

MASARYKOVA UNIVERZITA
FAKULTA INFORMATIKY



Editor letteristické stylizace rastrového obrazu

BAKALÁŘSKÁ PRÁCE

Jan Soukal

Brno, jaro 2006

Prohlášení

Prohlašuji, že tato bakalářská práce je mým původním autorským dílem, které jsem vypracoval samostatně. Všechny zdroje, prameny a literaturu, které jsem při vypracování používal nebo z nich čerpal, v práci řádně cituji s uvedením úplného odkazu na příslušný zdroj.

Vedoucí práce: Mgr. Tomáš Staudek, Ph.D.

Shrnutí

Náplní bakalářské práce je implementace prostředí pro kreativní práci s rastrovým obrazem. Filtr Letterist poskytuje možnosti pro editaci rastrového obrazu převážně pomocí jednotlivých znaků písma, ale také nabízí řadu dalších často využívaných grafických nástrojů a filtrů. Zjednodušeně by bylo možné říci, že podstatou programu je tvorba jakési koláže v letteristickém stylu.

Program je vytvořen v jazyce ObjectPascal pro platformu Win32. Jako vývojové prostředí bylo použito Borland Delphi 7. Práce je realizována ve formě dynamické knihovny, která slouží jako plug-in k již existujícímu grafickému editoru Elephant.

Klíčová slova

nefotorealistické zobrazování, abstrakce, koláž, letterismus, ASCII art

Obsah

1	Úvod	1
1.1	Computer-aided art	1
1.2	Přístup k obrazu	2
1.3	Font prostředkem stylizace	3
2	Letteristické zpracování obrazu	5
2.1	Přístup k pravděpodobnostem ve filtru	5
2.2	Typy jádra filtru	6
2.3	Grid - pracovní mřížka	8
2.4	Element náhody při zpracování	8
2.5	Font a množina znaků	9
3	Editor Letterist	10
3.1	Popis software	10
3.2	Instalace softwaru	10
3.3	Grafické rozhraní	11
3.3.1	Horizontální panel nástrojů	12
3.3.2	Vertikální panel nástrojů	13
4	Implementace	16
4.1	Knihovna Letterist	16
4.2	Jednotka MAINUnit	17
4.2.1	Funkce kategorie LAYERS	17
4.2.2	Funkce kategorie PANELS	18
4.2.3	Funkce kategorie FILTER	19
4.2.4	Funkce kategorie LETTERIST	19
4.2.5	Data v programu	20
4.3	Jednotka CHARSET	21
4.4	Algoritmy letteristického zpracování	21
4.4.1	Black Pixels only filtr	21
4.4.2	Black Pixels only filtr	22
5	Závěr	24
	Literatura	25
A	Obrazová příloha	26

Kapitola 1

Úvod

Abstraktní tvorba a stylizace skutečnosti jsou jistě nedílnou součástí nejen současného umění. Touhu určitým způsobem ozvláštnit a oživit realitu alespoň na plátně či jiném výtvarném médium často zažívá spousta umělců. Některým k uspokojení bohatě stačí štětec a barvy, jiní si vyhraji s nůžkami a spoustou novinových a časopisových fotografií. Prostředků a nástrojů pro práci s obrazem, případně s jeho samotnou tvorbou je skutečně mnoho. Svou bakalářskou práci se snažím přidat jeden další nástroj, který umožní kreativním lidem s touhou po abstrakci realizovat nápady a myšlenky třeba poněkud odlišnou formou, než na jakou byli dosud zvyklí.

Grafický filtr Letterist je primárně zameřený na editaci rastrového obrazu za pomoci písmen. Myslím, že velmi široká škála fontů a jistým způsobem symbolistická role písma jako takového ve společnosti dává myšlence abstrakce užitím znaků abecedy více než slušný potenciál. Snaha vyjadřovat obrazovou informaci použitím textu není žádná novinka, každý jistě alespoň na střední škole slyšel něco o kaligramech, případně při brouzdání po internetu narazil na "nějaký ten ASCII art", což ovšem neznamená, že by šlo o oblast, co se výtvarných prostředků týče, vyčerpanou, prozkoumanou a nemající co nabídnout. Podle mého názoru je to právě naopak, což se snažím ukázat právě implementací alespoň částečně originálního přístupu k písmenu co by štětcí.

Pracovní možnosti, které filtr Letteris nabízí, se více blíží možnostem grafického editoru než jen jediného filtru. Ve své podstatě totiž Letterist ani není filtr jako takový, jde spíš o rozhraní nabízející rozličné možnosti práce s rastrovým obrazem. Na Fakultě informatiky vytvořil v rámci bakalářské práce Juraj Šefara jednoduchý grafický editor Elephant, jehož hlavní síla spočívá v možnosti dodávání a pozdějšího vytváření zásuvných modulů. Proto jsem se rozhodl vytvořit filtr Letterist jako dynamickou knihovnu, jako plug-in, pro Elephant. V současné době již existuje několik filtrů, které byly napsány přímo pro tento editor a pokud budou přibývat další, z Elephantu se postupem času vyvine mocný nástroj pro práci s rastrovým obrazem.

1.1 Computer-aided art

Naivní je ten, kdo se domnívá, že umění nelze tvořit pomocí počítače. Najde se spousta takových, co budou zarytě zastávat přístup k umění zcela vyjímající jakoukoliv výpočetní techniku. Je třeba se však zamyslet a najít samotný princip umělecké tvorby, jednoduchý recept myšlenka, nástroj, plátno. Stejně jako je malířovým nástrojem štětec a barva, jsou v rámci

computer-aided artu (umění podporovaném počítačem) těmi samými elementy editor (program, algoritmus,...) a výsledný, byť binární zápis obrazové informace. To, co dělá opravdového umělce, je právě ona myšlenka, cit nebo nápad a ne skutečnost, zdali tluče dlátem do kusu dřeva nebo sedí před monitorem a zamyšleně se dívá na výsledek svojí práce[3].

Spousta lidí je přesvědčená, že tvorba pomocí počítačových programů práci velmi usnadňuje a proto se mylně domnívají, že umění tvořené za pomoci grafických editorů, je uměním do jisté míry laciným a tuctovým. Částečně by bylo možno s takovou myšlenkou souhlasit (ale opravdu jen částečně), neboť právě díky obrovským možnostem, které počítač při uměleckém zpracování obrazu nabízí, se objevuje velké množství nadšenců, co nějakým způsobem nastaví parametry filtru, počkají si na výsledek a přesvědčení o svém nedozírném talentu prezentují své výtvořky jako umělecká díla první kvality. Existuje ale i jiná skupina počítačových výtvarníků. Jsou to ti, kteří za počítač sedají s myšlenkou a myš a monitor jim slouží jen jako pomůcky vyjádření, jako prostor pro realizaci, který je jiný než ten s "klasickými" výtvarnými nástroji. Od takových tvůrců máme možnost se dočkat hodnotných výsledků, děl, která budou něco přinášet, budou mít určitou hloubku.

Filtr Letterist nestojí striktně ani na jedné straně barikády, i když je nutno poznamenat, že na práci metodou pokus-omyl je značně nevhodný. Přestože poskytuje uživateli určitou míru nejistoty, co přesně s obrázkem udělá, je nutné mít v hlavě alespoň rámcovou koncepci díla, aby měla práce smysl. Přístupů k filtru Letterist je jistě velmi mnoho, v rámci bakalářské práce se pokusím zmínit těch pár, na které jsem při práci s ním narazil a jež mě zaujaly.

1.2 Přístup k obrazu



Obrázek 1.1: Dvě vrstvy vzniklé detekcí hran (první zleva) a prahováním (druhá zleva). Dvě pravé vrstvy jsou letteristické

V každém obraze se vyskytují určitým způsobem výrazné rysy. Pokud na první pohled nejsou zřejmé, je možné je vytáhnout na povrch pomocí různých známých filtrů, nastavení či přizpůsobení. Letterist filtr se zabývá primárně úpravou šedotónních obrázků a předpo-

kládá dokonce černobílý výstup. Klíčovými atributy obrazu jsou proto mimo jiné ty, které lze do jisté míry zachovat i po ztrátě informace o stupni šedi při přechodu na binární obraz. Mezi takové vlastnosti patří například hrany nalezené vhodným filtrem. Jinou význačnou vlastností obrazu při práci s filtrem Letterist patří výrazné ohraničené černé a bílé plochy vzniklé trefným odprahováním obrazu. Na práci v editoru lze velmi zjednodušeně nahlížet jako na koláž. Základním konstrukčním kamenem filtru je písmeno, přesněji řečeno písmena na obraze. Ale i výše zmíněné vlastnosti obrazu se dají vhodně využít a navzdory poměrně nevzhlednému dojmu z každé vrstvy zvlášť, působí jejich kompozice příjemně odlišně.



Obrázek 1.2: Výsledná kompozice kytice

Při práci s filtrem Letterist je nutné mít na paměti, že filtr pracuje podle nastavení zadaného uživatelem. Bylo by příliš jednoduché, kdyby téměř nezáleželo na vstupním nastavení a přesto by filtr generoval plnohodnotné výstupy. Naneštěstí (nebo naštěstí) tomu tak není a správná a cílená práce v prostředí Letterist je velmi výrazně připoutána k adekvátnímu nastavení vstupních hodnot. Ve filtru je obsaženo několik klíčových parametrů, jejichž změny výrazným způsobem zasahují do celého procesu výpočtu a aplikace. Uživatel Letterist filtru by si měl přece jen udělat představu, co bude chtít s obrázkem udělat, jaké změny v samotné podstatě vstupu bude chtít provádět. Jednotlivé složky Letteristu jsou popsány v kapitole 2.

1.3 Font prostředkem stylizace

Zamyslíme-li se nad možnostmi využití písma jako výrazového prostředku, je zřejmých několi možností nazírání. Tím asi nejjednodušším je funkce, pro níž bylo písmo primárně vytvořeno, čili funkce komunikační. Avšak důležitý není jen obsah, ale i forma sdělení. Písmo proto procházelo určitými stádii vývoje, kdy byl kladen důraz na čitelnost a také na estetickou hodnotu jak jednotlivých znaků zvlášť tak i celé abecedy dohromady[7]. Filtr Letterist

přístupuje k písmu poněkud jiným způsobem.

Já jsem ukázka fontu
Já jsem ukázka fontu
Já jsem ukázka fontu

Obrázek 1.3: Z obrázku je patrná sestupná tendence "formálnosti" fontu. Použity byly shora dolů fonty: Times New Roman, Arial, ComicSans.

Téměř úplně je potlačena informační hodnota písma, dominantní postavení získává vizuální stránka fontu. Nejde jen o tvar, řez či rodinu písma. Jednotlivé fonty vnímají lidé odlišně, některá písma působí stroze, dalo by se říci konzervativně, jiná naopak evokují uvolnění a svobodu, viz. 1.3. Je proto velmi důležité brát v potaz i takové vlastnosti písma, které ovlivňují konečnou hodnotu upravovaného obrazu. Například při stylizaci fotografie gangstera z podsvětí Chicaga 20. let asi není nejvhodnější zvolit font Comic Sans, který nejen že uvolňuje atmosféru kompozice (a to ve většině případů u úpravy samopalem ozbrojeného mafiána asi není žádoucí), ale také působí hodně moderním, novým dojmem. Mnohem lépe by do této scény pasoval třeba font Courier, který krásně evokuje atmosféru staříčkových psacích strojů a sedí do prostředí počátku minulého století.

Kapitola 2

Letteristické zpracování obrazu

Základní sloužkou filtru Letterist je schopnost mapovat znaky zvoleného fontu podle uživatelem zadaných parametrů na rastrový obraz. Pro úplné zvládnutí práce v tomto grafickém editoru je proto nutné porozumět podstatě a principu samotných transformací. Jak již bylo zmíněno v 1.3, důležitým faktorem při stylizaci je zvolený font. Existují však další možná nastavení, která také významným způsobem zasahují do podoby výsledného obrazu. Mezi tyto atributy filtru patří nejnížší akceptovatelná pravděpodobnost shody, pravděpodobnost shody, která je dostatečná pro automatický výběr, a velikost mapovací mřížky (grid size). Velmi důležitou součástí je také použitý typ testování, zda se dané písmeno hodí na aktuální pozici mřížky, či nikoliv. Typy testování jsou v současné době implementovány dva, do budoucna se počítá s možným rozšířením. Jde o tzv. "Black Pixels only" metodu, která porovnává počty na stejném pixelu ležících černých bodů mřížky s tmavými místy matice písmene. Druhou testovací alternativou je "Black and White ratio" přístup, jenž porovnává šedý stupeň znaků zvoleného fontu a poměr černých a bílých pixelů v aktuálním políčku gridu. Neoddělitelnou součástí je v neposlední řadě také nastavení použití náhodného výběru písmen, avšak v rámci výše zmíněnými atributy ovlivněné pravděpodobnosti. Výsledek práce ve filtru významným způsobem ovlivňuje i nastavení množiny znaků (Charset), jež s sebou přenáší informaci, která písmena v rámci fontu budou použita při aplikaci a která se vůbec v potaz brát nebudou. Jde o soubor, podmnožinu kompletní abecedy uživatelem vybraného fontu určeného pro vykreslování.

2.1 Přístup k pravděpodobnostem ve filtru

Letterist používá dva typy uživatelem definovaných pravděpodobností, jež jsou v rámci aplikace filtru doslova klíčovými nastavením pro správnou práci. Tyto hodnoty udávají míru shody požadovanou uživatelem, jakýsi práh, podle níž algoritmus rozhoduje, které znaky do výsledné kompozice zařadit a které nechat stranou. Samotné využití prahů zadaných uživatelem ještě nezaručuje kvalitní práci. Podstatným je i fakt, jak k pravděpodobnostem přistupují jednotlivé typy filtru. V současné době implementované Black pixels only a Black and White ratio vnímají sice míry pravděpodobnosti stejně, ale v rámci jejich nároků se často může stávat, že pravděpodobnost zvolená pro první z nich vybere několik možných kandidátů z množiny znaků, zatímco při nastavení druhé možnosti typu filtru se žádné písmeno ani zdaleka nepřiblíží oné hranici, jež by umožnila jej vykreslit. Tento fakt je dobré mít na paměti při volbě parametrů pravděpodobnosti.

Jedním z těchto parametrů je tzv. nejnižší akceptovatelná pravděpodobnost. Tato hodnota udává určitý práh, která písmena jsou kandidáty na vykreslení do aktuální pozice v gridu. Omezuje rozsah celé zvolené množiny znaků na menší podmnožinu. Hodnota nejnižší akceptovatelné pravděpodobnosti je v obecném případě unikátní na každém poli mřížky zvlášť. V jednom případě může znamenat pravděpodobnost 100 % úplně stejný výsledek, jako na čtverci gridu o jiných souřadnicích nastavení hodnoty na pouhých 30 %.

Druhou zásadní možností volby pravděpodobnosti je pravděpodobnost pro automatický výběr. Tato hodnota je podobně jako výše zmíněná nejnižší akceptovatelná pravděpodobnost také svým způsobem jedinečná pro každou část gridu. Udává, jaký práh, jaká míra shody je nutná, aby byl znak bez dalších výpočtů a testů zapsán jako vyhovující a vykreslen na výsledný obraz. Principy použití různých nastavení pravděpodobností jsou popsány a vysvětleny v kapitole 4.4.

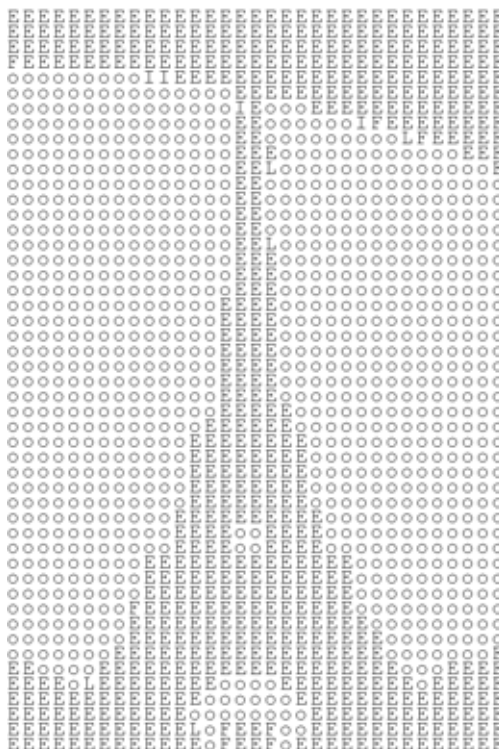
2.2 Typy jádra filtru

Ve filtru Letterist zajišťuje hlavní zpracování obrazu v souvislosti s fontem určitá základní funkce, dalo by se říci nějaký přístup k hodnotám. V současné době existují dva základní přístupy, jež filtr používá. Jsou jimi Black Pixels only a Black and White ratio. Oba tyto typy fungují velmi odlišně a jejich ani jejich výsledky si nejsou výrazně podobné. Při možném budoucím rozšiřování a zdokonalování filtru je možné vytvořit další přístupy, které budou fungovat opět trochu jiným způsobem.

První myšlenkou při tvorbě filtru byl přístup nazvaný Black pixels only porovnávající míru shody černých pixelů obrazu s černými body písma na korespondujících souřadnicích. Přesněji řečeno za každý černý pixel na zdrojovém obrazu a v matici znaku na shodných souřadnicích současně se zvyšuje šance, že testovaný znak dosáhne požadovaného limitu pravděpodobnosti a bude brán v potaz při výběru písmene, jež bude vykresleno do výsledného obrazu. Míra shody písma s podložkou je počítána vzhledem k celkovému počtu černých pixelů matice znaku, kdy spočítaná pravděpodobnost 100 % pro znak říká, že počet všech černých pixelů, které se na stejných souřadnicích vyskytnou současně je roven celkovému počtu černých bodů v matici daného písmene. Je dobré si uvědomit, že tento přístup přímo netestuje bílá místa. V praxi se totiž do souvislé černé plochy vykreslí jakékoliv písmeno, přestože matice znaku obsahuje i spoustu bílých míst, ta ovšem ve výpočtu zohledňována nejsou. Přístup testovat pozice jak černých tak bílých pixelů není ve filtru implementován. Jeho náročnost na pravděpodobnost shody je obrovská. Tento přístup byl odzkoušen při vývoji programu ASCIIInt[5], ale neosvědčil se. Do budoucna, po rozumném zvážení a úpravě takového typu filtru, je však jeho implementace možná.

Vyjma metody Black pixels only je v Letteristu implementován i přístup Black and White ratio. Jeho nosnou myšlenkou je srovnávat předlohu s maticí znaku na základě poměrného zastoupení černých pixelů. Jinými slovy Black and White ratio testuje jakýsi šedý stupeň jak na matici jednotlivých znaků tak i v rastrovém obraze, samozřejmě v závislosti na zvolené velikosti pracovní mřížky - gridu. Tento přístup ke srovnávání se oproti předchozímu Black pixels only výrazně liší.

Zamyslíme-li se nad principem práce obou typů filtru, asi je intuitivně zřejmé, v jakých oblastech obrazu bude který typ pracovat lépe a ve kterých hůře. Významnou vlastností Black and White ratio je fakt, že největší míru shody určuje na hraničních oblastech obrazu, na místech, kde se střídají výrazné bílé a černé plochy. V rámci Letteristu je tento přístup podobný běžnému hledání hran v rastrovém obraze. I když jeho výsledkem není hrana samotná, ale spíše její stylizace. Naproti tomu Black pixels only jsou mnohem vhodnější alternativou pro výplň, určité pseudo-šrafování rozsáhlých ploch. Při chytrém nastavení a vhodném zdrojovém obrázku ovšem také dokáže vytvořit zajímavé výsledky na hranách v obraze, podobně jako Black and White ratio. Co ovšem narušuje od něj nezvládně, je kompletní výplň obrazu znaky fontu malé velikosti a vytvoření obrazu, který neobsahuje nic jiného než jednu letteristickou vrstvu a přesto ponechává původní obrazovou informaci, byť modifikovanou, i v konečné kompozici.



Obrázek 2.1: Stylizace Eiffelovy věže. Obrázek byl vytvořen typem Black and White ratio nad abecedou (Charset) obsahující pouze písmena jména autora věže: E,I,F,L a pro okolí přidáného znaku 'o'.

2.3 Grid - pracovní mřížka

Důležitým faktorem při práci s filtrem je nastavení velikosti pracovní mřížky - gridu. Její velikost udává délku strany čtverce, na které se pomyslně rozdělí vstupní obraz. Levý horní roh každého čtverce potom ukazuje počátky jednotlivých míst, nad kterými se bude provádět filtrování. Nastavitelný rozměr gridu poskytuje uživateli silý nástroj pro ovlivnění filtru, ale přináší s sebou i jistá úskalí.

Při snaze o pochopení smyslu mřížky v pracovním prostředí je dobré si uvědomit několik faktů. Velikost gridu je udávána v pixelech, přičemž logicky nejnižší hodnota je rovna 1. Těžko si představit funkční filtr pracující nad čtverci o nulových rozměrech strany, které jsou navíc od sebe vzdáleny nulovou délkou. Dalším potřebnou informací je fakt, že velikost gridu není nijak svázána s velikostí fontu, což poskytuje uživateli možnost aplikovat malý grid na velký font či obráceně. Také je ovšem dobré mít na paměti, že velikost gridu není to samé, co velikost písma. V kombinaci s chytrým nastavením velikosti mřížky a fontu tak lze docílit zajímavých výsledků, zvláště v oblasti vyplňování větších souvislých ploch.

Nebezpečí, které číhá na neprozřetelného uživatele, tkví ve výrazným způsobem rostoucí časové náročnosti filtru se zmenšující se velikostí gridu. A navíc čím větší font je zvolen tím větší matice znaků je potřeba testovat na každém místě v mřížce. Pro běžné účely filtru je optimální nastavovat grid v toleranci do 15 % vzhledem k velikosti písma. Příliš hustá mřížka vede k dlouho trvajícímu výpočtu a ve většině případů bude hodně znaků vykresleno téměř přes sebe, zatímco při řídkém nastavení budou mezery mezi potenciálními sousedy příliš velké. Konečné nastavení však závisí výhradně na uživateli, jakým způsobem se rozhodne své myšlenky a nápady realizovat.

2.4 Element náhody při zpracování

Výsledek práce ve filtru Letterist není vždy předem jasně daný. Jedno z nastavení se týká použití náhody při zpracování. Je-li aktivována, částečně pozměňuje chování výpočtu. Stále platí stěžejní kritéria při určování pravděpodobnosti, tj nejnižší akceptovatelná pravděpodobnost, ale práh pro automatický výběr znaku svou úlohu ztrácí. Použití náhody svým způsobem vnáší trochu neurčitosti do míst, ve kterých se filtr primárně snaží o pravý opak. Veškerá možná nastavení Letteristu slouží k určení co nejlépe pasujícího kandidáta z množiny znaků pro následné vykreslení. Element náhody (Random) množinu adeptů lehce rozostřuje. Jeho princip je velmi jednoduchý.

Nastavení náhodnosti naprosto potlačí informační hodnotu definovanou pravděpodobností pro automatický výběr a místo snahy o hledání nejlepšího kandidáta, hledá množinu znaků, jež můžou mít sice rozdílné stupně shody s podkladem, ale všechny vyhovují kritériu definovanému uživatelem jako nejnižší akceptovatelná pravděpodobnost. Z takto vzniklého souboru znaků potom pomocí počítačem generované náhody vybere jeden znak jako vítěze a ten taky vykreslí.

Použití náhody se ukázalo jako velmi vhodné při vyplňování velkých souvislých ploch, u nichž bychom neradi viděli nějakou unifikovanou strukturu. Přestože přináší částečně

neočekávatelný výsledek, nepotlačuje výrazně charakter původního obrazu, pouze do jisté míry oživí zobrazení. Zda obsáhnout náhodu do obrazu nebo se jí radši vyhnout závisí velmi na struktuře obrazu a na tom, co autor touží vytvořit. V některých situacích se tato možnost přímo nabízí, v jiných jsou její výsledky přinejlepším rozpačité.

2.5 Font a množina znaků

V první kapitole byla naznačena důležitost typu fontu zvoleného pro stylizaci. V rámci jednoho projektu ve filtru Letterist však velmi často vystupuje více typů písma, případně se liší svým řezem nebo velikostí. Volba správného fontu je velmi důležitá, podstatná je ale také množina znaků (Charset), které budou použity při zpracování. Pracovní prostředí filtru nabízí ke každému fontu znaky od hodnoty 0 až po 255. Odpovědnost na konečné podobě výběru je však na uživateli. Optimální nastavení se liší případ od případu. Velmi častá a frekventovaná jsou písmena latinské abecedy, jak velká tak i malá. Ale svou hodnotu mají i čísla. Různé jiné znaky je občas vhodné použít, ale jejich symbolistická hodnota je přece jen poněkud posunutá a na potenciálního diváka může působit velmi různě. Při výběru množiny znaků je dobré se zamyslet nad vizuální hodnotou jednotlivých písmen a srovnat ji nejen s atmosférou a kontextem zdrojového obrazu, ale i s informací, kterou chceme výsledným dílem prezentovat. Například obraz složený jen z vhodně namapovaných nul a jedniček může někomu připadat docela "informatický" a třeba se najdou i tací, co budou pátrat po textu skrytém v binárním zápise. Jako nepříliš vhodné se ukázalo použití celé abecedy vybraného fontu. Ta totiž velmi často obsahuje širokou škálu znaků, takže se téměř vždy nějaký z nich vykreslí, ale přece jen informace sdělená drobnou horní tečkou se oproti trefně namapovanému velkému 'A' výrazně liší.

Kapitola 3

Editor Letterist

3.1 Popis software

Grafický editor Letterist je primárně určen pro abstraktní stylizaci obrazu v letteristickém stylu. Editor je navržen jako plug-in do již existujícího programu Elephant ve formě dynamické knihovny. V kontextu práce s Elephantelem je Letteris označen jako "filtr typu 3", což znamená, že má vlastní grafické rozhraní a používá nástroje ze svých palet a možností. Filtr by mohl fungovat i samostatně jako plnohodnotný program, implementace ve formě zásuvného modulu byla zvolena v rámci sdružování podobných prací, resp. jejich výsledků, dohromady, čímž poskytuje jednotlivým filtrům nejen šanci, že neupadnou v zapomnění, že si jich bude stále někdo nový všímat, ale také možnost jejich spojením vytvořit komplexní grafický editor se spoustou originálních možností a přístupů. Pro grafický filtr Letterist lze najít několik klíčových charakteristik:

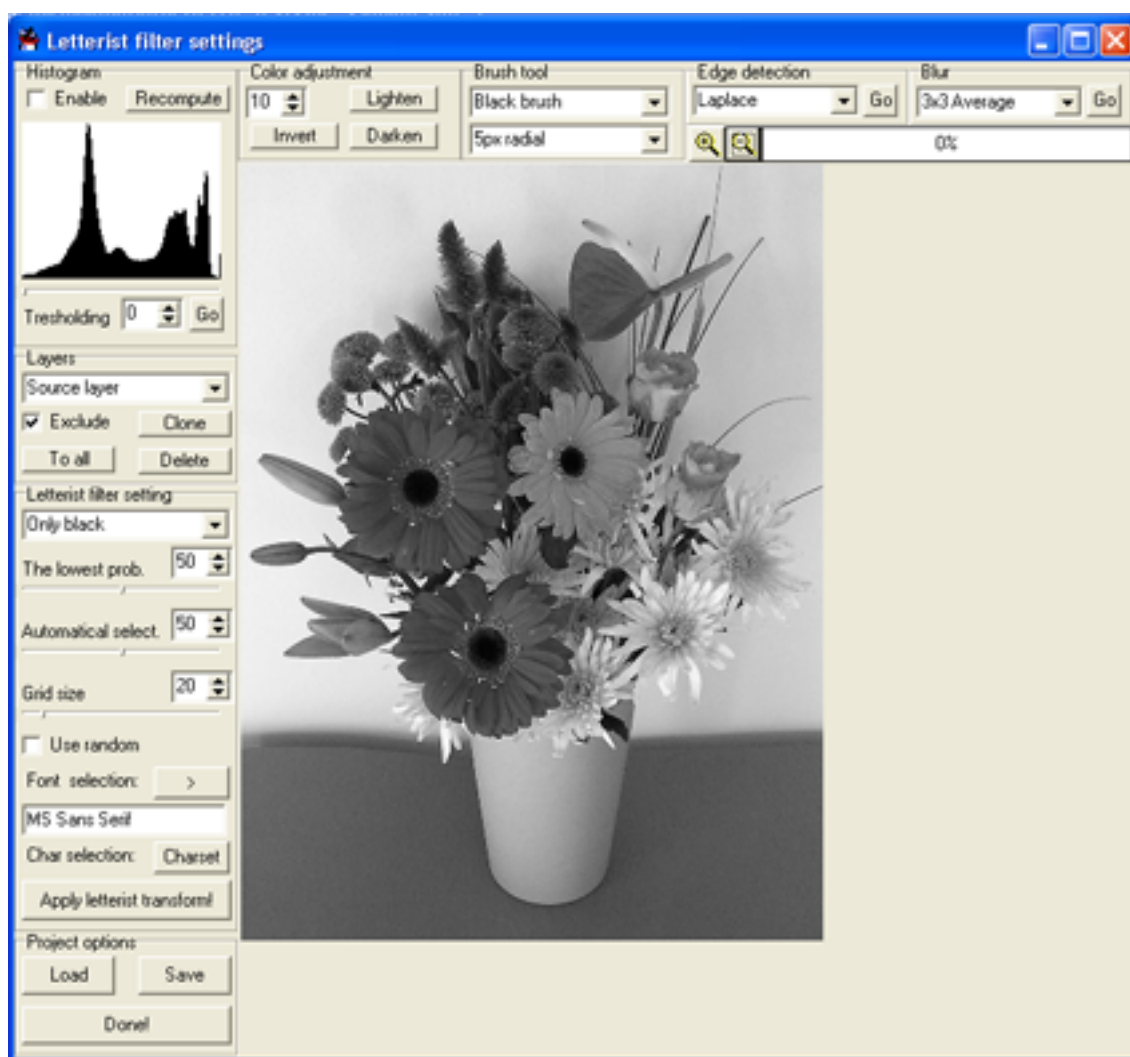
- možnost práce ve vrstvách, jednoduché přepínání a kontrola stavu
- rozsáhlé možnosti nastavení, široká škála možných výstupů z jednoho vstupního obrazu.
- široká paleta vlastních nástrojů pro editaci obrazu
- možnost uložit projekt se všemi vrstvami do souboru .lts a později znovu otevřít a pokračovat v rozpracovaném díle

3.2 Instalace softwaru

Zprovoznění filtru Letterist není nikterak obtížné. Abychom mohli filtr spustit, je nutné na počítač nahrát grafický editor Elephant, pod nímž Letterist běží jako jeho zásuvný modul. Editor Elephant je obsažen na CD přiloženém k bakalářské práci v adresáři Elephant, soubor Elephant.exe. V téže adresáři se také nacházejí další potřebné soubory společně s dynamickými knihovnami doposud vytvořených filtrů. Filtr Letterist je obsažen i v tomto umístění z důvodu snadné instalace. Stačí jen překopírovat adresář Elephant na počítač a spustit soubor Elephant.exe.

3.3 Grafické rozhraní

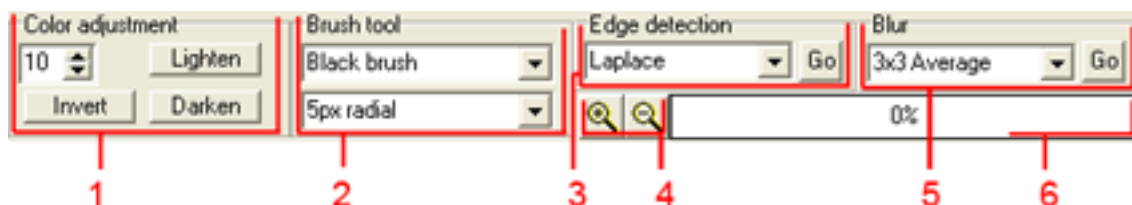
Letterist nabízí komplexní grafické rozhraní pohodlně ovladatelné myší a klávesnicí. Většina funkcí editoru je přímo dostupná z jeho pracovní plochy. Prostředí sestává ze dvou panelů nástrojů - horizontálního a vertikálního. Mezi panely nástrojů se nachází plocha pro zobrazování výsledků práce. Práce v editoru je směřována k používání vrstev, které lze jednoduše přepínat a průběžně sledovat změny při práci s filtrem



Obrázek 3.1: Náhled na pracovní prostředí filtru Letterist

3.3.1 Horizontální panel nástrojů

Horizontální panel nástrojů nabízí k použití celou řadu filtrů a transformací obrazu zahrnutých ve filtru Letterist. Prakticky všechny stylizační non-letteristické funkce jsou k dispozici právě zde. Jedinou výjimkou je prahování, to je součástí vertikálního panelu.



Obrázek 3.2: Ukázka horizontální pracovní lišty, rámcový popis: 1)Color adjustment options 2)Brush tool options 3)Edge detection options 4)Tlačítka Lupy 5)Blur options 6)Letterist progress bar

Panel je členěn na několik oddílů, převážně podle zaměření a typu činnosti funkcí, které se na něm nacházejí:

- **Color adjustment:** Slouží ke globálním změnám šedého stupně v rámci celého obrazu. Obsahuje tlačítko Invert, které slouží k barevné inverzi přes střed 256barevného intervalu. Další možností je aktivace filtrů Lighten a Darken stejnojmennými tlačítky. Tyto filtry následně provádějí celkovou změnu barvy podle hodnoty udané v políčku vlevo nahoře složky Color adjustment
- **Brush tool:** Zobrazuje informace o aktuálním nastaveném štětcí (vrchní combobox) a tvaru jeho stopy (dolní combobox). K dispozici je v Letteristu pouze omezená škála štětců, není to významný nástroj pro letteristickou stylizaci, ale do budoucna by jistě bylo vhodné nástroj štětec poněkud vylepšit.
- **Edge detection:** Je složka zaštiťující hledání hran v editoru. Roztažením roletky je možné zvolit některý z nabízených filtrů a tlačítkem GO! spustit samotný výpočet filtru.
- **Tlačítka lupy:** Tlačítka lupy slouží k přiblížení či oddálení pracovní plochy obrazu. Lupa je nastavena na konstantní poměr 2.00, což znamená dvojnásobné zvětšení při přiblížení a poloviční velikost při oddálení.
- **Blur:** Panel pro volbu rozmazávacích filtrů. V nabídce lze zvolit některý z daných filtrů a stejně jako u hledání hran jej spustit tlačítkem GO! Výhodou rozmazávacích filtrů je, že mohou být s úspěchem použity na odstranění šumu (kvůli čemuž ale do Letteristu implementovány nebyly).
- **Letterist Progress bar:** Stavový proužek, který ukazuje průběh a stav aktuálního výpočtu.

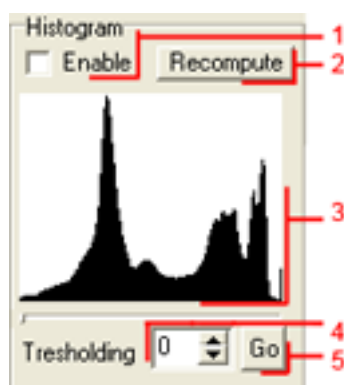
3.3. GRAFICKÉ ROZHRAŇÍ



Obrázek 3.3: Ukázka horizontální pracovní lišty, detail nástrojů: 1)Stupeň intenzity 2)Tlačítko zesvětlení 3)Tlačítko ztmavení 4)Tlačítko inverze 5)Velikosti štětce 6)Barva štětce 7)Přiblížení 8)Oddálení 9)Typ hranového filtru 10)Hledání hran 11)Typ rozmazávacího filtru 12)Rozmazání

3.3.2 Vertikální panel nástrojů

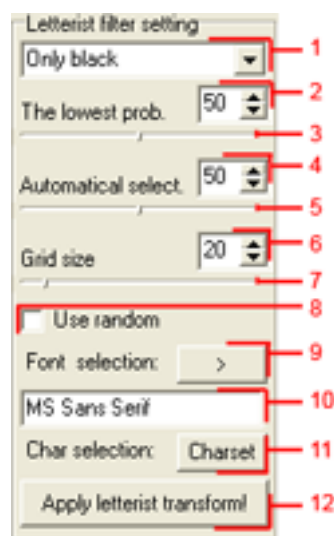
Na levé straně pracovního okna editoru Lettrist se nachází vertikální panel nástrojů. Obsahuje rozhraní pro práci s vrstvami, náhled histogramu, nastavení pro prahování nebo tlačítka pro ukládání a nahrávání dříve uložených rozpracovaných projektů.



Obrázek 3.4: Ukázka části vertikálního panelu nástrojů - histogram a obsluha prahování: 1)Zapnutí/vypnutí zobrazování histogramu 2)Přepočítání histogramu 3)Vizuální podoba histogramu 4)Hodnota prahu 5)Prahování

Horní část horizontálního panelu obsahuje informace o aktuálním obraze - histogram. Histogram je mnohdy užitečnou pomůckou, na druhou stranu často není vůbec potřeba a právě proto je zde možnost povolit/zakázat zobrazování (přesněji real-timové přepočítávání a zobrazení) histogramu. Tato volba se provádí zaškrtnutím políčka checkboxu Enable. Jelikož při vypnutém zobrazování program průběžně neaktualizuje hodnotu histogramu, je na liště zahrnuto také tlačítko recompute, které slouží k jednorázovému vypočítání a vykreslení histogramu, pokud si to uživatel přeje. Nedílnou součástí panelu s histogramem je také možnost prahování. Nastavit požadovaný práh lze buď pomocí číselné hodnoty nebo posuvnou lištou. Prahování se spouští tlačítkem GO!

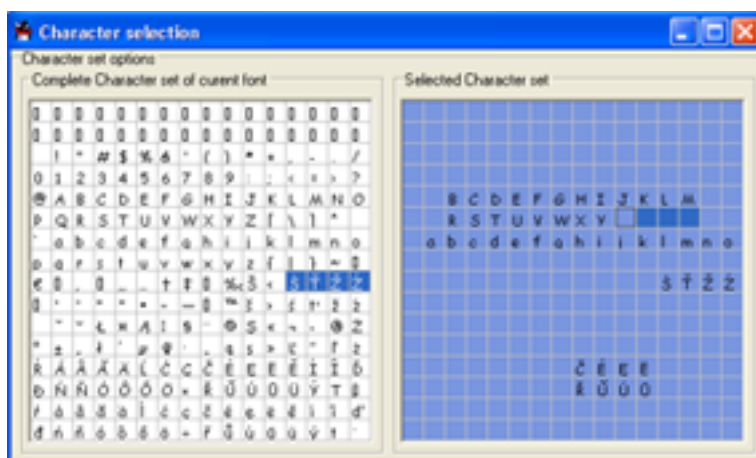
Nastavení parametru samotného letteristického filtru je obsaženo uprostřed vertikál-



Obrázek 3.5: Ukázka části vertikálního panelu nástrojů - nastavení pro letteristickou transformaci: 1)Typ filtru 2-3)Nejnižší akceptovatelná pravděpodobnost 4-5)Pravděpodobnost pro automatický výběr 6-7)Velikost mřížky - gridu 8)Potvrzení používání náhody 9)Výběr fontu 10)Zvolený typ fontu 11)Volba Charsetu 12)Aplikace filtru

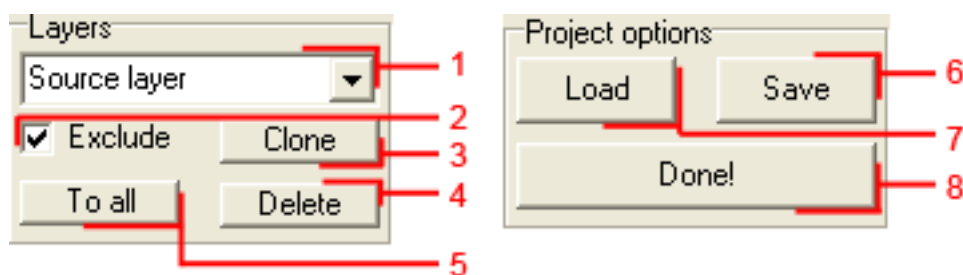
ního panelu. Nabízí širokou škálu možností a voleb. Jednou z nejdůležitějších vlastností stylizace je zvolený typ filtrování. Tato hodnota se nastavuje úplně nahoře v sekci Letterist filter setting. Dolů následují atributy The lowest probability (nejnižší akceptovatelná pravděpodobnost), Automatical selection (hranice pravděpodobnosti pro automatický výběr) a Grid size (velikost pracovní mřížky). Tato tři nastavení jsou editovatelná jak číselnou hodnotou tak prostřednictvím posuvné lišty. Checkbox Use random obstarává nastavení použití náhody při výpočtu filtru. Následující tlačítko Font selection spouští klasický Font-Dialog, v němž může uživatel zvolit požadovaný font, jeho velikost či řez. Tlačítkem Char selection se spustí okno zobrazující dvě čtvercové mřížky o hraně 16 polí. Levá mřížka ukazuje kompletní množinu znaků zvoleného fontu, pravá zobrazuje uživatelem definovanou abecedu, která bude použita při provádění filtru. Klikáním či obdélníkovým výběrem znaků ve čtverci lze vybírat znaky, případně také odebírat ty dříve zvolené.

Složka s možnostmi pro práci s vrstvami je často využívaným místem panelu. Rychlé a jednoduché prohlížení vytvořených vrstev usnadňuje práci a nechává více prostoru soustředění se na samotnou tvorbu. Ke změně aktuální vrstvy slouží roletka nahoře složky. Dále obsahuje panel také checkbox Exclude, jehož změnou lze nastavit, má-li být aktuální vrstva zahrnuta do finální kompozice či nikoliv, a sadu tří tlačítek. Tlačítko Clone slouží k rozmnožení aktuální vrstvy, Delete maže aktuální vrstvu, pokud není zdrojová nebo výsledná (ty jsou v každém projektu Letteristu zahrnuty napevno) a tlačítko To all aplikuje momentální nastavení vlastnosti Exclude na všechny ostatní vrstvy projektu. Vlastnost Exclude je nastavitelná nejen u pracovních vrstev, ale i u zdrojové a výsledné vrstvy, u posledních dvou



Obrázek 3.6: Ukázka výběru abecedy pro letteristickou stylizaci

zmíněných však nemá žádný efekt, kompozice je prováděna pouze nad pracovními vrstvami. Vhodná tvorba jednotlivých vrstev a správné nastavení této vlastnosti mají velký vliv na podobu výsledného obrazu. Jediná vrstva zahrnutá do kompozice navíc může radikálně změnit podobu i charakter vytvářeného díla.



Obrázek 3.7: Ukázka části vertikálního panelu nástrojů - lišta pro správu vrstev a lišta pro obsluhu celého projektu: 1)Aktuální vrstva 2)Ukazatel zahrnutí vrstvy do výsledné kompozice 3)Smazání vrstvy 4)Klonování vrstvy 6)Uložení projektu 7)Načtení projektu 8)Aplikace nastavení

Kapitola 4

Implementace

Grafický filtr je napsán pro prostředí Win32 v programovacím jazyce ObjectPascal. Jako vývojové prostředí bylo použito Borland Delphi7 Enterprise. Program je implementován jako zásuvný modul pro program Elephant ve formě dynamické knihovny.

Vlastní program se skládá z projektového souboru Letterist (letteris.dpr), a z jednotky MAINUnit (Unit MAINUnit), která obsahuje kompletní správu grafického rozhraní hlavního formuláře a řadu funkcí a procedur pro samotnou práci filtru, a jednotky Charset (Unit CHARSET), jež obsluhuje formulář odpovídající za zobrazování Charsetů.

4.1 Knihovna Letterist

Knihovna Letterist obsahuje několik funkcí, které jsou důležité pro komunikaci mezi programem Letterist a samotným grafickým editorem Elephant:

- **GetFilterName:** vrací jméno filtru
- **GetCreator:** předává informace o autorovi nebo skupině autorů, může obsahovat i doplňující informace
- **ExecFilter:** provádí samotné filtrování, vrací výsledný obraz po aplikaci filtru
- **CreateSettings:** zajišťuje vytvoření nastavení, vytváří paměťový prostor
- **DestroySettings:** ruší nastavení uchovávané v paměti, uvolňuje paměť
- **ShowSettingDialogPreview:** zobrazuje okno filtru s nastavením
- **NewImageLoaded:** rozpoznává, jestli byl otevřen nový obrázek
- **Minimize:** obsluhuje minimalizaci
- **Restore:** uvádí do stavu před minimalizací
- **GetFilterType:** vrací typ filtru

4.2 Jednotka MAINUnit

Jednotka UnitMAIN odpovídá za správu vlastního formuláře a správné zobrazování, ale také za aplikaci filtrů, letteristických transformací a jiných nástrojů použitých v programu. Jednotka obsahuje celou řadu funkcí a procedur, jež lze rozdělit do několika kategorií:

- **LAYERS:** zajišťují správu vrstev
- **PANELS:** řídí zobrazování na formuláři.
- **LETTERIST:** odpovídají za aplikace letteristických filtrů, také vytvářejí matice písmen
- **FILTER:** obstarávají aplikace non-letteristických filtrů obsažených v programu
- **HANDLE:** funkce spouštěné při událostech
- **ADD:** doplňkové funkce

Nejdůležitější funkce z výše uvedených kategorií jsou detailněji rozebrány. Ty, jež jsou uvedeny v kategorii HANDLE a ADD, nejsou pro samotnou podstatu programu nijak zásadní. Prvně zmíněné provádějí běžné řešení událostí, funkce ADD obsahují strojové zpracování dat (ADD_CreateArray_formBitmap, ADD_ImageToGrey) a funkci ADD_LayerSelection, která obsluhuje MessageBox s dotazem, zda transformaci aplikovat do aktuální nebo nové vrstvy.

4.2.1 Funkce kategorie LAYERS

Tyto funkce jsou odpovědné za práci s vrstvami uvnitř programu. Funkce nespravují správné zobrazování (k tomu jsou určeny funkce kategorie PANELS), ale pokud je potřeba, příslušné funkce na úpravu zobrazení volají. Mezi funkce LAYERS patří:

- **LAYER_NEW:** kompletně spravuje vytvoření nové vrstvy, nastavuje její parametry na požadované hodnoty
- **LAYER_DELETE:** odpovídá za vymazání vrstvy z paměti
- **LAYER_CLONE:** řídí zobrazování na formuláři
- **LAYER_TO_ALL_SET:** nastavuje všem vrstvám stejné zobrazovací atributy, jaké má aktuální vrstva. V současné době zahrnuje pouze atribut exclude, ale do budoucna je možné počítat s rozšířením množiny atributů. Pak by tato funkce obsluhovala i je.

4.2.2 Funkce kategorie PANELS

Funkce PANELS mají na starosti obsluhu grafického rozhraní a částečně i odpovídají za některé výpočty potřebné ke zobrazování stavu vrstev. Obsluhují také ukládání a načítání projektu a výpočty histogramu. Pro jednoduchost lze tyto funkce shrnout do několika skupin

- **správa rozhraní:** zajišťuje správné zobrazování v rámci grafického rozhraní. Obsahuje funkce `ChangeActualLayer`, která přenastavuje rozhraní podle atributů dané vrstvy, dále obsahuje dvě funkce `ComputeResultImage`, která je odpovědná za výslednou kompozici vrstev do výsledného obrázku a `ComputeWorkingImage`, jež počítá aktuální obraz. Mezi funkcemi PANELS jsou ještě další pomocné funkce na modifikace a zjišťování různých nastavení vrstev.
- **histogram:** obsahuje funkce `HistogramCompute_formBitmap`, která obsluhuje samotný výpočet histogramu a `HistogramDraw`, jež odpovídá za vykreslení histogramu na obrazovku.
- **načtení/uložení projektu:** odpovídá za načítání a ukládání projektu. Filtr Letterist má svůj vlastní projektový soubor s příponou `.lts`

Struktura .lts souboru: Typ `.lts` (`LeTterist`) je vlastním typem souboru určeného pro uchovávání informace o rozpracovaném projektu. Soubor neuchovává nastavení pracovních nástrojů, v prostředí filtru totiž není tak důležité jej držet v paměti, obsahuje pouze základní informace o vrstvách tak, aby je bylo možno znovu otevřít ve stavu, v jakém byly před uložením. Soubor ukládá informace implicitně po bytech a má následující strukturu:

- **Hlavička souboru [1B]:** Obsahuje číselný údaj o celkovém počtu vrstev v souboru v rozmezí 0-255.
- **Délka jména vrstvy ve znacích [1B]:** Udrží informaci o délce řetězce obsahujícím jméno aktuální vrstvy v rozsahu 0-255.
- **Jméno vrstvy [0-? B]:** Zde je zapsáno jméno vrstvy.
- **Typ vrstvy [1B]:** V současné době rozpoznává Letterist tři typy vrstev (`original`, `result`, `worknig`). Pro případé budoucí rozšíření je rozsah hodnot omezen jen velikostí 1B.
- **Vlastnosti vrstvy [1B]:** Podobně jako u typu vrstvy i zde se spíše počítá s budoucím rozšířením. Aktuálně rozpoznává Letterist pouze vlastnost `exclude`, udávající, bude-li daná vrstva zahrnuta do výsledné kompozice či nikoliv.
- **Šířka bitmapy vrstvy [4B]:** Určení šířky bitmapy vrstvy na hodnotě 4B.
- **Výška bitmapy vrstvy [4B]:** Určení výšky bitmapy vrstvy na hodnotě 4B.
- **Hodnoty intenzit jednotlivých pixelů [B]:** Pole délky šířka.výška uchovává stupeň šedi pixelů dané vrstvy směrem zleva doprava a shora dolů

4.2.3 Funkce kategorie FILTER

Funkce obsažené v této kategorii obsluhují ve filtru Letteris všechny aplikace filtrů vyjma těch letteristických a také některých nástrojů. Použity jsou základní jednoduché filtry [4][2], implementace složitějších filtrů není předmětem této bakalářské práce, proto jsou zahrnuty jen základní algoritmy pro nejzákladnější úpravu. Použité filtry jsou: Laplaceův a Sobelův filtr pro hledání hran, rozmazávací filtry průměr a Gaussův filtr. Další použité nástroje zahrnují zesvětlování a ztmavování, barevnou inverzi a prahování. Do sekce funkcí FILTER je zařazena i funkce starající se o kreslení černým nebo bílým štětcem. Funkce jsou většinou volány z příslušné HANDLE funkce.

- **FILTER_DRAW:** Obstarává ruční kreslení štětcem na vrstvu.
- **FILTER_INVERT:** Zajišťuje barevnou inverzi. Všechny stupně šedi jsou v rozsahu 0 až 255 "překlopeny" a výstupní obraz je výsledkem inverze.
- **FILTER_LIGHTEN/FILTER_DARKEN:** Tyto filtry obsluhují globální zesvětlení/ztmavení. Všechny pixely jsou upraveny o uživatelem zadanou hodnotu a "oříznuty" do intervalu od 0 do 255, pokud přesahují. Pro úplnost je dobré zmínit, že zde v podstatě nejde o dva různé filtry. DARKEN volá funkci LIGHTEN, jen jí hodnotu, o kterou má zesvětlovat, vrátí zápornou.
- **FILTER_TRESHOLDING:** Tresholding, jak už název napovídá, má na starosti prahování obrazu.
- **FILTER_LINEAR:** Tato funkce zaštiťuje aplikaci lineárních filtrů. Na vstupu dostane matici filtru a koeficient, kterým má být matice vynásobena a provádí samotné filtrování. Jedinou podmínkou je, že matice filtru musí být čtvercová. Právě přes tuto funkci se provádějí ony zmiňované filtry jako Laplace, Sobel či Gauss.

4.2.4 Funkce kategorie LETTERIST

Funkce obsažené kategorií letterist jsou samotným jádrem stylizace v letteristickém stylu. Jsou zde obsaženy jak některé inicializační funkce tak i samotné algoritmy zajišťující mapování znaků fontu na výsledný obraz. Tyto algoritmy vycházejí primárně z parametrů nastavených uživatelem a jejich výpočetní náročnost proto velmi kolísá.

- **LETTERIST_CREATE_MATRIX:** Tato funkce se volá vždy po zvolení nového fontu. Náplní její práce je tvorba matice pro každý znak písma, která obsahuje binární informaci o tvaru písmene. Kromě samotné tvorby matice také inicializuje a naplní pole, v němž je pro každé písmeno zvoleného fontu uložena touto funkcí spočítaná hodnota udávající počet černých pixelů na matici (dalo by se říci na jakési pseudo-kuželce) písma.

- **LETTERIST_DRAW_CHAR:** Z názvu funkce je patrné, že jejím hlavním úkolem je kreslit znak. V této kategorii je zařazena proto, neboť je přímo spojena s prováděním letteristických transformací a přistupuje při vykreslování k datům určeným právě pro letteristické zpracování.
- **LETTERIST_BPO_FILTER:** Jeden ze dvou hlavních algoritmů programu, filtr Black Pixels only (BPO). Struktura je vysvětlena níže.
- **LETTERIST_BnW_FILTER:** Black and White ratio, druhá velmi důležitá funkce, která je stejně jako BPO detailněji rozebrána níže.

4.2.5 Data v programu

Program obsahuje spoustu datových struktur, které ať už udržují informace o vrstvách, nebo jsou využívány při letteristických transformacích, jsou velmi důležité pro správný chod editoru. Zde jsou zmíněny ty nejdůležitější datové struktury Letterist fitru:

- **TLayer:** Je struktura uchovávající informace o jedné vrstvě. Pro každou nově vytvořenou vrstvu program alokuje paměť a vytváří další instanci typu TLayer. Jednotlivé atributy struktury:
 - **Layer_Type:** Udává typ vrstvy. V současné verzi programu jsou k dispozici 3 typy vrstvy: zdrojová (source), výsledná (result) a pracovní (working). Zdrojová vrstva je ta, která zpočátku obsahuje vstupní obraz. Zpravidla se aplikací filtrů a nástrojů nad touto vrstvou vytvoří pracovní vrstvy. Skládáním pracovních vrstev, které nemají vlastnost exclude zapnutou pak vzniká výsledná vrstva. Ta je předávána jako výsledek editoru Elephant.
 - **Layer_Name:** Obsahuje jméno vrstvy.
 - **Layer_BitmapImage:** Zde je uložena informace o obrazu vrstvy. Bitmapa vrstvy je typu TBitmap.
 - **Layer_BinaryImage:** Obsahuje binární pole reprezentující černé pixely bitmapy. Pokud je na bitmapě vrstvy černý pixel, na korespondujících souřadnicích binárního obrazu vrstvy se hodnota změní na True. V opačném případě jsou hodnoty pole False. Layer_BinaryImage je používán Letteristickými filtry pro zrychlení výpočtu.
 - **Layer_Histogram:** Udrží pole, v němž prvek [k] označuje počet k-šedých pixelů v obraze vrstvy.
 - **Layer_Exclude:** Hodnota typu Bool, která určuje, bude-li vrstva zahrnuta do finální kompozice či nikoliv. Nastavení proměnné na True vrstvu z konečného skládání vyloučí.

- **Layers:** Je dynamické pole instancí struktury TLayer. Rezervovanými indexy pole jsou 0 pro zdrojovou vrstvu a 1 pro výsledný obraz.
- **LETTERS_MATRIX:** Uchovává matici pro všechny znaky zvoleného fontu. Pole Letters_Matrix obsahuje 256 prvků z nich každý definuje obdélníkovou matici jednoho znaku písma.
- **LETTERS_BnW_RATIO:** Pole s informacemi o šedém indexu jednotlivých znaků.
- **LETTER_CHOOSE:** Určuje znaky, které mají být při výpočtu použity jako abeceda (Charset) letteristických filtrů editoru. Implicitně nastavená jako prázdná. Uživatel může vybrat libovolné znaky aktuálního písma a přidat je do Charsetu nebo je naopak odebrat.

4.3 Jednotka CHARSET

Tato jednotka obstarává v programu pouze jedinou úlohu. Tou je zobrazování a obsluha okna, jež zobrazuje dvě palety znaků (kompletní a vybranou). Jednotka přistupuje k proměnné LETTER_CHOOSE, které automaticky zapisuje změny v nastavení Charsetu.

4.4 Algoritmy letteristického zpracování

Podstatnou částí projektu jsou dva algoritmy zajišťující samotnou letteristickou stylizaci. Volba jednoho z nich při zpracování obrazu je velmi důležitá, je proto vhodné mít alespoň rámcovou představu o tom, jak tyto filtry pracují. Zatím jsou v Letteristu obsaženy dva typy transformace - Black Pixels only (BPO), Black and White ratio (BnW). Obě tyto metody jsou si v mnohém podobné, v určitých oblastech se však zásadně liší. Pozdější implementace dalších způsobů je možná a do budoucna zvažovaná.

4.4.1 Black Pixels only filtr

Filtr Black Pixels only je základním stavebním kamenem editoru Letterist. Přestože je často využíván i druhý zastoupený způsob transformace, Black and White ratio, práce s ním vede častěji k plnohodnotným výsledkům a na pohled příjemným stylizacím. Filtr Black Pixels only probíhá v několika do sebe zapouzdřených cyklech:

- **Průchod obrazem:** Filtr prochází aktuální vrstvu počínaje levým horním rohem směrem zleva doprava a shora dolů. Po každém průchodu se posunou souřadnice nového počátku o hodnotu určenou velikostí pracovní mřížky. Testuje se ale také, zda se při posunutí vejde celá matice znaku do obrazu. Pokud tomu tak není, cyklus končí.
 - **Průchod maticí znaku:** V tomto cyklu filtr testuje, na kterých souřadnicích binárního obrazu aktuální vrstvy je uložena hodnota True (černý pixel). Porovnává

4.4. ALGORITMY LETTERISTICKÉHO ZPRACOVÁNÍ

souřadnice s maticemi znaků a inkrementuje míru shody daného znaku s podložkou (přesněji řečeno udává přímo počet černých pixelů, v kterých se shoduje matice písmene s obrazem).

- **Výběr bez náhody:** Není-li zapnuta volba náhody, pochází filtr tímto cyklem pro hledání znaku. Pro každý znak počítá dosaženou pravděpodobnost (míru shody) jako podíl v předchozím cyklu spočítaného počtu pasujících černých pixelů s celkovým počtem černých pixelů znaku. Narazí-li na znak, který vyhovuje nejnižší akceptovatelné hranici, určí pro něj hodnotu `Actual_probability`, která udává konkrétní sumu pixelů, v nichž se daný znak shoduje s obrazem. Tato hodnota určuje kandidáta na výpis. Pokud je u nějakého znaku vyšší než byla doposud určena, kandidátem se stává procházený znak a počet jeho shodujících se černých pixelů je uložen do `Actual_probability`. Je-li míra shody větší nebo rovna pravděpodobnosti pro automatický výběr, cyklus končí a jeho výsledkem je právě zvolené písmeno. Na konci cyklu je buď vybrán znak pro výpis nebo je nastaveno, že žádný znak zvolený nebyl.
 - **Náhodný výběr:** Náhodný výběr funguje podobně jako předchozí cyklus s dvěma podstatnými odlišnostmi. Vůbec znak netestuje na pravděpodobnost pro automatický výběr, jediným určujícím faktorem je nejnižší akceptovatelná pravděpodobnost. Každé písmeno, které požadavek na ni splní, je zahrnuto jako možný kandidát. Na konci cyklu je buď řetězec kandidátů, nebo nastavení, že žádný kandidát nebyl nalezen. Oproti cyklu bez náhody se zde určuje konečný kandidát po skončení průchodů prostým náhodným výběrem nad množinou adeptů.
 - **Vykreslení znaku:** Byl-li v předchozích krocích zvolen znak určený k vykreslení, je tak učiněno na odpovídajících souřadnicích pomocné bitmapy filtru.
- **Ukončení filtru:** V této fázi dochází k samotnému překreslení výsledku filtru do zvolené vrstvy. Filtr jako takový zde končí.

4.4.2 Black Pixels only filtr

Black and White ratio je druhým možným typem letteristické stylizace použitý v editoru Letterist. Podstatou algoritmu je hledání míry shody prostřednictvím poměru černých a bílých pixelů jak na matici znaku tak na zdrojovém obrazu. Black and White ratio prochází podobným cyklem jako předchozí uvedený typ filtrování, avšak s drobnými odlišnostmi.

- **Průchod obrazem:** Tento průchod je totožný s odpovídajícím cyklem filtru Black Pixels only.
- **Průchod obrazu tvaru matice:** Zde filtr na rozdíl od předchozího nepočítá shodující se černé pixely vstupního obrazu a písmene. Zajímá jej pouze suma černých pixelů zdrojového obrazu. Hodnoty šedého stupně písma jsou předpočítány již v době výběru fontu.

4.4. ALGORITMY LETTERISTICKÉHO ZPRACOVÁNÍ

- **Výběr bez náhody:** Zde filtr počítá hranice pravděpodobnosti potřebné pro 100% shodu, což je pro algoritmus stav, kdy je počet černých pixelů obrazu a matice znaku totožný. Jakákoliv odchylka směrem nahoru či dolů míru shody snižuje. Proto jsou počítány hodnoty P_{lower} , P_{upper} , SEL_{lower} a SEL_{upper} . Jde o intervaly plnící hodnotu nejnižší akceptovatelné pravděpodobnosti (P) a hranice pro automatický výběr (SEL). V tomto typu filtru pak testování znaku probíhá dotazem, zda počet černých pixelů znaku leží v daném intervalu. Ostatní zpracování je totožné s filtrem Black Pixels only.
 - **Náhodný výběr:** Náhodný výběr filtru Black and White Ratio pracuje na shodném principu jako výše zmíněný cyklus u filtru Black Pixels only. Pouze kandidáty vybírá podle toho, zda pasují do spočítaného intervalu (P).
 - **Vykreslení znaku:** Shodné s filtrem BPO
- **Ukončení filtru:** Taktéž shodné s výše zmiňovaným typem filtru.

Kapitola 5

Závěr

Grafický filtr Letterist byl vytvořen pro abstraktní stylizaci rastrového obrazu převážně prostřednictvím písma. Role písma ve společnosti i jejím historickém vývoji představuje významný faktor, díky němuž člověk vnímá písmo jako něco zajímavého a svým způsobem krásného. Pokusy o doplňování obrazu pomocí znaků písma se objevují stále, filtr Letterist se snaží přijít s novým přístupem k abstraktnímu letteristickému zpracování a přinést nové možnosti jak pro výtvarníky tak i pro amatérské uživatele.

Editor Letterist nabízí vlastní intuitivní grafické rozhraní, díky němuž je práce s programem snadná a pohodlná. Možnost práce ve vrstvách dává uživateli velký prostor pro variabilitu práce a výsledku. Klíčovým algoritmem programu je samotný letteristický filtr. Pro pochopení systému práce letteristických filtrů je důležité porozumět jejich přístupu k obrazu a znakům. Program obsahuje i řadu dalších nástrojů a možností, které jsou určeny pro předzpracování obrazu před samotnou letteristickou stylizací, mohou však i samy vstupovat do procesu abstrakce a dodávat výslednému obrazu kvalitu.

Budoucí rozšíření filtru je možné. Je několik oblastí, z nichž by Letterist velmi těžil, pokud by byly implementovány. Velký prostor by uživateli jistě otevřely nové typy filtrovací funkce a širší nabídka existujících filtrů například pro hledání hran. Samostatná paleta znaku jako razítek by mohla přinést ještě větší míru zapojení autora do procesu stylizace a výrazně tak oživit obraz. Jistě by se našla i velká spousta dalších návrhů, co by bylo vhodné dále implementovat. Hranice nejsou určeny, ale v rámci jednoho zásuvného modulu je přece jen určitý, alespoň pomyslný strop. Při opravdu bohatém rozpracování by si Letterist zasloužil spíše vlastní program, než se krčit ve formě plug-inu.

Literatura

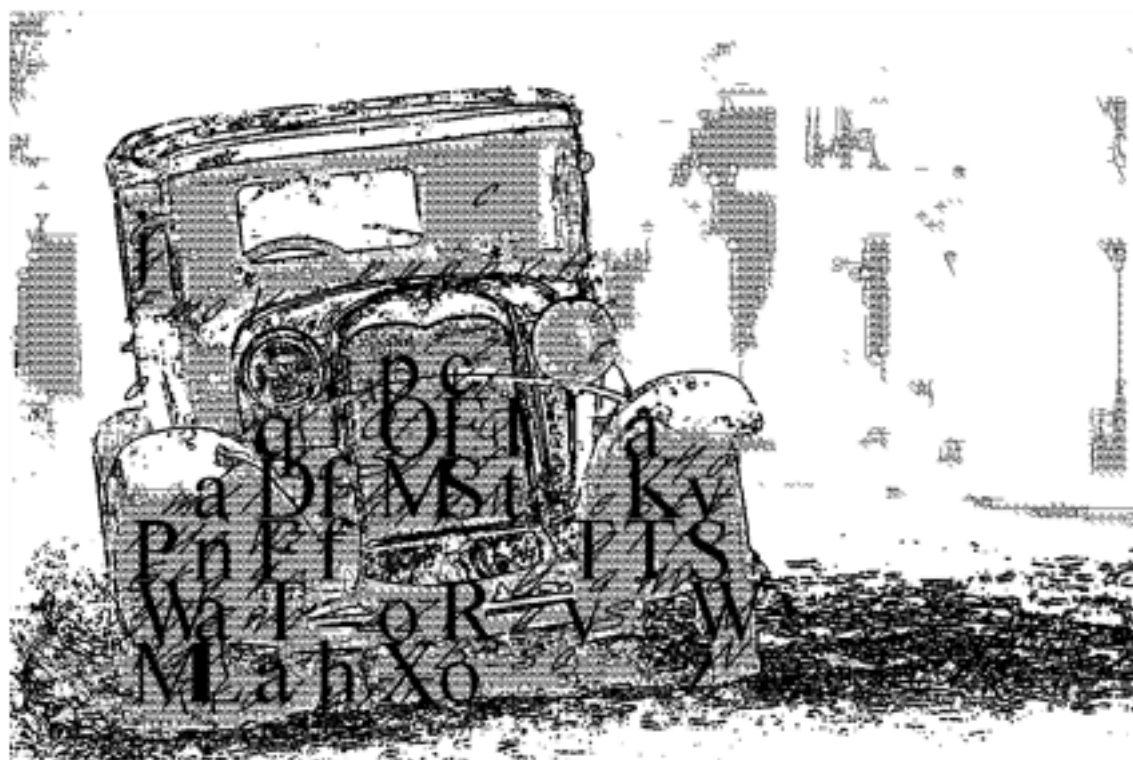
- [1] Old Car, <http://www.mrx.no/albums/colonia/Old_Car_2.jpg> . A.1, A.3
- [2] Kozubek, M.: *Digitální zpracování obrazu*, <<http://lom.fi.muni.cz/pages/education/DigImgProc.shtml>> . 4.2.3
- [3] Serba, I.: *Výtvarná informatika*, <<http://fosforos.fi.muni.cz/pv097/>> . 1.1
- [4] Sochor, J.: *Základy počítačové grafiky*, <<http://www.fi.muni.cz/~sochor/PB009/>> . 4.2.3
- [5] Soukal, J.: *ASCIInt*, <<http://www.fi.muni.cz/~xsoukal/bc/ASCIInt.zip>> . 2.2
- [6] Burned tree near the Lighthouse, <<http://www.3aw.com.au/morning/Wilson%20Prom/Burned%20tree%20near%20the%20Lighthouse.JPG>> . A.7, A.9
- [7] Švalbach, V.: *Základy vizuální komunikace*, <<https://is.muni.cz/auth/predmety/predmet.pl?fakulta=1433;obdobi=2825;kod=PV123>> . 1.3
- [8] Akt, <<http://www.latinartmuseum.com/images/Nudes2%20344.jpg>> . A.11

Příloha A

Obrazová příloha



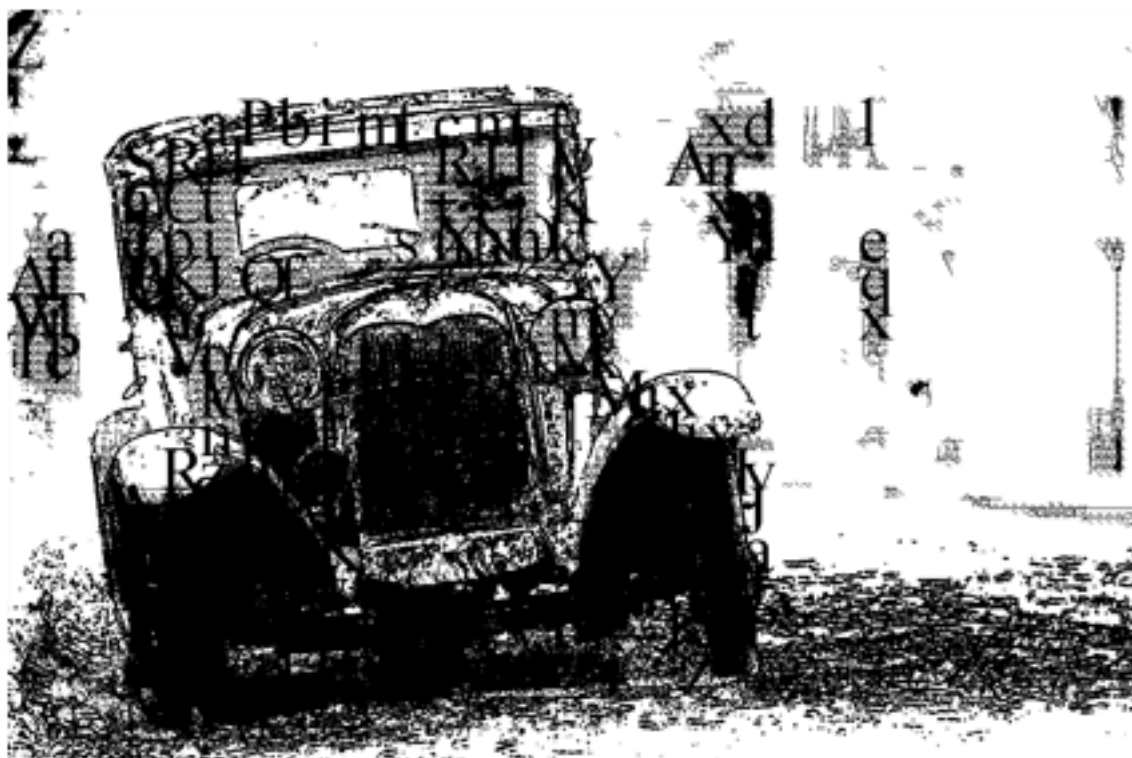
Obrázek A.1: Veterán zaparkovaný v uličce[1]



Obrázek A.2: Sylizace s vysokou mírou abstrakce připomínající skicu.



Obrázek A.3: Veterán zaparkovaný v uličce[1]



Obrázek A.4: Letteristicky stylizovaná scéna. Snaha o prohloubení obrazu výraznými černými plochami.



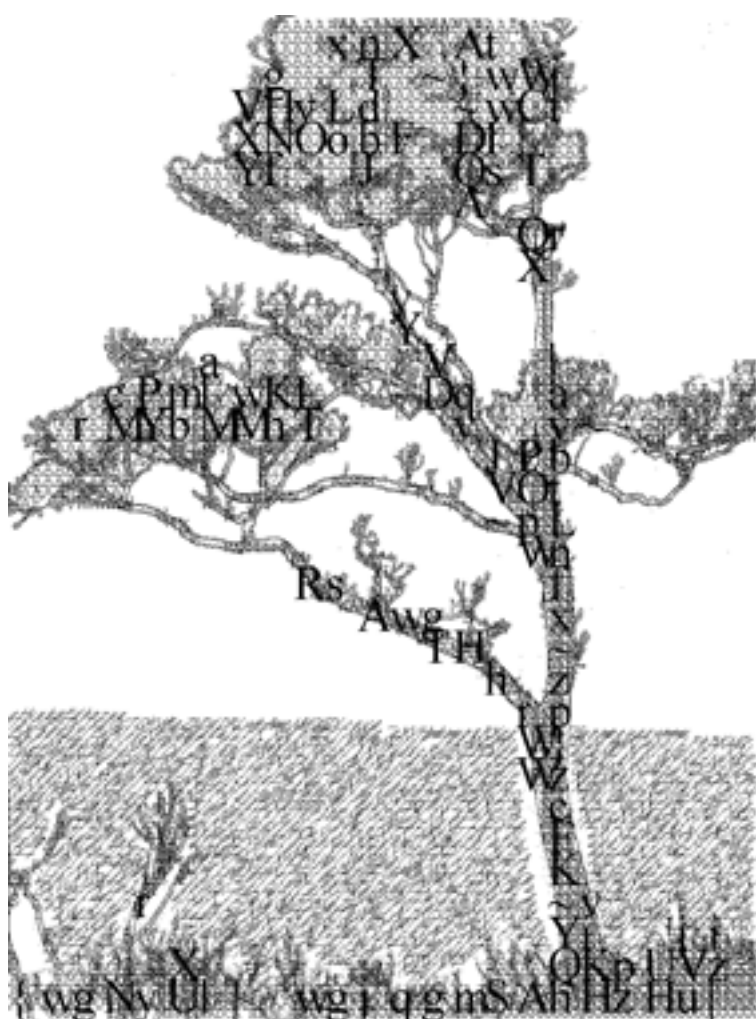
Obrázek A.5: Zamilování



Obrázek A.6: Standardní stylizace v letteristickém stylu. Výrazně jsou vytaženy záhyby rukávu na svetru, na hrudi je potlačena původní textura.



Obrázek A.7: Strom sklánějící se nad hladinu moře[6].



Obrázek A.8: Abstraktní pojetí sylizace stromu prostřednictvím textur a výrazných velkých písmen.



Obrázek A.9: Strom sklánějící se nad hladinu moře[6].



Obrázek A.10: Jednoduchá stylizace s využitím rozdílných způsobů tvorby textur (listoví a porost na břehu v kontrastu s hladinou moře).

