

PV251 Vizualizace

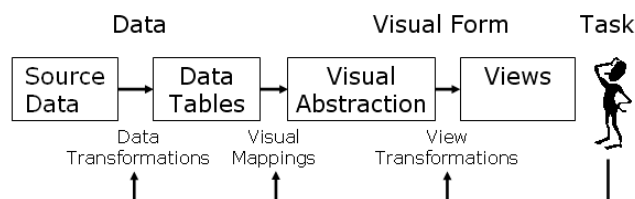
Jaro 2017

Výukový materiál

3. přednáška: Základní principy vizualizace

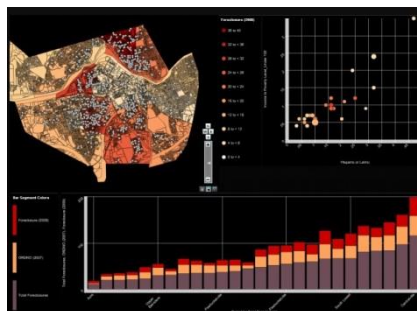
Visualization pipeline

Znovu se detailněji podívejme na celý vizualizační proces. Veškeré operace (např. transformace) prováděné v rámci procesu vizualizace je možné jednoduše mapovat na některou z fází této pipeline.



Tento obrázek reprezentuje nejjednodušší reprezentaci celého procesu. Je třeba si uvědomit dvě klíčové skutečnosti:

- 1) V ideálním případě by měl uživatel mít možnost interakce s kteroukoliv částí této pipeline. Tedy v jednotlivých stavech, ale rovněž v přechodech mezi nimi (v „uzlech“ i „hranách“).
- 2) Každá „hrana“ spojující jednotlivé stavy je typu „many-to-many“. To znamená, že například mnoho vizualizačních systémů poskytuje uživateli několik různých typů vizualizace najednou (viz obrázek), které jsou souběžně zobrazeny na obrazovce. To s sebou samozřejmě nese násobné mapování a renderování těchto různých reprezentací současně.



Práce s daty

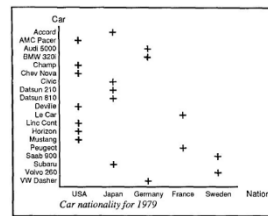
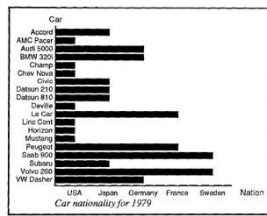
Nyní se podíváme na to, jakým způsobem jsou v této pipeline data zpracovávána – tedy rozšíříme si znalosti o transformacích a procesech, které mění vstupní data.

Předzpracování dat a transformace

Touto oblastí jsme se zabývali na minulé přednášce, kdy jsme se bavili o jednotlivých technikách používaných pro předzpracování dat. Jejich cílem je zpracovat surová vstupní data do podoby, která by byla využitelná zvoleným vizualizačním systémem. V první řadě se však musíme ujistit, že data jsou namapována na základní datové typy akceptované počítačem. Až ve druhém kroku můžeme s těmito daty pracovat a v této fázi přicházejí na řadu techniky zmíněné na minulé přednášce, které se dokáží vypořádat s chybějícími nebo chybnými záznamy nebo příliš velkými množinami vstupních dat, které není možné najednou zpracovat.

Mapování pro vizualizaci

Pokud máme data připravena pomocí předchozí fáze, je na místě uvažovat o způsobu jejich vizuální reprezentace. To v sobě zahrnuje mapování dané reprezentace – například geometrie, barvy nebo zvuku. Je velmi jednoduché navrhnout „nesmyslnou“ vizualizaci nebo takovou, která poskytuje špatné informace. Příklad vizualizace, která může vést k chybné interpretaci, je znázorněna na obrázcích.



Na levé straně je využit sloupcový graf, kde každý sloupec prochází přes všechny hodnoty na ose x, a končí na hodnotě příslušné země. Intuitivní vnímání této reprezentace nám říká, že každá hodnota na ose x, přes kterou sloupec prochází, odpovídá dané položce na ose y. Což je v tomto případě, kdy chceme zobrazit výrobce dané značky auta, zcela chybné. Například, Volvo by podle této reprezentace pocházelo z USA, Japonska, Německa, Francie i Švédska. Mnohem lepší reprezentace je zobrazena na pravém obrázku.

Poslední fází vizualizační pipeline je samotné mapování geometrických dat na výsledný obraz. Tato část je řešena pomocí interakce s některým z API počítačové grafiky. Průběh této fáze je velmi závislý na použité grafické knihovně. Příklady takovýchto knihoven jsou OpenGL, Processing, Flex, DirectX, ...

Metriky vizualizace

Nyní se pokusíme matematicky definovat dvě základní metriky vizualizace. Ty mohou být aplikovány ve všech fázích vizualizační pipeline. Jejich důležitost roste v momentě, kdy chceme měřit přenos informací sdělovaných pomocí dané vizualizace.

Tyto metriky jsou:

Expressiveness (expresivita, výraznost, významnost) a Effectiveness (efektivita).

Expressiveness

Expressiveness (expresivita) prezentuje danou informaci a právě jenom ji. Tedy do vizualizace nezanáší chybnou informaci. Pomocí expresivity tedy měříme koncentraci informace v dané vizualizaci.

Tato metrika se dá pro danou informaci, kterou chceme uživateli zobrazit, vyjádřit jako poměr M_{exp} této zobrazené informace k informaci, kterou chceme uživateli sdělit. Hodnota M_{exp} se pohybuje od 0 výše. Jestliže:

$M_{exp} = 1$, je expresivita ideální.

$M_{exp} < 1$, není naše vizualizace zcela efektivní, protože zobrazujeme méně informace, než bylo původně zamýšleno.

$M_{exp} > 1$, prezentujeme naopak více informací, než bychom měli. Toto je potenciálně nebezpečné, protože zobrazujeme nesprávné informace, které navíc mohou interferovat s interpretací základní informace, kterou jsme chtěli sdělit.

Effectiveness

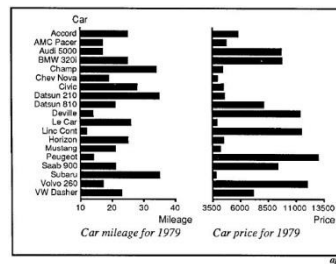
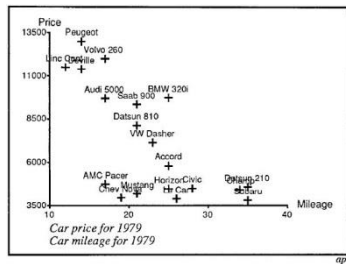
Vizualizace je efektivní, když je možné ji interpretovat přesně a rychle a jestliže je možné ji zobrazit (renderovat) „za rozumnou cenu“. Efektivita tedy měří „cenu“ daného vnímání informace.

Efektivitu lze definovat podobným způsobem jako expresivitu. V tomto případě poměr označujeme jako M_{eff} . Nicméně v tomto případě je situace trochu komplikovanější. Cílem je definovat metriku, která pro malé datové množiny měří čas potřebný k interpretaci – protože rendering malých dat je zpravidla velmi rychlý. Když se tento čas pro interpretaci zvyšuje, buď díky zvyšující se složitosti nebo velikosti dat, pak se hodnota M_{eff} snižuje, přičemž je kladen důraz i na čas potřebný pro renderování.

M_{eff} je definován následovně:

$$M_{eff} = 1 / (1 + \text{interpret} + \text{render})$$

Opět platí, že $0 \leq M_{\text{eff}} \leq 1$. Jestliže je hodnota M_{eff} malá, znamená to, že buď čas pro interpretaci nebo čas potřebný pro renderování je velký. Když se hodnota M_{eff} blíží jedničce, je čas potřebný pro interpretaci i renderování velmi krátký.



Jako příklad použití těchto dvou metrik si uveďme následující situaci. Oba obrázky zobrazují stejnou informaci, každá však jiným způsobem. Expresivita obou reprezentací může být považována za velmi podobnou až identickou – oba grafy zobrazují informaci o cenách aut a jejich najetých kilometrech a zobrazují veškeré informace a právě jen je. Navíc obě reprezentace je možné vykreslit velmi rychle (jedním z důvodů je malá datová sada).

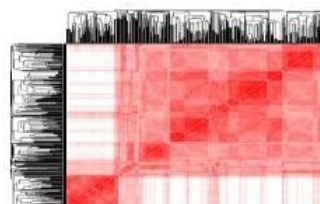
Avšak M_{eff} je různá. Informaci prezentovanou na grafu vpravo je možné pro některé dotazy interpretovat přesněji a rychleji než tu zobrazenou na levém grafu. Například pro dotaz „Které auto má nejméně najeto?“ je výsledek rychleji interpretovatelný z pravého grafu. Pokud ale otázku rozšíříme na „Které auto, jehož cena je nižší než 11 000 dolarů, má nejméně najeto?“, je hůře interpretovatelný pravý graf.

Grafické symboly

Jako grafický symbol označujeme nějaký vizuální objekt, který je jednoduše rozpoznatelný. Grafické symboly jsou často součástí složitějších vizualizací, jako například různé šipky, štítky (labels) a podobně. Podíváme se, jakým způsobem navrhovat grafické symboly a jak jsou vnímány. Oblast zabývající se grafickými symboly a značkami se v angličtině označuje jako **sémiotika (semiology)**.

Libovolný „výtvar“ ve 2D Euklidovském prostoru můžeme považovat za grafickou reprezentaci sestávající z grafických symbolů. Mohou to být diagramy, sítě, mapy, grafy a další běžně užívané vizualizace.

Nyní se soustředíme na různé charakteristiky vizualizace, obrázků a grafiky vytvořené z grafických symbolů.



Na obrázku vidíme dvě grafické symboly či reprezentace. Levý obrázek je velmi jednoduše rozpoznatelný, zejména proto, že jej pravděpodobně všichni známe – využíváme svoje předchozí znalosti a zkušenosti. Na druhou stranu, pro pochopení pravého obrázku potřebujeme mnohem více soustředění a pozornosti.

Proces vnímání a porozumění obrázku probíhá v několika krocích. Pochopení levého obrázku je díky jeho jednoduchosti a „známosti“ možné v jediném kroku, kdy si symbol asociujeme s jeho významem. U pravého obrázku je situace složitější, proto porozumění probíhá ve dvou krocích. V prvním kroku identifikujeme hlavní a význačné prvky obrázku a ve druhém kroku se pokusíme rozpoznat různé vztahy mezi těmito prvky. V této fázi se rovněž snažíme obrázek asociovat s předchozími znalostmi uloženými v dlouhodobé paměti. Ve výsledku definujeme v obrázku nejzajímavější informaci (například klastry, geny, státy, ...).

Bez vnější – kognitivní identifikace nemá žádná grafická reprezentace význam. Externí identifikace musí být jednoduše čitelná a pochopitelná.

Detekce vztahů a vzorů v dané grafické reprezentaci probíhá ve dvou základních krocích.

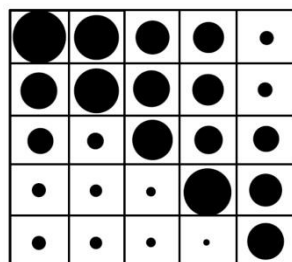
1. Mapování mezi vztahy grafických symbolů a daty, která tyto symboly reprezentují. Jinak řečeno, každý vzor na obrazovce musí odpovídat vzoru ve vstupních datech. Pokud tomu tak není, jedná se o rušivý artefakt vybrané vizuální reprezentace.
2. Uspořádání grafických symbolů, které odpovídá uspořádání na vstupních datech a naopak. Podobně jako u mapování, pokud v obrázku vnímáme uspořádání, které se ve vstupních datech nevyskytuje, opět jsme použili chybnou vizuální reprezentaci.

Obecně tedy musíme dbát na to, aby platilo:

- Podobnost mezi daty \Leftrightarrow vizuální podobnost odpovídajících grafických symbolů
- Uspořádání na datech \Leftrightarrow vizuální uspořádání mezi odpovídajícími grafickými symboly

Dimenzionalita 2D grafiky

2D grafika použitá při vizualizacích často zobrazuje tři i více dimenzí. Příkladem je daný obrázek, který zobrazuje matici s body různých velikostí umístěných v mřížce.



Každý z bodů může být interpretován jako vztah mezi pozicí v ose x a v ose y. Navíc se body liší ve velikosti, čímž dodávají do obrázku třetí rozměr. Ten můžeme považovat za osu z. Některé vztahy jsou zřejmé na první pohled, jako například velikost kruhů na diagonále (navíc kruhy v horní části diagonály jsou větší než ve spodní). Sada těchto bodů reprezentuje všechny vztahy mezi souřadnicemi x, y a z, které jsou přítomny ve vstupní datové množině.



Dalším příkladem je tento obrázek. Na první pohled vidíme strom se dvěma větvemi. Okamžitě víme, že je to strom, aniž bychom si museli detailně prohlížet všechny detaily obrázku, jako například listy. Grafika může obsahovat velké množství dat, která jsou sama o sobě rovněž grafickými prvky. Jediným omezením je, že tyto prvky musí být od sebe rozeznatelné. Lidské oko je pak schopné rozeznat rozložení jednotlivých grafických symbolů.

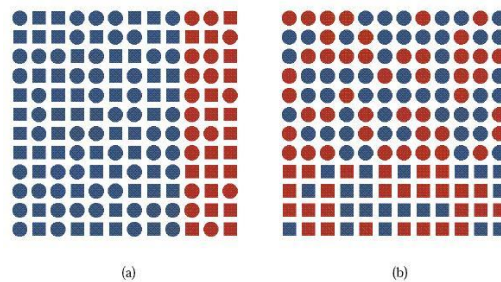
Analýza grafiky

Při analyzování grafiky nejdříve vnímáme seskupení objektů (preatentivně = podvědomým vnímáním). Podvědomé zpracování vizuální informace probíhá pomocí detekce základních vlastností objektů ve scéně. Mezi tyto vlastnosti patří například barva, délka, velikost, zakřivení atd. Obrovskou výhodou je, že tento způsob zpracování pomáhá velmi rychle zacílit pozornost na cíl pouze použitím jedinečného vizuálního prvku.

Tabulka ukazuje seznam známých vizuálních prvků, které vnímáme podvědomím, společně s referencí na studii, ve které bylo prokázáno, že jsou podvědomě vnímány.

Feature	Author
line (blob) orientation	Julézs & Bergen [1983]; Wolfe [1992]
length	Triesman & Gormican [1988]
width	Julézs [1985]
size	Triesman & Gelade [1980]
curvature	Triesman & Gormican [1988]
number	Julézs [1985]; Trick & Pylyshyn [1994]
terminators	Julézs & Bergen [1983]
intersection	Julézs & Bergen [1983]
closure	Enns [1986]; Triesman & Souther [1985]
colour [hue]	Triesman & Gormican [1988]; Nagy & Sanchez [1990]; D'Zmura [1991]
intensity	Beck et al. [1983]; Triesman & Gormican [1988]
flicker	Julézs [1971]
direction of motion	Nakayama & Silverman [1986]; Driver & McLeod [1992]
binocular lustre	Wolfe & Franzel [1988]
stereoscopic depth	Nakayama & Silverman [1986]
3-D depth cues	Enns [1990]
lighting direction	Enns [1990]

Obrázek pak znázorňuje příklad zobrazení segregace daného regionu pomocí tvaru a barvy. Hranice v regionu je vnímána lépe, pokud je oddělena barevně. Druhý případ oddělení pomocí tvaru již vyžaduje více pozornosti.



Ve druhém kroku se snažíme kognitivně tato seskupení charakterizovat. To zahrnuje proces poznání, který v sobě zahrnuje mimo jiné rozhodování a usuzování. Nakonec se zaměříme na speciální případy, které nejsou přímo v těchto seskupeních, ani v jejich vztazích, obsažena. Celý proces je možné provádět na mnoha úrovních a pomocí různých typů vizualizačních technik.

Osm vizuálních proměnných

Abychom mohli grafiku využít pro zprostředkování informace, je nutné nejdříve porozumět jednotlivým grafickým primitivům a jejich vlastnostem. Grafická primitiva jsou často označována termínem „značky“ – **marks**. Jedním ze způsobů kódování vstupních dat pro zobrazení je využití mapování různých typů dat na různé značky a jejich atributy. Avšak samotné značky nezobrazí požadovanou informaci – pouze jejich vzájemné prostorové uspořádání jim dodá požadovanou sdílnost. Umístění značek do prostoru obrazovky dodává význam dodatečným vlastnostem vstupních dat, jako například podobnosti a rozložení dat.

Pokud již máme definovány typy a rozložení jednotlivých značek, můžeme každé ze značek přidat doplňující grafické vlastnosti. Značky se mohou lišit například velikostí, barvou či orientací.

Obecně můžeme rozlišit osm způsobů, jakými mohou grafické objekty kódovat informaci – označujeme je jako 8 vizuálních proměnných:

- Pozice
- Tvar
- Velikost
- Jas
- Barva
- Orientace
- Textura
- Pohyb

Tyto proměnné mohou být využity pro maximalizaci efektivity dané vizualizační metody.

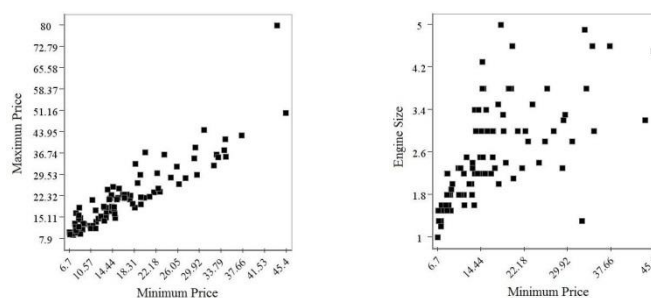
Na obrázku vidíte vizuální proměnné definované Jaquesem Bertinem. V jeho modelu chybí osmá proměnná – pohyb.

Bertin's Original Visual Variables	
Position changes in the x, y location	
Size change in length, area or repetition	
Shape infinite number of shapes	
Value changes from light to dark	
Colour changes in hue at a given value	
Orientation changes in alignment	
Texture variation in 'grain'	

Nyní se detailněji podíváme na jednotlivé proměnné.

Pozice

První a nejdůležitější vizuální proměnnou je pozice – umístění jednotlivých grafických prvků v prostoru obrazovky. Pozice má největší dopad na zobrazení dané informace, protože prostorové uspořádání grafiky je prvním krokem v porozumění vizualizaci. Platí, že čím větší je rozložení grafických prvků po obrazovce, tím větší množství informace jsme schopni předat. Nejhorším případem je právě vizualizace, při které jsou všechna data mapována na grafické symboly umístěné všechny do jednoho místa – ve výsledku vidíme jenom ten symbol, který byl zobrazen jako poslední a zbytek informací ztrácíme. Naopak nejlepší schéma mapuje každý grafický symbol na jeho unikátní pozici, kde se nepřekrývá s ostatními. Pro standardní obrazovky s rozlišením 1024 x 768 dostáváme počet individuálních pixelů pouze 786 432. Na takovéto obrazovce tedy nejsme schopni zobrazit milion individuálních symbolů. Navíc vzhledem k tomu, že pro rozumné zobrazení a porozumění dané informaci je potřeba více než jeden pixel, toto maximální číslo se ještě výrazně snižuje.



Příklad významu pozice je znázorněn na obrázku. Na levém obrázku jsou zobrazeny minimální vs. maximální ceny aut vyrobených v roce 1993. Rozložení jednotlivých bodů indikuje lineární vztah mezi minimální a maximální cenou.

Druhý obrázek ukazuje jinou sadu dat – porovnává minimální cenu a velikost motoru pro auta vyrobená v roce 1993. Na rozdíl od prvního obrázku zde není patrný žádný významný vztah mezi těmito proměnnými.

Výběr proměnných pro organizaci dat na obrazovce může zodpovědět mnoho otázek, například: Do které oblasti spadá nejvíce dat? Kopírují data některé ze známých rozložení? Jsou v datech rozpoznatelné nějaké trendy? Můžeme data nějakým způsobem klastrovat?

Další možností, jak objevit různé vztahy, souvislosti a struktury uvnitř zobrazovaných dat, je využití scalingu – změny velikosti. Obecně použití scalingu není nutné, nicméně v některých případech musí být použito.

Prvním typem je **lineární scale**, s jehož pomocí jednoduše zvětšujeme či zmenšujeme rozsah hodnot.

Druhým typem je **logaritmický scale**, který slouží pro mapování exponenciálně se měnících proměnných na více ucelený (kompaktní) rozsah.

Ačkoliv jsou oba tyto typy scalingu běžně aplikovány na jednotlivé proměnné, občas je nutné je využít v případech dvoudimenzionálního mapování – například v kartografii. V tomto případě mluvíme o projekci.

Ve většině případů je velmi výhodné do dané vizualizace dodat další doplňkovou grafiku, která pomůže popsat daný prostor. Příkladem takovýchto prvků jsou **osy**. Ty většinou obsahují dílky, které definují intervaly rozsahu hodnot a textové značky popisující samotné hodnoty. Každá osa navíc obsahuje popisek, ke kterým datům se vztahuje.

Tvar

Druhou vizuální proměnnou je tvar neboli značka: body, čáry, oblasti, objemy a jejich kombinace. Značky jsou grafická primitiva reprezentující data. Například obě vizualizace týkající se automobilů z roku 1993 v předchozí sekci využívaly grafické primitivum bod pro zobrazení jednotlivých hodnot. Jakýkoliv grafický objekt může být využit jako značka, včetně symbolů, písmen a slov.

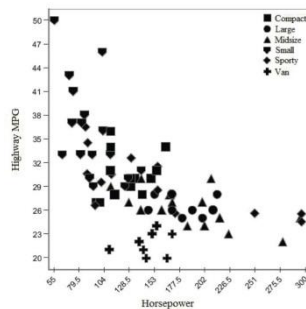
Příklady značek jsou na obrázku.



Když mluvíme o značkách, je důležité nezmiňovat jejich velikost, jasovou složku, orientaci a podobně, protože toto jsou další vizuální proměnné, kterými se budeme zabývat dále.

Při používání značek je důležité vzít v úvahu rozlišitelnost mezi použitými značkami. V jedné vizualizaci je někdy možné pozorovat i stovky až tisíce jednotlivých značek. Je nutné tedy vybírat takové značky, které si nejsou příliš podobné.

Jako příklad si uveďme využití značek, které jsme si ukázali na předchozím slajdu. Tato vizualizace používá různé tvary značek pro rozlišení různých typů aut v grafu porovnávajícím spotřebu a výkon motoru. Je možné pozorovat jednotlivé klastry.



Velikost

Předchozí dvě proměnné tvořily základ veškerých vizualizací. Bez nich by bylo nemožné vizualizaci vytvořit. Zbývající proměnné ovlivňují způsob zobrazení jednotlivých reprezentací.

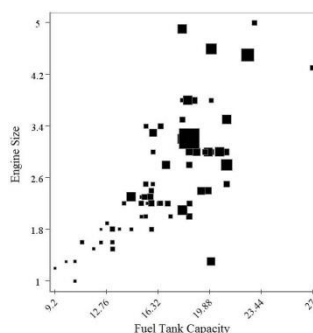
Třetí vizuální proměnnou je velikost. Ta určuje, jak malé či velké budou zobrazené značky (viz obrázek).



Velikost se dá jednoduše mapovat na interval a spojitě datové proměnné.

Velikost je použitelná zejména pro datové množiny s malou kardinalitou, protože je velmi těžké rozlišit značky, které se velikostně liší jen velmi málo.

Důležitým aspektem je vybrání typu značky. Pro body, čáry a křivky je použití velikosti velmi vhodné. To je vidět na obrázku, který opět ukazuje datovou množinu reprezentující modely aut z roku 1993. Nyní zobrazujeme velikost motoru vůči velikosti nádrže. Velikost značek odpovídá velikosti maximální ceny za danou kombinaci.



Pokud grafika obsahuje značky ve formě oblastí, je mapování dalších informací ve formě velikosti nevhodné.

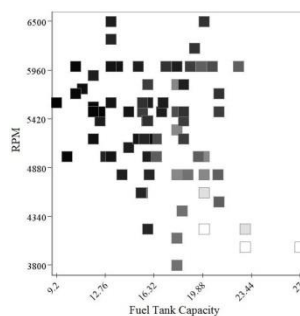
Jas

Čtvrtou vizuální proměnnou je jas neboli svítivost (brightness, luminance). Je druhou proměnnou, která slouží k modifikaci značek tím způsobem, aby prezentovaly další datové proměnné či vlastnosti.

Díky omezeným možnostem lidského vnímání, které nedokáže rozlišit všechny existující stupně jasu, se dá tato proměnná využít pro zobrazení spojitých datových proměnných nebo pro přesné rozlišení mezi značkami vykreslenými v navzorkovaných odstínech jasu. Dále je doporučováno, aby škála jasu byla lineární, což maximalizuje vnímatelnost rozdílů.



Příkladem využití jasové složky na již známém případě srovnání aut z roku 1993 je znázorněno na obrázku. Zde je jasová složka využita pro zobrazení šířky auta. Čím tmavší bod, tím širší auto.



Barva

Barvou a jejím vnímáním už jsme se zabývali dříve, proto si nyní pouze doplníme některé informace. Rozdíl mezi jasem a barvou je v tom, že barva je definována dvěma parametry – odstínem (hue) a sytostí (saturation).

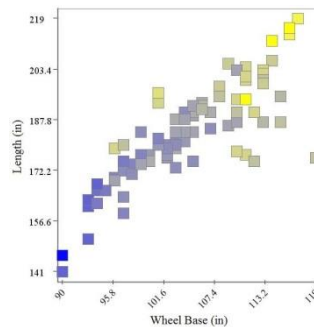
Obrázek znázorňující barevnou škálu je převzat z programů společnosti Microsoft, kde je odstín vynesena na horizontální osu a sytost na vertikální. Odstín je reprezentován všemi barvami vizuálního spektra. Sytost je definována stupněm odstínu relativně k šedé barvě a ovlivňuje „čistotu“ dané barvy.

Využití barvy pro zobrazení informace vyžaduje mapování datových hodnot na jednotlivé barvy. Pro tento účel je nejdříve nutné nadefinovat tzv. colormap (mapu barev), která

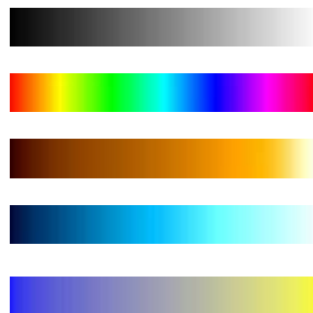
specifikuje vztah mezi rozsahem hodnot a jednotlivými hodnotami barev. Barevné mapy jsou vhodné pro vizualizaci diskrétních i spojitých datových proměnných.



Obrázek představuje příklad barevné mapy a jejího využití pro náš případ s auty. Zde je barva využita pro zobrazení délky auta. Tato délka je zároveň asociována s osou y. Modrá označuje kratší auta, žlutá pak ta delší.



Obrázek ukazuje časté a oblíbené typy barevných map.



Orientace

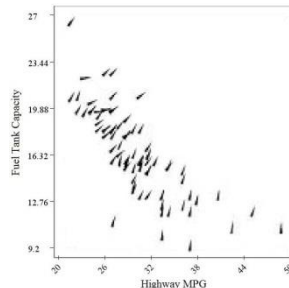
Šestou vizuální proměnnou je orientace neboli směr. Orientace je velmi spojena s podvědomým vnímáním. Tato vlastnost grafických prvků popisuje, jak je příslušná značka otáčena v závislosti na datové proměnné. Je zřejmé, že orientaci nelze využít u všech tvarů značek – například kruh nepodléhá změnám při rotaci. Nejvhodnější značky pro zobrazení orientace jsou takové, které mají svoji „hlavní osu“. Změna orientace pak probíhá kolem této osy. Takovéto značky pak mohou zobrazit celou škálu různých orientací, jako na příkladu na obrázku. Značka je reprezentována protáhlým trojúhelníkem, u kterého je zřejmá hlavní osa.



Obrázek ukazuje možnost využití této značky pouze do 90°, nicméně tato značka je použitelná pro celý rozsah 360°. Interpretace je taková, že nejnižší hodnoty dat jsou

mapovány na značku směřující hrotem nahoru (nejlevější) a zvyšující se hodnoty dat rotují značku ve směru hodinových ručiček.

Graf ukazuje opět vztah spotřeby a kapacity nádrže. Doplňková datová proměnná zobrazená pomocí našich trojúhelníkových značek je průměrná cena.



Textura

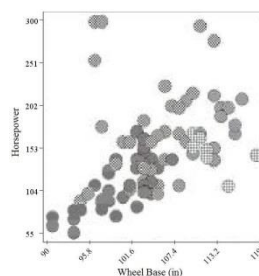
Předposlední vizuální proměnná je textura. Může být považována za kombinaci několika ostatních vizuálních proměnných, včetně značek (textely), barvy (spojené s každým pixelem v oblasti textury) a orientace (zprostředkována změnami v barvě).

Některé jednoduché textury mající lineární vzhled jsou tvořeny pomocí čárkovaných a tečkovaných čar. Je možné mezi nimi jednoduše rozlišovat, protože existuje pouze omezený počet takovýchto textur.



Textura je nejčastěji spojována s polygony, regiony a povrchy. Ve 3D je textura zpravidla považována za atribut geometrie. Podobně může být textura asociována s barvou grafické entity, s pravidelnými i nepravidelnými variacemi v barvě s různým rozsahem a rozložením.

Příkladem využití textury pro náš případ je mapování typu auta na různé textury.



Pohyb

Poslední vizuální proměnnou je pohyb. Ve skutečnosti může být pohyb asociován s kteroukoliv z dříve uvedených vizuálních proměnných. Jedním z běžných využití pohybu je při změně rychlosti, se kterou se daná změna provádí (například změna pozice nebo blikání).

Důležitým aspektem pohybu je jeho směr – v osách, diagonálně nebo libovolný. Pro ostatní vizuální proměnné je to změna: větší/menší, světlejší/tmavší, větší úhel/menší úhel atd.

Význam vizuálních proměnných

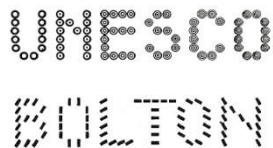
Různé vizuální proměnné slouží k různým účelům. Tyto účely lze kategorizovat několika způsoby. Nyní si uvedeme příklad takovéto kategorizace.

Selektivní vizuální proměnné – datové hodnoty jsou pozorovatelem rozděleny do příslušných skupin (pro vizualizaci nominálních dat):

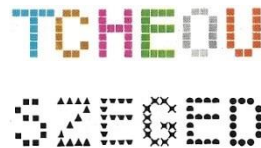
- velikost (délka, obsah/objem)
- jas
- textura
- barva (pouze základní barvy)
- směr/orientace

Asociativní vizuální proměnné – všechny faktory mají stejnou viditelnost (vizualizace nominálních dat):

- textura (viz obrázek vlevo nahoře)
- barva (viz obrázek vpravo nahoře)
- směr/orientace (viz obrázek vlevo dole)
- tvar (viz obrázek vpravo dole)



WESCO
SOLTON



TCHEOU
SAGEED

Ordinální vizuální proměnné – různé hodnoty dat jsou tříděny přímo pozorovatelem:

- textura
- velikost
- jas

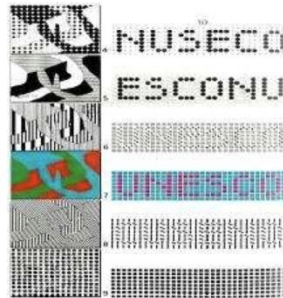
Proporcionální vizuální proměnné – tyto proměnné navíc obsahují přímou asociaci na relativní velikost (vizualizace ordinálních a kvantitativních dat):

- velikost (délka, obsah/objem)
- směr/orientace
- jas

Oddělovací vizuální proměnné – všechny prvky jsou viditelné.

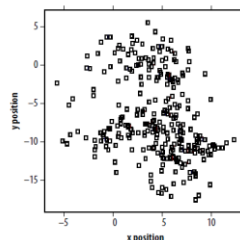
- textura (viz obrázek)

- barva
- směr/orientace
- tvar

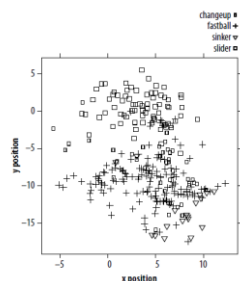


Příklad použití

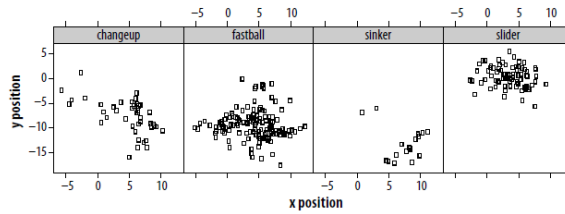
Význam jednotlivých vizuálních proměnných a jejich vliv na prezentované informace o vstupních datech si ukážeme na příkladu mapování 287mi odpalů baseballového hráče Oscara Villarreala v roce 2008. Použijeme-li pouze jediný grafický symbol stejných vlastností (barva, tvar, ...), můžeme ve 2D prezentovat pouze informaci o tom, kam jednotlivé odpaly směřovaly.



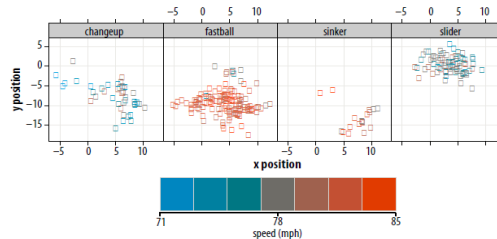
Přidáním zobrazení jednotlivých položek pomocí různých typů glyfů dostaneme do výsledné vizualizace další rozměr dat a to typ jednotlivých odpalů.



Pro zvýšení přehlednosti je možné na základě typu odpalů rozdělit vstupní graf na podgrafy, kdy každý reprezentuje pouze data odpovídající danému typu odpalu. Každý z grafů má stejný souřadný systém.



Jednoduchým přidáním informace o barvě do grafů můžeme reprezentaci doplnit o další informaci, zde o rychlost odpálených míčků.

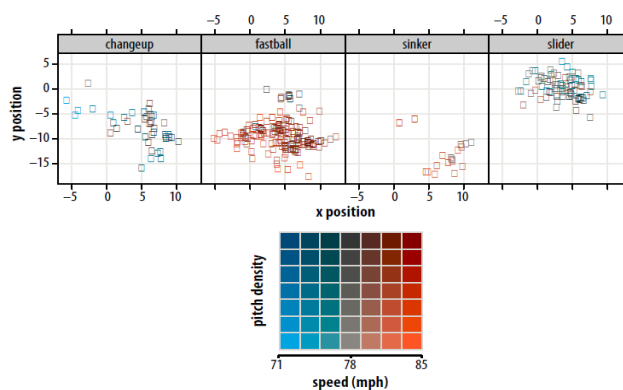


Barva je důležitým prvkem vizualizačních technik zejména proto, že rozdíly v barvě je možné detekovat již za 200 milisekund. Což je dokonce dříve, než si vůbec uvědomíme, že se na vizualizaci soustředíme. Tomu se říká tzv. preatentivní koncept.

Barva může být reprezentována ve třech dimenzích. Příkladem je RGB prostor, kde jsme schopni do jisté míry určit, zda je daná barva více či méně červenější, zelenější a modřejší. Běžně se však tato reprezentace příliš nepoužívá. Hlavními dvěma důvody jsou:

- Skutečnost, že poměrně velká část populace je zcela barvoslepá nebo má nějakou částečnou poruchu barevného vidění.
- Jednotlivé barvy mají různé rozsahy odstínů barev, které jsme schopni vnímat. Příkladem je žlutá a modrá barva, kdy ve žluté jsme schopni vnímat rozdíly mezi mnohem menším počtem odstínů než v modré.

Do našeho příkladu s baseballovými odpaly přidáme poslední informaci o hustotě odpalů v daném místě hřiště. Hustota je mapována na druhou dimenzi použité barevné škály a odpovídá odstínu dané barvy.



Krátký pohled do historie

I když jsme si již ukázali mnoho příkladů využití vizualizačních technik, stále jsme ještě nepopsali souhrnný aparát pro popis našich grafických reprezentací. Vizualizace je celá vědní disciplína, která konzistentně formalizuje obecné vizualizace. Výzkumníci v této oblasti přicházeli a přichází se svými myšlenkami v podobě specifikace různých modelů. Robertson jako první pojmenoval potřebu formálního modelu jako základu každého vizualizačního systému.

V této sekci ukážeme příklad formalizace oblasti vizualizace a seznam existujících formalizací.

Bertin (1967) Semiology of Graphics

V roce 1967 Jacques Bertin, pravděpodobně nejvýznamnější postava teorie vizualizace, publikoval knihu *Sémiologie Graphique*. Jednalo se o první rigorózní pokus o definici grafiky a její aplikace na zobrazení informace. Bertin zde prezentuje základy kódování informace prostřednictvím grafické reprezentace jako tzv. sémiotiku – vědu zabývající se značkovacími systémy.

První jeho klíčový poznatek je striktní oddělení obsahu (informace, kterou chceme kódovat) od „přenosového média“ (vlastnosti použitého grafického systému).

Abychom mohli zcela pochopit značkovací systém (sign system), je nutné nejdříve pochopit všechny základní prvky, které tento systém definují. Na základě tohoto poznatku Bertin sestavil tzv. grafický slovník.

Marks	Points, lines, and areas
Positional	Two planar dimensions
Retinal	Size, value, texture, color, orientation, and shape

Bertinův grafický slovník je znázorněn v tabulce. Jeho cílem je identifikovat některá velmi specifická grafická primitiva, která odpovídají vjemovým (percepčním) atributům – vlastnosti grafiky, které jsou vjemově identifikovatelné.

Bertin dále definuje pojem „plane“ - rovina, za kterou považuje čistý arch bílého papíru standardní velikosti. Dále poznamenává, že rovina je tak známým objektem, že jeho vlastnosti není třeba popisovat a dokazovat, avšak ty nejběžnější věci jsou často chápány pouze omezeně. Rovina je označena tzv. implantacemi – tvoří základní prvky rovinné geometrie (body, čáry, plochy). Tyto tři prvky jsou umístěny ve dvoudimenzionálním prostoru pomocí imposition (zavedení), které rozděluje grafiku do čtyř skupin: diagramy, sítě, mapy a symboly. Pomocí implantation a imposition je tedy definována vizualizace.

Třetí a poslední komponentou je grafická reprezentace využívající tzv. retinální (sítnicové) proměnné (retinal variables) – grafické odchylky navrženy pro vizuální vnímání. Tyto proměnné ovlivňují implantace. Experimentální psychologie definuje 6 typů retinálních proměnných:

Velikost (výška, obsah nebo číslo)

- Hodnota (sytost)
- Textura (jemnost nebo hrubost)
- Barva (jas)
- Orientace (úhlový posun)
- Tvar

Bertin dále definoval několik úrovní organizace specifikující, jak mohou být tyto proměnné kombinovány.

Další systémy:

- Mackinlay (1986) APT

<http://assassin.cs.rpi.edu/~cutler/classes/visualization/F10/papers/p110-mackinlay.pdf>

- Bergeron a Grinstein (1989) Visualization Reference Model
- Wehrend a Lewis (1990)
- Robertson (1990) Natural Scene Paradigm

Cílem Natural Scene Paradigm je mapování dat na vlastnosti realistických scén (například hory a údolí mohou reprezentovat hustotu či barvu). Robertson vzal v úvahu, že lidé mají vysoce vyvinutý smysl pro analýzu různých aspektů přírodních scénérií a pokusil se tuto skutečnost využít.

- Roth (1991) Visage and SAGE
- Casner (1991) BOZ
- Beshers a Feiner (1992) AutoVisual
- Senay a Ignatius (1994) VISTA
- Hibbard (1994) Lattice Model
- Golovchinsky (1995) AVE
- Card, Mackinlay a Shneiderman (1999) Spatial Substrate
- Kamps (1999) EAVE
- Wilkinson (1999) Grammar of Graphics
- Hoffman (2000) Table Visualizations

Shrnutí historie

Existuje celá řada grafických modelů, které formalizují grafické objekty používané pro zobrazení informace. Primárně tyto modely charakterizují grafiku pomocí tří komponent: značky, pozice a perceptuální proměnné. Výzkum těchto komponent byl iniciován Bertinem a neustále pokračuje. Poskytuje náhled na základy a teorii vizualizace. Většina nejnovějších výzkumů se shoduje, že Wilkinson vytvořil nejdůkladnější formalizaci statistické grafiky, která postihuje většinu základních prvků existujících vizualizačních technik.

Příklad využití grafických proměnných – Wordle

Jedním z příkladů specifického typu vizualizace jsou barevné slovní koláže. Jednou z hlavních aplikací tohoto typu je webová aplikace Wordle, kterou vytvořil Jonathan Feinberg. Primárně byla tato aplikace navržena pro zábavu. Je založena na principu tzv. tag clouds.

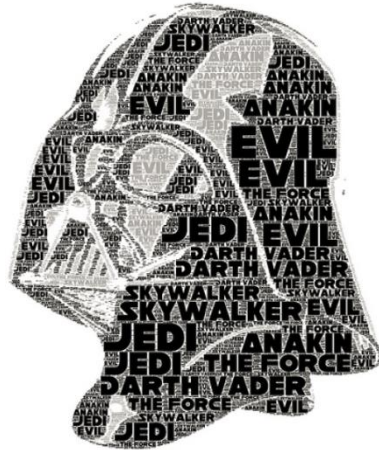


Původní myšlenka na systém zpracovávající tag clouds vznikla už v roce 2004, kdy vznikla aplikace nazvaná *dogear*. V tomto případě ještě nebyl kladen přílišný důraz na vizuální složku.



V stejném roce Matt Jones publikoval ručně vytvořený tag cloud, který naopak kladl velký důraz na typografické rozložení textu, které bylo navíc vizuálně přitažlivé. To motivovalo Jonathana k vytvoření programu, který by stejného výsledku mohl dosáhnout automaticky.





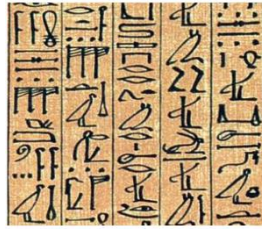
Jared Tarbell vytvořil vizualizaci, ve které je prostor rekurzivně dělen na stále menší náhodné obdélníky, čímž prostor vyplňujeme stále menšími slovy. Tento efekt vyžaduje velkou vstupní množinu kandidátních slov, kterým jsou přiřazeny náhodné váhy.



V případě, že nám nevadí deformace fontu protažením či smrštěním slov, můžeme použít další efekty. Příkladem je treemap, ve které je místo barevných obdélníků použita sada slov.



Z historického hlediska není tento princip ničím novým. Vezměme například hieroglyfy. Ty jsou důkazem toho, že ještě dříve, než se začaly dělat obrázky ze slov, lidé vytvářeli slova z obrázků.



Jak Wordle pracuje

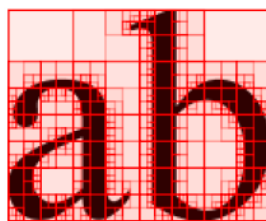
Principem aplikace Wordle je vykreslit slova s určitou vahou, která určuje velikost těchto slov. Na vstupní text je aplikován regulární výraz, který „rozseká“ vstupní text na jednotlivá slova. Poté jsou odstraněny tzv. *stop words*, což jsou slova, která se v jednotlivých jazycích vyskytují velmi často a přitom nemají žádný význam, který by měl vliv na informaci sdílenou pomocí tag cloudu. Příkladem těchto slov v angličtině jsou například „the“, „it“, „to“. Důležitým aspektem aplikace je funkce, která zjistí, jaký je jazyk vstupního textu.

Určení typu jazyka vstupního textu probíhá takovým způsobem, že je vybráno 50 nejčastějších slov ze vstupního textu a ta jsou porovnána se stop words podporovaných jazyků. Jednotlivým slovům jsou přiřazeny váhy, které odpovídají frekvenci výskytu daných slov ve vstupním textu.

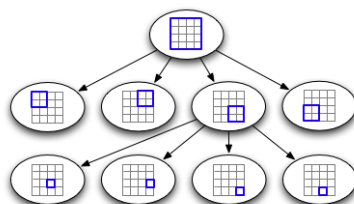
Dále je pro každé slovo určen font, velikost je definována právě přiřazenou vahou. Poté je každému slovu přiřazen bounding box a jejich součet určuje rozmístění a celkovou plochu, kterou výsledná vizualizace zabere. Podle této hodnoty je pak určen poměr, ve kterém musí být tato plocha škálována dle výstupních požadavků (např. velikost obrazovky).

Jak již bylo zmíněno, rozmístění slov je prováděno na základě náhodnostního hladového algoritmu. Když je dané slovo umístěno, jeho pozice už se nezmění.

Pro detekci průniků jednotlivých slov se využívají hierarchické bounding boxy. Při jejich protnutí na vyšší úrovni sestupujeme do nižší úrovně hierarchie a průsečík testujeme znovu.



Druhou možností, jak detekovat průsečíky mezi slovy, je testování kolize aktuálně vkládaného slova s již umístěnými slovy za využití pomocné struktury *quad-tree*. Poté, co je slovo umístěno, je určen nejmenší čtverec, který dané slovo plně obklopuje a tato informace je vložena do příslušného uzlu.



Wordle nebere v potaz délku jednotlivých slov, slova o stejné váze zobrazuje stejnou velikostí. To může vést k dojmu, že delší slovo je více časté, protože zabírá větší prostor.

Další vlastností je, že barva ve Wordle nemá žádný skrytý význam. Uživatel si může zvolit z několika přednastavených barevných palet nebo vytvořit vlastní paletu.

Wordle byl využit pro různé účely, například v business prezentacích, disertačních pracích, novinových článcích či televizních spotech. Galerie obsahuje více než 1 400 000 vytvořených word clouds. Dokonce byla vytvořena studie, jakým způsobem vnímají uživatelé práci s word clouds. Výsledky jsou shrnuty v následující tabulce.

	Agree %	Neutral %	Disagree %
I felt creative	88	9	4
I felt an emotional reaction	66	22	12
I learned something new about the text	63	24	13
It confirmed my understanding of the text	57	33	10
It jogged my memory	50	35	15
The Wordle confused me	5	9	86