filter or an effect). One moment we see almost unprocessed live imagery; the next moment it becomes a completely abstract pattern; the following moment parts of the live image again become visible, and so on.

In *Untitled (Pink Dot)* the general condition of media hybridity is realized as a permanent metamorphosis. True, we still see some echoes of movement through space, which was the core method of pre-digital animation. (Here this is the movement of the figures in the live footage from *Rambo.*) But now the real change that matters is the one between different media aesthetics: between the texture of a film and the pulsating abstract patterns of flowing patches of color, between the original "liveness" of human figures in action as captured on film and the highly exaggerated artificial liveness they generate when processed by a machine.

Visually, *Untitled (Pink Dot)* and *Sodium Fox* do not have much in common*.* However, as we can see, both films share the same strategy: creating a visual narrative through continuous transformations of image layers, as opposed to discrete movements of graphical marks or characters, which was common to both the classic commercial animation of Disney and the experimental classics of Norman McLaren, Oskar Fischinger, and others. Although we can assume that neither Blake nor Murata has aimed to achieve this consciously, in different ways each artist stages for us the key technical and conceptual change that defines the new era of media hybridity. Media software allows the designer to combine any number of visual elements regardless of their original media and to control each element in the process. This basic ability can be explored through numerous visual aesthetics. The films of Blake and Murata, with their different temporal rhythms and different logics of media combination, exemplify this diversity. Blake layers over various still graphics, text, animation, and effects, dissolving elements in and out. Murata processes live footage to create a constant image flow in which the two layers—live footage and its processed result—seem to constantly push each other out.

Deep Remixability

I believe that "media hybridity" constitutes a new fundamental stage in the history of media. It manifests itself in different areas of culture and not only moving images – although the later does offer a particularly striking example of this new cultural logic at work. Here media authoring software environment became a kind of Petri dish where the techniques and tools of computer animation, live cinematography, graphic design, 2D animation, typography, painting and drawing can interact, generating new hybrids. And as the examples above demonstrate, the result of this process of hybridity are new aesthetics and new "media species" which cannot be reduced to the sum of media that went into them.

Can we understand the new hybrid language of moving image as a type of remix? I believe so—if we make one crucial distinction. Typical remix combines content within the same media or content from different media. For instance, a music remix may combine music elements from any number of artists; anime music videos may combine parts of anime films and music taken from a music video. Professionally produced motion graphics and other moving-image projects also routinely mix together content in the same media and/or from different media. For example, in the beginning of the "Go" music video, the video rapidly switches between live-action footage of a room and a 3D model of the same room. Later, the live-action shots also incorporate a computer-generated plant and a still photographic image of mountain landscape. Shots of a female dancer are combined with elaborate animated typography. The human characters are transformed into abstract animated patterns. And so on.

Such remixes of content from different media are definitely common today in moving-image culture. In fact, I begun discussing the new visual language by pointing out that in the case of short forms such remixes now constitute a rule rather than exception. But this type of remix is only one aspect of “hybrid revolution” For me, its essence lies in something else. Let’s call it “deep remixability.” For *what gets remixed today is not only content from different media but also their fundamental techniques, working methods, and ways of representation and expression*. United within the common software environment, the languages of cinematography, animation, computer animation, special effects, graphic design, and typography have come to form a new *metalanguage*. A work produced in this new metalanguage can use all the techniques, or any subset of these techniques, that were previously unique to these different media.

We may think of this new metalanguage of moving images as a large library of all previously known techniques for creating and modifying moving images. A designer of moving images selects techniques from this library and combines them in a single sequence or a single frame. But this clear picture is deceptive. How exactly she combines these techniques? When you remix content, it is easy to imagine: different texts, audio samples, visual elements, or data streams are positioned side by side. Imagine a typical 20th century collage except that it is now moves and changes over time. But how do you remix the techniques?

In the cases of hybrid media interfaces which we have already analyzed (such as Acrobat interface), “remix” means simple combination. Different techniques literally appear next to each in application UI. Thus, in Acrobat, a forward and backward buttons, a zoom button, a “find” tool and others are positioned one after another on a toolbar above the open document. Other techniques appear as tools listed in vertical pull-down menus: spell, search, email, print, and so on. We find the same principles in interfaces of all media authoring and access applications. The techniques borrowed from various media and the new born-digital techniques are presented side-by-side using tool bars, pull-down menus, toolboxes and other conventions of UI.

Such “addition of techniques” which exist in a single space side by side without any deep interactions is also indirectly present in remixes of content well familiar to us, be it fashion designs, architecture, collages, or motion graphics. Consider a hypothetical example of a visual design which combines drawn elements, photos, and 3D computer graphics forms. Each of these visual elements is a result of the use of particular media techniques of drawing, photography and computer graphics. Thus, while we may refer to such cultural objects as remixes of content, we are also justified in thinking about them as remixes of techniques. This applies equally well to pre-digital design when designer would use separate physical tools or machines, and to contemporary software-driven design where she has access to all these tools in a few compatible software applications.

As long as the pieces of content, interface buttons, or techniques are simply added rather than integrated, we don’t need a special term such as “deep remix. This, for me, is still “remix” the way this term is used commonly used. But in the case of moving image aesthetics we also encounter something like. Rather than a simple addition, we also find interactions between previously separate techniques of cell animation, cinematography, 3D animation, design, and so on – interactions which were unthinkable before. (The same argument can be made in relation to other types of cultural objects and experiences created with media authoring software such as visual designs and music.)

I believe that this is something that neither pioneers of computer media of the 1960s-1970s nor the designers of first media authoring applications that started to appear in the 1980s were planning. However, once all media techniques met within the same software environment—and this was gradually accomplished throughout the 1990s—they started interacting in ways that could never have been predicted or even imagined previously.

For instance, while particular media techniques continue to be used in relation to their original media, they can also be applied to other media. Here are a few examples of this "crossover effect." Type is choreographed to move in 3D space; motion blur is applied to 3D computer graphics; algorithmically generated fields of particles are blended with live-action footage to give it an enhanced look; a virtual camera is made to move around a virtual space filled with 2D drawings. In each of these examples, the technique that was originally associated with a particular medium—cinema, cel animation, photorealistic computer graphics, typography, graphic design—is now applied to a different media type. Today a typical short film or a sequence may combine many of such pairings within the same frame. The result is a hybrid, intricate, complex, and rich media language – or rather, numerous languages that share the logic of deep remixabilty.

In fact, such interactions among virtualized media techniques define the aesthetics of contemporary moving image culture. This is why I have decided to introduce a special term—*deep remixability*. I wanted to differentiate more complex forms of interactions between techniques (such as cross-over) from the simple remix (i.e. addition) of media content and media techniques with which we are all familiar, be it music remixes, anime video remixes, 1980s postmodern art and architecture, and so on.

For concrete examples of the "crossover effect," which exemplifies deep remixability, we can return to the same "Go" video and look at it again, but now from a new perspective. Previously I have pointed the ways in which this video – typical for short format moving images works today – combines visual elements of different media types: live action video, still photographs, procedurally generated elements, typography, etc. However, exactly the same shots also contain rich examples of the interactions between techniques, which are only possible in a software-driven design environment.

As the video begins, a structure made up from perpendicular monochrome blocks and panels simultaneously rapidly grows in space and rotates to settle into a position which allows us to recognize it as a room (00:07 – 00:11). As this move is being completed, the room is transformed from an abstract geometric structure into a photorealistically rendered once: furniture pops in, wood texture roils over the floor plane, and a photograph of a mountain view fills a window. Although such different styles of CG rendering have been available in animation software since the 1980s, a particular way in which this video opens with a visually striking abstract monochrome 3D structure is a clear example of deep remixability. When in the middle of the 1990s graphic designers started to use computer animation software, they brought their training, techniques and sensibilities to computer animation that until that time was used in the service of photorealism. The strong diagonal compositions, the deliberate flat rendering, and the choice of colors in the opening of "Go" video subordinates CG photorealistic techniques to a visual discipline specific to modern graphic design. The animated 3D structure references suprematism of Malevich and Lissitzky that played a key role in shaping the grammar of modern design – and which, in our example, has become a conceptual "filter" which transformed CG field.

After a momentary stop to let us take in the room, which is now largely completed, a camera suddenly rotates 90^0 (00:15 – 00:17). This physically impossible camera move is another example of deep remixability. While animation software implements the standard grammar of 20th century cinematography – a pan, a zoon, a dolly, etc. – the software, of course, does not have the limitations of a physical world. Consequently a camera can move in arbitrary direction, follow any imaginable curve and do this at any speed. Such impossible camera moves become standard tools of contemporary media design and 21st century cinematography, appearing with increased frequency in feature films since the middle of 2000s. Just as Photoshop filters which can be applied to any visual composition, virtual camera moves can also be superimposed, so to speak, on any visual scene regardless of whether it was constructed in 3D, procedurally generated, captured on video, photographed, or drawn - or, as in the example of the room from "Go" video, is a combination of these different media.

Playing video forward (00:15 – 00:22), we notice yet another previously impossible interaction between media techniques. The interaction in question is a lens reflection, which is slowly moving across the whole scene. Originally an artifact of a camera technology, lens reflection was turned into a filter – i.e., a technique which can now be "drawn" over any image constructed with all other techniques available to a designer. (This important type of software techniques which originated as artifacts of physical or electronic media technologies will be discussed in more details in the concluding section of this chapter.) If you wanted more proof that we are dealing here with a visual technique, note that this "lens reflection" is moving while the camera remains perfectly still (00:17 – 00:22) – a logical impossibility, which is sacrificed in favor of a more dynamic visual experience.

Metalanguage and Metamedium

I referred to the new language of moving imaging as a "metalanguage." What does that mean? What is the connection between this term as I am using here and a "computer metamedium"?

The acceleration of the speed of social, technological and cultural changes in the second part of the 20th century has led to the frequent use of meta-, hyper-, and super- in cultural theory and criticism. From 1960s Superstudio (a conceptual architectural group), Ted Nelson's Hypermedia and Alan Kay's metamedium to more recent Supermodernism and

Hypermodernity[108], these terms may be read as attempts to capture the feeling that we have passed a point of singularity and are now moving at warp speed. Like the cosmonauts of the 1960s observing the Earth from the orbits of their spaceships and seeing it for the first time as a single object, we are looking down at human history from a new higher orbit while moving forward. This connotation seems to fit Alan Kay's conceptual and practical redefinition of a digital computer as a "metamedium" which contains most of the existing medium technologies and techniques and also allows invention of many new ones.

While the term "metalanguage" has precise meanings in logic, linguistics and computing, here I am using in a sense similar to Alan Kay's use of "meta" in "computer metamedium." Normally a "metalanguage" refers to a separate formal system for describing mediums or cultural languages - the way a grammar describes how a particular natural language works. But this not how Kay uses "meta" in "metamedium." As he uses it, it stands for gathering / including / collecting – in short, bringing previously separate things together.

Let us imagine this computer metamedium as a large and continuously expanding set of resources. It includes all media creation and manipulation techniques, interaction techniques and data formats available to programmers and designers in the current historical moment. Everything from sort and search algorithms and pull-down menus to hair and water rendering techniques, video games AI, and multi-touch interface methods – its all there.

If we look at how these resources are used in different cultural areas to create particular kinds of contents and experiences, we will see that each of them only uses a subset of these resources. For example, today Graphical interfaces which come with all popular computer operating systems (Windows, Lunix, Mac OS) use static icons. In contrast, in some consumer electronics interfaces (such as certain mobile phones) all icons are animated loops.

Moreover, the use of a subset of all existing elements is not random but follows particular conventions. Some elements always go together. In other cases, the use of one element means that we are unlikely to find some other element. In other words, not only different forms of digital media use different subsets from a complete set which makes a computer metamedium but this use also follows distinct patterns.

If you notice a parallel with what cultural critics usually call an "artistic language," a "style," or a "genre," you are right. Any single work of literature or works of a particular author or a literary movement uses only some of the all existing literary techniques and this use follows some patterns. The same goes for cinema, music and all other recognized cultural forms. This allows us to talk about a style of a particular novel or a film, or a style of an author as a whole, or a style of a whole artistic school. (Film scholars David Bordwell and Kristin Thompson call this a "stylistic system" which they define as a "patterned and significant use of techniques." They divide these techniques into four categories: mise-en-scene, cinematography, editing, and sound.[109]) When a whole cultural field can be divided into a small number of distinct groups of works with each group sharing some patterns, we usually talk about "genres." For instance, theoreticians of Ancient Greek theatre distinguished between comedies and tragedies and prescribed the rules each genre should follow, while today companies use automatic software to classify blogs into different genres.

If by medium we mean a set of standard technological resources, be it a physical stage or a film camera, lights and film stock, we can see that each medium usually supports multiple artistic languages / styles / genres. For example, a medium of 20th century filmmaking supported Russian

Montage of the 1920s, Italian Neorealism of the 1940s, French New Wave of the 1960s, Hong Kong fantasy kong-fu films of the 1980s, Chinese, "fifth-generation" films of the 1980s-1990s, etc.

Similarly, a computer metamedium can support multiple cultural or artistic metalanguages. In other words, in the theoretical scheme I am proposing, there is only one metamedium - but many metalanguages.

So what is a metalanguage? If we define artistic language as a patterned use of a selected number of a subset of the techniques available in a given medium[110], *a metalanguage is a patterned use of a subset of all the techniques available in a computer metamedium*. But not just any subset. It only makes sense to talk about a metalangauge (as opposed to a language) if the techniques it uses come from previously distinct cultural languages. As an example, consider a metalanguage of popular commercial virtual globes (Google Earth and Microsoft Virtual Earth). These applications 1) systematically combine different types of media formats and media navigation techniques that previously existed separately; and 2) these combinations follow common patterns. Another example will be a metalanguage common to many graphical user interface uses (recall my analysis of Acrobat interface which combines metaphors drawn from different media traditions).

Since moving images today systematically combine techniques of different visual media which almost never met until middle of the 1990s, we are justified in using the term "metalanguage" in their case. Visual design today has its own metalanguge, which is a subset of the metalanguage of moving images. The reason is that a designer of moving images has access to all the techniques of a visual designer plus additional techniques since she is working with additional dimension of time. These two metalanguages also largely overlap in patterns that are common to them – but there are also some important differences. For instance, today moving image works often feature a continuous movement through a 3D space that may contain various 2D elements. In contrast, visual designs for print, web, products or other applications are usually 2D – they assemble elements either over an imaginary flat surface. (I think that the main reason for this insistence on flatness is that these designs often exist next to large blocks of text that already exist in 2D.)

Layers, Transparency, Compositing

So far I have focused on describing the aesthetics of moving images that emerged from the Velvet Revolution. While continuing this investigation, we will now pay more attention to the analysis of new software production environment that made this aesthetics possible. The following sections of this chapter will look at the tools offered by After Effects and other media authoring applications, their user interfaces, and the ways these applications are used together in production (i.e., design workflow). Rather than discussing all of tools and interface features, I will highlight a number of fundamental assumptions behind them – ways of understanding of what a moving image project is, which, as we will see, are quite different from how it was understood during the 20th century.

Probably the most dramatic among the changes that took place during 1993-1998 was the new ability to combine together multiple levels of imagery with varying degree of transparency via digital compositing. If you compare a typical music video or a TV advertising spot circa 1986 with their counterparts circa 1996, the differences are striking. (The same holds for other areas of visual design.) As I already noted, in 1986 "computerized memory banks" were very limited in their storage capacity and prohibitively expensive, and therefore designers could not quickly and easily cut and paste multiple image sources. But even when they would assemble multiple visual references, a designer only could place them next to, or on top of each other. She could not modulate these juxtapositions by precisely adjusting transparency levels of different images. Instead, she had to resort to the same photocollage techniques popularized in the 1920s. In other words, the lack of transparency restricted the number of different images sources that can be integrated within a single composition without it starting to look like certain photomontages or photocollages of John Heartfield, Hannah Hoch, or Robert Rauschenberg – a mosaic of fragments without any strong dominant.[111]

Compositing also made trivial another operation that was very cumbersome previously. Until the 1990s, different media types such as hand-drawn animation, lens-based recordings, i.e. film and video, and typography practically never appeared within the same frame. Instead, animated commercials, publicity shorts, industrial films, and some feature and experimental films that did include multiple media usually placed them in separate shots. A few directors have managed to build whole aesthetic systems out of such temporal juxtapositions – most notably, Jean-Luc Godard. In his 1960s films such as *Week End* (1967) Godard cut bold typographic compositions in between live action creating what can be called a "media montage" (as opposed to a montage of live action shots as dveloped by the Russians in the 1920s.) In the same 1960s pioneering motion graphics designer Pablo Ferro who has appropriately called his company Frame Imagery created promotional shorts and TV graphics that played on juxtapositions of different media replacing each other in a rapid succession.[112] In a number of Ferro's spots, static images of different letterforms, line drawings, original hand painted artwork, photographs, very short clips from newsreels, and other visuals would come after another with machine gun speed.

Within cinema, the superimposition of different media within the same frame were usually limited to the two media placed on top of each other in a standardized manner – i.e., static letters appearing on top of still or moving lens-based images in feature film titles. Both Ferro and another motion graphics pioneer Saul Bass have created a few remarkable title sequences where visual elements of different origin were systematically overlaid together – such as the opening for Hitchcock's *Vertigo* designed by Bass (1958). But I think it is fare to say that such complex juxtapositions of media within the same frame (rather than in edited sequence) were rare exceptions in the overwise "unimedia" universe where filmed images appeared in feature films and hand drawn images appeared in animated films. The only twentieth century feature film director I know of who has build his unique aesthetics by systematically combining different media within the same frame is Czech Karel Zeman. Thus, a typical shot by Zeman may contain filmed human figures, an old engraving used for background, and a miniature model.[113]

The achievements of these directors and designers are particularly remarkable given the difficulty of combing different media within the same frame during film era. To do this required utilizing the services of a special effects departments or separate companies which used optical printers. The techniques that were cheap and more accessible such as double exposure were limited in their precision. So while a designer of static images could at least cut and paste multiple elements within the same composition to create a photomontage, to create the equivalent effect with moving images was far from trivial.

To put this in more general terms, we can say that before computerization of the 1990s, the designer's capacities to access, manipulate, remix, and filter visual information, whether still of moving, were quite restricted. In fact, they were practically the same as hundred years earlier - regardless of whether filmmakers and designers used in-camera effects, optical printing, or video keying. In retrospect, we can see they were at odds with the flexibility, speed, and precision of data manipulation already available to most other professional fields which by that time were computerized – sciences, engineering, accounting, management, etc. Therefore it was only a matter of time before all image media would be turned into digital data and illustrators, graphic designers, animators, film editors, video editors, and motion graphics designers start manipulating them via software instead of their traditional tools. But this is only obvious today – after Velvet Revolution has taken place.

In 1985 Jeff Stein directed a music video for the new wave band *Cars*. This video had a big attempt in the design world, and MTV gave it the first prize in its first annual music awards.[114] Stein managed to create a surreal world in which a video cutout of the singing head of the band member was animated over different video backgrounds. In other words, Stein took the aesthetics of animated cartoons – 2D animated characters superimposed over a 2D background – and recreated it using video imagery. In addition, simple computer animated elements were also added in some shots to enhance the surreal effect. This was shocking because nobody ever saw such juxtapositions this before. Suddenly, modernist photomontage came alive. But ten years later, such moving video collages not only became commonplace but they also became more complex, more layered, and more subtle. Instead of two or three, a composition could now feature hundreds and even thousands of layers. And each layer could have its own level of transparency.

In short, *digital compositing* now allowed the designers to easily *mix any number of visual elements regardless of the media in which they originated and to control each element in the process.* Here we can make an analogy between multitrack audio recording and digital compositing. In multitrack recording, each sound track can be manipulated individually to produce the desired result. Similarly, in digital compositing each visual element can be independently modulated in a variety of ways: resized, recolored, animated, etc. Just as the music artist can focus on a particular track while muting all other tracks, a designer often turns of all visual tracks except the one she is currently adjusting. Similarly, both a music artist and a designer can at any time substitute one element of a composition by another, delete any elements, and add new ones. Most importantly, just as multitrack recording redefined the sound of popular music from the 1970s onward, once digital compositing became widely available during the 1990s, it fundamentally changed the visual aesthetics of most moving images forms.

This brief discussion only scratched the surface of my subject in this section, i.e. layers and transparency. For instance, I have not analyzed the actual techniques of digital compositing and the fundamental concept of an alpha channel which deserves a separate and detailed treatment. I have also did not go into the possible media histories leading to digital compositing, nor its relationship to optical printing, video keying and video effects technology of the 1980s. These histories and relationships were discussed in "Compositing" chapter in *The Language of New Media* but from a different perspective than the one used here. At that time (1999) I was looking at compositing from the point of view of the questions of cinematic realism, practices of montage, and the construction of special effects in feature films. Today, however, it is clear to me that in addition to disrupting the regime of cinematic realism in favor of other visual aesthetics, compositing also had another, even more fundamental effect.

By the end of the 1990s digital compositing has become the basic operation used in creating *all* forms of moving images, and not only big budget features. So while it was originally developed as a technique for special effects in the 1970s and early 1980s[115], compositing had a much broader effect on contemporary visual and media cultures beyond special effects. Compositing played the key part in turning digital computer into a kind of experimental lab (or a Petri dish) where different media can meet and there their aesthetics and techniques can be combined to create new species. In short, digital compositing was essential in enabling the development of a new hybrid visual language of moving images which we see everywhere today.

Defined at first as a particular digital technique designed to integrate two particular media of live action film and computer graphics in special effects sequences, composing later become a "universal media integrator." And although compositing was originally created to support the aesthetics of cinematic realism, over time it actually had an opposite effect. Rather that forcing different media to fuse seamlessly, compositing led to the flourishing of numerous media hybrids where the juxtapositions between live and algorithmically generated, two dimensional and three dimensional, raster and vector are made deliberately visible rather than being hidden.

From "Time-based" to a "Composition-based"

My thesis about media hybridity applies both to the cultural objects and the software used to create them. Just as the moving image media made by designers today mix formats, assumptions, and techniques of different media, the toolboxes and interfaces of the software they use are also remixes. Let us see use again After Effects as the case study to see how its interface remixes previously distinct working methods of different disciplines.

When moving image designers started to use compositing / animation software such as After Effects, its interface encouraged them to think about moving images in a fundamentally new way. Film and video editing systems and their computer simulations that came to be known as non-linear editors (today exemplified by Avid and Final Cut[116]) have conceptualized a media project as a sequence of shots organized in time. Consequently, while NLE (the standard abbreviation for non-linear video editing software) gave the editor many tools for adjusting the edits, they took for granted the constant of a film language that came from its industrial organization – that all frames have the same size and aspect ratio. This is an example of a larger trend. During the first stage of the development of cultural software, its pioneers were exploring the new possibilities of a computer metamedium going in any direction they were interested, since commercial use (with a notable exception of CAD) was not yet an option. However, beginning in the 1980s new generation of companies – Aldus, Autodesk, Macromedia, Adobe, and others - started to produce GUI-based software media authoring software aimed at particular industries: TV production, graphic design, animation, etc. As a result, many of the workflow principles, interface conventions and constraints of media technologies standard in these industries were already using were methodically re-created in software – even though software medium itself has no such limitations. NLE software is a case in point. In contrast, from the beginning After Effects interface put forward a new concept of moving image – as a composition organized both in time and 2D space.

The center of this interface is a Composition window conceptualized as a large canvas that can contain visual elements of arbitrary sizes and proportions. When I first started using After Effects soon after it came out, I remember feeling shocked that software did not automatically resized the graphics I dragged into Composition window to make them fit the overall frame. The fundamental assumption of cinema that accompanied it throughout its whole history – that film consists from many frames which all have the same size and aspect ratio – was gone.

In film and video editing paradigms of the twentieth century, the minimal unit on which the editor works on is a frame. She can change the length of an edit, adjusting where one film or video segment ends and another begins, but she cannot directly modify the contents of a frame. The frame functions as a kind of “black box” that cannot be “opened.” This was the job for special effects departments and companies. But in After Effects interface, the basic unit is not a frame but a visual element placed in the Composition window. Each element can be individually accessed, manipulated and animated. In other words, each element is conceptualized as an independent object. Consequently, a media composition is understood as a set of independent objects that can change over time. The very word “composition” is important in this context as it references 2D media (drawing, painting, photography, design) rather than filmmaking – i.e. space as opposed to time.

Where does After Effects interface came? Given that this software is commonly used to create animated graphics and visual effects, it is not surprising that its interface elements can be traced to three separate fields: animation, graphic design, and special effects. And because these elements are integrated in intricate ways to offer the user a new experience that can’t be simply reduced to a sum of working methods already available in separate fields, it makes sense to think of After Effects UI as an example of “deep remixability.”

In a 20th century cell animation practice, an animator places a number of transparent cells on top of each other. Each cell contains a different drawing – for instance, a body of a character on one cell, the head on another cell, eyes on the third cell. Because the cells are transparent, the drawings get automatically “composited” into a single composition. While After Effects interface does not use the metaphor of a stack of transparent cells directly, it is based on the same principle. Each element in the Composition window is assigned a “virtual depth” relative to all other elements. Together all elements form a virtual stack. At any time, the designer can change the relative position of an element within the stack, delete it, or add new elements.

We can also see a connection between After Effects interface and stop motion that was another popular twentieth century animation technique – stop motion. To create stop motion shot, puppets or any other 3D objects are positioned in front of a film camera and manually animated one step at a time. For instance, an animator may be adjusting a head of character, progressively moving its head left to right in small discrete steps. After every step, the animator exposes one frame of film, then makes another adjustment, exposes another frame, and so on. (The twentieth century animators and filmmakers who used this technique with great inventiveness include Ladyslaw Starewicz, Oscar Fishinger, Aleksander Ptushko, Jiri Rmka, Jan Svankmajer, and Brothers Quay.)

Just as both cell and stop-motion animation practices, After Effects does not make any assumptions about the size or positions of individual elements. Instead of dealing with standardized units of time – i.e. film frames containing fixed visual content - a designer now works with separate visual elements. An element can be a digital video frame, a line of type, an arbitrary geometric shape, etc. The finished work is the result of a particular arrangement of these elements in space and time. Consequently, a designer who uses After Effects can be compared to a choreographer who creates a dance by "animating" the bodies of dancers - specifying their entry and exit points, trajectories through space of the stage, and the movements of their bodies. (In this respect it is relevant that although After Effects interface did not evoke this reference, another equally important 1990s software that was commonly used to author multimedia - Macromedia Director - did explicitly the metaphor of the theatre stage in its UI.)

While we can link After Effects interface to traditional animation methods as used by commercial animation studios, the working method put forward by software is more close to graphic design. In commercial animation studio of the twentieth century all elements – drawings, sets, characters, etc. – were prepared beforehand. The filming itself was a mechanical process. Of course, we can find exceptions to this industrial-like separation of labor in experimental animation practice where a film was usually produced by one person. This allowed a filmmaker to invent a film as he went along, rather than having to plan everything beforehand. A classical example of this is Oscar Fishinger*'s Motion Painting 1* created in 1949. Fishinger made this eleven-minute film *Motion Painting 1* by continuously modifying a painting and exposing film one frame at a time after each modification. This process took 9 months. Because Fishinger was shooting on film, he had to wait a long time before seeing the results of his work. As the historian of abstract animation William Moritz writes, "Fischinger painted every day for over five months without being able to see how it was coming out on film, since he wanted to keep all the conditions, including film stock, absolutely consistent in order to avoid unexpected variations in quality of image."[117] In other words, in the case of this project by Fischinger, creating animation and seeing the result were even more separated than in a commercial animation process.

In contrast, a graphic designer works in true "in real time." As the designer introduces new elements, adjusts their locations, colors and other properties, tries different images, changes the size of the type, and so on, she can immediately see the result of her work.[118] After Effects adopts this working method by making Composition window the center of its interface. Like a traditional designer, After Effects user interactively arranges the elements in this window and can immediately see the result. In short, After Effects interface makes filmmaking into a design process, and a film is re-conceptualized as a graphic design that can change over time.

As we saw when we looked of the history of cultural software, when physical or electronic media are simulated in a computer, we do not simply end with the same media as before. By adding new properties and working methods, computer simulation fundamentally changes the identity of a given media. For example, in the case of "electronic paper" such as a Word document or a PDF file, we can do many things which were not possible with ordinary paper: zoom in and out of the document, search for a particular phrase, change fonts and line spacing, etc. Similarly, current (2008) online interactive maps services provided by Mapquest, Yahoo, and Google augment the traditional paper map in multiple and amazing ways.

A significant proportion of contemporary software for creating, editing, and interacting with media was developed in this way. Already existing media technology were simulated in a computer and augmented with new properties. But if we consider media authoring software such as Maya (3D modeling and computer animation) or After Effects (motion graphics, compositing and visual effects), we encounter a different logic. These software applications do not simulate any single physical media that existed previously. Rather, they borrow from a number of different media combining and mixing their working methods and specific techniques. (And, of course, they also add new capabilities specific to computer – for instance, the ability to automatically calculate the intermediate values between a number of keyframes.) For example, 3D modeling software mixes form making techniques which previously were "hardwired" to different physical media: the ability to change the curvature of a rounded form as though it is made from clay, the ability to build a complex 3D object from simple geometric primitives the way buildings were constructed from identical rectangular bricks, cylindrical columns, pillars, etc.

Similarly, as we saw, After Effects original interface, toolkit, and workflow drew on the techniques of animation and the techniques of graphic design. (We can also find traces of filmmaking and 3D computer graphics.) But the result is not simply a mechanical sum of all elements that came from earlier media. Rather, as software remixes the techniques and working methods of various media they simulate, the result are new interfaces, tools and workflow with their own distinct logic. In the case of After Effects, the working method that it puts forward is neither animation, nor graphic design, nor cinematography, even though it draws from all these fields. It is a new way to make moving image media. Similarly, the visual language of media produced with this and similar software is also different from the languages of moving images that existed previously.

Consequently, the Velvet Revolution unleashed by After Effects and other software did not simply made more commonplace the animated graphics artists and designers – John and James Whitney, Norman McLaren, Saul Bass, Robert Abel, Harry Marks, R/Greenberg, and others – were creating previously using stop motion animation, optical printing, video effects hardware of the 1980s, and other custom techniques and technologies. Instead, it led to the emergence of numerous new visual aesthetics that did not exist before. And if the common feature of these aesthetics is "deep remixability," it is not hard to see that it mirrors "deep remixabilty" in After Effects UI.

Three-dimensional Space as a New Platform for Media

As I was researching what the users and industry reviewers has been saying about After Effects, I came across a somewhat condescending characterization of this software as "Photoshop with keyframes." I think that this characterization is actually quite useful.[119] Think about all the different ways of manipulating images available in Photoshop and the degree of control provided by its multiple tools. Think also about Photoshop's concept of a visual composition as a stack of potentially hundreds of layers each with its transparency setting and multiple alpha channels. If we are able to animate such a composition and continue using Photoshop tools to adjust visual elements over time on all layers independently, this is indeed constitutes a new paradigm for creating moving images. And this is what After Effects and other animation, visual effects and compositing software make possible today.[120] And while idea of working with a number of layers placed on top of each other itself is not new – consider traditional cell animation, optical printing, video switchers, photocollage, graphic design, – going from a few non-transparent layers to hundreds and even thousands, each with its controls, fundamentally changes not only how a moving image looks but also what it can say. From being a special effect reserved for particular shots, 2D compositing became a part of the standard animation and video editing interface.

But innovative as 2D composting paradigm was, by the beginning of the 2000s already came to be supplemented by a new one: 3D compositing. If 2D compositing can be thought as an extension of already familiar media techniques, the new paradigm does not come from any previous physical or electronic media. Instead, it takes the new born-digital media which was invented in the 1960s and matured by early 1990s – interactive 3D computer graphics and animation – and transforms it into a general platform for moving media design.

The language used in professional production milieu today reflects an implicit understanding that 3D graphics is a new medium unique to a computer. When people use terms “computer visuals,” “computer imagery,” or “CGI” (which is an abbreviation for “computer generated imagery”) everybody understands that they refer to 3D graphics as opposed to other image source such as “digital photography.” But what is my own reason for thinking of 3D computer graphics as a new media – as opposed to considering it as an extension of architectural drafting, projection geometry, or set making? Because it offers a new method for representing three-dimensional reality - both objects which already exists and objects which are only imagined. This method is fundamentally different from what has been offered by main representational media of the industrial era: lens-based capture (still photography, film recording, video) and audio recording. With 3D computer graphics, we can represent three-dimensional structure of the world – versus capturing only a perspectival image of the world, as in lens-based recording. We can also manipulate our representation using various tools with ease and precision which is qualitatively different from a much more limited “manipulability” of a model made from any physical material (although nanotechnology promises to change this in the future.) And, as contemporary architectural aesthetics makes it clear, 3D computer graphics is not simply a faster way of working with geometric representations such as plans and cross-sections used by draftsmen for centuries. When the generations of young architects and architectural students started to systematically work with 3D modeling and animation software such as Alias in the middle of the 1990s, the ability to directly manipulate a 3D shape (rather than only dealing with its projections as in traditional drafting) quickly led to a whole new language of complex non-rectangular curved forms. In other words, architects working with the media of 3D computer graphics started to imagine different things than their predecessors who used pencils, rules, and drafting tables.

When Velvet Revolution of the 1990s made possible to easily combine multiple media sources in a single moving image sequence using multi-layer interface of After Effects, CGI was added to the mix. Today, 3D models are routinely used in media compositions created in After Effects and similar software, along with all other media sources. But in order to be a part of the mix, these models need to be placed on their own 2D layers and thus treated as 2D images. This was the original After Effects paradigm: all image media can meet as long as they are reduced to 2D.[121]

In contrast, in 3D compositing paradigm all media types are placed within a single virtual 3D space.

One advantage of this representation is that since 3D space is "native" to 3D computer graphics, 3D models can stay as they are, i.e. three-dimensional. An additional advantage is that the designer can now use all the techniques of virtual cinematography as developed in 3D computer animation. She can define different kinds of lights, fly the virtual camera around and through the image planes at any trajectory, and use depth of field and motion blur effects.[122]

While 3D computer-generated models already "live" in this space, how do you bring there two-dimensional visual elements – video, digitized film, typography, drawn images? If 2D compositing paradigm treated everything as 2D images – including 3D computer models – 3D compositng treats everything as 3D. So while two-dimension elements do not inherently have a 3rd dimension, it has to be added to enable these elements enter the three-dimensional space. To do that, a designer places flat cards in this space in particular locations, and situates two-dimensional images on these cards. Now, everything lives in a common 3D space. This condition enables "deep remixability" between techniques which I have illustrated using the example of "Go" video. The techniques of drawing, photography, cinematography and typography which go into capturing or creating two-dimensional visual elements can now "play" together with all the techniques of 3D computer animation (virtual camera moves, controllable depth of field, variable lens, etc.)

3D Compositing, or How Cinema Became Design

In 1995 I published the article *What is Digital Cinema?* where I tried to think about how the changes in moving image production I was witnessing were changing the concept of "cinema." In that article I proposed that the logic of hand-drawn animation, which throughout the twentieth century was marginal in relation to cinema, became dominant in a software era. Because software allows the designer to manually manipulate any image regarding of its source as though it was drawn in the first place, the ontological differences between different image media become irrelevant. Both conceptually and practically, they all reduced to hand-drawn animation.

After Effects and other animation/video editing/2D compositig software by default treats a moving image project as a stack of layers. Therefore, I can extend my original argument and propose that animation logic moves from the marginal to the dominant position also in yet another way. The paradigm of a composition as a stack of separate visual elements as practiced in cell animation becomes the default way of working with all images in a software environment – regardless of their origin and final output media. In other words, a "moving image" is now understood as a composite of layers of imagery – rather than as a still flat picture that only changes in time, as it was the case for most of the 20th century. In the word of animation, editing, and compositing software, such "single layer image" becomes an exception.

The emergence of 3D compositing paradigm can be also seen as following this logic of historical reversal. The new representational structure as developed within computer graphics field – a 3D virtual space containing 3D models – has gradually moved from a marginal to the dominant role. In the 1970s and 1980s computer graphics were used only occasionally in a dozen of feature films such as *Alien* (1979), *Tron* (1981), *The Last Starfighter* (1984), and *Abyss* (1989), and selected television commercials and broadcast graphics. But by the beginning of the 2000s, the representation structure of computer graphics, i.e. a 3D virtual space, came to function as an umbrella within can hold all other image types regardless of their origin. An example of an application which implements this paradigm is Flame, enthusiastically described by one user as "a full 3D compositing environment into which you can bring 3D models, create true 3D text and 3D particles, and distort layers in 3D space."[123]

This does not mean that 3D animation itself became visually dominant in moving image culture, or that the 3D structure of the space within which media compositions are now routinely constructed is necessary made visible (usually it is not.) Rather, the way 3D computer animation organizes visual data – as objects positioned in a Cartesian space – became the way to work with all moving image media. As already stated above, a designer positions all the elements which go into a composition – 2D animated sequences, 3D objects, particle systems, video and digitized film sequences, still images and photographs – inside the shared 3D virtual space. There these elements can be further animated, transformed, blurred, filtered, etc. So while all moving image media has been reduced to the status of hand-drawn animation in terms of their manipulability, we can also state that all media have become layers in 3D space. In short, the new media of 3D computer animation has "eaten up" the dominant media of the industrial age – lens-based photo, film and video recording.

Since we just discovered that software has redefined the concept of a "moving image" as a composite of multiple layers, this is a good moment to pause and consider other possible ways software changed this concept. When cinema in its modern form was born in the end of the nineteenth century, the new medium was understood as an extension of already familiar one – that is, as a photographic image which is now moving. This understanding can be found in the press accounts of the day and also in at least one of the official names given to the new medium - "moving pictures." On the material level, a film indeed consisted from separate photographic frames which when they were quickly replacing each created the effect of motion for the viewer. So the concept used to understand cinema indeed fit with the structure of the medium.

But is this concept still appropriate today? When we record video and play it, we are still dealing with the same structure: a sequence of frames. But for the professional media designers, the terms have changed. The importance of these changes is not just academic and purely theoretical. Because designers understand their media differently, they are creating films and sequences that also look very different from 20th century cinema or animation.

Consider what I referred to as new paradigms – essentially, new ways of creating "moving images" – which we have discussed so far. (Although theoretically they are not necessary all compatible with each other, in production practice these different paradigms are used in a complementary fashion.) A "moving image" became a hybrid which can combine all different visual media invented so far – rather than holding only one kind of data such as camera recording, hand drawing, etc. Rather than being understood as a singular flat plane – the result of light focused by the lens and captured by the recording surface – it is now understood as a stack of potentially infinite number of separate layers. And rather than "time-based," it becomes "composition-based," or "object oriented." That of, instead of being treated as a sequence of frames arranged in time, a "moving image" is now understood as a two-dimensional composition that consists from a number of objects that can be manipulated independently. Alternatively, if a designer uses 3D compositing, the conceptual shift is even more dramatic: instead of editing "images," she is working in a virtual three-dimensional space that holds both CGI and lens-recorded flat image sources.

Of course, frame-based representation did not disappear – but it became simply a recoding and output format rather than the space where a film is being put together. And while the term "moving image" can be still used as an appropriate description for how the output of a production process is experienced by the viewers, it is no longer captures how the designers think about what they create. Because their production environment - workflow, interfaces, and the tools – has changed so much, they are thinking today very differently than twenty years ago.

If we focus on what the different paradigms summarized above have in common, we can say that filmmakers, editors, special effects artists, animators, and motion graphics designers are working on *a composition in 2D or a 3D space that consists from a number of separate objects*. The spatial dimension became as important as temporal dimension. From the concept of a "moving image" understood as a sequence of static photographs we have moved to a new concept: *a modular media composition*. And while a person who directs a feature or a short film that is centered around actors and live action can be still called "filmmaker," in all other cases where most of production takes place in a software environment, it is more appropriate to call the person a "designer." This is yet another fundamental change in the concept of "moving images": today more often than not they are not "captured," "directed," or "animated." Instead, they are "designed."

Import/Export: Design Workflow And Contemporary Aesthetics

In our discussions of After Effects interface and workflow as well as the newer paradigm of 3D compositing we have already came across the crucial aspect of software-based media production process. Until the arrival of the software-based tools in the 1990s, to combine different types of time-based media together was either time consuming, or expensive, or in some cases simply impossible. Software tools such as After Effects have changed this situation in a fundamental way. Now a designer can import different media into her composition with just a few mouse clicks.

However, the contemporary software-based design of moving images – or any other design process, for that matter – does not simply involve combining elements from different sources within a single application. In this section we will look at the whole workflow typical of contemporary design – be it design of moving images, still illustrations, 3D objects and scenes, architecture, music, web sites, or any other media. (Most of the analysis of software-based production of moving images which I already presented also applies to graphic design of still images and layouts for print, the web, packaging, physical spaces, mobile devices, etc. However, in this section I want to make this explicit. Therefore the examples below will include not only moving images, but also graphic design.)

Although ”import”/”export” commands appear in most modern media authoring and editing software running under GUI, at first sight they do not seem to be very important for understanding software culture. When you “import,” you are not authoring new media or modifying media objects or accessing information across the globe, as in web browsing. All these two commands allow you to do is to move data around between different applications. In other words, they make data created in one application compatible with other applications. And that does not look so glamorous.

Think again. What is the largest part of the economy of greater Los Angeles area? It is not entertainment. From movie production to museums and everything is between only accounts for 15%). It turns out that the largest part of the economy is import/export business more than 60%. More generally, one commonly evoked characteristic of globalization is greater connectivity – places, systems, countries, organizations etc. becoming connected in more and more ways. And connectivity can only happen if you have certain level of compatibility: between business codes and procedures, between shipping technologies, between network protocols, between computer file formats, and so on.

Let us take a closer look at import/export commands. As I will try to show below, these commands play a crucial role in software culture, and in particular in media design – regardless of what kind of project a design is working on.

Before they adopted software tools in the 1990s, filmmakers, graphic designers, and animators used completely different technologies. Therefore, as much as they were influenced by each other or shared the same aesthetic sensibilities, they inevitably created differently looking images. Filmmakers used camera and film technology designed to capture three-dimensional physical reality. Graphic designers were working with offset printing and lithography. Animators were working with their own technologies: transparent cells and an animation stand with a stationary film camera capable of making exposures one frame at a time as the animator changed cells and/or moved background.

As a result, twentieth century cinema, graphic design and animation (I am talking here about standard animation techniques used by most commercial studios) developed distinct artistic languages and vocabularies both in terms of form and content. For example, graphic designers worked with a two dimensional space, film directors arranged compositions in three-dimensional space, and cell animators worked with a 'two-and-a-half' dimensions. This holds for the overwhelming majority of works produced in each field, although of course exceptions do exist. For instance, Oscar Fishinger made one abstract film that consisted from simple geometric objects moving in an empty space – but as far as I know, this is the only film in the whole history of abstract animation, which is taking place in three-dimensional space.

The differences in technology influenced what kind of content would appear in different media. Cinema showed "photorealistic" images of nature, built environments and human forms articulated by special lighting. Graphic designs featured typography, abstract graphic elements, monochrome backgrounds and cutout photographs. And cartoons showed hand-drawn flat characters and objects animated over hand-drawn but more detailed backgrounds. The exceptions are rare. For instance, while architectural spaces frequently appear in films because directors they could explore their three dimensionality in staging scenes, they practically never appear in animated films in any detail – until animation studios start using 3D computer animation.

Why was it so difficult to cross boundaries? For instance, in theory one could imagine making an animated film in the following way: printing a series of slightly different graphics designs and then filming them as though they were a sequence of animated cells. Or a film where a designer simply made a series of hand drawings that used the exact vocabulary of graphic design and then filmed them one by one. And yet, to the best of my knowledge, such a film was never made. What we find instead are many abstract animated films that have certain connection to various styles of abstract painting. For example, Oscar Fishinger's films and paintings share certain forms. We can also find abstract films and animated commercials and movie titles that have certain connection to graphic design aesthetics popular around the same times. For instance, some moving image sequences made by motion graphics pioneer Pablo Ferro around 1960s display psychedelic aesthetics which can be also found in posters, record covers, and other works of graphic design in the same period.

And yet, despite these connections, works in different media never used exactly the same visual language. One reason is that projected film could not adequately show the subtle differences between typeface sizes, line widths, and grayscale tones crucial for modern graphic design. Therefore, when the artists were working on abstract art films or commercials that adopted design aesthetics (and most major 20th abstract animators worked both on their own films and commercials), they could not simply expand the language of a printed page into time dimension. They had to invent essentially a parallel visual language that used bold contrasts, more easily readable forms and thick lines – which, because of their thickness, were in fact no longer lines but shapes.

Although the limitations in resolution and contrast of film and television image in comparison to a printed page contributed to the distance between the languages used by abstract filmmakers and graphic designers for the most of the twentieth century, ultimately I do not think it was the decisive factor. Today the resolution, contrast and color reproduction between print, computer screens, television screens, and the screens of mobile phones are also substantially different – and yet we often see exactly the same visual strategies deployed across these different display media. If you want to be convinced, leaf through any book or a magazine on contemporary 2D design (i.e., graphic design for print, broadcast, and the web). When you look at pages featuring the works of a particular designer or a design studio, in most cases its impossible to identify the origins of the images unless you read the captions. Only then do you find that which image is a poster, which one is a still from a music video, and which one is magazine editorial.

I am going to use Tashen's *Graphic Design for the 21st Century: 100 of the World's Best Graphic Designers* (2001) for examples. Peter Anderson's design showing a line of type against a cloud of hundred of little letters in various orientations turns out to be the frames from the title sequence for Channel Four documentary. His other design which similarly plays on the contrast between jumping letters in a larger font against irregularly cut planes made from densely packed letters in much smaller fonts turns to be a spread from IT Magazine. Since the first design was made for broadcast while the second was made for print, we would expect that the first design would employ bolder forms - however, both designs use the same scale between big and small fonts, and feature texture fields composed from hundreds of words in such a small font that they clear need to be read. A few pages later we encounter a design by Philippe Apeloig that uses exactly the same technique and aesthetics as Anderson. In this case, tiny lines of text positioned at different angles form a 3D shape floating in space. On the next page another design by Apeloig creates a field in perspective - made not from letters but from hundreds of identical abstract shapes.

These design rely on software's ability (or on the designer being influenced by software use and recreating what she did with software manually) to treat text as any graphical primitive and to easily create compositions made from hundreds of similar or identical elements positioned according to some pattern. And since an algorithm can easily modify each element in the pattern, changing its position, size, color, etc., instead of the completely regular grids of modernism we see more complex structures that are made from many variations of the same element. (This strategy is explored particularly imaginatively in Zaha Hadid's designs such as Louis Vuiiton Icone Bag, 2006, and in urban masterplans for Singapore and Turkey which use what Hadid calls a "variable grid.")

Each designer included in the book was asked to provide a brief statement to accompany the portfolio of their work, and Lust studio has put this phrase as their motto: "Form-follows-process." So what is the nature of design process in the software age and how does it influence the forms we see today around us?

If you practically involved in design or art today, you already knows that contemporary designers use the same small set of software tools to design just about everything. I have already named them repeatedly, so you know the list: Photoshop, Illustrator, Flash, Maya, etc. However, the crucial factor is not the tools themselves but the workflow process, enabled by "import" and "export" operations and related methods ("place," "insert object," "subscribe," "smart object," etc.), which ensure coordination between these tools.

When a particular media project is being put together, the software used at the final stage depends on the type of output media and the nature of the project – After Effects for motion graphics projects and video compositing, Illustrator or Freehand for print illustrations, InDesign for graphic design, Flash for interactive interfaces and web animations, 3ds Max or Maya for 3D computer models and animations, and so on. But these programs are rarely used alone to create a media design from start to finish. Typically, a designer may create elements in one program, import them into another program, add elements created in yet another program, and so on. This happens regardless whether the final product is an illustration for print, a web site, or a motion graphics sequence; whether it is a still or a moving image, interactive or non-interactive, etc.

The very names which software companies give to the products for media design and production refer to this defining characteristic of software-based design process. Since 2005, Adobe has been selling its different applications bundled together into "Adobe Creative Suite." The suite collects the most commonly used media authoring software: Photoshop, Illustrator, inDesign, Flash, Dreamweaver, After Effects, Premiere, etc. Among the subheadings and phrases used to accompany this band name, one in particular is highly meaningful in the context of our discussion: "Design Across Media." This phrase accurately describes both the capabilities of the applications collected in a suite, and their actual use in the real world. Each of the key applications collected in the suite – Photoshop, Illustrator, InDesign, Flash, Dreamweaver, After Effects, Premiere – has many special features geared for producing a design for particular output media. Illustrator is set up to work with professional-quality printers; After Effects and Premiere can output video files in a variety of standard video formats such as HDTV; Dreamweaver supports programming and scripting languages to enable creation of sophisticated and large-scale dynamic web sites. But while a design project is finished in one of these applications, most other applications in Adobe Creative Suite will be used in the process to create and edit its various elements. Thus is one of the ways in which Adobe Creative Suite enables "design across media." The compatibility between applications also means that the elements (called in professional language "assets") can be later re-used in new projects. For instance, a photograph edited in Photoshop can be first used in a magazine ad and later put in a video, a web site, etc. Or, the 3D models and characters created for a feature film are reused for a video game based on the film. This ability to re-use the same design elements for very different projects types is very important because of the widespread practice in creative industries to create products across the range of media which share the same images, designs, characters, narratives, etc. An advertising campaign often works "across media" including web ads, TV ads, magazine ads, billboards, etc. And if turning movies into games and games into movies has been already popular in Hollywood for a while, a new trend since approximately middle of 2000s is to create a movie, a game, a web site or maybe other media products at the same time – and have all the products use the same digital assets both for economic reasons and to assure aesthetic continuity between these products. Thus, a studio may create 3D backgrounds and characters and put them both in a movie and in a game, which will be released simultaneously. If media authoring applications were not compatible, such practice would simply not be possible.

All these examples illustrate the intentional reuse of design elements "across media." However, the compatibility between media authoring applications also has a much broader and non-intentional effect on contemporary aesthetics. Given the production workflow I just described, we may expect that the same visual techniques and strategies will also appear in all types of media projects designed with software without this being consciously planned for. We may also expect that this will happen on a much more basic level. This is indeed the case. *The same software-enabled design strategies, the same software-based techniques and the same software-generated iconography are now found across all types of media, all scales, and all kinds of projects*.

We have already encountered a few concrete examples. For instance, the three designs by Peter Anderson and Philip Apeloig done for different media use the same basic computer graphic technique: automatic generation of a repeating pattern while varying the parameters which control the appearance of each element making up the pattern's element – its size, position, orientation, curvature, etc. (The general principle behind this technique can also be used to generate 3D models, animations, textures, make plants and landscapes, etc. It is often referred to as "parametric design," or "parametric modeling.") The same technique is also used by Hadid's studio for Louis Vuiiton Icone Bag. In another example, which will be discussed below, Gregg Lynn used particle systems technique – which at that time was normally used to simulate fire, snow, waterfalls, and other natural phenomena in cinema – to generate the forms of a building.

To use the biological metaphor, we can say that compatibility between design applications creates very favorable conditions for the propagation of media DNAs between species, families, and classes. And this propagation happens on all levels: the whole design, parts of a design, the elements making up the parts, and the "atoms" which make up the elements. Consider the following hypothetical example of propagation on a lower level. A designer can use Illustrator to create a 2D smooth curve (called in computer graphics field called a “spline.”) This curve becomes a building block that can be used in any project. It can form a part of an illustration or a book design. It can be imported into animation program where it can be set to motion, or imported into 3D program where it can be extruded in 3D space to define a solid object.

Over time software manufacturers worked to developed tighter ways of connecting their applications to make moving elements from one to another progressively easier and more useable. Over the years, it became possible to move a complex project between applications without loosing anything (or almost anything). For example, in describing the integration between Illustrator CS3 and Photoshop CS3, Adobe’s web site states that a designer can “Preserve layers, layer comps, transparency, editable files when moving files between Photoshop and Illustrator.”[124] Another important development has been the concept that Microsoft Office calls “linked objects.” If you link all of a part of one file to another file (for instance, linking an excel document to a PowerPoint presentation), any time information changes in the first file, it automatically gets updated in the second file. Many media applications implement this feature. To use the same example of Illustrator CS3, a designer can “Import Illustrator files into Adobe Premiere Pro software, and then use Edit Original command to open the artwork in Illustrator, edit it, and see your changes automatically incorporated into your video project.”[125]

Each of the type of programs used by media designers – 3D graphics, vector drawing, image editing, animation, compositing – excel at particular design operations, i.e. particular ways of creating design elements or modifying already existing elements. These operations can be compared to the different types of blocks of a Lego set. You can create an infinite number of projects by just using the limited number of block types provided in the set. Depending on the project, these block types will play different functions and appear in different combinations. For example, a rectangular red block may become a part of the tabletop, a part of the head of a robot, etc.

Design workflow that uses a small number of compatible software programs works in a similar way – with one important difference. The building blocks used in contemporary design are not only different kinds of visual elements one can create – vector patterns, 3D objects, particle systems, etc. – but also *various ways of modifying these elements*: blur, skew, vectorize, change transparency level, spherisize, extrude, etc. This difference is crucial. If media creation and editing software did not include these and many other modification operations, we would have seen an altogether different visual language at work today. We would have seen "multimedia," i.e. designs that simply combine elements from different media. Instead, we see "deep remixability" – the "deep" interactions between working methods and techniques of different media within a single project.

In a "cross-over" use, the techniques which were previously specific in one media are applied to other media types (for example, a lens blur filter). This often can be done within a single application – for instance, applying After Effects's blur filter to a composition which can contain graphic elements, video, 3D objects, etc. However, being able to move a whole project or its elements between applications opens many more possibilities because each application offers many unique techniques not available in other applications. As the media data travels from one application to the next, is being transformed and enhanced using the operations offered by each application. For example, a designer can take her project she has been editing in Adobe Premiere and import in After Effects where she can use advanced compositing features of this program. She can then import the result back into Premiere and continue editing. Or she can create artwork in

Photoshop or Illustrator and import into Flash where it can be animated. This animation can be then imported into a video editing program and combined with video. A spline created in Illustrator becomes a basis for a 3D shape. And so on.

The production workflow specific to the software era that I just illustrated has two major consequences. Its first result is the partcular visual aesthetics of hybridity which dominates contemporary design universe. The second is the use of the same techniques and strategies across this universe - regardless of the output media and type of project.

As I already stated more than once, a typical design today combines techniques coming from multiple media. We now in a better position to understand why this is the case. As designer works on a project, she combines the results of the operations specific to different software programs that were originally created to imitate work with different physical media (Illustrator was created to make illustrations, Photoshop - to edit digitized photographs, Premiere – to edit video, etc.) While these operations continue to be used in relation to their original media, most of them are now also used as part of the workflow on any design job.

The essential condition that enables this new design logic and the resulting aesthetics is compatibility between files generated by different programs. In other words, "import," "export" and related functions and commands of graphics, animation, video editing, compositing and modeling software are historically more important than the individual operations these programs offer. The ability to combine raster and vector layers within the same image, to place 3D elements into a 2D composition and vice versa, and so on is what enables the production workflow with its reuse of the same techniques, effects, and iconography across different media.

The consequences of this compatibility between software and file formats, which was gradually achieved during the 1990s, are hard to overestimate. Besides the hybridity of modern visual aesthetics and reappearance of exactly the same design techniques across all output media, there are also other effects. For instance, the whole field of motion graphics as it exists today came into existence to a large extent because of the integration between vector drawing software, specifically Illustrator, and animation/compositing software such as After Effects. A designer typically defines various composition elements in Illustrator and then imports them into After Effects where they are animated. This compatibility did not exist when the initial versions of different media authoring and editing software initially became available in the 1980s. It was gradually added in particular software releases. But when it was achieved around the middle of the 1990s[126], within a few years the whole language of contemporary graphical design was fully imported into the moving image area – both literally and metaphorically.

In summary, the compatibility between graphic design, illustration, animation, video editing, 3D modeling and animation, and visual effects software plays the key role in shaping visual and spatial forms of the software age. On the one hand, never before have we witnessed such a variety of forms as today. On the other hand, exactly the same techniques, compositions and iconography can now appear in any media.

The Variable Form

As the films of Blake and Murata discussed earlier illustrate, in contrast to twentieth-century animation, in contemporary motion graphics the transformations often affect the frame as a whole. Everything inside the frame keeps changing: visual elements, their transparency, the texture of the image, etc. In fact, if something stays the same for a while, that is an exception rather than the norm.

Such *constant change on many visual dimensions* is another key feature of motion graphics and design cinema produced today. Just as we did it in the case of media hybridity, we can connect this preference for constant change to the particulars of software used in media design.

Digital computers allow us to represent any phenomenon or structure as a set of variables. In the case of design and animation software, this means that all possible forms—visual, temporal, spatial, interactive—are similarly represented as sets of variables that can change continuously. This new logic of form is deeply encoded in the interfaces of software packages and the tools they provide. In 2D animation/compositing software such as After Effects, each new object added to the scene by a designer shows up as a long list of variables—geometric position, color, transparency, and the like. Each variable is immediately assigned its own channel on the timeline used to create animation.[127] In this way, the software literally invites the designer to start animating various dimensions of each object in the scene. The same logic extends to the parameters that affect the scene as a whole, such as the virtual camera and the virtual lighting. If you add a light to the composition, this immediately creates half a dozen new animation channels describing the colors of the lights, their intensity, position, orientation, and so on.

During the 1980s and 1990s, the general logic of computer representation—that is, representing everything as variables that can have different values—was systematically embedded throughout the interfaces of media design software. As a result, although a particular software application does not directly prescribe to its users what they can and cannot do, the structure of the interface strongly influences the designer’s thinking. In the case of moving image design, the result of having a timeline interface with multiple channels all just waiting to be animated is that a designer usually does animate them. If previous constraints in animation technology—from the first optical toys in the early nineteenth century to the standard cel animation system in the twentieth century—resulted in an aesthetics of discrete and limited temporal changes, the interfaces of computer animation software quickly led to a new aesthetics: the continuous transformations of all visual elements appearing in a frame (or of the singular image filling the frame).

This change in animation aesthetics deriving from the interface design of animation software was paralleled by a change in another field—architecture. In the mid-1990s, when architects started to use software originally developed for computer animation and special effects (first Alias and Wavefront; later Maya and others), the logic of animated form entered architectural thinking as well. If 2D animation/compositing software such as After Effects enables an animator to change any parameter of a 2D object (a video clip, a 2D shapes, type, etc.) over time, 3D computer animation allows the same for any 3D shape. An animator can set up keyframes manually and let a computer calculate how a shape changes over time. Alternatively, she can direct algorithms that will not only modify a shape over time but can also generate new ones. (3D computer animation tools to do this include particle systems, physical simulation, behavioral animation, artificial evolution, L-systems, etc.) Working with 3D animation software affected architectural imagination both metaphorically and literally. The shapes, which started to appear in the projects by young architects and architecture students in the second part of the 1990s looked as they were in the process of being animated, captured as they were transforming from one state to another. The presentations of architectural projects and research begin to feature multiple variations of the same shape generated by varying parameters in software. Finally, in projects such as Gregg Lynn’s New York Port Authority Gateway (1995),[128] the paths of objects in an animation were literally turned into an architectural design. Using a particle system (a part of Wavefront animation software), which generates a cloud of points and moves them in space to satisfy a set of constraints, Lynn captured these movements and turned them into a curves making up his proposed building.

Equally crucial was the exposure of architects to the new generation of modeling tools in the commercial animation software of the 1990s. For two decades, the main technique for 3D modeling was to represent an object as a collection of flat polygons. But by the mid-1990s, the faster processing speeds of computers and the increased size of computer memory made it practical to offer another technique on desktop workstations—spline-based modeling. This new technique for representing form pushed architectural thinking away from rectangular modernist geometry and toward the privileging of smooth and complex forms made from continuous curves. As a result, since the second part of 1990s, the aesthetics of "blobs" has come to dominate the thinking of many architecture students, young architects, and even already well-established "star" architects such as Hadid, Eric Moss, and UN Studio.

But this was not the only consequence of the switch from the standard architectural tools and CAD software (such as AutoCAD) to animation/special effects software. Traditionally, architects created new projects on the basis of existing typology. A church, a private house, a railroad station all had their well-known types—the spatial templates determining the way space was to be organized. Similarly, when designing the details of a particular project, an architect would select from the various standard elements with well-known functions and forms: columns, doors, windows, etc.[129] In the twentieth century, mass-produced housing only further embraced this logic, which eventually became encoded in the interfaces of CAD software.

But when in the early 1990s, Gregg Lynn, the firm Asymptote, Lars Spuybroek, and other young architects started to use 3D software that had been created for other industries—computer animation, special effects, computer games, and industrial design—they found that this software came with none of the standard architectural templates or details. In addition, if CAD software for architects assumed that the basic building blocks of a structure are rectangular forms, 3D animation software came without such assumptons. Instead it offered splined curves and smooth surfaces and shapes constructed from these surves — which were appropriate for the creation of animated and game characters and industrial products. (In fact, splines were originally introduced into computer graphics in 1962 by Pierre Bézier for the use in computer-aided car design.)

As a result, rather than being understood as a composition made up of template-driven standardized parts, a building could now be imagined as a single continuous curved form that can vary infinitely. It could also be imagined as a number of continuous forms interacting together. In either case, the shape of each of these forms was not determined by any kind of a priori typology.

(In retrospect, we can think of this highly productive "misuse" of 3D animation and modeling software by architects as another case of media hybridity – in particular, what I called the "crossover effect" In this case, it is a crossover between the conventions and the tools of one design field—character animation and special effects—and the ways of thinking and knowledge of another field, namely, architecture.)

Relating this discussion of architecture to the main subject of this chapter—production of moving images—we can see now that by the 1990s both fields were affected computerization in a structurally similar way. In the case of commercial animation in the West, previously all temporal changes inside a frame were limited, discrete, and usually semantically driven – i.e., connected to the narrative. When an animated character moved, walking into a frame, turned his head, or extended his arm, this was used to advance the story.[130] After the switch to software-based production process, moving images came to feature constant changes on many visual dimensions that were no longer limited by the semantics. As defined by numerous motion-graphics sequences and short films of the 2000s, contemporary temporal visual form constantly changes, pulsates, and mutates beyond the need to communicate meanings and narrative. (The films of Blake and Murata offer striking examples of this new aesthetics of a variable form; many other examples can easily be found by surfing websites that collect works by motion graphics studios and individual designers.)

A parallel process took place in architectural design. The differentiations in a traditional architectural form were connected to the need to communicate meaning and/or to fulfill the architectural program. An opening in a wall was either a window or a door; a wall was a boundary between functionally different spaces. Thus, just as in animation, the changes in the form were limited and they were driven by semantics. But today, the architectural form designed with modeling software can change continuously, and these changes no longer have to be justified by function.

The Yokohama International Port Terminal (2002) designed by Foreign Office Architects illustrates very well the aesthetics of variable form in architecture. The building is a complex and continuous spatial volume without a single right angle and with no distinct boundaries that would break the form into parts or separate it from the ground plane. Visiting the building in December 2003, I spent four hours exploring the continuities between the exterior and the interior spaces and enjoying the constantly changing curvature of its surfaces. The building can be compared to a Mobius strip - except that it is much more complex, less symmetrical, and more unpredictable. It would be more appropriate to think of it as a whole set of such strips smoothly interlinked together.

To summarize this discussion of how the shift to software-based representations affected the modern language of form: All constants were substituted by variables whose values can change continuously. As a result, culture went through what we can call the *continuity turn*. Both the temporal visual form of motion graphics and design cinema and the spatial form of architecture entered the new universe of continuous change and transformation. (The fields of product design and space design were similarly affected.) Previously, such aesthetics of "total continuity" was imagined by only a few artists. For instance, in the 1950s, architect Friedrich Kiesler conceived a project titled *Continuous House* that, as the name implies, a single continuously curving spatial form unconstrained by the usual divisions into rooms. But when architects started to work with the 3D modeling and animation software in the 1990s, such thinking became commonplace. Similarly, the understanding of a moving image as a continuously changing visual form without any cuts, which previously could be found only in a small number of films made by experimental filmmakers throughout the twentieth century such as Fischinger's *Motion Painting* (1947), now became the norm.

Scaling Up Aesthetics of Variability

Today, there are many successful short films under a few minutes and small-scale building projects are based on the aesthetics of continuity – i.e., a single continuously changing form, but the next challenge for both motion graphics and architecture is to discover ways to employ this aesthetics on a larger scale. How do you scale-up the idea of a single continuously changing visual or spatial form, without any cuts (for films) or divisions into distinct parts (for architecture)?

In architecture, a number of architects have already begun to successfully address this challenge. Examples include already realized projects such as the Yokohama International Port Terminal or the Kunsthaus in Graz by Peter Cook (2004), as well as those that have yet to be built, such as Zaha Hadid's Performing Arts Centre on Saadiyat Island in Abu

Dhabi, United Arab Emirates (proposed in 2007). In fact, given the current construction book in China, Dubai, Eastern Europe and a number of other "developing countries," and their willingness to take risks and embrace the new, the architectural designs made from complex continuosly changing curves are getting build on a larger scale, in more numbers, and faser than it was possible to imagine even a few yeas before.

What about motion graphics? So far Blake has been one of the few artists who have systematically explored how hybrid visual language can work in longer pieces. *Sodium Fox* is 14 minutes; an earlier piece, *Mod Lang* (2001), is 16 minutes. The three films that make up *Winchester Trilogy* (2001–4) run for 21, 18, and 12 minutes. None of these films contain a single cut.

Sodium Fox and *Winchester Trilogy* use a variety of visual sources, which include photography, old film footage, drawings, animation, type, and computer imagery. All these media are weaved together into a continuous flow. As I have already pointed out in relation to *Sodium Fox*, in contrast to shorter motion-graphics pieces with their frenzy of movement and animation, Blake's films contain very little animation in a traditional sense. Instead, various still or moving images gradually fade in on top of each other. So while each film moves through a vast terrain of different visuals—color and monochrome, completely abstract and figurative, ornamental and representational—it is impossible to divide the film into temporal units. In fact, even when I tried, I could not keep track of how the film got from one kind of image to a very different one just a couple of minutes later. And yet these changes were driven by some kind of logic, even if my brain could not compute it while I was watching each film.

The hypnotic continuity of these films can be partly explained by the fact that all visual sources in the films were manipulated via graphics software. In addition, many images were slightly blurred. As a result, regardless of the origin of the images, they all acquired a certain visual coherence. So although the films skillfully play on the visual and semantic differences between live-action footage, drawings, photographs with animated filters on top of them, and other media, these differences do not create juxtaposition or stylistic montage.131 Instead, various media seem to peacefully coexist, occupying the same space. In other words, Blake's films seem to suggest that media hybridization is not the only possible result of softwarization.

We have already discussed in detail Alan Kay's concept of a computer metamedium. According to Kay's proposal made in the 1970s, we should think of the digital computer as a metamedium containing all the different "already existing and non-yet-invented media."132 What does this imply for the aesthetics of digital projects? In my view, it does not imply that the different media necessarily fuse together, or make up a new single hybrid, or result in "multimedia," "intermedia," "convergence," or a totalizing Gesamtskunstwerk. As I have argued, rather than collapsing into a single entity, different media (i.e., different techniques, data formats, data sources and working methods) start interacting producing a large number of hybrids, or new "media species." In other words, just as in biological evolution, media evolution in a software era leads to differentiation and increased diversity – more species rather than less.

In the world dominated by hybrids, Blake's films are rare in presenting us with relatively "pure" media appearances. We can either interpret this as the slowness of the art world, which is behind the evolutionary stage of professional media – or as a clever strategy by Blake to separate himself from the usual frenzy and over stimulation of motion graphics. Or we can read his aesthetics as an implicit statement against the popular idea of "convergence." As demonstrated by Blake's films, while different media has become compatible, this does not mean that their distinct identities have collapsed. In

Sodium Fox and *Winchester Trilogy*, the visual elements in different media maintain their defining characteristics and unique appearances.

Blake's films also expand our understanding of what the aesthetics of continuity can encompass. Different media elements are continuously added on top of each other, creating the experience of a continuous flow, which nevertheless preserves their differences. Danish artist Ann Lislegaard also belongs to the "continuity generation." A number of her films involve continuous navigation or an observation of imaginary architectural spaces. We may relate these films to the works of a number of twentieth-century painters and filmmakers which were concerned with similar spatial experiences: Giorgio de Chirico, Balthus, the Surrealists, Alan Resnais (*Last Year at Marienbad*), Andrei Tarkovsky (*Stalker*). However, the sensibility of Lislegaard's films is unmistakably that of the early twenty-first century. The spaces are not clashing together as in, for instance, *Last Year at Marienbad*, nor are they made uncanny by the introduction of figures and objects (a practice of Réne Magritte and other Surrealists). Instead, like her fellow artists Blake and Murata, Lislegaard presents us with forms that continuously change before our eyes. She offers us yet another version of the aesthetics of continuity made possible by software such as After Effects, which, as has already been noted, translates the general logic of computer representation—the substitution of all constants with variables—into concrete interfaces and tools.

The visual changes in Lislegaard's *Crystal World (after J. G. Ballard)* (2006) happen right in front of us, and yet they are practically impossible to track. Within the space of a minute, one space is completely transformed into something very different. And it is impossible to say how exactly this happened.

Crystal World creates its own hybrid aesthetics that combines photorealistic spaces, completely abstract forms, and a digitized photograph of plants.

(Although I don't know the exact software Lislegaard's assistant used for this film, it is unmistaken some 3D computer animation package.) Since everything is rendered in gray scale, the differences between media are not loudly announced. And yet they are there. It is this kind of subtle and at the same time precisely formulated distinction between different media that gives this video its unique beauty. In contrast to twentieth-century montage, which created meaning and effect through dramatic juxtapositions of semantics, compositions, spaces, and different media, Lislegaard's aesthetics is in tune with other cultural forms. Today, the creators of minimal architecture and space design, web graphics, generative animations and interactives, ambient electronic music, and progressive fashions similarly assume that a user is intelligent enough to make out and enjoy subtle distinctions and continuous modulations.

Lislegaard's *Bellona (after Samuel R. Delany)* (2005) takes the aesthetics of continuity in a different direction. We are moving through and around what appears to be a single set of spaces. (Historically, such continuous movement through a 3D space has its roots in the early uses of 3D computer animation first for flight simulators and later in architectural walk-throughs and first-person shooters.) Though we pass through the same spaces many times, each time they are rendered in a different color scheme. The transparency and reflection levels also change. Lislegaard is playing a game with the viewer: while the overall structure of the film soon becomes clear, it is impossible to keep track of which space we are in at any given moment. We are never quite sure if we have already been there and it is now simply lighted differently, or if it is a space that we have not yet visited.

Bellona can be read as an allegory of "variable form." In this case, variability is played out as seemingly endless color schemes and transparency settings. It does not matter how many times we have already seen the same space, it always can appear in a new way.

To show us our world and ourselves in a new way is, of course, one of the key goals of all modern art, regardless of the media. By substituting all constants with variables, media software institutionalizes this desire. Now everything can always change and everything can be rendered in a new way. But, of course, simple changes in color or variations in a spatial form are not enough to create a new vision of the world. It takes talent to transform the possibilities offered by software into meaningful statements and original experiences. Lislegaard, Blake, and Murata—along with many other talented designers and artists working today—offer us distinct and original visions of our world in the stage of continuous transformation and metamorphosis: visions that are fully appropriate for our time of rapid social, technological, and cultural change.

Amplification of the Simulated Techniques

Although the discussions in this chapter did not cover all the changes that took place during Velvet Revolution, the magnitude of the transformations in moving image aesthetics and communication strategies should by now be clear. While we can name many social factors that all could have and probably did played some role – the rise of branding, experience economy, youth markets, and the Web as a global communication platform during the 1990s – I believe that these factors alone cannot account for the specific design and visual logics which we see today in media culture. Similarly, they cannot be explained by simply saying that contemporary consumption society requires constant innovation, constant novel aesthetics, and effects. This may be true – but why do we see these particular visual languages as opposed to others, and what is the logic that drives their evolution? I believe that to properly understand this, we need to carefully look at media creation, editing, and design software and their use in production environment - which can range from a single laptop to a number of production companies around the world with thousands of people collaborating on the same large-scale project such as a feature film. In other words, we need to use the perspective of Software Studies.

The makers of software used in media production usually do not set out to create a revolution. On the contrary, software is created to fit into already existing production procedures, job roles, and familiar tasks. But software are like species within the common ecology – in this case, a shared environment of a digital computer. Once "released," they start interacting, mutating, and making hybrids. Velvet Revolution can therefore be understood as the period of systematic hybridization between different software species originally designed to do work in different media. By 1993, designers has access to a number of programs which were already quite powerful but mostly incompatible: Illustrator for making vector-based drawings, Photoshop for editing of continuous tone images, Wavefront and Alias for 3D modeling and animation, After Effects for 2D animation, and so on. By the end of the 1990s, it became possible to use them in a single workflow. A designer could now combine operations and representational formats such as a bitmapped still image, an image sequence, a vector drawing, a 3D model and digital video specific to these programs within the same design. I believe that the hybrid visual language that we see today across "moving image" culture and media design in general is largely the outcome of this new production environment. While this language supports seemingly numerous variations as manifested in the particular media designs, its key aesthetics feature can be summed up in one phrase: deep remixability of previously separate media languages.

As I already stressed more than once, the result of this hybridization is not simply a mechanical sum of the previously existing parts but new "species." This applies both to the visual language of particular designs, and to the operations themselves. *When a* *pre-digital media operation is integrated into the overall digital production environment, it often comes to function in a new way*. I would like to conclude by analyzing in detail how this process works in the case of a particular operation - in order to emphasize once again that media remixability is not simply about adding the content of diffirent media, or adding together their techniques and languages. And since remix in contemporary culture is commonly understood as these kinds of additions, we may want to use a different term to talk about the kinds of transformations the example below illustrates. I called this provisonally "deep remixability," but what important is the idea and not a particular term. (So if you have a suggestion for a better one, send me an email.)

What does it mean when we see depth of field effect in motion graphics, films and television programs which use neither live action footage nor photorealistic 3D graphics but have a more stylized look? Originally an artifact of lens-based recording, depth of field was simulated in software in the 1980s when the main goal of 3D compute graphics field was to create maximum "photorealism," i.e. synthetic scenes not distinguishable from live action cinematography. But once this technique became available, media designers gradually realized that it can be used regardless of how realistic or abstract the visual style is – as long as there is a suggestion of a 3D space. Typography moving in perspective through an empty space; drawn 2D characters positioned on different layers in a 3D space; a field of animated particles – any spatial composition can be put through the simulated depth of field.

The fact that this effect is simulated and removed from its original physical media means that a designer can manipulate it a variety of ways. The parameters which define what part of the space is in focus can be independently animated, i.e. they can be set to change over time – because they are simply the numbers controlling the algorithm and not something built into the optics of a physical lens. So while simulated depth of field maintains the memory of the particular physical media (lens-based photo and film recording) from which it came from, it became an essentially new technique which functions as a "character" in its own right. It has the fluidity and versatility not available previously. Its connection to the physical world is ambiguous at best. On the one hand, it only makes sense to use depth of field if you are constructing a 3D space even if it is defined in a minimal way by using only a few or even a single depth cue such as lines converging towards the vanishing point or foreshortening. On the other hand, the designer is now able to "draw" this effect in any way desirable. The axis controlling depth of field does not need to be perpendicular to the image plane, the area in focus can be anywhere in space, it can also quickly move around the space, etc.

Following Velvet Revolution, the aesthetic charge of many media designs is often derived from more "simple" remix operations – juxtaposing different media in what can be called "media montage." However, for me the essence of this Revolution is the more fundamental "deep remixability" illustrated by this example of how depth of field was greatly amplified when it was simulated in software.

Computerization virtualized practically all media creating and modification techniques, "extracting" them from their particular physical media and turning them into algorithms. This means that in most cases, *we will no longer find any of the pre-digital techniques in their pure original state*. This is something I already discussed in general when we looked at the first stage in cultural software history, i.e. 1960s and 1970s. In all cases we examined - Sutherland's work on fist interactive graphical editor (Sketchpad), Nelson's concepts of hypertext and hypermedia, Kay's discussions of an electronic book – the inventors of cultural software systematically emphasized that they were not aiming at simply simulating existing media in software. To quite Kay and Goldberg once again when they write about the possibilities of a computer book, "It need not be treated as a simulated paper book since this is a new medium with new properties."

We have now seen how this general idea articulated already in the early 1960s made its way into the details of the interfaces and tools of applications for media design which eventually replaced most of traditional tools: After Effects (which we analyzed in detail), Illustrator, Photoshop, Flash, Final Cut, etc. So what is true for depth of field effect is also true for most other tools offered by media design applications.

What was a set of theoretical concepts implemented in a small number of custom software systems accessible mostly to their own creators in the 1960s and 1970s (such as Sketchpad or Xerox PARC workstation) later became a universal production environment used today throughout all areas of culture industry. The ongoing interactions between the ideas coming from software industry and the desires of users of their tools (media designers, graphic designers, film editors, and so on) – along with new needs which emerge than these tools came to used daily by hundreds of thousands of individuals and companies led to the further evolution of software - for instance, the emergence of an new category of "digital asset management" systems around early 2000s, or the concept of "production pipeline" which becomes important in the middle of this decade. In this chapter I highlighted just one among many directions in which evolution – making software applications, their tools, and media formats compatible with each other. As we saw, the result of this trend was anything but minor: an emergence of fundamentally new type of aesthetics which today dominates visual and media culture.

One of the consequences of this software compatibility is that the 20th century concepts that we still use by inertia to describe different cultural fields (or different areas of culture industry, if you like) – "graphic design," "cinema," "animation" and others – are in fact no longer adequately describe the reality. If each of the original media techniques has been greatly expanded and "super-charged" as a result of its implementation in a software, if the practioners in all these fields have access to a common set of tools, and if these tools can be combined in a single project and even a single image or a frame, are these fields really still distinct from each other? In the next chapter I will wrestle with this theoretical challenge by looking at a particularly interesting case of media hybridity – the techniques of Total Capture and originally developed for *Matrix* films. I will also ask how one of the terms from the list above which in the twentieth-century was used to refer to a distinct medium – "animation" - functions in a new software-based "post-media" universe of hybridity.

Chapter 4. Universal Capture

Introduction

For the larger part of the twentieth century, different areas of commercial moving image culture maintained their distinct production methods and distinct aesthetics. Films and cartoons were produced completely differently and it was easy to tell their visual languages apart. Today the situation is different. Softwarization of all areas of moving image production created a common pool of techniques that can be used regardless of whether one is creating motion graphics for television, a narrative feature, an animated feature, or a music video. The abilities to composite many layers of imagery with varied transparency, to place 2D and 3D visual elements within a shared 3D virtual space and then move a virtual camera through this space, to apply simulated motion blur and depth of field effect, to change over time any visual parameter of a frame are equally available to the creators of all forms of moving images.

The existence of this common vocabulary of software-based techniques does not mean that all films now look the same. What it means, however, is that while most live action films, animated features and motion graphics do look quite distinct today, this is the result of a deliberate choices rather than the inevitable consequence of differences in production methods and technology.

Given that all techniques of previously distinct media are now available within a single software-based production environment, what is the meaning of the terms that were used to refer to these media in the twentieth century – such as "animation"? From the industry point of view, the answer is simple. Animation not only continues to exist as a distinct area of media industry but its also very successful – its success in no small part fueled by new efficiency of software-based global production workflow. 2D and 3D animated features, shorts and series are produced today in larger numbers than ever before; students can pursue careers in "animation"; Japanese anime and animated features continue to grow in popularity; China is building whole cities around mega-size animation and rendering studios and production facilities.

Certainly, the aesthetics of many contemporary feature-length 3D animated features largely relies on the visual language of twentieth-century commercial animation. So while everything may be modeled and animated in 3D computer animation program, the appearance of the characters, their movements, and the staging of scenes conceptually owe more to mid 20th century Disney than to 21st century Autodesk (producer of industry-standard Maya software). Similarly, hybrid looking short-form films (exemplified by but not limited to "motion graphics") also often feature sequences or layers that look very much like character animation we know from the 20th century.

The examples above illustrate just one, more obvious, role of animation in contemporary post-digital visual landscape. In this chapter I will explore its other role: as a generalized tool set that can be applied to any images, including film and video. Here, animation functions not as a medium but as a set of general-purpose techniques – used together with other techniques in the common pool of options available to a filmmaker/designer. Put differently, what has been "animation" has become a part of the computer metamedium.

I have chosen a particular example for my discussion that I think will illustrate well this new role of animation. It is an especially intricate method of combining live action and CG (a common abbreviation for "compute graphics.") Called "Universal Capture" (U-cap) by their creators, it was first systematically used on a large scale by ESC Entertainment in *Matrix 2* and *Matrix 3* films from *The Matrix* trilogy. I will discuss how this method is different from the now standard and older techniques of integrating live action and computer graphics elements. The use of Universal Capture also leads to visual hybrids – but they are quite distinct from the hybrids found in motion graphics and other short-form moving image productions being created today. With Universal Capture, different types of imagery are "fused" together to create a new kind of image. This image combines "the best of" qualities of two types of imagery that we normally understand as being ontologically the opposites: live action recording and 3D computer animation. I will suggest that such image hybrids are likely to play a large role in future visual culture while the place of "pure" images that are not fused or mixed with anything is likely to gradually diminish.

Uneven Development

What kinds of images would dominate visual culture a number of decades from now? Would they still be similar to the typical images that surround us today – photographs that are digitally manipulated and often combined with various graphical elements and type? Or would future images be completely different? Would photographic code fade away in favor of something else?

There are good reasons to assume that the future images would be photograph-like. Like a virus, a photograph turned out to be an incredibly resilient representational code: it survived waves of technological change, including computerization of all stages of cultural production and distribution. One of the reason for this persistence of photographic code lies in its flexibility: photographs can be easily mixed with all other visual forms - drawings, 2D and 3D designs, line diagrams, and type. As a result, while photographs continue to dominate contemporary visual culture, most of them are not pure photographs but various mutations and hybrids:

photographs which went through different filters and manual adjustments to achieve a more stylized look, a more flat graphic look, more saturated color, etc.; photographs mixed with design and type elements; photographs which are not limited to the part of the spectrum visible to a human eye (night vision, x-ray); simulated photographs created with 3D computer graphics; and so on. Therefore, while we can say that today we live in a "photographic culture," we also need to start reading the word "photographic" in a new way. "Photographic" today is really photo-GRAPHIC, the photo providing only an initial layer for the overall graphical mix. (In the area of moving images, the term "motion graphics" captures perfectly the same development: the subordination of live action cinematography to the graphic code.)

One way in which change happens in nature, society, and culture is inside out. The internal structure changes first, and this change affects the visible skin only later. For instance, according to Marxist theory of historical development, infrastructure (i.e., mode of production in a given society – also called "base") changes well before superstructure (i.e., ideology and culture in this society). To use a different example, think of the history of technology in the twentieth century. Typically, a new type of machine was at first fitted within old, familiar skin: for instance, early twentieth century cars emulated the form of horse carriage. The popular idea usually ascribed to Marshall McLuhan – that the new media first emulates old media – is another example of this type of change. In this case, a new mode of media production, so to speak, is first used to support old structure of media organization, before the new structure emerges. For instance, first typeset book were designed to emulate hand-written books; cinema first emulated theatre; and so on.

This concept of uneven development can be useful in thinking about the changes in contemporary visual culture. Since this process started in the middle of the 1950s, computerization of photography (and cinematography) has by now completely changed the internal structure of a photographic image. Yet its "skin," i.e. the way a typical photograph looks, still largely remains the same. It is therefore possible that at some point in the future the "skin" of a photographic image would also become completely different, but this did not happen yet. So we can say at present our visual culture is characterized by a new computer "base" and old photographic "superstructure."

The Matrix films provide us with a very rich set of examples perfect for thinking further about these issues. The trilogy is an allegory about how its visual universe is constructed. That is, the films tell us about *The Matrix*, the virtual universe that is maintained by computers – and of course, visually the images of The Matrix trilogy that we the viewers see in the films were all indeed assembled using help software. (The animators sometimes used Maya but mostly relied on custom written programs). So there is a perfect symmetry between us, the viewers of a film, and the people who live inside *The Matrix* – except while the computers running *The Matrix* are capable of doing it in real time, most scenes in each of *The Matrix* films took months and even years to put together. (So *The Matrix* can be also interpreted as the futuristic vision of computer games in the future when it would become possible to render *The Matrix*-style visual effects in real time.)

The key to the visual universe of *The Matrix* is the new set of computer graphic techniques that over the years were developed by Paul Debevec**,** Georgi Borshukov, John Gaeta, and a number of other people both in academia and in the special effects industry.[133] Their inventors coined a number of names for these techniques: "virtual cinema," "virtual human," "virtual cinematography," "universal capture." Together, these techniques represent a true milestone in the history of computer-driven special effects. They take to their logical conclusion the developments of the 1990s such as motion capture, and simultaneously open a new stage. We can say that with *The Matrix*, the old "base" of photography has finally been completely replaced by a new computer-driven one. What remains to be seen is how the "superstructure" of a photographic image – what it represents and how – will change to accommodate this "base."

Reality Simulation versus Reality Sampling

Before proceeding, I should note that not all of special effects in *The Matrix* rely on Universal Capture. Also, since the Matrix, other Hollywood films and video games (EA SPORT Tiger Woods 2007) already used some of the same strategies. However, in this chapter I decided to focus on the use of this process in the second and third films of *the Matrix* for which the method of Universal Capture was originally developed. And while the complete credits for everybody involved in developing Universal Capture would run for a whole page, here I will identify it with Gaeta. The reason is not because, as a senior special effects supervisor for *The Matrix Reloaded* and *The Matrix Revolutions* he got most publicity. More importantly, in contrast to many others in the special effects industry, Gaeta has extensively reflected on the techniques he and his colleagues have developed, presenting it as a new paradigm for cinema and entertainment and coining useful terms and concepts for understanding it.

In order to understand better the significance of Gaeta's method, lets briefly run through the history of 3D photo-realistic image synthesis and its use in the film industry. In 1963 Lawrence G. Roberts (who later in the 1960s became one of the key people behind the development of Arpanet but at that time was a graduate student at MIT) published a description of a computer algorithm to construct images in linear perspective. These images represented the objects' edges as lines; in contemporary language of computer graphics they would be called "wire frames." Approximately ten years later computer scientists designed algorithms that allowed for the creation of shaded images (so-called Gouraud shading and Phong shading, named after the computer scientists who create the corresponding algorithms). From the middle of the 1970s to the end of the 1980s the field of 3D computer graphics went through rapid development. Every year new fundamental techniques were created: transparency, shadows, image mapping, bump texturing, particle system, compositing, ray tracing, radiosity, and so on.[134] By the end of this creative and fruitful period in the history of the field, it was possible to use combination of these techniques to synthesize images of almost every subject that often were not easily distinguishable from traditional cinematography. ("Almost" is important here since the creation of photorealistic moving images of human faces remained a hard to reach a goal – and this is in part what Total Capture method was designed to address.)

All this research was based on one fundamental assumption: in order to re-create an image of visible reality identical to the one captured by a film camera, we need to systematically simulate the actual physics involved in construction of this image. This means simulating the complex interactions between light sources, the properties of different materials (cloth, metal, glass, etc.), and the properties of physical film cameras, including all their limitations such as depth of field and motion blur. Since it was obvious to computer scientists that if they exactly simulate all this physics, a computer would take forever to calculate even a single image, they put their energy in inventing various short cuts which would create sufficiently realistic images while involving fewer calculation steps. So in fact each of the techniques for image synthesis I mentioned in the previous paragraph is one such "hack" – a particular approximation of a particular subset of all possible interactions between light sources, materials, and cameras.

This assumption also meant that you are re-creating reality step-by-step starting from a blank canvas (or, more precisely, an empty 3D space.) Every time you want to make a still image or an animation of some object or a scene, the story of creation from The Bible is being replayed.

(I imagine God creating Universe by going through the numerous menus of a professional 3D modeling, animation, and rendering program such as Maya. First he has to make all the geometry: manipulating splines, extruding contours, adding bevels…Next for every object and creature he has to choose the material properties: specular color, transparency level, image, bump and reflexion maps, and so on. He finishes one set of parameters, wipes his forehead, and starts working on the next set. Now on defining the lights: again, dozens of menu options need to be selected. He renders the scene, looks at the result, and admires his creation. But he is far from being done: the universe he has in mind is not a still image but an animation, which means that the water has to flow, the grass and leaves have to move under the blow of the wind, and all the creatures also have to move. He sights and opens another set of menus where he has to define the parameters of algorithms that simulate the physics of motion. And on, and on, and on. Finally the world itself is finished and it looks good; but now God wants to create the Man so he can admire his creation. God sights again, and takes from the shelf a particular Maya manuals from the complete set which occupies the whole shelf…)

Of course we are in somewhat better position than God was. He was creating everything for the first time, so he could not borrow things from anywhere. Therefore everything had to be built and defined from scratch. But we are not creating a new universe but instead visually simulating universe that already exists, i.e. physical reality. Therefore computer scientists working on 3D computer graphics techniques have realized early on that in addition to approximating the physics involved they can also sometimes take another shortcut. Instead of defining something from scratch through the algorithms, they can simply *sample* it from existing reality and incorporate these samples in the construction process.

The examples of the application of this idea are the techniques of texture mapping and bump mapping which were introduced already in the second part of the 1970s. With texture mapping, any 2D digital image – which can be a close-up of some texture such as wood grain or bricks, but which can be also anything else, for instance a logo, a photograph of a face or of clouds – is wrapped around a 3D model. This is a very effective way to add visual richness of a real world to a virtual scene. Bump texturing works similarly, but in this case the 2D image is used as a way to quickly add complexity to the geometry itself. For instance, instead of having to manually model all the little cracks and indentations which make up the 3D texture of a concrete wall, an artist can simply take a photograph of an existing wall, convert into a grayscale image, and then feed this image to the rendering algorithm. The algorithm treats grayscale image as a depth map, i.e. the value of every pixel is being interpreted as relative height of the surface. So in this example, light pixels become points on the wall that are a little in front while dark pixels become points that are a little behind. The result is enormous saving in the amount of time necessary to recreate a particular but very important aspect of our physical reality: a slight and usually regular 3D texture found in most natural and many human-made surfaces, from the bark of a tree to a weaved cloth.

Other 3D computer graphics techniques based on the idea of sampling existing reality include reflection mapping and 3D digitizing. Despite the fact that all these techniques have been always widely used as soon as they were invented, many people in the computer graphics field always felt that they were cheating. Why? I think this feeling was there because the overall conceptual paradigm for creating photorealistic computer graphics was to simulate everything from scratch through algorithms. So if you had to use the techniques based on directly sampling reality, you somehow felt that this was just temporary - because the appropriate algorithms were not yet developed or because the machines were too slow. You also had this feeling because once you started to manually sample reality and then tried to include these samples in your perfect algorithmically defined image, things rarely would fit exactly right, and painstaking manual adjustments were required. For instance, texture mapping would work perfectly if applied to a flat surface, but if the surface were curved, inevitable distortion would occur.

Throughout the 1970s and 1980s the "reality simulation" paradigm and "reality sampling" paradigms co-existed side-by-side. More precisely, as I suggested above, sampling paradigm was "imbedded" within reality simulation paradigm. It was a common sense that the right way to create photorealistic images of reality is by simulating its physics as precisely as one could. Sampling existing reality and then adding these samples to a virtual scene was a trick, a shortcut within over wise honest game of mathematically simulating reality in a computer.

Building *The Matrix*

So far we looked at the paradigms of 3D computer graphics field without considering the uses of the 3D images? So what happens if you want to incorporate photorealistic images produced with CG into a film? This introduces a new constraint. Not only every simulated image has to be consistent internally, with the cast shadows corresponding to the light sources, and so on, but now it also has to be consistent with the cinematography of a film. The simulated universe and live action universe have to match perfectly (I am talking here about the "normal" use of computer graphics in narrative films and not the hybrid aesthetics of TV graphics, music videos, etc. which deliberately juxtaposes different visual codes). As can be seen in retrospect, this new constraint eventually changed the relationship between the two paradigms in favor of sampling paradigm. But this is only visible now, after films such as *The Matrix* made the sampling paradigm the basis of its visual universe.[135]

At first, when filmmakers started to incorporate synthetic 3D images in films, this did not have any effect on how computer scientists thought about computer graphics. 3D computer graphics for the first time briefly appeared in a feature film in 1980 (*Looker*). Throughout the 1980s, a number of films were made which used computer images but always only as a small element within the overall film narrative. (One exception was *Tron*; released in 1982, it can be compared to *The Matrix* since its narrative universe is situated inside computer and created through computer graphics – but this was an exception.) For instance, one of *Star Track* films contained a scene of a planet coming to life; it was created using CG. (In fact, now commonly used "particle system" was invented for to crate this effect). But this was a single scene, and it had no interaction with all other scenes in the film.

In the early 1990s the situation has started to change. With pioneering films such as *The Abyss* (James Cameron, 1989*)*, *Terminator 2* (James Cameron, 1991*)*, and *Jurassic Park* (Steven Spielberg, 1993) computer generated characters became the key protagonists of feature films. This meant that they would appear in dozens or even hundreds of shots throughout a film, and that in most of these shots computer characters would have to be integrated with real environments and human actors captured via live action photography (such shots are called in the business "live plates.") Examples are the T-100 cyborg character in *Terminator 2: Judgment Day,* or dinosaurs in *Jurassic Park.* These computer-generated characters are situated inside the live action universe that is the result of capturing physical reality via the lens of a film camera. The simulated world is located inside the captured world, and the two have to match perfectly.

As I pointed out in *The Language of New Media* in the discussion of compositing, perfectly aligning elements that come from different sources is one of fundamental challenges of computer-based realism. Throughout the 1990s filmmakers and special effects artists have dealt with this challenge using a variety of techniques and methods. What Gaeta realized earlier than others is that the best way to align the two universes of live action and 3D computer graphics was to build *a single new universe*.[136]

Rather than treating sampling reality as just one technique to be used along with many other "proper" algorithmic techniques of image synthesis, Gaeta and his colleagues turned it into the key foundation of Universal Capture process. The process systematically takes physical reality apart and then systematically reassembles the elements together to create a new software-based representation. The result is a new kind of image that has photographic / cinematographic appearance and level of detail yet internally is structured in a completely different way.

Universal Capture was developed and refined over a three-year period from 2000 to 2003.[137] How does the process work? There are actually more stages and details involved, but the basic procedure is the following.[138] An actor's performance is recorded using five synchronized high-resolution video cameras. "Performance" in this case includes everything an actor will say in a film and all possible facial expressions.[139] (During the production the studio was capturing over 5 terabytes of data each day.) Next special algorithms are used to track each pixel's movement over time at every frame. This information is combined with a 3D model of a neutral expression of the actor captured via a 3D scanner. The result is an animated 3D shape that accurately represents the geometry of the actor's head as it changes during a particular performance. The shape is mapped with color information extracted from the captured video sequences. A separate very high resolution scan of the actor's face is used to create the map of small-scale surface details like pores and wrinkles, and this map is also added to the model. (How is that for hybridity?)

After all the data has been extracted, aligned, and combine, the result is what Gaeta calls a "virtual human" - a highly accurate reconstruction of the captured performance, now available as a 3D computer graphics data – with all the advantages that come from having such representation. For instance, because actor's performance now exists as a 3D object in virtual space, the filmmaker can animate virtual camera and "play" the reconstructed performance from an arbitrary angle. Similarly, the virtual head can be also lighted in any way desirable. It can be also attached to a separately constructed CG body.[140] For example, all the characters which appeared the Burly Brawl scene in *The Matrix* 2 were created by combining the heads constructed via Universal Capture done on the leading actors with CG bodies which used motion capture data from a different set of performers. Because all the characters along with the set were computer generated, this allowed the directors of the scene to choreograph the virtual camera, having it fly around the scene in a way not possible with real cameras on a real physical set.

The process was appropriately named Total Capture because it captures all the possible information from an object or a scene using a number of recording methods – or at least, whatever is possible to capture using current technologies. Different dimensions – color, 3D geometry, reflectivity and texture – are captured separately and then put back together to create a more detailed and realistic representation.

Total Capture is significantly different from the commonly accepted methods used to create computer-based special effects such as keyframe animation and physically based modeling. In the first method, an animator specifies the key positions of a 3D model, and the computer calculates in-between frames. With the second method, all the animation is automatically created by software that simulates the physics underlying the movement. (This method thus represents a particular instance of "reality simulation" paradigm.) For instance, to create a realistic animation of moving creature, the programmers model its skeleton, muscles, and skin, and specify the algorithms that simulate the actual physics involved. Often the two methods are combined: for instance, physically based modeling can be used to animate a running dinosaur while manual animation can be used for shots where the dinosaur interacts with human characters.

When the third Matrix film was being released, the most impressive achievement in physically based modeling was the battle in *The Lord of the Rings: Return of the King* (Peter Jackson, 2003) which involved tens of thousands of virtual soldiers all driven by *Massive* software.[141] Similar to the Non-human Players (or bots) in computer games, each virtual soldier was given the ability to "see" the terrain and other soldiers, a set of priorities and an independent "brain," i.e. a AI program which directs character's actions based on the perceptual inputs and priorities. But in contrast to games AI, Massive software does not have to run in real time. Therefore it can create the scenes with tens and even hundreds of thousands realistically behaving agents (one commercial created with the help of *Massive* software featured 146,000 virtual characters.)

Universal Capture method uses neither manual animation nor simulation of the underlying physics. Instead, it directly samples physical reality, including color, texture and the movement of the actors. Short sequences of an actor's performances are encoded as 3D computer animations; these animations form a library from which the filmmakers can then draw as they compose a scene. The analogy with musical sampling is obvious here. As Gaeta pointed out, his team never used manual animation to try to tweak the motion of character's face; however, just as a musician may do it, they would often "hold" particular expression before going to the next one.[142]

This suggests another analogy – analog video editing. But this is a second-degree editing, so to speak: instead of simply capturing segments of reality on video and then joining them together, Gaeta's method produces complete virtual recreations of particular phenomena – self-contained micro-worlds – which can be then further edited and embedded within a larger 3D simulated space.

Animation as an Idea

The brief overview of the methods of computer graphics that I presented above in order to explain Universal Capture offers good examples of the multiplicity of ways in which animation is used in contemporary moving image culture. If we consider this multiplicity, it is possible to come to a conclusion that "animation" as a separate medium in fact hardly exists anymore. At the same time, the general principles and techniques of putting objects and images into motion developed in nineteenth and twentieth century animation are used much more frequently now than before computerization. But they are hardly ever used by themselves – usually they are combined with other techniques drawn from live action cinematography and computer graphics.

So where does animation start and end today? When you see a Disney or Pixar animated feature or many graphics shorts it is obvious that you are seeing "animation." Regardless of whether the process involves drawing images by hand or using 3D software, the principle is the same: somebody created the drawings or 3D objects, set keyframes and then created in-between positions. (Of course in the course of commercial films, this is not one person but large teams.) The objects can be created in multiple ways and inbetweening can be done manually or automatically by the software, but this does not change the basic logic. The movement, or any other change over time, is defined manually – usually via keyframes (but not always). In retrospect, the definition of movement via keys probably was the essence of twentieth century animation. It was used in traditional cell animation by Disney and others, for stop motion animation by Starevich and Trnka, for the 3D animated shorts by Pixar, and it continues to be used today in animated features that combine traditional cell method and 3D computer animation. And while experimental animators such as Norman McLaren refused keys / inbetweens system in favor of drawing each frame on film by hand without explicitly defining the keys, this did not change the overall logic: the movement was created by hand. Not surprisingly, most animation artists exploited this key feature of animation in different ways, turning it into aesthetics: for instance, exaggerated squash and stretch in Disney, or the discontinuous jumps between frames in McLaren.

What about other ways in which images and objects can be set into motion? Consider for example the methods developed in computer graphics: physically based modeling, particle systems, formal grammars, artificial life, and behavioral animation. In all these methods, the animator does not directly create the movement. Instead it is created by the software that uses some kind of mathematical model. For instance, in the case of physically based modeling the animator may sets the parameters of a computer model which simulates a physical force such as wind which will deform a piece of cloth over a number of frames. Or, she may instruct the ball to drop on the floor, and let the physics model control how the ball will bounce after it hits the floor. In the case of particle systems used to model everything from fireworks, explosions, water and gas to animal flocks and swarms, the animator only has to define initial conditions: a number of particles, their speed, their lifespan, etc.

In contrast to live action cinema, these computer graphics methods do not capture real physical movement. Does it mean that they belong to animation? If we accept that the defining feature of traditional animation was manual creation of movement, the answer will be no. But things are not so simple. With all these methods, the animator sets the initial parameters, runs the model, adjusts the parameters, and repeats this production loop until she is satisfied with the result. So while the actual movement is produced not by hand by a mathematical model, the animator maintains significant control. In a way, the animator acts as a film director – only in this case she is directing not the actors but the computer model until it produces a satisfactory performance. Or we can also compare her to a film editor who is selecting among best performances of the computer model.

James Blinn, a computer scientist responsible for creating many fundamental techniques of computer graphics, once made an interesting analogy to explain the difference between manual keyframing method and physically based modeling.[143] He told the audience at a SIGGRAPH panel that the difference between the two methods is analogous to the difference between painting and photography. In Blinn's terms, an animator who creates movement by manually defining keyframes and drawing inbetween frames is like a painter who is observing the world and then making a painting of it. The resemblance between a painting and the world depends on painter's skills, imagination and intentions. Whereas an animator who uses physically based modeling is like a photographer who captures the world as it actually is. Blinn wanted to emphasize that mathematical techniques can create a realistic simulation of movement in the physical world and an animator only has to capture what is created by the simulation.

Although this analogy is useful, I think it is not completely accurate. Obviously, the traditional photographer whom Blinn had in mind (i.e. before Photoshop) chooses composition, contrast, depth of field, and many other parameters. Similarly, an animator who is using physically based modeling also has control over a large number of parameters and it depends on her skills and perseverance to make the model produce a satisfying animation. Consider the following example from the related area of software art that uses some of the same mathematical methods. Casey Reas, an artist who is well-know both for his own still images and animations and for Processing graphics programming environment he helped to develop, told me that he may spend only a couple of hours writing a software program to create a new work – and then another two years working with the different parameters of the same program and producing endless test images until he is satisfied with the results.[144] So while at first physically based modeling appears to be opposite of traditional animation in that the movement is created by a computer, in fact it should be understood as a hybrid between animation and computer simulation. While the animator no longer directly draws each phase of movement, she is working with the parameters of the mathematical model that “draws” the actual movement.

And what about Universal Capture method as used in *The Matrix*? Gaeta and his colleagues also banished keyframing animation – but they did not used any mathematical modes to automatically generate motion either. As we saw, their solution was to capture the actual performances of an actor (i.e., movements of actor’s face), and then reconstruct it as a 3D sequence. Together, these reconstructed sequences form a library of facial expressions. The filmmaker can then draw from this library, editing together a sequence of expressions (but not interfering with any parameters of separate sequences). It is important to stress that a 3D model has no muscles, or other controls traditionally used in animating computer graphics faces - it is used “as is.”

Just as it is the case when animator employs mathematical models, this method avoids drawing individual movements by hand. And yet, its logic is that of animation rather than of cinema. The filmmaker chooses individual sequences of actors’ performances, edits them, blends them if necessary, and places them in a particular order to create a scene. In short, the scene is actually constructed by hand even though its components are not. So while in traditional animation the animator draws each frame to create a short sequence (for instance, a character turning his head), here the filmmaker "draws" on a higher level: manipulating whole sequences as opposed to their individual frames.

To create final movie scenes, Universal Capture is combined with Virtual Cinematography: staging the lighting, the positions and movement of a virtual camera that is "filming" the virtual performances. What makes this Virtual Cinematography as opposed to simply "computer animation" as we already know it? The reason is that the world as seen by a virtual camera is different from a normal world of computer graphics. It consists from reconstructions of the actual set and the actual performers created via Universal Capture. The aim is to avoid manual processes usually used to create 3D models and sets. Instead, the data about the physical world is captured and then used to create a precise virtual replica.

Ultimately, ESC's production method as used in *Matrix* is neither "pure" animation, nor cinematography, nor traditional special effects, nor traditional CG. Instead, it is "pure" example of hybridity in general, and "deep remixability" in particular. With its complex blend of the variety of media techniques and media formats, it is also typical of moving image culture today. When the techniques drawn from these different media traditions are brought together in a software environment, the result is not a sum of separate components but a variety of hybrid methods - such as Universal Capture. As I already noted more than once, I think that this how different moving image techniques function now in general. After computerization virtualized them – "extracting" them from their particular physical media to turn into algorithms – they start interacting and creating hybrids. While we have already encountered various examples of hybrid techniques, Total Capture and

Virtual Cinematography illustrate how creative industries today develop whole production workflow based on hybridity.

It is worthwhile here to quote Gaeta who himself is very clear that what he and his colleagues have created is a new hybrid. In 2004 interview, he says: "If I had to define virtual cinema, I would say it is somewhere between a live-action film and a computer-generated animated film. It is computer generated, but it is derived from real world people, places and things."[145] Although Universal Capture offers a particularly striking example of such "somewhere between," most forms of moving image created today are similarly "somewhere between," with animation being one of the coordinate axises of this new space of hybridity.

"Universal Capture": Reality Re-assembled

The method which came to be called "Universal Capture" combines the best of two worlds: visible reality as captured by lens-based cameras, and synthetic 3D computer graphics. While it is possible to recreate the richness of the visible world through manual painting and animation, as well as through various computer graphics techniques (texture mapping, bump mapping, physical modeling, etc.), it is expensive in terms of labor involved. Even with physically based modeling techniques endless parameters have to be tweaked before the animation looks right. In contrast, capturing visible reality via lens-based recording (the process which in the twentieth century was called "filming") is cheap: just point the camera and press "record" button.

The disadvantage of such lens-based recordings is that they lack flexibility demanded by contemporary remix culture. *Remix culture demands not self-contained aesthetic objects or self-contained records of reality but smaller units - parts that can be easily changed and combined with other parts in endless combinations.* However, lens-based recording process flattens the semantic structure of reality.

Instead of a set of unique objects which occupy distinct areas of a 3D physical space, we end up with a flat field of made from pixels (or film grains in the case of film-based capture) that do not carry any information of where they came from, i.e., which objects they correspond to. Therefore, any kind of spatial editing operation – deleting objects, adding new ones, compositing, etc – becomes quite difficult. Before anything can be done with an object in the image, it has to be manually separated from the rest of the image by creating a mask. And unless an image shows an object that is properly lighted and shot against a special blue or green background, it is practically impossible to mask the object precisely.

In contrast, 3D computer generated worlds have the exact flexibility one would expect from media in information age. (It is not therefore accidental that 3D computer graphics representation – along with hypertext and other new computer-based data representation methods – was conceptualized in the same decade when the transformation of advanced industrialized societies into information societies became visible.) In a 3D computer generated worlds everything is discrete. The world consists from a number of separate objects. Objects are defined by points described as XYZ coordinates; other properties of objects such as color, transparency and reflectivity are similarly described in terms of discrete numbers. As a result, while a 3D CG representation may not have the richness of a lens-based recording, it does contain a semantic structure of the world. This structure is easily accessible at any time. A designer can directly select any object (or any object part) in the scene. Thus, to duplicate an object hundred times requires only a few mouse clicks or typing a short command; similarly, all other properties of a world can be always easily changed. And since each object itself consists from discrete components (flat polygons or surface patches defined by splines), it is equally easy to change its 3D form by selecting and manipulating its components. In addition, just as a sequence of genes contains the code that is expanded into a complex organism, a compact description of a 3D world that contains only the coordinates of the objects can be quickly transmitted through the network, with the client computer reconstructing the full world (this is how online multi-player computer games and simulators work).

Universal Capture brings together the complementary advantages of lens-based capture and CG representation in an ingenious way. Beginning in the late 1970s when James Blinn introduced CG technique of texture mapping[146], computer scientists, designers and animators were gradually expanding the range of information that can be recorded in the real world and then incorporated into a computer model. Until the early 1990s this information mostly involved the appearance of the objects: color, texture, light effects. The next significant step was the development of motion capture. During the first half of the 1990s it was quickly adopted in the movie and game industries. Now computer synthesized worlds relied not only on sampling the visual appearance of the real world but also on sampling of movements of animals and humans in this world. Building on all these techniques, Gaeta’s method takes them to a new stage: capturing just about everything that at present can be captured and then reassembling the samples to create a digital - and thus completely malleable - recreation. Put in a larger context, the resulting 2D / 3D hybrid representation perfectly fits with the most progressive trends in contemporary culture which are all based on the idea of a hybrid.

The New Hybrid

It is my strong feeling that the emerging “information aesthetics” (i.e., the new cultural features specific to information society) already has or will have a very different logic from what modernism. The later was driven by a strong desire to erase the old - visible as much in the avant-garde artists’ (particularly the futurists) statements that museums should be burned, as well as in the dramatic destruction of all social and spiritual realities of many people in Russia after the 1917 revolution, and in other countries after they became Soviet satellites after 1945. Culturally and ideologically, modernists wanted to start with "tabula rasa," radically distancing them from the past. It was only in the 1960s that this move started to feel inappropriate, as manifested both in loosening of ideology in communist countries and the beginnings of new post-modern sensibility in the West. To quote the title of a famous book by Robert Venturi, Denise Scott Brown, and Steven Izenour (published in 1972, it was the first systematic manifestation of new sensibility), *Learning from Las Vegas* meant admitting that organically developing vernacular cultures involves bricolage and hybridity, rather than purity seen for instance in "international style" which was still practiced by architects world-wide at that time. Driven less by the desire to imitate vernacular cultures and more by the new availability of previous cultural artifacts stored on magnetic and soon digital media, in the 1980s commercial culture in the West systematically replaced purity by stylistic heterogeneity. Finally, when Soviet Empire collapsed, post-modernism has won world over.

Today we have a very real danger of being imprisoned by new "international style" - something which we can call the new "global style" The cultural globalization, of which cheap airline flights, the web, and billions of mobile phones are two most visible carriers, erases some dimensions of the cultural specificity with the energy and speed impossible for modernism. Yet we also witness today a different logic at work: the desire to creatively place together old and new – local and transnational - in various combinations. It is this logic, for instance, which made cities such as Barcelona (where I talked with John Gaeta in the context of Art Futura 2003 festival which led to this article), such a "hip" and "in" place at the turn of the century (that is, 20th to 21st). All over Barcelona, architectural styles of many past centuries co-exist with new "cool" spaces of bars, lounges, hotels, new museums, and so on. Medieval meets multi-national, Gaudy meets Dolce and Gabana, Mediterranean time meets global time. The result is the invigorating sense of energy which one feels physically just walking along the street. It is this hybrid energy, which characterizes in my view the most interesting cultural phenomena today.[147] The hybrid 2D / 3D image of *The Matrix* is one such hybrids.

The historians of cinema often draw a contrast between the Lumières and Marey. Along with a number of inventors in other countries all working independently from each other, the Lumières created what we now know as cinema with its visual effect of continuous motion based on the perceptual synthesis of discrete images. Earlier Maybridge already developed a way to take successive photographs of a moving object such as horse; eventually the Lumières and others figured out how to take enough samples so when projected they perceptually fuse into continuous motion. Being a scientist, Marey was driven by an opposite desire: not to create a seamless illusion of the visible world but rather to be able to understand its structure by keeping subsequent samples discrete. Since he wanted to be able to easily compare these samples, he perfected a method where the subsequent images of moving objects were superimposed within a single image, thus making the changes clearly visible.

The hybrid image of *The Matrix* in some ways can be understand as the synthesis of these two approaches which for a hundred years ago remained in opposition. Like the Lumières, Gaeta’s goal is to create a seamless illusion of continuous motion. In the same time, like Marey, he also wants to be able to edit and sequence the individual recordings of reality.

In the beginning of this chapter I evoked the notion of uneven development, pointing that often the structure inside (“infrastructure”) completely changes before the surface (“superstructure”) catches up. What does this idea imply for the future of images and in particular 2D / 3D hybrids as developed by Gaeta and others? As Gaeta pointed out in 2003, while his method can be used to make all kinds of images, so far it was used in the service of realism as it is defined in cinema – i.e., anything the viewer will see has to obey the laws of physics.[148] So in the case of *The Matrix,* its images still have traditional "realistic" appearance while internally they are structured in a completely new way. In short, we see the old "superstructure" which stills sits on top of "old" infrastructure. What kinds of images would we see then the superstructure" would finally catch up with the infrastructure?

Of course, while the images of Hollywood special effects movies so far follow the constraint of realism, i.e. obeying the laws of physics, they are also continuously expanding the boundaries of what "realism" means. In order to sell movie tickets, DVDs, and all other merchandise, each new special effects film tries to top the previous one showing something that nobody has seen before. In *The Matrix* 1 it was "bullet time"; in *The Matrix* 2 it was the Burly Brawl scene where dozens of identical clones fight Neo; in *Matrix 3* it was the Superpunch.[149] The fact that the image is constructed differently internally does allow for all kinds of new effects; listening to Gaeta it is clear that for him the key advantage of such image is the possibilities it offers for virtual cinematography. That is, if before camera movement was limited to a small and well-defined set of moves – pan, dolly, roll – now it can move in any trajectory imaginable for as long as the director wants. Gaeta talks about the Burly Brawl scene in terms of *virtual choreography*: both choreographing the intricate and long camera moves impossible in the real word and also all the bodies participating in the flight (all of them are digital recreations assembled using Total Capture method). According to Gaeta, creating this one scene took about three years. So while in principle Total Capture represents one of the most flexible way to recreate visible reality in a computer so far, it will be years before this method is streamlined and standardized enough for these advantages to become obvious. But when it happens, the artists will have an extremely flexible hybrid medium at their disposal: completely virtualized cinema. Rather than expecting that any of the present pure forms will dominate the future of visual culture, I think this future belongs to such hybrids. In other words, the future images would probably be still photographic – although only on the surface.

And what about animation? What will be its future? As I have tried to explain, besides animated films proper and animated sequences used as a part of other moving image projects, animation has become a set of principles and techniques which animators, filmmakers and designers employ today to create new techniques, new production methods and new visual aesthetics. Therefore, I think that it is not worthwhile to ask if this or that visual style or method for creating moving images which emerged after computerization is "animation" or not. It is more productive to say that most of these methods were born from animation and have animation DNA – mixed with DNA from other media. I think that such a perspective which considers "animation in an extended field" is a more productive way to think about animation today, and that it also applies to other modern media fields which "donated" their genes to a computer metamedium.

PART 3: Webware

Chapter 5. What Comes After Remix?

Introduction

It is always more challenging to think theoretically about the present than the past. But this challenge is what also makes it very exiting.

In Part 2 we looked at the interface and tools of professional media authoring software that were largely shaped in the 1990s. While each major release of Photoshop, Flash, Maya, Flame, and other commonly used applications continues to introduce dozens of new features and improvements, in my view these are incremental improvements rather than new paradigms.

The new paradigms that emerge in the 2000s are not about new types of media software per ce. Instead, they have to with the exponential expansion of the number of people who now use it – and the web as a new universal platform for non-professional media circulation. "Social software," "social media," "user-generated content," "Web 2.0," "read/write Web" are some of the terms that were coined in this decade to capture these developments.

If visual communication professionals have adopted software-based tools and workflows throughout the 1990s, in the next decade "media consumers" were gradually turned into "media producers." The decline in prices and increase in the media capabilities of consumer electronics (digital cameras, media players, mobile phones, laptops) combined with the ubiquity of the internet access combined with the emergence of new social media platforms have created a whole new media ecology and dynamics. In retrospect, if we can designate 1995 as the year of professional media revolution (for example, version 3 of After Effects released this year added Illustrator and Photoshop layers import), I would center consumer media revolution on 2005. During this year, photo and video blogging have exploded; the term "user-generated content" entered mainstream; YouTube was started; and both Flickr was bought by Yahoo, while MySpace wer acquired by larger companies (Yahoo and Rupert Murdoch's News Corporation, respectively.)

If the professional media revolution of the 1990s can be identified with a small set of software applications, the cultural software which enables new media ecology emerging in the middle of 2000s is much more diverse and heterogeneous. Media sharing sites (Flickr), social networking sites (Facebook), webware such as Google Docs, APIs of major Web 2.0 companies, RSS readers, blog publishing software (Blogger), virtual globes (Google Earth, Microsoft Virtual Earth), consumer-level media editing and cataloging software (iPhoto), media and communication software running on mobile phones and other consumer electronics devices, and, last but not least, search engines are just some of the categories. (Of course, each brand name appearing in brackets in the preceding sentence is just one example of a whole software category.) Add to these other software categories which are not directly visible to consumers but which are responsible for networked-based media universe of sharing, remixing, collaboration, blogging, reblogging, and so on – everything from web services and client-server architecture to Ajax and Flex – and the task of tracking cultural software today appears to be daunting. But not impossible.

The two chapters of this part of the book consider different dimensions of the new paradigm of user-generated content and media sharing which emerged in 2000s. As before, my focus is on the relationships between the affordances provided by software interfaces and tools, the aesthetics and structure of media objects created with their help, and the theoretical impact of software use on the very concept of media. (In other words: what is "media" after software?) One key difference from Part 2, however, is that instead of dealing with separate media design applications, we now have to consider larger media environments which integrates the functions of creating media, publishing it, remixing other people' media, discussing it, keeping up with friends and interest groups, meeting new people, and so on.

I look at the circulation, editing and experience of media as structured by web interfaces. Given that the term *remix* has already been widely used in discussing social media, I use it as a starting point in my own investigation. Similarly to how I did this in the discussion of software-based media design in Part 2, here I am also interested in both revealing the parallels and highlighting the differences between "remix culture" in general and software-enabled remix operations in particular. (If we don't do this and simply refer to everything today as "remix," we are not really trying to explain things anymore – we are just labeling them.) I also discuss other crucial dimensions of the new universe of social media: *modularity* and *mobility*. (Mobility here refers not to the movement of individuals and groups or accessing media from mobile devices, but to something else which so far has not been theoretically acknowledged: *the movement of media objects between people, devices, and the web*.)

I continue by examining some of the new types of user-to-user visual media communication which emerged on social media platforms. I conclude by asking how the explosion of user-generated content challenges professional cultural producers – not the media industries (since people in the industry, business and press are already discussing this all the time) - but rather another cultural industry which has been the slowest to respond to the social web – professional art world.

Given the multitude of terms already widely used describe the new developments of 2000s and the new concepts we can develop to fill the gaps, is there a single concept that would sum it all? The answers to this question would of course vary widely, but here is mine. For me, this concept is *scale*. The exponential growth of a number of both non-professional and professional media producers during 2000s has created a fundamentally new cultural situation. Hundreds of millions of people are routinely created and sharing cultural content (blogs, photos, videos, online comments and discussions, etc.). This number is only going to increase. (During 2008 the number of mobile phones users' is projected to grow from 2.2 billion to 3 billion).

A similar explosion in the number of media professionals has paralleled this explosion in the number of non-professional media producers. The rapid growth of professional, educational, and cultural institutions in many newly globalized countries, along with the instant availability of cultural news over the web, has also dramatically increased the number of "culture professionals" who participate in global cultural production and discussions. Hundreds of thousands of students, artists and designers now have access to the same ideas, information and tools. It is no longer possible to talk about centers and provinces. In fact, the students, culture professionals, and governments in newly globalized countries are often more ready to embrace latest ideas than their equivalents in "old centers" of world culture.

Before, cultural theorists and historians could generate theories and histories based on small data sets (for instance, "classical Hollywood cinema," "Italian Renaissance," etc.) But how can we track "global digital culture" (or cultures), with its billions of cultural objects, and hundreds of millions of contributors? Before you could write about culture by following what was going on in a small number of world capitals and schools. But how can we follow the developments in tens of thousands of cities and educational institutions?

If the shift from previous media technologies and distribution platforms to software has challenged our most basic concepts and theories of "media," the new challenge in my view is even more serious. Let's say I am interested in thinking about cinematic strategies in user-generated videos on YouTube. There is no way I can manually look through all the billions of videos there. Of course, if I watch some of them, I am likely to notice some patterns emerging.. but how do I know which patterns exist in all the YouTube videos I never watched? Or, maybe I am interested in the strategies in the works of design students and young professionals around the world. The data itself is available: every design school, studio, design professional and a student have their stuff on the web. I can even consult special web sites such as colorflot.com that contains (as of this writing) over 100,000 design portfolios submitted by designers and students from many countries. So how do I go about studying 100,000+ portfolios?

I don't know about you, but I like challenges. In fact, my lab is already working on how we can track and analyze culture at a new scale that involve hundreds of millions of producers and billions of media objects. (You can follow our work at softwarestudies.com and culturevis.com.) The first necessary step, however, is to put forward some conceptual coordinates for the new universe of social media – an initial set of hypothesis about its new features which later can be improved on.

And this is what this chapter is about. Let's dive in.

"The Age of Remix"

It is a truism that we live in a “remix culture.” Today, many cultural and lifestyle arenas - music, fashion, design, art, web applications, user created media, food - are governed by remixes, fusions, collages, and mash-ups. If post-modernism defined 1980s, remix definitely dominates 1990s and 2000s, and it will probably continue to rule the next decade as well. (For an expanding resource on remix culture, visit remixtheory.net by Eduardo Navas.) Here are just a few examples. In his winter collection John Galliano (a fashion designer for the house of Dior) mixes vagabond look, Yemenite traditions, East-European motifs, and other sources that he collects during his extensive travels around the world (2004 collection). DJ Spooky creates a feature-length remix of D.W. Griffith's 1912 "Birth of a Nation” which he appropriately names "Rebirth of a Nation." The group BOOM BOOM SATELLITES initiates a remix competition aimed at bringing together two cultures: “the refined video editing techniques of AMV enthusiasts” and “the cutting-edge artistry of VJ Culture” (2008).[150] The celebrated commentator on copyright law and web culture Lawrence Lessig names his new book *Remix: Making Art and Commerce Thrive in the Hybrid Economy* (2008.)

The Web in particular has become a breeding ground for variety of new remix practices. In April 2006 Annenberg Center at University of Southern California run a conference on “Networked Politics” which put forward a useful taxonomy of some of these practices: political remix videos, anime music videos, machinima, alternative news, infrastructure hacks.[151] In addition to these cultures that remix media content, we also have a growing number of “software mash-ups,” i.e. software applications that remix data. (In case you skipped Part 1, let me remind you that, in Wikipedia definition, a mash-up as “a website or application that combines content from more than one source into an integrated experience.”[152] As of March 1, 2008, the web site www.programmableweb.com listed the total of 2814 software mash-ups, and approximately 100 new mash-ups were created every month.[153]

Yet another type of remix technology popular today is RSS. With RSS, any information source which is periodically updated – a personal blog one's collection of photos on Flickr, news headlines, podcasts, etc. – can be published in a standard format, i.e., turned into a "feed.") Using RSS reader, an individual can subsribe to such feeds - create her custom mix selected from many millions of feeds available. Alternatively, you can use widget-based feed readers such as iGoogle, My Yahoo, or Netvibes to create a personalized home page that mixes feeds, weather reports, Facebook friends updates, podcasts, and other types of information sources. (Appropriately, Netvibes includes the words "re(mix) the web" in its logo.)

Given the trends towards ubiquitous computing and "Internet of things," it is inevitable that remixing paradigm will make its way into physical space as well. Bruce Sterling's brilliant book *Shaping Things* describes a possible future scenario where objects publish detailed information about their history, use, and impact on the environment, and ordinary consumers track this information.[154] I imagine a future RSS reader may give you a choice of billions of objects to track. (If you were already feeling overwhelmed by 112 million blogs tracked by Technorati [xxx check the spelling] - as of December 2007 - this is just a beginning.[155])

For a different take on how a physical space – in this case, a city - can reinvent itself via remix, consider coverage of Buenos Aires by *The,* the journal by "trend and future consultancy" The Future Laboratory.[156] *The* enthusiastically describes the city in remix terms – and while the desire to project a fashionable term on everything in site is obvious, the result is actually mostly convincing. The copy reads as follows: "Buenos Aires has gone mash-up. The portefnos are adopting their traditions with some American sauce and European pepper." A local DJ

Villa Diamante released an album that "mixes electronic music with cumcia, South American peasant music." A clothing brand 12-na "mixes flea-market finds with modern materials. A non-profit publication project Eloisa Cartonea "combines covers painted by kids who collect the city's cardboard with the work of emerging writers and poets."

Remix practices extend beyond particular technologies and areas of culture. *Wired* magazine devoted its July 2005 issue to the theme Remix Planet. The introduction boldly stated: "From *Kill Bill* to Gorillaz, from custom Nikes to *Pimp My Ride*, this is the age of the remix."[157] Another top IT trend watcher in the world – the annual O'Reilly Emerging Technology conferences (ETECH) similarly adopted Remix as the theme for its 2005 conference. Attending the conference, I watched in amazement how top executives from Microsoft, Yahoo, Amazon, and other leading IT companies not precisely known for their avant-garde aspirations described their recent technologies and research projects using the concept of remix. If I had any doubts that we are living not simply in Remix Culture but in a Remix Era, they disappeared right at that conference.

Remix, Appropriation, Quotation, Montage

"Remixing" originally had a precise and a narrow meaning limited to music. Although precedents of remixing can be found earlier, it was the introduction of multi-track mixers that made remixing music a standard practice. With each element of a song – vocals, drums, etc. – available for separate manipulation, it became possible to "re-mix" the song: change the volume of some tracks or substitute new tracks for the old ounces. Gradually the term became more and more broad, today referring to any reworking of already existing cultural work(s).

In his book *DJ Culture* Ulf Poscardt singles out different stages in the evolution of remixing practice. In 1972 DJ Tom Moulton made his first disco remixes; as Poscard points out, they "show a very chaste treatment of the original song. Moulton sought above all a different weighting of the various soundtracks, and worked the rhythmic elements of the disco songs even more clearly and powerfully…Moulton used the various elements of the sixteen or twenty-four track master tapes and remixed them."[158] By 1987, "DJs started to ask other DJs for remixes" and the treatment of the original material became much more aggressive. For example, "Coldcut used the vocals from Ofra Hanza's 'Im Nin Alu' and contrasted Rakim's ultra-deep bass voice with her provocatively feminine voice. To this were added techno sounds and a house-inspired remix of a rhythm section that loosened the heavy, sliding beat of the rap piece, making it sound lighter and brighter."[159]

Around the turn of the century (20th to 21st) people started to apply the term "remix" to other media besides music: visual projects, software, literary texts. Since, in my view, electronic music and software serve as the two key reservoirs of new metaphors for the rest of culture today, this expansion of the term is inevitable; one can only wonder why it did no happen earlier. Yet we are left with an interesting paradox: while in the realm of commercial music remixing is officially accepted[160], in other cultural areas it is seen as violating the copyright and therefore as stealing. So while filmmakers, visual artists, photographers, architects and Web designers routinely remix already existing works, this is not openly admitted, and no proper terms equivalent to remixing in music exist to describe these practices.

One term that is sometimes used to talk about these practices in non-music areas is "appropriation." The term was first used to refer to certain New York-based "post-modern" artists of the early 1980s who re-worked older photographic images – Sherrie Levine, Richard Prince, Barbara Kruger, and a few others. But the term "appropriation" never achieved the same wide use as "remixing." In fact, in contrast to “remix,” “appropriation” never completely left its original art world context where it was coined. I think that “remixing” is a better term anyway because it suggests a systematic re-working of a source, the meaning which “appropriation” does not have. And indeed, the original “appropriation artists” such as Richard Prince simply copied the existing image as a whole rather than re-mixing it. As in the case of Duchamp’s famous urinal, the aesthetic effect here is the result of a transfer of a cultural sign from one sphere to another, rather than any modification of a sign.

The other older term commonly used across media is “quoting” but I see it as describing a very different logic than remixing. If remixing implies systematically rearranging the whole text, quoting refers to inserting some fragments from old text(s) into the new one. Therefore, I don’t think that we should see quoting as a historical precedent for remixing. Rather, we can think of it as a precedent for another new practice of authorship practice that, like remixing, was made possible by electronic and digital technology – sampling.

Music critic Andrew Goodwin defined sampling as “the uninhibited use of digital sound recording as a central element of composition. Sampling thus becomes an aesthetic programme.”[161] It is tempting to say that the arrival of sampling technologies has industrialized the practices of montage and collage that were always central to twentieth century culture. Yet we should be careful in applying the old terms to new technologically driven cultural practices. While it is comforting to see the historical continuities, it is also too easy to miss new distinctive features of the present. The use of terms “montage” and “collage” in relation to the sampling and remixing practices is a case in point. These two terms regularly pop up in the writings of music theorists from Poscardt to DJ Spooky and Kodwo Eshun. (In 2004 Spooky published brilliant book *Rhythm Science*[162] which ended up on a number of “best 10 books of 2004” lists and which put forward "unlimited remix" as the artistic and political technique of our time).

The terms "montage" and "collage" come to us from literary and visual modernism of the early twentieth century – think for instance of works by Moholy-Nagy, Sergey Eisenstein, Hannah Hooch or Raoul Hausmann. In my view, they do not always adequately describe contemporary electronic music. Let me note just three differences. Firstly, musical samples are often arranged in loops. Secondly, the nature of sound allows musicians to mix pre-existent sounds in a variety of ways, from clearly differentiating and contrasting individual samples (thus following the traditional modernist aesthetics of montage/collage), to mixing them into an organic and coherent whole. To borrow the terms from Roland Barthes we can say that if modernist collage always involved a "clash" of element, electronic and software collage also allows for "blend."[163] Thirdly, the electronic musicians now often conceive their works beforehand as something that will be remixed, sampled, taken apart and modified. In other words, rather than sampling from mass media to create a unique and final artistic work (as in modernism), contemporary musicians use their own works and works by other artists in further remixes.

It is relevant to note here that the revolution in electronic pop music that took place in the second part of the 1980s was paralleled by similar developments in pop visual culture. The introduction of electronic editing equipment such as switcher, keyer, paintbox, and image store made remixing and sampling a common practice in video production towards the end of the decade. First pioneered in music videos, it eventually later took over the whole visual culture of TV. Other software tools such as Photoshop (1989) and After Effects (1993) had the same effect on the fields of graphic design, motion graphics, commercial illustration and photography. And, a few years later, World Wide Web redefined an electronic document as a mix of other documents. Remix culture has arrived.

The question that at this point is really hard to answer is what comes after remix? Will we get eventually tired of cultural objects - be they dresses by Alexander McQueen, motion graphics by MK12 or songs by Aphex Twin – made from samples which come from already existing database of culture? And if we do, will it be still psychologically possible to create a new aesthetics that does not rely on excessive sampling? When I was emigrating from Russia to U.S. in 1981, moving from grey and red communist Moscow to a vibrant and post-modern New York, me and others living in Russia felt that Communist regime would last for at least another 300 years. But already ten years later, Soviet Union caused to exist. Similarly, in the middle of the 1990s the euphoria unleashed by the Web, collapse of Communist governments in Eastern Europe and early effects of globalization created an impression that we have finally Cold War culture behind – its heavily armed borders, massive spying, and the military-industrial complex. And once again, only ten years later it appeared that we are back in the darkest years of Cold War - except that now we are being tracked with RFID chips, computer vision surveillance systems, data mining and other new technologies of the twenty first century. So it is very possible that the remix culture, which right now appears to be so firmly in place that it can’t be challenged by any other cultural logic, will morph into something else sooner than we think.

I don’t know what comes after remix. But if we now try now to develop a better historical and theoretical understanding of remix era and the technological platforms which enable it, we will be in a better position to recognize and understand whatever new era which will replace it.

Communication in a “Cloud”

During 2000s remix gradually moved from being one of the options to being treated as practically a new cultural default. The twentieth century paradigm in which a small number of professional producers send messages over communication channels that they also controlled to a much larger number of users was replaced by a new paradigm.[164] In this model, *a much large number of producers publish content into "a global media cloud"; the users create personalized mixes by choosing from this cloud.*[165] A significant percentage of these producers and users overlap - i.e. they are the same people. Furthermore, a user can also select when and where to view her news – a phenomenon that has come to be known as "timeshifting" and "placeshifting." Another feature of the new paradigm, which I will discuss in detail below, is what I call "media mobility." A message never arrives at some final destination as in broadcasting / mass publishing model. Instead, a message continues to move between sites, people, and devices. As it moves, it accumulates comments and discussions. Frequently, its parts are extracted and remixed with parts of other messages to create new messages.

The arrival of a new paradigm has been reflected in and supported by a set of new terms. Twentieth century terms "broadcasting" and "publishing" and "reception" have been joined (and in many contexts, replaced), by new terms that describe new operations now possible in relation to media messages. They include "embed," "annotate," "comment," "respond," "syndicate," "aggregate," "upload," "download," "rip," and "share."

There are a number of interesting things worth noting in relation to this new vocabulary, Firstly, the new terms are more discriminating than the old ones as they now name many specific operations involved in communication. You don't simply "receive" a message; you can also annotate it, comment on it, remix it, etc. Secondly, most of the new terms describe new types of users' activities which were either not possible with the old media or were strictly marginal (For instance, a marginal practice of "slash" videos made by science fiction fans.) Thirdly, if old terms such as "read," "view" and "listen" were media-specific, the new ones are not. For instance, you can “comment” on a blog, a photo, a video, a slide show, a map, etc. Similarly, you can “share” a video, a photo, an article, a map layer, and so on. This media-indifference of the terms indirectly reflects the media-indifference of the underlying software technologies. (As I have already discussed in depth earlier, the important theme in the development of cultural software has been the development of new information management principles and techniques – such as Englebardt’s “view control” – which work in the same way on many types of media.)

Among these new terms, “remix” (or “mix”) occupies a major place. As the user-generated media content (video, photos, music, maps) on the Web exploded in 2005, an important semantic switch took place. The terms “remix” (or “mix”) and “mashup” started to be used in contexts where previously the term “editing” had been standard – for instance, when referring to a user editing a video. When in the spring of 2007 Adobe released video editing software for users of the popular media sharing web site Photobucket, it named the software Remix. (The software was actually a stripped down version of one of the earliest video editing applications for PCs called Premiere.[166]) Similarly, Jumpcut, a free video editing and hosting site, does not use the word “edit.”[167] Instead, it puts forward “remix” as the core creative operation: “You can create your own movie by remixing someone else's movie.” Other online video editing and hosting services which also use the term “remix”, or “mashup” instead of “edit” (and which existed at least when I was writing this chapter in the Spring 2008) include eyespot and Kaltura.[168]

The new social communication paradigm where millions are publishing “content” into the “cloud” and an individual curates her personal mix of content drawn from this cloud would be impossible without new types of consumer applications, new software features and underlying software standards and technologies such as RSS. To make a parallel with the term "cloud computing," we can call this paradigm "communication in a cloud." If "cloud computing" enables users and developers to utilize [IT] services without knowledge of, expertise with, nor control over the technology infrastructure that supports them,"[169] software developments of 2000s similarly enable content creators and content receivers to communicate without having to deeply understand underlying technologies.

Another reason why a metaphor of a " cloud" – which at first appears vague – may also be better for describing communication patterns communication in 2000s than the "web" has changed do with the changes in the patterns of information flow between the original Web and so-called Web 2.0. In the original web model, information was published in the form of web pages collected into web sites. To receive information, a user had to visit each site individually. You could create a set of bookmarks for the sites you wanted to come back to, or a separate page containing the links to these sites (so-called "favorites") - but this was all. The lack of a more sophisticated technology for "receiving" the web was not an omission on the part of the web's architect Tim Berners-Lee – it is just that nobody anticipated that the number of web sites will explode exponentially. (This happened after first graphical browsers were introduced in 1993. In 1998 First Google index collected 26 million pages; in 2000 it already had one billion; on June 25, 2008, Google engineers announced on Google blog that they collected one trillion unique URLs…[170])

In the new communication model that has been emerging after 2000, information is becoming more *atomized*. *You can access individual atoms of information without having to read/view the larger packages in which it is enclosed* (a TV program, a music CD, a book, a web site, etc.) Additionally, *information is gradually becoming presentation and device independent* – it can be received using a variety of software and hardware technologies and stripped from its original format. Thus, while web sites continue to flourish, it is no longer necessary to visit each site individually to access their content. With RSS and other web feed technologies, any periodically changing or frequently updated content can be syndicated (i.e., turned into a feed, or a channel), and any user can subscribe to it. Free blog software such as Blogger and WordPress automatically create RSS feeds for elements of a blog (posts, comments). Feeds can be also be created for parts of web sites (using tools such as feedity.com), weather data, search results, Flickr's photo galleries, YouTube channels, and so on. For instance, let's say you go and register for a Flickr account. After you do that, Flickr automatically creates a feed for your photos. So when you upload photos to your Flickr account – which you can do from your laptop, mobile phone or (in some cases) directly from a digital camera – people who subscribed to your feed will automatically get all your new photos.

The software technologies used to send information into the cloud are complemented by software that allows people to curate (or "mix") the information sources they are interested in. Software in this category is referred to as newsreaders, feed readers, or aggregators. Examples include separate web-based feed readers such as Bloglines and Google Reader; all popular web browsers that also provide functions to read feeds; desktop-based feed-readers such as NetNewsWire; and personalized home pages such as live.com, iGoogle, my Yahoo!

Finally, If feed technologies turned the original web of interlinked web pages sites into a more heterogeneous and atomized global "cloud" of content, other software developments helped to make this cloud rapidly grow in size.[171] It is not accidental that during the period when "user generated media" started to grow exponentially, the interfaces of most consumer-level media applications came to prominently feature buttons and options which allow for to move new media documents into the "cloud" – be they PowerPoint presentations, PDF files, blog posts, photographs, video, etc. For example, iPhoto '08 groups functions which allow the user to email photos, or upload them to her blog or website (under a top level "Share" menu). Similarly, Windows Live Photo Gallery includes "Publish" and "E-mail" among its top menu bar choices. Meanwhile, the interfaces of social media sites were given buttons to easily move content around the "cloud," so to speak – emailing it to others, embedding it in one's web site or blog, linking it, posting to one's account on other popular social media sites, etc.

Regardless of how easy it is to create one personal mix of information sources – even if only takes a single click – the practically unlimited number of these sources now available in the "cloud" means that manual ways of selecting among these sources become limited in value. Enter the automation. From the very beginning, computers were used to automate various processes. Over time, everything - factory work, flying planes, financial trading, or cultural processes - is gradually subjected to automation.[172] However, algorithmic automated reasoning on the Web arrived so quickly that it hardly even been publically discussed. We take it for granted that Google and other search engines automatically process tremendous amounts of data to deliver search results. We also take it for granted that Google's algorithms automatically insert ads in web pages by analyzing pages' content. Flickr uses its own algorithm to select the photos it calls "interesting."[173] Pandora, Musicovery, OWL music search, and many other similar web services automatically create music programs based on the users' musical likes. Digg automatically pushes the stories up based on how many people have voted for them. Amazon and Barnes & Noble use collaborative filtering algorithms to recommend books; Last.fm and iTunes – to recommend music, Netflix – to recommend movies; StumbleUpon – to recommend websites; and so on.[174] (iTunes 8 calls its automation feature Genius sidebar; it is designed to make "playlists in your song library that go great together" and also to recommend "music from the iTunes

Stores that you don't already have.) In contrast to these systems which provide recommedations by looking at the users which have similar rating patterns, Mufin is fully automatic recommendatio system for music which works by matching songs based on 40 attributes such as tempo, instruments, and percussion.[175]

As I write this in the summer of 2008, the use of automation to create mixes from hundreds of millions information sources is just beginning. One already popular service is Google News site that algorithmically assembles "news" by remixing material gathered from thousands of news publications. (As it is usually the case with algorithms used by web companies, when I checked last there was no information on the Google News web site about the algorithm used, so we know nothing about its selection criteria or what counts as important and relevant news.) Newspond similarly automatically aggregates news, and it similarly discloses little about the process. According to its web site, "Newspond's articles are found and sorted by real-time global popularity, using a fully automated news collection engine."[176] Spotplex assembles news from blogosphere using yet another type of automation: counting most read articles within a particular time frame.[177] Going further, news.ask.com not only automatically selects the news but it also provides BigPicture pages for each news story containing relevant articles, blog posts, images, videos, and diggs.[178] News.ask.com also tells us that it selects news stories based on four factors – breaking, impact, media, and discussion – and it actually shows how each story rates in terms of these factors Another kind of algorithmic "news remix" is performed by the web-art application 10x10 by Jonathan Harris. It presents a grid of news images based on the algorithmic analysis of news feeds from *The New York Times*, the BBC, and Reuters.[179]

Remixability And Modularity

The dramatic increase in quantity of information greatly speeded up by the web has been accompanied by another fundamental development. Imagine water running down a mountain. If the quantity of water keeps continuously increasing, it will find numerous new paths and these paths will keep getting wider. Something similar is happening as the amount of information keeps growing - except these paths are also all connected to each other and they go in all directions; up, down, sideways. Here are some of these new paths which facilitate movement of information between people, listed in no particular order: SMS, forward and redirect buttons in email applications, mailing lists, Web links, RSS, blogs, social bookmarking, tagging, publishing (as in publishing one's playlist on a web site), peer-to-peer networks, Web services, Firewire, Bluetooth. These paths stimulate people to draw information from all kinds of sources into their own space, remix and make it available to others, as well as to collaborate or at least play on a common information platform (Wikipedia, Flickr). Barb Dybwad introduces a nice term "collaborative remixability'" to talk about this process: "I think the most interesting aspects of Web 2.0 are new tools that explore the continuum between the personal and the social, and tools that are endowed with a certain flexibility and modularity which enables collaborative remixability — a transformative process in which the information and media we've organized and shared can be recombined and built on to create new forms, concepts, ideas, mashups and services."[180]

If a traditional twentieth century model of cultural communication described movement of information in one direction from a source to a receiver, now the reception point is just a temporary station on information's path. If we compare information or media object with a train, then each receiver can be compared to a train station. Information arrives, gets remixed with other information, and then the new package travels to other destination where the process is repeated.

We can find precedents for this “remixability” – for instance, in modern electronic music where remix has become the key method since the 1980s. More generally, most human cultures developed by borrowing and reworking forms and styles from other cultures; the resulting “remixes” were later incorporated into other cultures. Ancient Rome remixed Ancient Greece; Renaissance remixed antiquity; nineteenth century European architecture remixed many historical periods including the Renaissance; and today graphic and fashion designers remix together numerous historical and local cultural forms, from Japanese Manga to traditional Indian clothing.

At first glance it may seem that remixability as practiced by designers and other culture professionals is quite different from “vernacular” remixability made possible by the software-based techniques described above. Clearly, a professional designer working on a poster or a professional musician working on a new mix is different from somebody who is writing a blog entry or publishing her bookmarks.

But this is a wrong view. The two kinds of remixability – professional and vernacular - are part of the same continuum. For the designer and musician (to continue with the sample example) are equally affected by the same software technologies. Design software and music composition software make the technical operation of remixing very easy; the web greatly increases the ease of locating and reusing material from other periods, artists, designers, and so on. Even more importantly, since every company and freelance professionals in all cultural fields, from motion graphics to architecture to fashion, publish documentation of their projects on their Web sites, everybody can keep up with what everybody else is doing. Therefore, although the speed with which a new original architectural solution starts showing up in projects of other architects and architectural students is much slower than the speed with which an interesting blog entry gets referenced in other blogs, the difference is quantitative than qualitative. Similarly, when H&M or Gap can "reverse engineer" the latest fashion collection by a high-end design label in only two weeks, this is an example of the same cultural remixability speeded up by software and the web. In short, a person simply copying parts of a message into the new email she is writing, and the largest media and consumer company recycling designs of other companies are doing the same thing – they practice remixability.

The remixability does not require modularity (i.e., organization of a cultural objects into clearly separable parts) - but it greatly benefits from it. For example, as already discussed above, remixing in music really took after the introduction of multi-track equipment. With each song element available on its own track, it was not long before substituting tracks become commonplace.

In most cultural fields today we have a clear-cut separation between libraries of elements designed to be sampled – stock photos, graphic backgrounds, music, software libraries – and the cultural objects that incorporate these elements. For instance, a design for a corporate report or an ad may use photographs that the designer purchased from a photo stock house. But this fact is not advertised; similarly, the fact that this design (if it is successful) will be inevitably copied and sampled by other designers is not openly acknowledged by the design field. The only fields where sampling and remixing are done openly are music and computer programming, where developers rely on software libraries in writing new software.

Will the separation between libraries of samples and "authentic" cultural works blur in the future? Will the future cultural forms be deliberately made from discrete samples designed to be copied and incorporated into other projects? It is interesting to imagine a cultural ecology where all kinds of cultural objects regardless of the medium or material are made from Lego-like building blocks. The blocks come with complete information necessary to easily copy and paste them in a new object – either by a human or machine. A block knows how to couple with other blocks – and it even can modify itself to enable such coupling. The block can also tell the designer and the user about its cultural history – the sequence of historical borrowings which led to the present form. And if original Lego (or a typical twentieth century housing project) contains only a few kinds of blocks that make all objects one can design with Lego rather similar in appearance, software can keep track of unlimited number of different blocks.

One popular twentieth century notion of cultural modularity involved artists, designers or architects making finished works from the small vocabulary of elemental shapes, or other modules. Whether we are talking about construction industry, Kandinsky's geometric abstraction, or modular furniture systems, the underlying principle is the same. The scenario I am entertaining proposes a very different kind of modularity that may appear like a contradiction in terms. It is modularity without a priori defined vocabulary. In this scenario, any well-defined part of any finished cultural object can automatically become a building block for new objects in the same medium. Parts can even "publish" themselves and other cultural objects can "subscribe" to them the way you subscribe now to RSS feeds or podcasts.

When we think of modularity today, we assume that a number of objects that can be created in a modular system is limited. Indeed, if we are building these objects from a very small set of blocks, there are a limited number of ways in which these blocks can go together. (Although as the relative physical size of the blocks in relation to the finished object get smaller, the number of different objects which can be built increases: think IKEA modular bookcase versus a Lego set.) However, in my imaginary scenario modularity does not involve any reduction in the number of forms that can be generated. On the contrary, if the blocks themselves are created using one of many already developed software-based designed methods (such as parametric design), every time they are used again they can modify themselves automatically to assure that they look different. In other words, if pre-software modularity leads to repetition and reduction, post-software modularity can produce unlimited diversity.

I think that such "real-time" or "on-demand" modularity can only be imagined today after various large-scale projects created at the turn of the century - online stores such as Amazon, blog indexing services such as Technorati, buildings such as Yokohama International Port Terminal by Foreign Office Architects and Walt Disney Concert Hall in Los Angeles by Frank Gehry - visibly demonstrated that we can develop hardware and software to coordinate massive numbers of cultural objects and their building blocks: books, bog entries, construction parts. Whether we will ever have such a cultural ecology is not important. We often look at the present by placing it within long historical trajectories. But I believe that we can also productively use a different, complementary method. We can imagine what will happen if the contemporary techno-cultural conditions which are already firmly established are pushed to their logical limit. In other words, rather than placing the present in the context of the past, we can look at it in the context of a logically possible future. This "look from the future" approach may illuminate the present in a way not possible if we only "look from the past." The sketch of a logically possible cultural ecology I just made is a little experiment in this method: futurology or science fiction as a method of contemporary cultural analysis.

So what else can we see today if we will look at it from this logically possible future of a "total remixability" and universal modularity? If my scenario sketched above looks like a "cultural science fiction," consider the process that is already happening at one end of remixability continuum. This process is gradual atomization of information on the web that we already touched on earlier in this chapter. New software technologies separate content from particular presentation formats, devices, and the larger cultural "packages" where it is enclosed by the producers. (For instance, consider how iTunes and other online music stores changed the unit of music consumption from a record/CD to a separate music track.) In particular, wide adoption and standardization of feed formats allows cultural bits to move around more easily – changing a web into what I called a "communication cloud." The increased modularity of content allowed for a wide adoption of remix as a preferred way of receiving it (although, as we saw, in many cases it is more appropriate to call the result a collection rather than a true remix.)

The Web was invented by the scientists for scientific communication, and at first it was mostly text and "bare-bones" HTML. Like any other markup language, HTML was based on the principle of modularity (in this case, separating content from its presentation). And of course, it also brought a new and very powerful form of modularity: the ability to construct a single document from parts that may reside on different web servers. During the period of web's commercialization (second part of the 1990s), twentieth century media industries that were used to producing highly structured information packages (books movies, records, etc.) similarly pushed the web towards highly coupled and difficult to take apart formats such as Shockwave and Flash. However, since approximately 2000, we see a strong move in the opposite direction: from intricately packaged and highly designed "information objects" (or "packages") which are hard to take apart – such as web sites made in Flash – to "strait" information: ASCII text files, RSS feeds, blog posts, KML files, SMS messages, and microcontent. As Richard MacManus and Joshua Porter put it in 2005, "Enter Web 2.0, a vision of the Web in which information is broken up into "microcontent" units that can be distributed over dozens of domains. The Web of documents has morphed into a Web of data. We are no longer just looking to the same old sources for information. Now we're looking to a new set of tools to aggregate and remix microcontent in new and useful ways."[181] And it is much easier to "aggregate and remix microcontent" if it is not locked by a design. An ASCII file, a JPEG image, a map, a sound or video file can move around the Web and enter into user-defined remixes such as a set of RSS feed subscriptions; cultural objects where the parts are locked together (such as Flash interface) can't. In short, in the era of Web 2.0, we can state that *information wants to be ASCII*.

This very brief and highly simplified history of the web does not do justice to many other important trends in web evolution. But I do stand by its basic idea. That is, a contemporary "communication cloud" is characterized by a constantly present tension between the desires to "package" information (for instance, use of Flash to create "splash" web pages) and to strip it from all packaging so it can travel easier between different sites, devices, software applications, and people. Ultimately, I think that in the long run, the future will belong to the word of information that is more atomized and more modular, as opposed to less. The reason I think that is because we can observe a certain historical correspondence between the structure of cultural "content" and the structure of the media that carries it. Tight packaging of the cultural products of mass media era corresponds to the non-discrete materiality of the dominant recording media – photographic paper, film, and magnetic tape used for audio and later video recording. In contrast, the growing modularity of cultural content in the software age perfectly corresponds the systematic modularity of modern software which manifest itself on all levels: "structured programming" paradigm, "objects" and "methods" in object-oriented programming paradigm, modularity of Internet and web protocols and formats, etc. – all the way to the bits, bytes, pixels and other atoms which make up digital representations in general.

If we approach the present from the perspective of a potential future of "ultimate modularity / remixability," we can see other incremental steps towards this future which are already occurring.

Creative Commons developed a set of flexible licenses that give the producers of creative work in any field more options than the standard copyright terms. The licenses have been widely used by individuals, non-profits and companies – from MIT Open Course Initiative and Australian Government to Flickr and blip.tv. The available types include a set of Sampling Licenses which "let artists and authors invite other people to use a part of their work and make it new."[182]

In 2005 a team of artists and developers from around the world set out to collaborate on an animated short film *Elephants Dream* using only open source software[183]; after the film was completed, all production files from the move (3D models, textures, animations, etc.) were published on a DVD along with the film itself.[184]

Flickr offers multiple tools to combine multiple photos (not broken into parts – at least so far) together: tags, sets, groups, Organizr. Flickr interface thus position each photo within multiple "mixes." Flickr also offers "notes" which allows the users to assign short notes to individual parts of a photograph. To add a note to a photo posted on Flickr, you draw a rectangle on any part of the phone and then attach some text to it. A number of notes can be attached to the same photo. I read this feature as another a sign of modularity/remixability paradigm, as it encourages users to mentally break a photo into separate parts. In other words, "notes" break a single media object – a photograph – into blocks.

In a similar fashion, the common interface of DVDs breaks a film into chapters. Media players such as iPod and online media stores such as iTunes break music CDs into separate tracks – making a track into a new basic unit of musical culture. In all these examples, what was previously a single coherent cultural object is broken into separate blocks that can be accessed individually. In other words, if “information wants to be ASCII,” “content wants to be modular.” And culture as a whole? Culture has always been about remixability – but now this remixability is available to all participants of web culture.

Since the introduction of first Kodak camera, “users” had tools to create massive amounts of vernacular media. Later they were given amateur film cameras, tape recorders, video recorders...But the fact that people had access to "tools of media production" for as long as the professional media creators until recently did not seem to play a big role: the amateur’ and professional’ media pools did not mix. Professional photographs traveled between photographer’s darkroom and newspaper editor; private pictures of a wedding traveled between members of the family. But the emergence of multiple and interlinked paths which encourage media objects to easily travel between web sites, recording and display devices, hard drives and flash drives, and, most importantly, people changes things. Remixability becomes practically a built-in feature of digital networked media universe. In a nutshell, what maybe more important than the introduction of a video iPod (2001), YouTube (2005), first consumer 3-CCD camera which can record full HD video (HD Everio GZ-HD7, 2007), or yet another exiting new device or service is how easy it is for media objects to travel between all these devices and services - which now all become just temporary stations in media’s Brownian motion.

Modularity and “Culture Industry”

Although we see a number of important new types of cultural modularity emerged in software era, it is important to remember that modularity is something that only applies to RSS, social bookmarking, or Web Services. We are talking about the larger cultural logic that extends beyond the Web and digital culture.

Modularity has been the key principle of modern mass production. That is, mass production is possible because of the standardization of parts and how they fit with each other - i.e. modularity. Although there are historical precedents for mass production, until twentieth century they have been separate historical cases. But after Ford installs first moving assembly lines at his factory in 1913, others follow. ("An assembly line is a manufacturing process in which interchangeable parts are added to a product in a sequential manner to create an end product."[185]) Soon modularity permeates most areas of modern society. The great majority of products we use today are mass produced, which means they are modular, i.e. they consist from standardized mass produced parts which fit together in standardized way. But modularity was also taken up outside of factory. For instance, already in 1932 – long before IKEA and Logo sets – Belgian designer Louis Herman De Kornick developed first modular furniture suitable for smaller council flats being built at the time.

Today we are still leaving in an era of mass production and mass modularity, and globalization and outsourcing only strengthen this logic. One commonly evoked characteristic of globalization is greater connectivity – places, systems, countries, organizations, etc. becoming connected in more and more ways. Although there are ways to connect things and processes without standardizing and modularizing them – and the further development of such mechanisms is probably essential if we ever want to move beyond all the grim consequences of living in a standardized modular world produced by the twentieth century – for now it appears so much easier just to go ahead and apply the twentieth century logic. Because society is so used to it, it is not even thought of as one option among others.

In November 205 I was at a Design Brussels event where a well-known designer Jerszy Seymour speculated that once Rapid Manufacturing systems become advanced, cheap and easy, this will give designers in Europe a hope for survival. Today, as Seymour pointed out, as soon as some design becomes successful, a company wants to produce it in large quantities – and its production goes to China. He suggested that when Rapid Manufacturing and similar technologies would be installed locally, the designers can become their own manufactures and everything can happen in one place. But obviously this will not happen tomorrow, and it is also not at all certain that Rapid Manufacturing will ever be able to produce complete finished objects without any humans involved in the process, whether its assembly, finishing, or quality control.

Of course, modularity principle did not stayed unchanged since the beginning of mass production a hundred years ago. Think of just-in-time manufacturing, just-in-time programming or the use of standardized containers for shipment around the world since the 1960s (over %90 of all goods in the world today are shipped in these containers). The logic of modularity seems to be permeating more layers of society than ever before, and software – which is great to keeping track of numerous parts and coordinating their movements – only help this process.

The logic of culture often runs behind the changes in economy (recall the concept of “uneven development” I already evoked in Part 2) – so while modularity has been the basis of modern industrial society since the early twentieth century, we only start seeing the modularity principle in cultural production and distribution on a large scale in the last few decades. While Adorno and Horkheimer were writing about "culture industry" already in early 1940s, it was not then - and it is not today - a true modern industry.[186] In some areas such as large-scale production of Hollywood animated features or computer games we see more of the factory logic at work with extensive division of labor. In the case of software engineering, software is put together to a large extent from already available software modules - but this is done by individual programmers or teams who often spend months or years on one project – quite different from Ford production line model used assembling one identical car after another in rapid succession. In short, today cultural modularity has not reached the systematic character of the industrial standardization circa 1913.

But this does not mean that modularity in contemporary culture simply lags behind industrial modularity. Rather, cultural modularity seems to be governed by a different logic. In terms of packaging and distribution, "mass culture" has indeed achieved complete industrial-type standardization. In other words, all the material carriers of cultural content in the 20th century have been standardized, just as it was done in the production of all other goods - from first photo and films formats in the end of the nineteenth century to game cartridges, DVDs, memory cards, interchangeable camera lenses, and so on today. But the actual making of content was never standardized in the same way. In "Culture industry reconsidered," Adorno writes:

> The expression "industry" is not to be taken too literally. It refers to the standardization of the thing itself — such as that of the Western, familiar to every movie-goer — and to the rationalization of distribution techniques, but not strictly to the production process… it [culture industry] is industrial more in a sociological sense, in the incorporation of industrial forms of organization even when nothing is manufactured — as in the rationalization of office work — rather than in the sense of anything really and actually produced by technological rationality.[187]

So while culture industries, at their worst, continuously put out seemingly new cultural products (fims, television programs, songs, games, etc.) which are created from a limited repertoire of themes, narratives, icons and other elements using a limited number of conventions, these products are conceived by the teams of human authors on a one-by-one basis – not by software. In other words, while software has been eagerly adopted to help automate and make more efficient lower levels of the cultural production (such as generating in-between frames in an animation or keeping track of all files in a production pipeline), humans continue to control the higher levels. Which means that the semiotic modularity of cultural industries' products – i.e., their Lego-like construction from mostly pre-existent elements already familiar to consumers – is not something which is acknowledged or thought about.

The trend toward the reuse of cultural assets in commercial culture, i.e. media franchising – characters, settings, icons which appear not in one but a whole range of cultural products – film sequels, computer games, theme parks, toys, etc. – this does not seem to change this basic "pre-industrial" logic of the production process. For Adorno, this individual character of each product is part of the ideology of mass culture: "Each product affects an individual air; individuality itself serves to reinforce ideology, in so far as the illusion is conjured up that the completely reified and mediated is a sanctuary from immediacy and life."[188]

Neither fundamental re-organization of culture industries around software-based production in the 1990s nor the rise of user-generated content and social media paradigms in 2000s threatened the Romantic ideology of an artist-genius. However, what seems to be happening is that the "users" themselves have been gradually "modularizing" culture. In other words, modularity has been coming into mass culture from the outside, so to speak, rather than being built-in, as in industrial production. In the 1980s musicians start sampling already published music; TV fans start sampling their favorite TV series to produce their own "slash films," game fans start creating new game levels and all other kinds of game modifications, or "mods". (Mods "can include new items, weapons, characters, enemies, models, modes, textures, levels, and story lines."[189]) And of course, from the very beginning of mass culture in early twentieth century, artists have immediately starting sampling and remixing mass cultural products – think of Kurt Schwitters, collage and particularly photomontage practice which becomes popular right after WWI among artists in Russia and Germany. This continued with Pop Art, appropriation art, video art, net art...

Enter the computer. In *The Language of New Media* I named modularity as one of the trends I saw in a culture undergoing computerization. If before modularity principle was applied to the packaging of cultural goods and raw media (photo stock, blank videotapes, etc.), computerization modularizes culture on a structural level. Images are broken into pixels; graphic designs, film and video are broken into layers in Photoshop, After Effects, and other media design software. Hypertext modularizes text. Markup languages such as HTML and media formats such as QuickTime modularize multimedia documents in general. This all already happened by 1999 when I was finishing *The Language of New Media*; as we saw in this chapter, soon thereafter the adoption of web feed formats such as RSS further modularized media content available on the web, breaking many types of packaged information into atoms…

In short: in culture, we have been modular already for a long time already. But at the same time, "we have never been modular"[190] - which I think is a very good thing.

Chapter 6. Social Media: Tactics as Strategies

From Mass Consumption to Mass (Cultural) Production

The evolution of cultural software during 2000s is closely linked to from the rise a web as the platform for media publishing, sharing, and social communication. The key event in this evolution has been the shift from the original web to the so-called Web 2.0 (the term was introduced by Tim O'Reilly in 2004.) This term refers to a number of different technical, economical, and social developments which were given their own terms: *social media*, *user-generated content*, *long tail*, *network as platform*, *folksonomy*, *syndication*, *mass collaboration*, etc. We have already discussed a number of these developments directly or indirectly in relation to the topics of remixability and modularity. What I want to do now is to approach them from a new perspective. I want to ask how the phenomena of social media and user-generated content reconfigure the relationships between cultural "amateurs" and official institutions and media industries, on the one hand, and "amateurs" and professional art world, on the other hand.

To get the discussion started, let's simply summarize these two Web 2.0 themes. Firstly, in 2000s, we see a gradual shift from the majority of web users accessing content produced by a much smaller number of professional producers to users increasingly accessing content produced by other non-professional users.[191] Secondly, if 1990s Web was mostly a publishing medium, in 2000s it increasingly became a communication medium. (Communication between users, including conversations around user-generated content, take place through a variety of forms besides email:

posts, comments, reviews, ratings, gestures and tokens, votes, links, badges, photo, and video.[192])

What do these trends mean for culture in general and for professional art in particular? First of all, they do not mean that every user has become a producer. According to 2007 statistics, only between 0.5% – 1.5% users of most popular (in the U.S.) social media sites - Flickr, YouTube, and Wikipedia - contributed their own content. Others remained consumers of the content produced by this 0.5 - 1.5%. Does this mean that professionally produced content continues to dominate in terms of where people get their news and media? If by "content" we mean typical twentieth century mass media - news, TV shows, narrative films and videos, computer games, literature, and music – then the answer is often yes. For instance, in 2007 only 2 blogs made it into the list of 100 most read news sources. At the same time, we see emergence of the "long-tail" phenomenon on the net: not only "top 40" but most of the content available online - including content produced by individuals - finds some audiences.[193] These audiences can be tiny but they are not 0. This is best illustrated by the following statistics: in the middle of 2000s every track out of a million of so available through iTunes sold at least once a quarter. In other words, every track no matter how obscure found at least one listener. This translates into new economics of media: as researchers who have studied the long tail phenomena demonstrated, in many industries the total volume of sales generated by such low popularity items exceeds the volume generated by "top forty" items.[194]

Let us now consider another set of statistics showing that people increasingly get their information and media from social media sites. In January 2008, Wikipedia has ranked as number 9 most visited web site; Myspace was at number 6, Facebook was at 5, and MySpace was at 3. (According to the company that collects these statistics, it is more than likely that these numbers are U.S. biased, and that the rankings in other countries are different.[195] However, the general trend towards increasing use of social media sites – global, localized, or local - can be observed in most countries. In fact, according to 2008 report, the growth in social media has been accelerating outside of U.S., with a number of countries in Asia significantly outpacing Western Countries in areas – reading and writing blogs, watching and making video and photos, etc. For instance, while only %26.4 of Internet users in the U.S. started a blog at some point, this number was %60.3 for Mexico, %70.3 for China, and %71.7 for South Korea. Similarly, while in the U.S. the percentage of Internet users who also use social networks was %43, it was %66 for India, %71.1 for Russia, %75.7 for Brazil, and %83.1 for Philippines.[196])

The numbers of people participating in these social networks, sharing media, and creating "user generated content" are astonishing – at least from the perspective of early 2008. (It is likely that in 2012 or 2018 they will look trivial in comparison to what will be happening then). MySpace: 300,000,000 users.[197] Cyworld, a Korean site similar to MySpace: 90 percent of South Koreans in their 20s, or 25 percent of the total population of South Korea.[198] Hi4, a leading social media site Central America: 100,000,000 users.[199] Facebook: 14,00,000 photo uploads daily.[200] The number of new videos uploaded to YouTube every 24 hours (as of July 2006): 65,000.[201] The number of videos watched by 79 million visitors to YouTube during January 2008: more than 3 billion.[202]

If these numbers are already amazing, consider another platform for accessing, sharing, and publishing media: a mobile phone. In Early 2007, 2.2 billion people have mobile phones; by the end of the year this number was expected to be 3 billion. Obviously, today people in an Indian village who all sharing one mobile phone do not make video blogs for global consumption – but this is today. Think of the following trend: in the middle of 2007, Flickr contained approximately 600 million images. By early 2008, this number has already doubled.

These statistics are impressive. The more difficult question is: how to interpret them? First of all, they don't tell us about the actual media diet of users (obviously these diets vary between places and demographics). For instance, we don't have exact numbers (at least, they are not freely available) regarding what exactly people watch on sites such as YouTube – the percentage of user-generated content versus commercial content such as music videos, anime, game trailers, movie clips, etc.[203] Secondly, we also don't have exact numbers regarding which percentage of peoples' daily media/information intake comes from big news organization, TV, commercially realized films and music versus non-professional sources.

These numbers are difficult to establish because today commercial media does not only arrive via traditional channels such as newspapers, TV stations and movie theatres but also via the same channels which carry user-generated content: blogs, RSS feeds, Facebook's posted items and notes, YouTube videos, etc. Therefore, simply counting how many people follow a particular communication channel is no longer tells you what they are watching.

But even if we knew precise statistics, it still would not be clear what are the relative roles between commercial sources and user-produced content in forming people understanding of the world, themselves, and others. Or, more precisely: what are the relative weights between the ideas expressed in large circulation media and alternative ideas available elsewhere? And, if one person gets all her news via blogs, does this automatically mean that her understanding of the world and important issues is different from a person who only reads mainstream newspapers?

The Practice of Everyday Media Life: Tactics as Strategies

For different reasons, media, businesses, consumer electronics and web industries, and academics converge in celebrating content created and exchanged by users. In U.S. academic discussions, in particular, the disproportional attention was given to certain genres such as "youth media," "activist media," "political mash-ups" – which are indeed important but do not represent more typical usage of hundreds of millions of people.

In celebrating user-generated content and implicitly equating "user-generated" with "alternative" and "progressive," academic discussions often stay away from asking certain basic critical questions. For instance: To what extent the phenomenon of user-generated content is driven by consumer electronics industry – the producers of digital cameras, video cameras, music players, laptops, and so on? Or: To what extent the phenomenon of user-generated content is generated by social media companies themselves – who, after all, are in the business of getting as much traffic to their sites as possible so they can make money by selling advertising and their usage data?

Here is another question. Given that the significant percentage of user-generated content either follows the templates and conventions set up by professional entertainment industry, or directly re-uses professionally produced content (for instance, anime music videos), does this means that people's identities and imagination are now even more firmly colonized by commercial media than in the twentieth century? In other words: Is the replacement of *mass consumption of commercial culture* in the 20th century by *mass production of cultural objects by users* in the early 21st century is a progressive development? Or does it constitutes a further stage in the development of "culture industry" as analyzed by Theodor Adorno and Max Horkheimer in their 1944 book *The Culture Industry: Enlightenment as Mass Deception*? Indeed, if the twentieth century subjects were simply consuming the products of culture industry, 21st century prosumers and "pro-ams" are passionately imitating it. That is, they now make their own cultural products that follow the templates established by the professionals and/or rely on professional content.

The case in point is anime music videos (often abbreviated as AMV). My search for "anime music videos" on YouTube on April 7, 2008 returned 275,000 videos.[204] Animemusicvideos.org, the main web portal for anime music video makers (before the action moved to YouTube) contained 130,510 AMVs as of February 9, 2008. AMV are made mostly by fans of anime in the West. They edit together clips from one or more anime series to music, which comes from different sources such as professional music videos. Sometimes, AMV also use cut-scene footage from video games. From approximately 2002-2003, AMV makers also started to increasingly add visual effects available in software such as After Effects. But regardless of the particular sources used and their combination, in the majority of AMV all video and music comes from commercial media products. AMVs makers see themselves as editors who re-edit the original material, rather than as filmmakers or animators who create from scratch.[205]

To help us analyze AMV culture, let us put to work the categories set up by Michel de Certeau in his 1980 book *The Practice of Everyday Life*.[206] De Certeau makes a distinction between "strategies" used by institutions and power structures and "tactics" used by modern subjects in their everyday life. The tactics are the ways in which individuals negotiate strategies that were set for them. For instance, to take one example discussed by de Certeau, city's layout, signage, driving and parking rules and official maps are strategies created by the government and companies. The ways an individual is moving through the city, taking shortcuts, wondering aimlessly, navigating through favorite routes and adopting others constitute tactics. In other words, an individual can't physically reorganize the city but she can adopt itself to her needs by choosing how she moves through it. A tactic "expects to have to work on things in order to make them its own, or to make them 'habitable'."[207]

As De Certeau points out, in modern societies most of the objects which people use in their everyday life are mass produced goods; these goods are the expressions of strategies of producers, designers, and marketers. People build their worlds and identities out of these readily available objects by using different tactics: bricolage, assembly, customization, and – to use the term which was not a part of De Certeau's vocabulary but which has become important today – remix. For instance, people rarely wear every piece from one designer's collection as they appear in fashion shows: they usually mix and match different pieces from different sources. They also wear clothing pieces in different ways than it was intended, and they customize the cloves themselves through buttons, belts, and other accessories. The same goes for the ways in which people decorate their living spaces, prepare meals, and in general construct their lifestyles.

While the general ideas of *The Practice of Everyday Life* still provide an excellent intellectual paradigm for thinking about the vernacular culture, since the book was published in 1980s many things also changed. These changes are less drastic in the area of governance, although even there we see moves towards more transparency and visibility. (For instance, most government agencies operate detailed web sites.) But in the area of consumer economy, the changes have been quite substantial. Strategies and tactics are now often closely linked in an interactive relationship, and often their features are reversed. This is particularly true for "born digital" industries and media such as software, computer games, web sites, and social networks. Their products are *explicitly designed to be customized by the users.* Think, for instance, of the original Graphical User Interface (popularized by Apple's Macintosh in 1984) designed to allow a user to customize the appearance and functions of the computer and the applications to her liking. The same applies to recent web interfaces – for instance, iGoogle which allows the user to set up a custom home page selecting from many applications and information sources. Facebook, Flickr, Google and other social media companies encourage others to write applications, which mash-up their data and add new services (as of early 2008, Facebook hosted over 15,000 applications written by outside developers.) The explicit design for customization is not limited to the web: for instance, many computer games ship with level editor that allows the users to create their own levels. And *Spore* (2008) designed by celebrated Will Write went much further: most of the content of the game is created by users themselves: "The content that the player can create is uploaded automatically to a central database (or a peer-to-peer system), cataloged and rated for quality (based on how many users have downloaded the object or creature in question), and then re-distributed to populate other players' games."[208]

Although the industries dealing with the physical world are moving much slower, they are on the same trajectory. In 2003 Tayota introduced Scion cars. Scion marketing was centered on the idea of extensive customization. Nike, Adidas, and Puma all experimented with allowing the consumers to design and order their own shoes by choosing from a broad range of shoe parts. (In the case of Puma Mongolian Barbeque concept, a few thousand unique shoes can be constructed.)[209] In early 2008 Bug Labs introduced what they called "the Lego of gadgets": open sourced consumer electronics platform consisting from a minicomputer and modules such as a digital camera or a LCD screen.[210] The celebration of DIY practice in various consumer industries from 2005 onward is another example of this growing trend. Other examples include the idea of co-creation of products and services between companies and consumers (*The Future Of Competition: Co-Creating Unique Value with Customers* by C.K. Prahalad and Venkat Ramaswamy[211]), as well as the concept of crowdsourcing in general.

In short: since the original publication of *The Practice of Everyday Life*, companies have developed new kinds of strategies. These strategies mimic people's tactics of bricolage, re-assembly and remix. In other words: the logic of tactics has now become the logic of strategies.

According to De Certeau original analysis from 1980, tactics do not necessary result in objects or anything stable or permanent: "Unlike the strategy, it <tactic> lacks the centralized structure and permanence that would enable it to set itself up as a competitor to some other entity… it renders its own activities an "unmappable" form of subversion."[212] Since the early 1980s, however, consumer and culture industries have started to systematically turn every subculture (particularly every youth subculture) into products. In short, the cultural tactics evolved by people were turned into strategies now sold to them. If you want to "oppose the mainstream," you now had plenty of lifestyles available – with every subculture aspect, from music and visual styles to clothes and slang – available for purchase.

This adaptations, however, still focused on distinct subcultures: bohemians, hip hop and rap, Lolita fashion, rock, punk, skin head, Goth, etc.[213] However, in 2000s, the transformation of people's tactics into business strategies went into a new direction. The developments of the previous decade – the Web platform, the dramatically decreased costs of the consumer electronics devices for media capture and playback, increased global travel, and the growing consumer economies of many countries which after 1990 joined the global economy – led to the explosion of user-generated content available in digital form: Web sites, blogs, forum discussions, short messages, digital photo, video, music, maps, etc. Responding to this explosion, Web 2.0 companies created powerful platforms designed to host this content. MySpace, Facebook, Orkut, Livejournal, Blogger, Flickr, YouTube, h5 (Central

America), Cyworld (Korea), Wretch (Taivan), Orkut (Brasil), Baidu (China), and thousands of other social media sites make this content instantly available worldwide (except, of course, in a small number of countries which block or filter these sites). Thus, not just particular features of particular subcultures but the details of everyday life of hundreds of millions of people who make and upload their media or write blogs became public.

What before was ephemeral, transient, umappable, and invisible become permanent, mappable, and viewable. Social media platforms give users unlimited space for storage and plenty of tools to organize, promote, and broadcast their thoughts, opinions, behavior, and media to others. As I am writing this, you can already directly stream video from your laptop or mobile phone's camera, and it is only a matter of time before constant broadcasting of one's live becomes as common as email. If you follow the evolution from MyLifeBits project (2001-) to Slife software (2007-) and Yahoo! Live personal broadcasting service (2008-), the trajectory towards continuous capture and broadcasting of one's everyday life is clear.

According to De Certeau's 1980 analysis, strategy "is engaged in the work of systematizing, of imposing order… its ways are set. It cannot be expected to be capable of breaking up and regrouping easily, something which a tactical model does naturally." The strategies used by social media companies today, however, are the exact opposite: they are focused on flexibility and constant chance. Of course, all businesses in the age of globalization had to become adaptable, mobile, flexible, and ready to break up and regroup – but the companies involved in producing and handling physical objects rarely achieve the flexibility of web companies and software developers.[214] According to the Tim O'Reilly (in case you don't remember – he originated the term Web 2.0 in 2004), an important feature of Web 2.0 applications is "design for 'hackability' and remixability."[215] Indeed, Web 2.0 era has truly got under way when major web companies - Amazon, eBay, Flickr, Google, Microsoft, Yahoo and YouTube - make available some of their services (APIs) and data to encourage others to create new applications using this data.[216]

In summary, today strategies used by social media companies often look more like tactics in the original formulation by De Certeau – while tactics look strategies. Since the companies which create social media sites make money from having as many users as possible visiting their sites as often as possible – because they sell ads, sell data about site usage to other companies, selling ad-on services, etc. – they have a direct interest in having users pouring their lives into these platforms. Consequently, they give users unlimited storage space to store all their media, the ability to customize their "online lives" (for instance, by controlling what is seen by who) and expand the functionality of the platforms themselves.

All this, however, does not mean strategies and tactics have completely exchanged places. If we look at the actual media content produced by users, here strategies/tactics relationship is different. As I already mentioned, for a few decades now companies have been systematically turning the elements of various subcultures developed by people into commercial products. But these subcultures themselves, however, rarely develop completely from scratch – rather, they are the result of cultural appropriation and/or remix of earlier commercial culture by consumers and fans.[217] AMV subculture is a case in point. On the other hand, it exemplifies new "strategies as tactics" phenomenon: AMVs are hosted on mainstream social media sites such as YouTube, so they can't be described as "transient" or "unmappable" - you can use search to find them, see how others users rated them, save them as favorites, etc. On the other hand, on the level of content, it is "practice of everyday life" as before: the great majority of AMVs consist from segments sampled from commercial anime programs and commercial music. This does not mean that best AMVs are not creative or original – only that their creativity is different from the romantic/modernist model of "making it new." To borrow De Certeau's terms, we can describe it as *tactical creativity* that "expects to have to work on things in order to make them its own, or to make them 'habitable.'"

Media Conversations

"Creativity" is not the only term impacted by the phenomena of social media. Other very basic terms – content, a cultural object, cultural production, cultural consumption, communication – are similarly being expanded or redefined. In this section we will look at some of the most interesting developments in social media which are responsible for these redefinitions.

One of the characteristics of social media is that it is often hard to say where "content" ends and the discussions of this content begin. Blog writing offers plenty of examples. Frequently, blog posts are comments by a blog writer about an item that s/he copied from another source. Or, consider comments by others that may appear below a blog post. The original post may generate a long discussion which goes into new and original directions, with the original post itself long forgotten. (Discussions on Forums often follow the same patterns.)

Often "content," "news" or "media" become *tokens* used to initiate or maintain a conversation. Their original meaning is less important than their function as such tokens. I am thinking here of people posting pictures on each other pages on MySpace, or exchanging gifts on Facebook. What kind of gift you get is less important than the act of getting a gift, or posting a comment or a picture. Although it may appear at first that such conversation simply foreground Roman Jakobson's emotive and/or phatic communication functions which he described already in 1960[218], it is also possible that a detailed analysis will show them to being a genuinely new phenomenon.

The beginnings of such analysis can be found in the writing of social media designer Adrian Chan. As he points out, "All cultures practice the exchange of tokens that bear and carry meanings, communicate interest and count as personal and social transactions." Token gestures "cue, signal, indicate users' interests in one another." While the use of tokens in not unique to networked social media, some of the features pointed by Chan do appear to be new. For instance, as Chan notes, the use of tokens in net communication is often "accompanied by ambiguity of intent and motive (the token's meaning may be codified while the user's motive for using it may not). This can double up the meaning of interaction and communication, allowing the recipients of tokens to respond to the token or to the user behind its use."[219]

Consider another very interesting new communication situation: *a conversation around a piece of media* – for instance, comments added by users below somebody's Flickr photo or YouTube video which do not only respond to the media object but also to each other. According to a survey conducted in 2007, %13 of Internet users who watch video also post comments about the videos.[220] (The same is often true of comments, reviews and discussions on the web in general – the object in question can be software, a film, a previous post, etc.) Of course, such conversation structures are also common in real life. However, web infrastructure and software allow such conversations to become distributed in space and time – people can respond to each other regardless of their location and the conversation can in theory go forever. (The web is millions of such conversations taking place at the same time – as dramatized by the installation *Listening Post* created by Ben Rubin and Mark Hansen[221]). These conversations are quite common: according to the report by Pew internet & American Life Project (12/19/2007), among U.S.

teens who post photos online, %89 reported that people comment on these photos at least some of the time.[222]

Equally interesting *is conversations which takes place through images or video* – for instance, responding to a video with a new video. This phenomenon of "conversation through media" was first pointed to me by UCSD graduate student Derek Lomas in 2006 in relation to comments on MySpace pages that often consists of only images without any accompanying text. Soon thereafter, YouTube UI "legitimized" this new type of communication by including "post a video response" button and along with other tools that appear below a rectangle where videos are played. It also provides a special places for videos created as responses. (Note again that all examples of interfaces, features, and common uses of social media sites here refer to middle of 2008; obviously some of the details may change by the time you read this.) Social media sites contain numerous examples of such "conversations through media" and most of them are not necessary very interesting – but enough are. One of them is a conversation around a five-minute "video essay" *Web 2.0 ... The Machine is Us/ing Us* posted by a cultural anthropologist Michael Wesch on January 31, 2007.[223] A year later this video was watched 4,638,265 times.[224] It has also generated 28 video responses that range from short 30-second comments to equally theoretical and carefully crafted longer videos.

Just as it is the case with any other feature of contemporary digital culture, it is always possible to find some precedents for any of these communication situations. For instance, modern art can be understood as conversations between different artists or artistic schools. That is, one artist/movement is responding to the works produced earlier by another artist/movement. For instance, modernists react against classical nineteenth century salon art culture; Jasper John and other pop-artists react to abstract expressionism; Godard reacts to Hollywood-style narrative cinema; and so on. To use the terms of YouTube, we can say that Godard posts his video response to one huge clip called "classical narrative cinema." But the Hollywood studios do not respond – at least not for another 30 years.

As can be seen from these examples, typically these conversations between artists and artistic schools were not full conversations. One artist/school produced something, another artist/school later responded with their own productions, and this was all. The first art/school usually did not respond. But beginning in the 1980s, professional media cultures begin to respond to each other more quickly and the conversations are no longer go one way. Music videos affect the editing strategies of feature films and television; similarly, today the aesthetics of motion graphics is slipping into narrative features. Cinematography, which before only existed in films, is taken up in video games, and so on. But these conversations are still different from the *communication between individuals through media* in a networked environment. In the case of Web 2.0, it is individuals directly talking to each other using media rather than only professional producers.

New Media Technologies and the Arts: a History of Diminishing Options

It has become a cliché to discuss new communication and media technologies in terms of "new possibilities they offer for artists." Since I started writing about new media art in the early 1990s, I have seen this stated countless times in relation to each new technology which came along – virtual reality and virtual worlds, Internet, Web, networks in general ("network art"), computer games, locative media, mobile media, and social media.

But what if instead of automatically accepting this idea of "expanding possibilities," we imagine its opposite? What if new media technologies impact professional arts in a very different way? Let us explore the thesis that, instead of offering arts new options, each new modern media technology has put further limits on the kinds of activities and strategies for making media that artists can claim as unique.

As an example, consider a well-known and extensively discussed episode in the history of arts and technology: the effect of photography on painting in the 19th century. According to a common interpretation, the new medium of photography liberated painting from its documentary function. By taking over the job of recording visible reality, photography set painters free to discover new functions for their artworks. As a result, painters gradually moved away from representation towards abstraction. A two-dimensional canvas came to be understood as an object in itself rather than as a window into an illusionary space. From there, modern artists took the next step of moving from a flat painting to a three-dimensional object (constructivism, pop art). Artists also came up with a variety of new techniques for making both representational and non-representational images that opposed the automatic generation of an image in photography and film - for example, expressionism of 1910s and 1920s and post-war abstract expressionism. They also stared to use mass produced objects and their own bodies as both subjects and materials of art (pop art, performance, and other new forms which emerged in the 1960s).

But it is also possible to reinterpret these developments in visual arts in a different way. By taking over the documentary function of painting, photography has taken away painters’ core business - portraits, family scenes, landscapes, and historical events. As a result, paintings suddenly lost the key roles they played both in religious and in secularized societies until that time – encoding social and personal memories, constructing visual symbols, communicating foundational narratives and world views – all in all, carrying over society’s semiotic DNAs. So what could painters do after this? In fact, they never recovered. They turned towards examining the visual language of painting (abstraction), the material elements of their craft and the conventions of painting's existence ("white on white" paintings, stretched canvases exhibited with their back facing the viewer, and so on), and the conditions of art institutions in general (from Duchamp to Conceptual Art to Institutional Critique.) And if at first these explorations were generating fresh and socially useful results - for instance, geometric abstraction was adopted as the new language of visual communication, including graphic design, packaging, interior design, and publicity - eventually they degenerated, turning into painful and self-absorbed exercises. In other words, by the 1980s professional art more often than not was chasing its own tale.

Thus, rather than thinking of modern art as a liberation (from representation and documentation), we can see it as a kind of psychosis – an intense, often torturous examination of the contents of its psyche, the memories of its glamorous past lives, and the very possibilities of speaking. At first this psychosis produced brilliant insights and inspired visions but eventually, as the mental illness progressed, it degenerated into endless repetitions.

This is only to be expected, given that art has given up its previously firm connection to outside reality. Or, rather, it was photography that forced art into this position. Having severed its connection to visible reality, art became like a mental patient whose mental processing is no longer held in check by sensory inputs. What eventually saved art from this psychosis was globalization of the 1990s. Suddenly, the artists in newly "globalized" countries – China, India, Pakistan, Vietnam, Malaysia, Kazakhstan, Turkey, Poland, Macedonia, Albania, etc. – had access to global cultural markets – or rather, the global market had now access to them. Because of the newness of modern art and the still conservative social norms in many of these countries, the social functions of art that by that time lost their effectiveness in the West – representation of sexual taboos, critique of social and political powers, the ironic depiction of new middle classes and new rich – still had relevance and urgency in these contexts. Deprived from quality representational art, Western collectors and publics rushed to admire the critical realism produced outside of the West – and thus realism returned to become if not the center, than at least one of the key focuses of contemporary global art.

Looking at the history of art between middle of the nineteenth century and the end of Cold War (1990), it is apparent that painting did quite well for itself. If you want the proof, simply take a look at the auction prices for 20th century paintings that in 2000s became higher than the prices for the classical art. But not everybody was able to recover as well as painters from the impact of new media technologies. Probably the main reason for their success was the relatively slow development of photographic technology in the nineteenth and first third of the twentieth century. From the moment painters perceived the threat – let us say in 1839 when Daguerre developed his daguerreotype process – it took about a hundred years before color photography got to the point there it could compete with painting in terms of visual fidelity. (The relevant date here is 1935 when Kodak introduced first mass-marketed still color film Kodachrome). So painters had a luxury of time to work out new subjects and new strategies. In the last third of the twentieth century, however, the new technologies have been arriving at an increasing pace, taking over more and more previously unique artistic strategies within a matter of a few years.

For instance, in the middle of the 1980s more sophisticated video keyers and early electronic and digital effects boxes designed to work with professional broadcast video – Quantel Paintbox, Framestore, Harry, Mirage and others – made possibly to begin combining at least a few layers of video and graphics together, resulting in a kind of use video collage.[225] As a result, the distinctive visual strategies which previously clearly marked experimental film – superimposition of layers of imagery, juxtaposition of unrelated objects of filmed reality and abstract elements – quickly became the standard strategies of broadcast video post-production. In the 1990s the wide adoption of a Video Toaster, an Apple Macintosh and a PC, which could do such effects at a fraction of a cost, democratized the use of such visual strategies. By the middle of the 1990s most techniques of modernist avant-garde become available as standard features of software such as Adobe Premiere (1991), After Effects (1993), Flash (1996), and Final Cut (1999).

As a result, the definition of experimental film, animation and video radically shifted. If before their trademark was an unusual and often "difficult" visual form, they could no longer claim any formal uniqueness. Now experimental video and films could only brand themselves through content – deviant sexuality, political views which would be radical or dangerous in the local context, representations of all kinds of acts which a viewer would not see on TV (of course, this function was also soon to be taken over by YouTube), social documentary, or the use of performance strategies focused on the body of an artist. Accordingly, we see a shift from experimentation with forms to the emphasis on "radical" content," while the term "experimental" gradually replaced by the term "independent." The latter term accurately marks the change from a definition based on formal difference to a definition based on economics: an independent (or "art") project is different from a "commercial" project mainly because it is not commissioned and paid by a commercial client (i.e., a company). Of course, in reality things are not so neatly defined: many independent films and other cultural projects are either explicitly commissioned by some organization or made for a particular market. In the case of filmmaking, the difference is even smaller: any film can be considered independent as long as its producer is *not* an older large Hollywood studio.

In summary: from the early days of modern media technologies in the middle of the nineteenth century until now, modern artists were able to adopt to competition from these media creatively, inventing new roles for themselves and redefining what art was. (This, in fact, is similar to how today globalization and outsourcing pushes companies and professionals in different fields to redefine themselves: for instance, graphic designers in the West are turning into design consultants and managers). However, the emergence of *social media* - free web technologies and platforms which enable normal people to share their media and easily access media produced by others – combined with the rapidly fallen cost for professional-quality media devices such as HD video cameras – brings fundamentally new challenges.[226]

Is Art After Web 2.0 still possible?

How does art world responds to these challenges? Have professional artists benefited from the explosion of media content online being produced by regular users and the easily availability of media publishing platforms? Is the fact that we now have such platforms where anybody can publish their videos mean that artists have a new distribution channel for their works? Or is the world of social media – hundreds of millions of people daily uploading and downloading video, audio, and photographs; media objects produced by unknown authors getting millions of downloads; media objects easily and rapidly moving between users, devices, contexts, and networks – makes professional art simply irrelevant? In short, while modern artists have so far successfully met the challenges of each generation of media technologies, can professional art survive extreme democratization of media production and access?

On one level, this question is meaningless. Surely, never in the history of modern art it has been doing so well commercially. No longer a pursuit for a few, in 2000s *contemporary art became another form of mass culture*. Its popularity is often equal to that of other mass media. Most importantly, contemporary art has become a legitimate investment category, and with the all the money invested into it, today it appears unlikely that this market will ever completely collapse.

In a certain sense, since the beginnings of globalization in the early 1990s, the number of participants in the institution called "contemporary art" has experienced a growth that parallels the rise of social media in 2000s. Since 1990s, many new countries entered the global economy and adopted western values in their cultural politics. Which includes supporting, collecting, and promoting "contemporary art." When I first visited Shanghai in 2004, it already had has not just one but three museums of contemporary art plus more large-size spaces that show cotemporary art than New York or London. Starchitects rank Gehry, Jean Nouvel, Tadao Ando (above) and Zaha Hadid are now building museums and cultural centers on Saadiyat Island in Abu Dhabi.[227] Rem Koolhaus is building new museum of contemporary art in Riga, a capital of tiny Latvia (2007 population: 2.2 million). I can continue this list but you get the idea.

In the case of social media, the unprecedented growth of numbers of people who upload and view each other media led to lots of innovation. While the typical diary video or anime on YouTube may not be that special, enough are. In fact, in all media where the technologies of productions were democratized - music, animation, graphic design, (and also software development itself) - I have came across many projects available online which not only rival those produced by most well-known commercial companies and most well-known artists but also often explore the new areas not yet touched by those with lots of symbolic capital.

Who is creating these projects? In my observations, while some of them do come from prototypical "amateurs," "prosumers" and "pro-ams," most are done by young professionals, or professionals in training. The emergence of the Web as the new standard communication medium in the 1990s means that today in most cultural fields, every professional or a company, regardless of its size and geo location, has a web presence and posts new works online. Perhaps most importantly, young design students can now put their works before a global audience, see what others are doing, and develop together new tools and projects (for instance, see processing.org community).

Note that we are not talking about "classical" social media or "classical" user-generated content here, since, at least at present, many of such portfolios, sample projects and demo reels are being uploaded on companies' own web sites and specialized aggregation sites known to people in the field (such as archinect.com for architecture), rather than Flickr or YouTube. Here are some examples of such sites that I consult regularly: xplsv.tv (motion graphics, animation), coroflot.com (design portfolios from around the world), archinect.com (architecture students projects), infosthetics.com (information visualization projects). In my view, the significant percentage of works you find on these web sites represents the most innovative cultural production done today. Or at least, they make it clear that the world of professional art has no special license on creativity and innovation.

But perhaps the most amount of conceptual innovation is to be found today in software development for the web medium itself. I am thinking about all the new creative software tools - web mashups, Firefox plug-ins, Processing libraries, etc. – coming out from large software companies, small design firms, individual developers, and students.

Therefore, the true challenge posed to art by social media may be not all the excellent cultural works produced by students and non-professionals which are now easily available online – although I do think this is also important. The real challenge may lie in the dynamics of web culture – its constant innovation, its energy, and its unpredictability.

To summarize: Alan Kay was deeply right in thinking of a computer as generation engine which would enable invention of many new media. And yet, the speed, the breadth, and the sheer number of people now involved in constantly pushing forward *what media is* would be very hard to imagine thirty years ago when a computer metamedium was only coming into existence.