VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA INFORMAČNÍCH TECHNOLOGIÍ
FACULTY OF INFORMATION TECHNOLOGY

ÚSTAV POČÍTAČOVÉ GRAFIKY A MULTIMÉDIÍ
DEPARTMENT OF COMPUTER GRAPHICS AND MULTIMEDIA

PŘESNÉ ÚPRAVY OBRAZU - POSOUVÁNÍ A DEFORMACE

PRECISE IMAGE MODIFICATIONS - SHIFTING AND DEFORMATION

BAKALÁŘSKÁ PRÁCE
BACHELOR'S THESIS

AUTOR PRÁCE **JIŘÍ HEJTMÁNEK**
AUTHOR

VEDOUCÍ PRÁCE **Doc. Dr. Ing. PAVEL ZEMČÍK**
SUPERVISOR

BRNO 2019

Vysoké učení technické v Brně
Fakulta informačních technologií

Ústav počítačové grafiky a multimédií (UPGM) Akademický rok 2018/2019

Zadání bakalářské práce

Student: **Hejtmánek Jiří**
Program: Informační technologie
Název: **Přesné úpravy obrazu - posouvání a deformace**
Precise Image Modifications - Shifting and Deformation
Kategorie: Zpracování obrazu

Zadání:

1. Prostudujte literaturu a existující software na téma vzorkování signálů a geometrické transformace obrazu.
2. Vytipujte vhodnou sadu funkcí realizující základní funkce posouvání (převzorkování) a deformace, například pro korekce zkreslení objektivu.
3. Implementujte sadu funkcí podle předchozího bodu zadání.
4. Demonstrujte funkčnost na vhodném příkladě.
5. vyhodnoťte dosažené výsledky a možnosti dalšího pokračování práce.

Literatura:

- Dle pokynů vedoucího

Podrobné závazné pokyny pro vypracování práce viz http://www.fit.vutbr.cz/info/szz/

Vedoucí práce: **Zemčík Pavel, prof. Dr. Ing.**
Vedoucí ústavu: Černocký Jan, doc. Dr. Ing.
Datum zadání: 1. listopadu 2018
Datum odevzdání: 15. května 2019
Datum schválení: 9. května 2019

Abstrakt

Tato bakalářská práce se zabývá realizací systému pro provádění deformací nad digitálními obrázky. Hotový systém tento požadavek realizuje za pomoci matic a interpolací. Ve své první teoretické části popisuje matematické základy těchto deformací, ze kterých dále čerpá později. Je rovněž vypsáno několik již existujících programů provádějících, mimo jiné, právě tyto deformace. Poté je vysvětlena implementace systému a závěrem je práce doplněna o sadu testů pro validaci chování.

Abstract

This bachelor thesis deals with the realisation of a system for digital image deformation. The finished system implements this requirement with matrices and interpolations. In its first theoretical part, it describes the mathematical basis for these deformations on which it builds later on. A few of already existing programs which implement the deformations and many more are listed. The system implementation is explained after that. Lastly, a set of test to validate the behaviour is provided.

Klíčová slova

C++, RGB, RGBA, digitální obraz, interpolace, deformace, matice

Keywords

C++, RGB, RGBA, Digital Image, Interpolation, Deformation, Matrices

Citace

HEJTMÁNEK, Jiří. *Přesné úpravy obrazu - posouvání a deformace*. Brno, 2019. Bakalářská práce. Vysoké učení technické v Brně, Fakulta informačních technologií. Vedoucí práce Doc. Dr. Ing. Pavel Zemčík

Přesné úpravy obrazu - posouvání a deformace

Prohlášení

Prohlašuji, že jsem tuto bakalářskou práci vypracoval samostatně pod vedením pana Doc. Dr. Ing. Pavla Zemčíka. Uvedl jsem všechny literární prameny a publikace, ze kterých jsem čerpal.

.....
Jiří Hejtmánek
15. května 2019

Poděkování

Tímto bych chtěl poděkovat svému vedoucímu práce panu Doc. Dr. Ing. Pavlu Zemčíkovi za odbornou pomoc a rady při tvorbě této práce.

Obsah

Kapitola 1

Úvod

Počítačově upravené obrazy jsou v dnešní době všude kolem nás. Bez zásahu grafických expertů se obejde málokterá modelka, publikum v kině se dívá na počítačem zpracované záběry už v prakticky každém filmu, aniž by o tom někdy nutně věděli. Oblast počítačového vidění by bez tohoto konceptu nemohla ani existovat. Využití předzpracovaných obrázků je prakticky nevyčerpatelné. Avšak i ty nejpokročilejší úpravy mají základ v několika principiálně velmi jednoduchých operací, které jsou známé již desítky let.

Tato práce popisuje návrh a implementaci systému, který provádí jednoduché operace deformace obrazu. Stejnými (a ve větším počtu) operacemi disponují mnohé rozšířené softwary pro úpravu obrázků, ať už komerčního či svobodného původu. Mnohé programovací knihovny toto rovněž umožňují, pouze algoritmy realizující dané operace bývají optimalizované pro konkrétní hardware platformy, na které má pracovat.

V kapitole 2 je několik takových softwarů představeno a popsáno.

Následující kapitola popisuje digitální obraz, jeho druhy a reprezentace. Ve své druhé části se kapitola zabývá způsoby získání digitálního obrazu z analogového signálu.

Speciální pozornost je věnována interpolaci v kapitole 4, jelikož interpolace je v implementaci práce velkou částí. Konvoluční jádro, které je rozdílné pro každou interpolaci, je též popsáno.

V kapitole 5 jsou již popsány konkrétní geometrické transformace z obecného pohledu doplněné o příslušné matematické zápisy, které je definují.

Kapitola 6 staví na předchozích kapitolách a popisuje implementaci již zmíněných operací. Na začátku jsou vypsány použité nástroje, zmíněn je též způsob práce programu.

Poslední kapitola 7 se celá zabývá testováním systému. Jednotlivé testy jsou vysvětleny a doprovázeny relevantními obrázky.

Na samotný závěr je potom celý systém zhodnocen a jsou popsána možná rozšíření.

Kapitola 2

Existující software

Tématika zpracování obrazu je velice oblíbená problematika již několik desítek let. Zájem o operace pro úpravy obrazu jako jsou změny barev, deformace, škálování a další dal prostor pro vznik mnoha grafickým editorům sloužícím pro tyto účely. Cílem takových editorů je uživateli poskytnout grafické uživatelské rozhraní, které mu usnadní práci s programem a zpřístupní mu k používání sadu funkcí, jejichž funkcionalita je uživateli skryta, jelikož často přesahuje jeho rámec chápání problematiky. V dnešní době je již mezi uživateli jakési očekávání určitých minimálních operací, které by měl jednoduchý software pro úpravu obrazu nabízet. Zde se pak dá mluvit o neoficiálním standardu pro editory obrazu, které jsou z drtivé většiny dodržovány i u těch nejjednodušších programů. Je pak víceméně na preferencích uživatele, který software bude využívat. Často však uživatelé nevyužívají jediný grafický editor, ale svoji práci provádí postupně mezi více editory k dosažení požadovaného výsledku. Některé z populárních variant, pro účely práce zásadnějších bitmapových grafických editorů, tedy editorů pracujících s rastrovou grafikou 3.1.2 budou popsány v následujících podkapitolách.

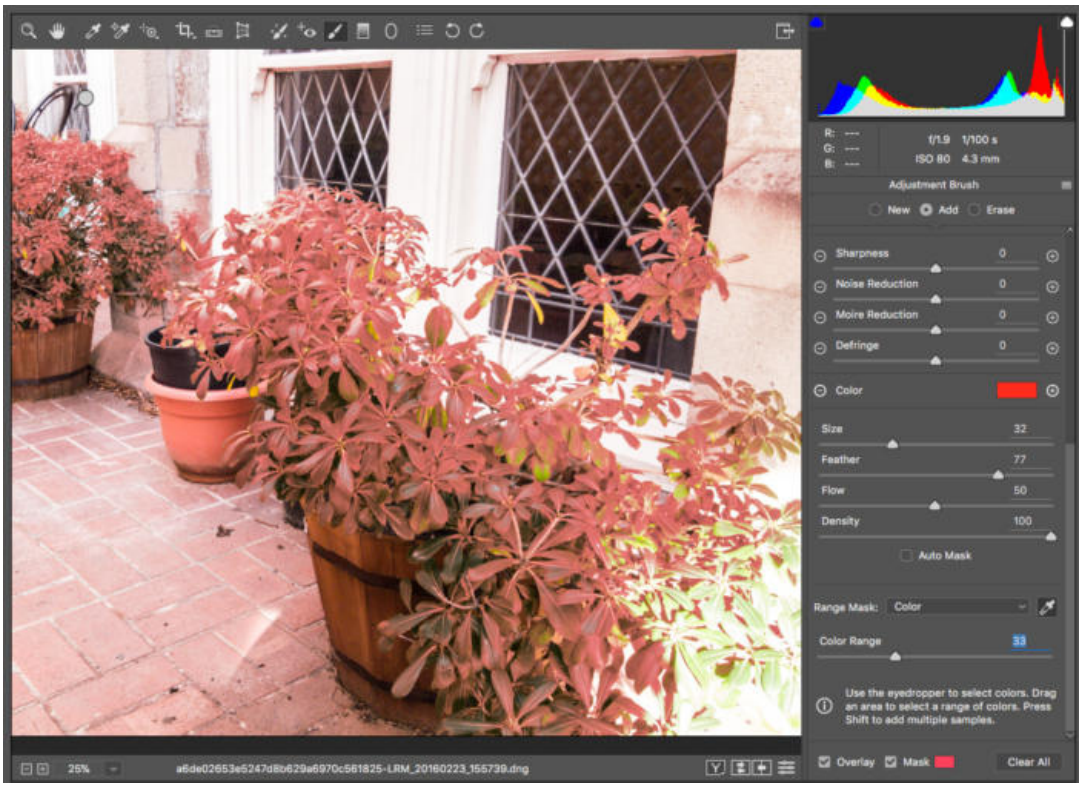
2.1 Adobe Photoshop

Adobe Photoshop je jedním z celosvětově nejoblíbenějších grafických editorů pro zpracování obrazu tvořeného rastrovou grafikou, a to i přestože se jedná o poměrně drahý komerční software dopstupný pouze pro operační systémy Windows a MAC OS. Podle mnohých uživatelů, mezi kterými najdeme velkou řadu designerů a fotografů, je Adobe Photoshop dokonce standardem, bez kterého se lze v dnešní době jen těžko obejít a který nemá ve své kvalitě přílišné konkurence.

V roce 1988 jej vytvořili bratři Thomas a John Knollovi a o rok později ho prodali společnosti Adobe Systems, která ho spravuje až do současnosti. Photoshop umožňuje svým uživatelům upravovat digitální rastrové snímky, ale rovněž i vytvářet úplně nové pomocí velmi pokročilých funkcí pro kresbu, tuto možnost oceňují především ilustrátoři. Klíčovou funkcí je několikavrstvé zpracování grafického obsahu, což zpřístupňuje nadstandardní operace jako je například překrývání fotografií nebo separátní práci s pozadím snímku. Od roku 2005 lze v tomto grafickém editoru pracovat i s 3D grafickými objekty. Zajímavým prvkem mohou být funkce pro práci s videem, které jsou součástí implementace aplikace.

Jedná se o profesionální grafický editor, který tedy může být poměrně náročný na pochopení, z tohoto důvodu existuje velká řada výukových programů, ať už se jde o formu prostého textu nebo video návodů. Je možné využít i oficiální výukové materiály, které lze

najít přímo na webových stránkách autorské společnosti. Adobe Photoshop přesto může zaujmout i neprofesionální a amatérské tvůrce grafického obsahu, protože se zde můžeme setkat s řadou automatických úprav, které nevyžadují hlubší pochopení problematiky, a přesto se staly velmi oblíbeným nástrojem pro korekci digitálních snímků a videa. [1] [13]



Uživatelské grafické rozhraní grafického editoru Adobe Photoshop; Převzato z [13].

2.2 GIMP

Další velmi rozšířený editor, který jistě stojí za zmínku je GIMP. Svou popularitu částečně navýšil z důvodu, že se jedná o bezplatný software s dostupností všech zdrojových kódu pod svobodnou licencí GNU. Tato skutečnost zpřístupňuje každému dobrovolnému zájemci možnost pomoci s budoucím vývojem a rozšířeními samotné aplikace, a to ať už formou prostého testování aplikace a případného nahlašování nalezených chyb anebo i tvorbou nových funkcí pro grafickou editaci, popřípadě dále zmíněných výukových nástrojů. GIMP je dostupný rovněž na operačním systému Linux, což nebývá u grafických editorů vždy zvykem.

GIMP pracuje převážně s rastrovým obrazem a umožňuje editaci digitálních snímků, nikoliv však už úpravy videa jako je tomu u Adobe Photoshop, za kterým GIMP v profesionální sféře využití zaostává. Přesto však za dobu své existence tento grafický editor prošel velkou řadou vylepšení a dokáže naplnit očekávání mnoha svých uživatelů. Aktuálně umožňuje práci s velkou skupinou souborových formátů, práci s vrstvami a kanály a všechny další základní operace pro zpracování obrazu. GIMP poskytuje pokročilé možnosti skriptování v programovacích jazycích jako jsou C, C++, Python a mnoho dalších.

O práci v GIMPu byla napsána řada knih, výukových tutoriálů a oficiální manuál přeložený do sedmnácti populárních jazyků, lze tedy tvrdit, že podpora pro nové uživatele je značně dostatečná. [3]

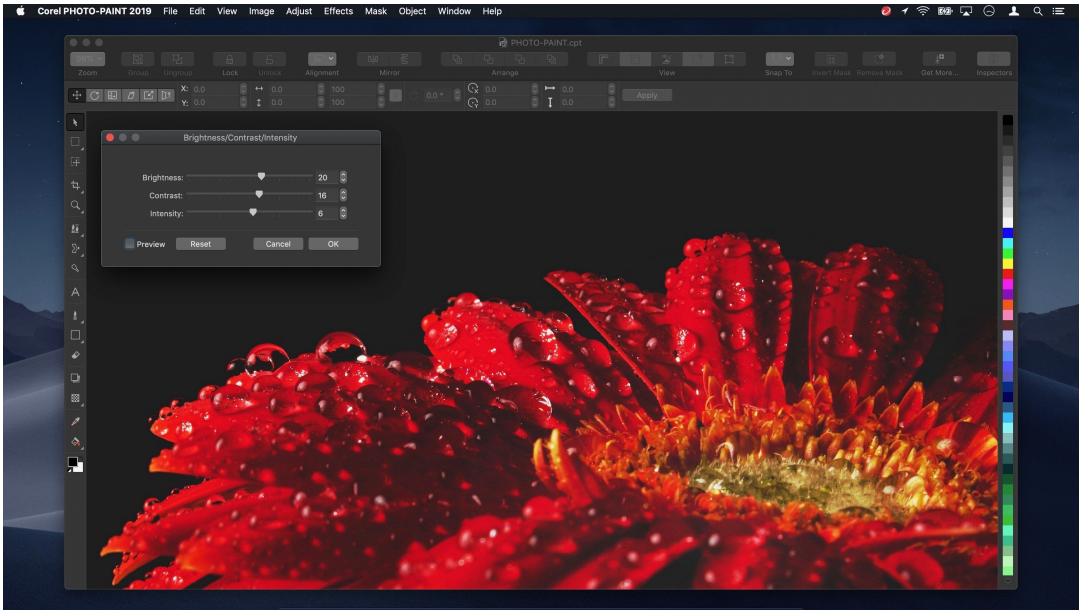
Uživatelské grafické rozhraní grafického editoru GIMP; Převzato z [17].

2.3 Corel PHOTO-PAINT

Corel PHOTO-PAINT je komerční bitmapový grafický editor pro úpravu fotografií, který překvapivě vznikl pouze jako doplněk pro svého úspěšného sourozence, kterým je vektorový editor CorelDRAW používaný například i na skicování a tvorbu průmyslových modelů, společně potom tvoří jako celek sadu nástrojů Graphics Suite.

Corel je dostupný pouze pro operační systémy Windows a Mac OS, přičemž se zaměřuje především na práci se souborovým formátem RAW pro více než 300 typů fotoaparátů a za cílovou skupinu lze tedy předpokládat profesionální i neprofesionální fotografy, společnost Corel se však snaží podporovat flexibilní možnosti licencování pro firmy a vzdělávací instituce.

Grafický editor umožňuje kreslení obsahu pro návrháře, tvorbu komplexních kompozic, retušování, izolaci oblastí obrazu, vícevrstvé editování přezdívané jako práce s objekty a mnoho dalších očekávaných funkcí pro zpracování obrazu. Úkony mohou být rovněž automatizovány pomocí skriptů a maker, psaných v jazyku Corel Script případně Visual Basic. K dispozici je celá řada manuálů a uživatelských příruček, je však nutné logicky oddělovat funkcionality mezi balíky celé sady nástrojů Corel. [2]



Uživatelské grafické rozhraní grafického editoru Corel PHOTO-PAINT; Převzato z [4].

Kapitola 3

Digitální obraz

Tématem této práce jsou transformace digitálního obrazu. Než dojde na samotné transformace, je potřeba vědět, jakým způsobem je směsice různých analogových signálů tvořící obraz reprezentována digitálně v počítači. K tomuto úkonu sloužící digitalizace a následné vzorkování jsou popsány v této kapitole, stejně jako i samotný výsledek digitalizace a tedy způsob, kterým lze zdigitalizovaný obraz uchovávat v paměti počítače.

3.1 Reprezentace digitálního obrazu

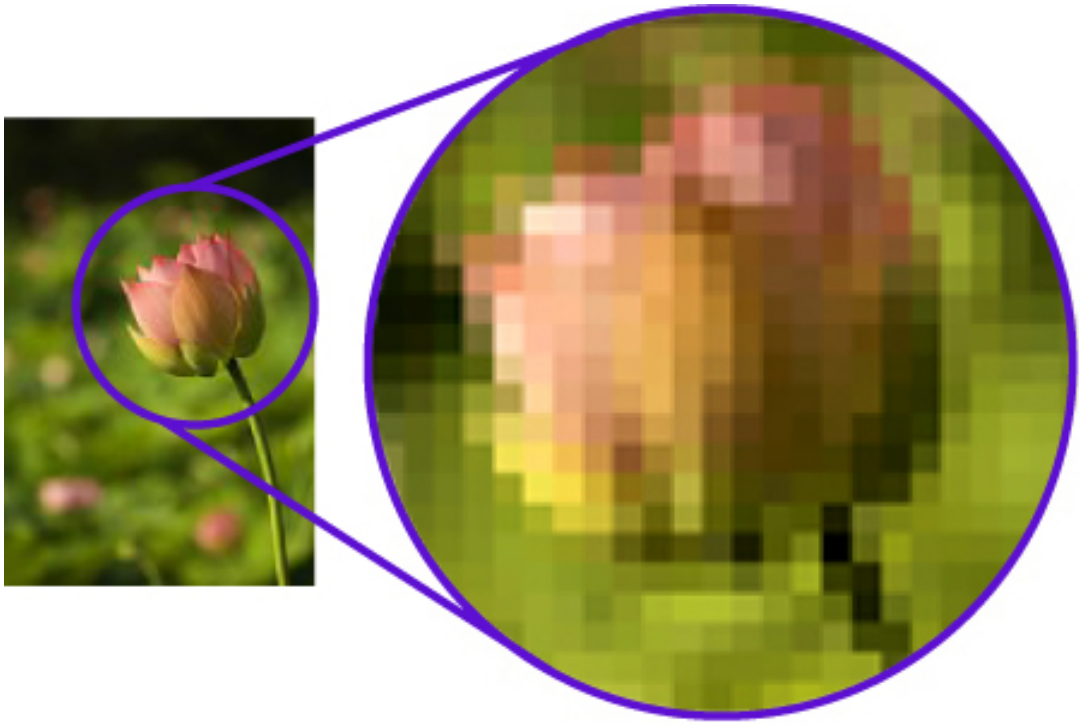
Lidské oko dokáže zachytit pomocí vizuálních vjemů velké množství informací, které pro nás představují pozorovaný reálný obraz. Velmi často však chceme taková data reprezentovat v digitální podobě, což umožňuje práci s nimi za pomocí výpočetní techniky. Je zde nutné zmínit, že existují dvě hlavní varianta, s kterými se dá k dané problematice přistupovat. Digitální obraz se skládá z binárních dat reprezentujících vektorový či rastrový přístup zobrazení, tyto techniky jsou popsány v následujících podkapitolách.

3.1.1 Vektorová grafika

Přestože se tato práce vektorovou grafikou nezabývá, bude velmi stručně zmíněna pro hlubší pochopení rozebírané problematiky. Vektorový přístup je popsán pomocí matematických rovnic, vždy je dán počáteční a cílový bod pro zobrazení a objekt, který je spojuje. Takový objekt nejčastěji představuje přímka či křivka. Hlavní výhodou vektorové grafiky je možnost přiblížení obrazu beze ztráty jeho kvality.

3.1.2 Rastrová grafika

Rastrová grafika (někdy rovněž označováno jako bitmapová grafika) popisuje nejčastější způsob reprezentace digitálního obrazu, k tomu přispívá i skutečnost, že každý kamerou zachycený digitální snímek je automaticky rastrový. Binární data reprezentujicí nespojitou obrazovou funci $f(x, y)$ jsou zde ukládána do pixelů, představujících jeden obrazový bod na displeji pro zobrazení. Tyto pixely jsou uspořádány do obrazové mřížky, jejíž velikost odpovídá rozlišení obrazu a pomocí které můžeme jednoznačně rozlišovat jednotlivé pixely. Jako celek tvoří matici intenzit jednotlivých barevných kanálů, o kterých bude zmínka v následující podkapitole. 3.1.3 U rastrového obrazu, na rozdíl od vektorového, dochází při jeho přiblížení k degradaci kvality zvané rozpad na artefakty obrazu, tuto situaci zachycuje obrázek 3.1.2. [9]



Vznik nežádoucích artefaktů při přiblížení rastrové grafiky; Převzato z [9].

Obrovskou výhodou rastrové reprezentace je možnost každému jednotlivému pixelu přiřadit odlišnou barvu a tvořit tak bohaté obrazové kompozice.

3.1.3 Barevný model RGBA

V rastrovém digitálním obraze každý obrazový bod definuje intenzitu zobrazované barvy, a jakým způsobem je tato hodnota kódována určuje použitý barevný model. Popis jednotlivých barevných modelů není předmětem této práce, a proto zde bude přiblížen pouze model RGBA, který je použit. RGBA je klasický RGB model obsahující barevné složky R – červená, G – zelená, B – modrá barva, rozšířený o takzvaný alfa kanál, který definuje výslednou průhlednost. Tyto čtyři složky jsou ukládány pro každý pixel ve výše zmíněném pořadí a to v binární podobě bytu.

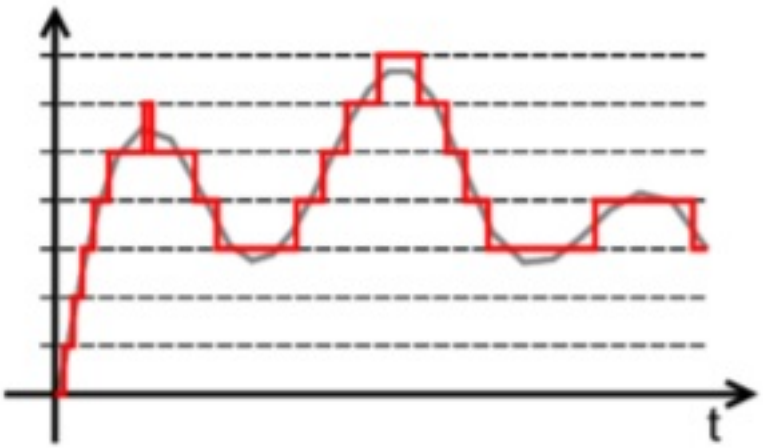
3.2 Digitalizace

Digitalizace je proces převodu analogového signálu na signál digitální, lze ji vnímat rovněž jako převod spojitého signálu na diskrétní. Tuto operaci je možné provádět pro zvukové, obrazové i textové vstupy a jejím výstupem jsou binárně reprezentovaná data. Pro úspěšné provedení digitalizace se používá technik kvantování a vzorkování.

3.2.1 Kvantizace obrazu

Kvantizace obrazu představuje zaokrouhlování hodnot původní barevné hloubky pixelů na nejbližší hodnoty odpovídající konečnému počtu referenčních úrovní a jejím cílem je tedy redukce barevné palety výsledného obrazu. Taková operace logicky vede k částečné ztrátě

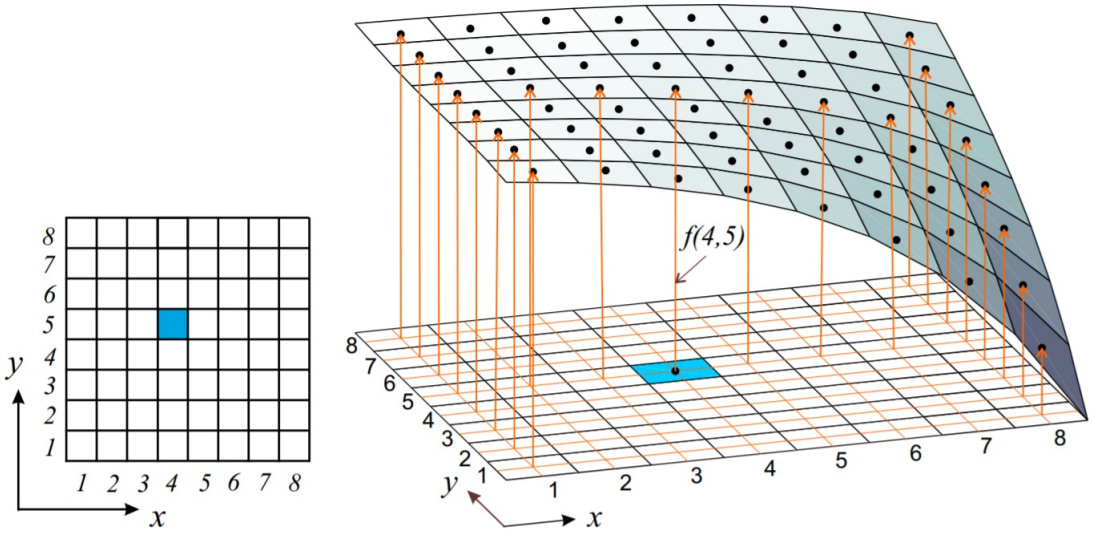
počáteční informace, přičemž velikost kvantizační chyby pak přímo závisí na zvoleném počtu referenčních úrovní, kterých je obvykle 256 pro každou intenzitu barvy. Kvantizace se dá rozdělit na dva druhy, na uniformní a neuniformní. Uniformní kvantizace je častější a jednodušší variantou, při níž je vzájemná vzdálenost referenčních úrovní jednotná. Opačnou situaci pak popisuje neuniformní kvantizace. [18]



Proces kvantizace signálu; Převzato z [18].

3.2.2 Vzorkování obrazu

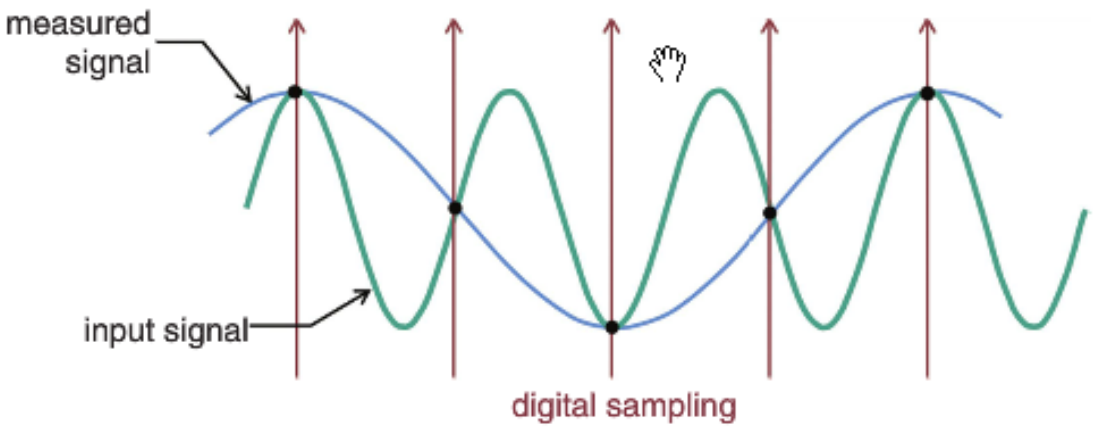
Vzorkování je proces zachytávání hodnot nebo také vzorků ze spojité funkce a to v předem zvoleném stejném intervalu. Čím je frekvence vzorkování vyšší, tím lepší je výsledná kvalita obrazu, ovšem za cenu větší výpočetní náročnosti. Dochází tedy cíleně k významné redukci celkového počtu vzorků, což rovněž jako kvantizace vede k degradaci kvality původních vstupních dat. Vzorkování v teorii zpracování obrazu představuje proces snižování výsledného rozlišení obrazu. [14]



Proces vzorkování v teorii zpracování obrazu; Převzato z [14]

3.2.3 Vzorkovací teorém

Základní myšlenkou vzorkování je, že spojitá funkce, která se mění velmi rychle, musí být výrazně rychleji vzorkována než by tomu bylo u funkce, která se mění pozvolna. Tento jev se dá velmi dobře popsat pomocí v oblasti vzorkování známého Nyquist-Shannonova teorému, který praví, že frekvence vzorkování musí být alespoň dvakrát vyšší než dvojnásobek nejvyšší harmonické složky vzorkovaného signálu. V opačném případě dochází k nežádoucímu jevu, který nazýváme Aliasing a představuje zkreslení signálu způsobené tzv. podvzorkováním. [11] [21]



Způsob vzniku podvzorkování vedoucího na aliasing; Převzato z [21]

3.2.4 Převzorkování

Proces převzorkování je využíván k transformaci vzorkovaného obrazu do nového souřadnicového systému s odlišnou velikostí oproti původnímu souřadnicovému systému, přičemž se může jednat o zvětšení i zmenšení. Vztah mezi nimi pak popisuje mapovací funkce, kterou můžeme dělit na dopřednou a zpětnou. Jelikož ve vzorkovaném obraze se nachází pixely vždy na pozicích náležících řádu přirozených čísel, tak dochází převzorkováním ke vzniku neceločíselných pozic. Tento nežádoucí jev představuje problém pouze u dopředného přístupu a v takovém případě se odstraňuje pomocí rozšiřování na celočíselný vzorek. [10]

Kapitola 4

Interpolace

Interpolace v teorii zpracování obrazu představuje proces určení intenzit jednotlivých barevných kanálu pro bod v obraze, který není znám. Za předpokladu, že jsou zmíněné hodnoty okolních bodů známé, můžeme tyto body proložit funkcí u které již můžeme určit funkční hodnotu neboli hledanou hodnotu bodu v obraze. V této kapitole je popsáno určení funkční hodnoty bodu za použití konvolučního jádra 4.1, v kterém se zde zmíněné metody liší.

Interpolační techniky můžeme dělit na adaptivní a neadaptivní. Jejich rozdíl pak spočívá v práci se zdrojovým obrazem, zatímco adaptivní metody se zaměřují na interpretaci obsahu obrazu a hledají v něm objekty ohraničené hranami nad nimiž pak provádí různou kombinaci interpolačních metod, například v oblasti hran metody pro zachování jejich ostrosti, neadaptivní metody význam obsahu snímku nerozlišují a nad celým obsahem používají jednotnou interpolační metodu. Hlavní účel této práce je zabývat se v pořadí druhými zmíněnými neadaptivními metodami. [5]

4.1 Konvoluce

Pojem konvoluce popisuje matematickou operaci spojování dvou funkcí, čímž dojde ke vzniku třetí výsledné sdružené funkce.Operace konvoluce se provede nad zdrojovými daty s použitím filtru, někdy též nazývaného jako jádro či kernel, který se posouvá nad celou oblastí vstupních dat a získá se tak výsledná mapa.

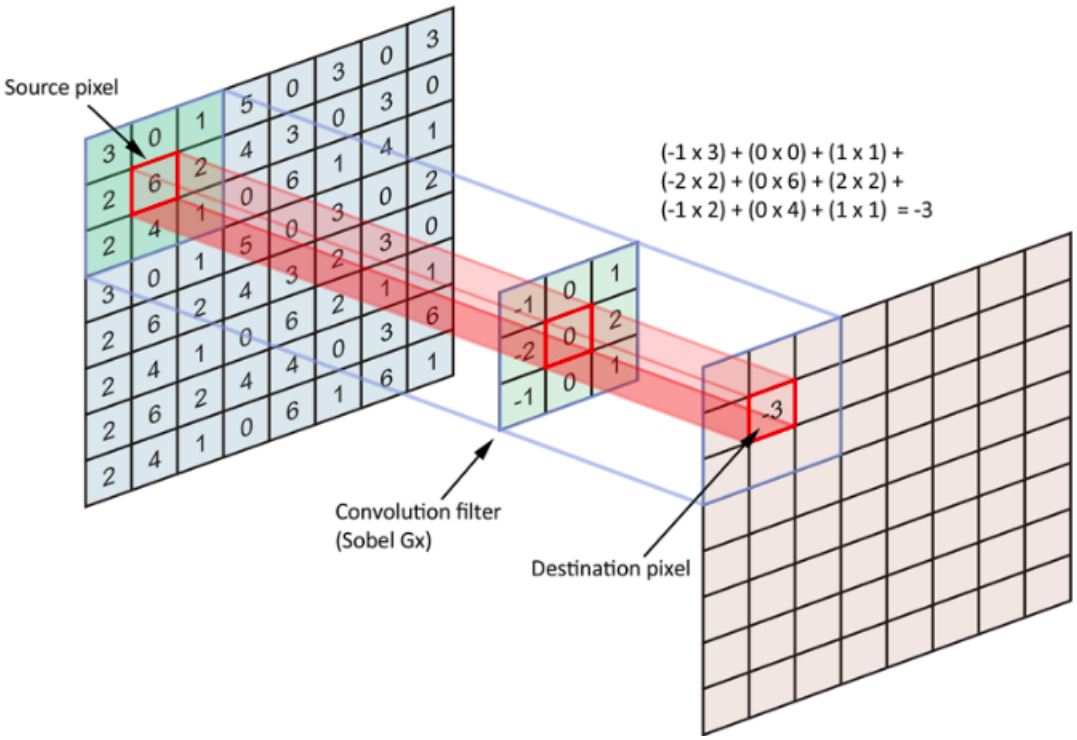
Vzorec diskretni konvoluce ma tvar:

$$(f * h)(x, y) = \sum_{i=-k}^{k} \sum_{j=-k}^{k} f(x - i, y - j) \cdot h(i, j). \tag{4.1}$$

kde f(x,y) je spojitá obrazová funkce a h(x,y) je konvoluční jádro.

V případě diskrétní konvoluce lze jádro vnímat jako čtvercovou konvoluční masku, kterou položíme na příslušné místo vstupního obrazu. Každý pixel překrytý touto maskou vynásobíme koeficientem v příslušné buňce masky a provedeme součet všech těchto hodnot, kterému je pak roven jeden výsledný pixel. Zajímavou situací je přiložení středu masky na okrajové řádky či sloupce vstupního obrazu, pak dochází k jejímu přesahu. Tento jev se v praxi řeší několika způsoby; první z nich mírně naivní způsob je ignorace této skutečnosti a výpočet nového pixelu pouze z přístupných vstupních bodů, druhý z nich je rozšíření originálních dat o nulové hodnoty a dále pak technika rozkopírování hodnoty okrajové buňky do rozšířené oblasti.

Vhodně zvolená jádra jsou vždy čtvercová o liché horizontální a vertikální velikosti, tedy obsahující jasně definovanou středovou oblast. Pochopení této skutečnosti usnadňuje následující obrázek 4.1 [12].



mapování vstupního obrazu pomocí konvolučního jádra; Převzato z [12]

4.2 Nejbližší soused

Metoda nejbližšího souseda je nejjednodušší varianta interpolace, při které je hodnota každého interpolovaného pixelu přiřazena podle hodnoty nejbližšího sousedního pixelu ve zdrojovém obraze. Přestože je metodu poměrně jednoduché realizovat, tak výsledná kvalita obrazu zpravidla nebývá dostatečně uspokojivá. Konvoluční jádro vrací váhu bodu w v závisloti vzdálenosti pixelu x od hledaného bodu. Zdroj: word + obrazek

$$w_{nn}(x) = \begin{cases} 1 & pro \quad -0.5 \leq x < 0.5 \\ 0 & jinak \end{cases} \tag{4.2}$$

4.3 Lineární interpolace

Lineární interpolace popisuje situaci v jednorozměrném prostoru a hodnota získaného interpolovaného pixel u je vypočítávána jako průměr dvou okolních pixelů zdrojového obrazu. Velmi podobným případem je pak bilineární interpolace, které je využíváno v dvourozměrném prostoru a průměr je počítán ze čtyř okolních pixelů. Lineární a bilineární interpolace poskytuje obraz s vyhlazenými hranami a snímek je tedy jemnější nežli je tomu při použití metody nejbližšího souseda, to vše však za cenu mírného rozmazání výsledné reprezentace. Zdroj a obrazek word

$$w_{lin}(x) = \begin{cases} 1-x & pro \quad |x| < 1 \\ 0 & pro \quad |x| \geq 1 \end{cases} \tag{4.3}$$

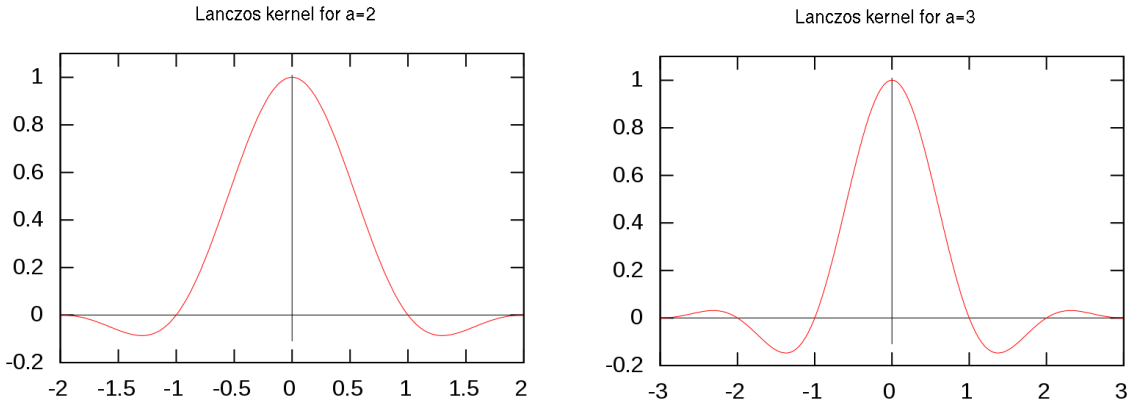
4.4 Bikubická interpolace

Bikubická interpolace rozšiřuje interpolaci bilineární, s tím rozdílem, že nyní je průměr vytvářen z okolních šestnácti pixelů. Přičemž blíže vzdáleným pixelům je přikládána větší váha než pixelům vzdálenějším Bikubická metoda interpolace je výpočetně náročnější než předchozí varianty, poskytuje však přesnější výsledky, u kterých se rovněž setkáváme s mírným rozostřením výsledného obrazu, tyto problémy lze však řešit využitím dalších doostřovacích filtrů. S využitím tohoto principu interpolace se setkáváme i v řadě programů pro zpracování programu jako je například Photoshop.

$$w_{cub}(x,a) = \begin{cases} (-a+2)\cdot|x|^3 + (a-3)\cdot|x|^2 + 1 & pro \quad 0 \leq |x| < 1 \\ -a\cdot|x|^3 + 5a\cdot|x|^2 - 8a\cdot|x| + 4a & pro \quad 1 \leq |x| < 2 \\ 0 & pro \quad |x| \geq 2 \end{cases} \tag{4.4}$$

4.5 Lanczosova interpolace

Lanczosova interpolace někdy také označováno jako Lanczosovo okno nebo Lanczosovo filtrování je filtr pojmenovaný po stejnojmenném maďarském autorovi a patří mezi oblíbené omezené funkce z třídy sinc. Je zde využíváno Lanczasova jádra, které je možné vidět na obrázku 4.5. Tato interpolace se podobá bikubické interpolaci, využívá principiálně velmi podobného konvolučního jádra, avšak Lanczosova metoda dokáže zachovat lepší ostrost hran a tónové přechody jsou viditelně jemnější, to s sebou přináší vyšší výpočetní náročnost, a proto nemusí být vždy preferovanou volbou v oblasti této tématiky, přestože bývá někdy označován jako nejlepší kompromisní volba. [20]



Lanczosovo jádro; Převzato [6]

$$w(x) = \psi(x)\cdot Sinc(x). \tag{4.5}$$

$$Sinc(x) = \begin{cases} 1 & pro \quad |x| = 0 \\ \frac{sin(\pi x)}{\pi x} & pro \quad |x| > 0 \end{cases} \tag{4.6}$$

$$\psi_{Ln} = \begin{cases} 1 & pro \quad |x| = 0 \\ \frac{sin(\pi \frac{x}{n})}{\pi \frac{x}{n}} & pro \quad 0 < |x| < n \\ 0 & pro \quad |x| \geq n \end{cases} \tag{4.7}$$

Kapitola 5

Geometrické transformace

Jedněmi z nejčastěji používaných operací v počítačové grafice jsou geometrické transformace, které jsou mnohdy prováděny v řádu tisíců až milionu opakování a z toho důvodu bývají většinou akcelerovány hardwarou implementací. Bylo uvedeno, že diskrétní obraz je reprezentován konečným počtem bodů, transformace nad digitálním obrazem je pak provádění operací nad konečnou množinou těchto bodů. Podle využité operace se pak diskrétní transformace rozdělují na lineární a nelineární.

Lineární diskrétní geometrické transformace během výpočtů zachovávají korelaci mezi proměnnými. Řečeno jednoduššeji, lineární operace jako násobení, dělení či sčítaní a odčítání nad body lze považovat za lineární geometrické transformace.

Neliární transformace jsou pak takové, které korelaci mezi proměnnými nezachovávají a tedy operace jako převrácení hodnoty či druhá odmocnina jsou nelineární geometrické transformace.

5.1 Homogenní souřadnice

Pro usnadnění výpočtu geometrických lineárních transformací se používá reprezentace bodů pomocí homogenních souřadnic. Díky této reprezentaci lze vyjádřit všechny mnou popisované transformace jednotnou transformační maticí. Uspořádaná trojice čísel [x,y,w] představuje homogenní souřadnice bodu P s kartézskými souřadnicemi [X,Y] ve dvou rozměrech, platí-li:

$$X = \frac{x}{w}, Y = \frac{y}{w}, w \neq 0. \tag{5.1}$$

Bod P je svými homogenními souřadnicemi určen jednoznačně. Souřadnici w též nazýváme váhou bodu. Obecnou matici 3×3 reprezentující lineární transformaci bodu $P = [x, y, w]$ budeme označovat A, dále pak podle druhu transformace, např. T(translace), R(rotace). Transformaci souřadnic zapíšeme

$$P' = \begin{bmatrix} x' \\ y' \\ w' \end{bmatrix} = A_{3\times3} * P = A * \begin{bmatrix} x \\ y \\ w \end{bmatrix}. \tag{5.2}$$

5.2 Posunutí

Transformaci posunutí v rovině s homogenními souřadnicemi P(x,y,1) nebo také translace bodu P je určena vektorem posunutí $\vec{p} = (X_t, Y_t)$. Matice posunutí T a inverzní matice T^{-1}, kterou je možno interpretovat i jako posunutí zpětné posunutí opačným směrem, mají tvar

$$T(X_t, Y_t) = \begin{bmatrix} 1 & 0 & X_t \\ 0 & 1 & Y_t \\ 0 & 0 & 1 \end{bmatrix}, T^{-1}(X_t, Y_t) = \begin{bmatrix} 1 & 0 & -X_t \\ 0 & 1 & -Y_t \\ 0 & 0 & 1 \end{bmatrix}. \tag{5.3}$$

5.3 Změna měřítka

Změna měřítka (scale) v rovině s homogenními souřadnicemi P(x,y,1) ovlivňuje současně polohu i velikost transformovaného objektu ve směru souřadnicových os. Příslušné transformační matice jsou

$$S(s_x, s_y) = \begin{bmatrix} s_x & 0 & 0 \\ 0 & s_y & 0 \\ 0 & 0 & 1 \end{bmatrix}, S^{-1}(s_x, s_y) = \begin{bmatrix} 1/s_x & 0 & 0 \\ 0 & 1/s_y & 0 \\ 0 & 0 & 1 \end{bmatrix}. \tag{5.4}$$

kde s_x je koeficient změny měřítka ve směru souřadnicové osy x a s_y je koeficient změny měřítka ve směru souřadnicové osy y. Pokud je absolutní hodnota koeficientu v intervalu (0,1), dojde ke zmenšení a přiblížení transformovaného objektu. Je-li absolutní hodnota větší než jedna, dojde k prodloužení. V případě, kdy je znaménko záporné, dochází k prodloužení či zmenšení v opačném směru.

5.4 Rotace

Rotace (rotation) bodu v rovině s homogenními souřadnicemi P(x,y,1) kolem počátku soustavy souřadnic $O = [0, 0]$ o úhel α získáme bod P' o souřadnicích. Příčemž kladný směr otáčení je proti směru hodinových ručiček, podle pravidla pravé ruky.

$$\begin{aligned} X' &= X\cos\alpha - Y\sin\alpha \\ Y' &= X\sin\alpha + Y\cos\alpha. \end{aligned} \tag{5.5}$$

Matice transformace otáčení R a inverzní matice R^{-1} mají tvar

$$R(\alpha) = \begin{bmatrix} \cos\alpha & -\sin\alpha & 0 \\ \sin\alpha & \cos\alpha & 0 \\ 0 & 0 & 1 \end{bmatrix}, R^{-1}(\alpha) = \begin{bmatrix} \cos\alpha & \sin\alpha & 0 \\ -\sin\alpha & \cos\alpha & 0 \\ 0 & 0 & 1 \end{bmatrix}. \tag{5.6}$$

5.5 Zkosení

Zkosení (shear) v rovině s homogenními souřadnicemi P(x,y,1) zkresluje tvar objektu. Zkosení provádíme ve směru souřadnicových os. Transformační matice mají tvar

$$Sh_x(sh_x) = \begin{bmatrix} 1 & sh_x & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}, Sh_y(sh_y) = \begin{bmatrix} 1 & 0 & 0 \\ sh_y & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}. \tag{5.7}$$

kde koeficienty sh_x respektive sh_y udávají míru zkosení. Pro inverzní matice platí $Sh_x^{-1}(sh_x) = Sh_x(-sh_x)$ a $Sh_y^{-1}(sh_y) = Sh_y(-sh_y)$

5.6 Skládání transformací

Složitější transformace se skládají ze dvou či více lineárních geometrických transformací. Při postupném aplikování jednotlivých transformací na body objektu záleží na pořadí, v jakém se transformace provádějí. Pokud objekt například prvně zkosíme a potom otočíme, vznikne objekt jiný, než kdyby se tyto transformace provedly naopak. Jelikož je důležité pořadí provedení transformací, tak je důležité i pořadí násobení matic. V případě, kdy jsou aplikovány obecné transformace v pořadí A_1, A_2, je bod P transformován vztahem $P' = A_2 * A_1 * P$, tedy pozdější transformace je vložena do složené transformace zleva.

5.7 Geometrické transformace diskrétního obrazu

Již bylo uvedeno, že diskrétní obraz je reprezentován konečným počtem bodů. Aplikováním obecné transformace na bod určený diskrétními souřadnicemi v obraze vstupním může vzniknout bod o neceločíselných souřadnicích v obraze výstupním. Z toho vyplývá, že zde není možné se vždy omezit pouze na diskrétní souřadnice a způsobem přiřazení těchto souřadnic se budou zabývat metody mapování.

5.8 Warping

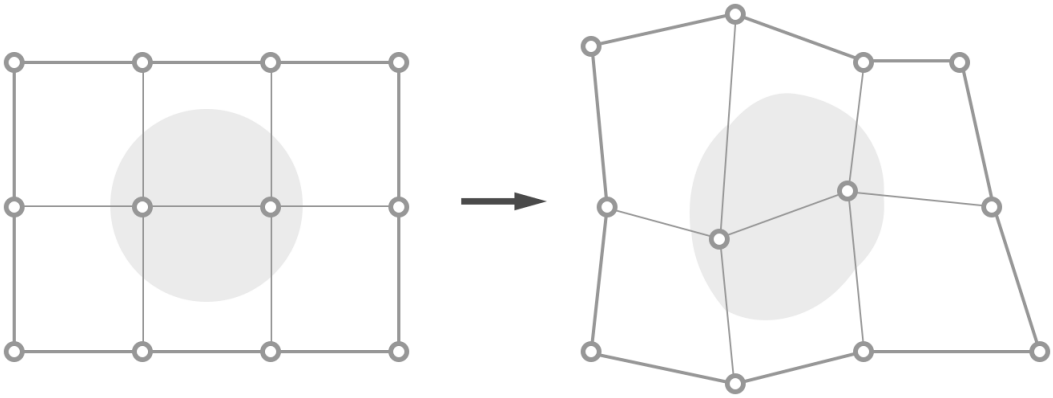
Proces warpingu obrazu spočívá ve vizuální úpravě, pokřivení tvarů, počátečního zpracovávaného obrazu, a to za pomoci geometrických transformačních metod, dochází zde tedy ke změně pozice jednoho či více bodů vstupního obrazu na pozici odlišnou, a to za pomoci mapovací funkce. Důležitou skutečností, které je třeba si povšimnout, je fakt, že jsou pozměněny pouze pozice pixelů, nikoliv však už jejich hodnota barevných kanálů nebo jejich intenzita, ty zůstávají stejné. Metoda warpingu se používá k záměrné deformaci obrazu a s touto funkcí je možné se setkat ve velké řadě grafických editorů.

Warping nachází uplatnění v oblasti pro kreativní tvorbu grafického obsahu, ale také více přínosné odstranění optického zkreslení způsobené kamerou nebo perspektivou zobrazení. Takové mnohdy nežádoucí zkreslení je možné vidět na následujícím obrázku 5.8 pořízeném pomocí Nikon D3S s čočkou širokoúhlého objektivu (zde 14 mm), jejíž ohnisková vzdálenost je výrazně nižší než ohnisková vzdálenost čočky normálního objektivu. Levá část automobilu na obrázku se jeví v porovnání se zbytkem obrazové kompozice podstatně rozměrově větší, než tomu ve skutečnosti je. [15] [19]

Ukázka nežádoucího zkreslení zobrazení; Převzato z [16]

Síťový warping

Síťový warping patří mezi jeden z často používaných způsobů, jak k této problematice přistupovat a prakticky se jím zabývá i tato práce. Principem této metody je zpřístupnění sítě čtyřúhelníků zobrazované jako mřížky s jednotlivými body ležícími na ní, přičemž právě posuny bodů dochází k deformaci původního obrazu. Síťový warping tedy umožňuje pokřivení pouze některých oblastí obrazu a zachování všech ostatních.[7]



Transformace - síťový warping; Převzato z [8]

Kapitola 6

Implementace

V této kapitole je popsána samotná implementace požadavků zadání práce a využité technologie během vývoje. Speciální pozornost je věnována využitým technologiím, neboť spousta základních knihoven či sada nástrojů obsahuje mimo jiné i prvky, které by mohly vykonat z části nebo někdy i úplně celou jednu transformaci. Kapitola se dále zmiňuje o strukturách a třídách, které byly pro implementaci transformací potřeba vytvořit. Poté je vysvětlen základní postup programu a důvod k jeho sestavení určitým způsobem. Nakonec je popsán způsob implementace matematických operací potřebných k provedení libovolné transformace.

6.1 Využité technologie

Dle zadání byl pro implementaci požadovaných funkcí zvolen programovací jazyk `C++` podle standardu `C++11`. Jazyk `C++` je kompilovaný, a tedy pro účely výpočtů, kterých je v oblasti manipulace s počítačovou grafikou vždy dost, velmi rychlý. `C++` byl preferován před jazykem `C`, který zadání práce též nabízelo, z prostého důvodu možnosti programovat projekt objektově. Objektové programování se také více podobalo modelování jednotlivých problémů než programování procedurální.

Projekt sám o sobě byl pojat ve dvou na sobě nezávislých částech, které jednak reflektují postup vývoje, ale lze také tyto dvě části samostatně sestavit.

První část umožňuje uživateli skrze příkazovou řádku či terminál manipulovat s obrázky podle popsaných lineárních geometrických transformací, což z ní činí limitovanou verzi oproti druhé, která umožňuje plnou manipulaci dle všech geometrických transformací, jenž jsou v této práci popsané. Důvodem této limitace je deformace podle mřížky, kterou nebylo v rozumném pojetí možné zrealizovat pro verzi, která bere od uživatele vstup pouze v textové podobě. Protože předmětem této práce nebylo psaní vlastního parseru pro různé formáty obrázků, byla pro jejich načítání zvolena svobodná[1] knihovna `lodepng`, která načítá obrázky ve formátu PNG a jejím výstupem je pak prosté pole čtveřic 8bitových hodnot reprezentující barevné kanály RGBA. Formát obrázků PNG byl zároveň zvolen jako jediný podporovaný formát pro tuto práci. Pro validaci vstupu a tisku nápovědy na standardní vstup byla zvolena svobodná[2] knihovna `cxxopts`. Obě využité knihovny stačí přiložit k projektu a během překladu se přeloží společně s projektem, není tedy potřeba externího linkování knihoven.

[1] zlib licence
[2] MIT licence

Druhou část pak tvoří stejné jádro kódu, avšak interakce s uživatelem je zprostředkována skrze grafické rozhraní. Právě tento rozdíl umožňuje vykreslení a manipulaci s mřížkou, která určuje styl deformace obrázku. Pro realizaci této části bylo zvoleno prostředí a sada knihoven `Qt`. `Qt` je svobodná[3] sada knihoven pro vytváření grafického prostředí nehledě na používanou platformu. Momentálně mimo rodiny operačních systémů Microsoft Windows podporuje také například Linux, macOS a mobilní operační systémy Android, iOS a Windows Phone. Uživateli je zde nabídna i možnost tvořit grafické rozhraní s pomocí značkovacího jazyka `QML` a velice snadno propojit rozhraní s jádrem, které je psané v C++. Knihovna byla zvolena právě díky tomu, že umožňuje přenést projekt na jinou platformu s minimálními úpravami. Zároveň má od svého vzniku již celkem dobrou pověst.

Drtivá většina vývoje a testování probíhala na počítačích s operačním systémem Windows, nícméně díky zmíněným technologiím není žádný velký problém projekt přenést na téměř libovolnou platformu za předpokladu, že pro ni existuje příslušný překladač.

6.2 Zpracování vstupního obrazu

Než bude možné provádět operace nad obrazem, je třeba jej mít uložený v určité podobě v paměti. K tomuto úkonu slouží již zmíněná knihovna `lodepng`, která dekóduje formát PNG a výsledek v surových bajtech uloží do operační paměti počítače. Její návratovou hodnotou je pak ukazatel na pole těchto bajtů.

Pro lepší manipulaci s takto načteným obrazem bylo vytvořeno několik struktur. Struktura `Pixel` v sobě uchovává údaje o hodnotách jednotlivých kanálů RGBA, šablona třídy `Matrix` umožnuje tvořit matice z libovolného typu, který využívá jako svůj podkres. Nakonec třída `Image`, která využívá pro svůj chod instanci šablony `Matrix` právě pro typ `Pixel`.

Struktura `Pixel` zmíněné kanály implementuje jako 16bitovou hodnotu. Důvodem jsou výsledky určitých operací, které mohou být záporné anebo větší než 255 a jsou důležité k navazujícím výpočtům. V závěru operací jsou však tyto hodnoty vždy zapraveny do rozsahu 0 až 255, aby bylo možné s touto strukturou pracovat pro vykreslení.

Třída `Matrix` implementuje jednoduché matice o rozměrech $m \times n$. Implemetuje jednoduchou funkci násobení matic a zkopírování matice. Veškeré dodatečné funkce (jako např. zdali je symetrická, jednotková atd. . .) se podle potřeby řeší externě.

Skrze třídu `Image` se pak dá manipulovat s jednotlivými pixely v podkreslující matici. Mimo jiné se zde načítá obsah do matice z operační paměti a z matice do výsledného výstupního souboru. Rovněž nabízí možnost vytvoření kopie obrázku.

6.3 Základní tok programu

Základní snahou bylo vytvořit částečně modulární systém, do kterého by bylo možné přidávat či odebírat funkce, které vykonávají určitou transformaci. V názvosloví projektu se takové funkce jmenují `Modifier`, kterýžto je prostým ukazatelem na funkci. Pro účel libovolného přidávání či odebírání transformačních funkcí existuje v projektu fronta těchto modifikátorů. Inspirací pro tento nápad (a názvosloví) byl `Modifier Stack` (zásobník modifikátorů) z 3D modelovacího softwaru Blender, ačkoliv podobnou koncept existuje i u mnoho jiných programů v jedné či druhé formě.

[3] GPL licence

Po spuštění programu se v konzolové verzi nejprve naparsují parametry, poté se projede fronta modifikátorů a jeden po druhém jak jdou za sebou se vykonají. Jakmile již není další modifikátor, program uloží výsledek a skončí. V grafické verzi se čeká na uživatelovo potvrzení, že má program zpracovat načtený obrázek, který uživatel musí rovněž specifikovat přes dialogové okno výběru. Výsledek se neukládá automaticky; opět se čeká na vybraný výstupní soubor od uživatele.

6.4 Matematické operace

Veškeré popsané transformace jsou implementovány pouze s pomocí vlastnoručně naimplementovaných struktur a základních knihovních funkcí (sinus, cosinus, odmocniny, ...). Ačkoliv `Qt` samozřejmě dovede manipulovat s obrázky, nejsou tyto funkce využity, neboť by to bylo v rozporu se podstatou zadání práce.

Zároveň nejsou žádné z prováděných transformací optimalizovány pro výkon tak, jako tomu je u knihoven k těmto účelům určeným. Takové optimalizace by si žádaly důkladnou znalost konkrétních procesorových architektur včetně jejich instrukčních sad. Ponecháním operací v C++ se zaručuje, že kód půjde přeložit všude tam, kde existuje překladač C++.

6.5 Grafické rozhraní

Přestože zadání práce přímo nespecifikovalo, jak má program s uživatelem komunikovat, bylo s pomocí sady nástrojů `Qt` vytvořeno velmi triviální uživatelské prostředí. V této sadě bylo využito rozhraní `QtQuick` 2.7, které si žádá minimální verzi Qt 5.12.2 pro správné fungování. Pointou grafické verze programu je pouze ukázka deformací za pomocí mřížky, nicméně byly doplněny i ostatní modifikace pro úplnost. Pro zachování jednoduchosti je výstupem grafické verze programu vždy soubor pojmenovaný jako **`output.png`**, který lze nalézt ve stejné složce jako vstupní soubor. Výstupní soubor je pak znovu načten do zobrazovací plochy.

Jako vstup se očekávají následující hodnoty:

- Zvětšení: Hodnoty mezi 0,00 až teoreticky libovolná hodnota (podle limitace hardwaru)
- Zkosení: Hodnoty od 0,00 až po 1,00
- Posun: Celé číslo reprezentující počet pixelů, o které má být obraz posunut
- Rotace: Celé číslo reprezentující úhel otočení ve směru hodinových ručiček. Očekávaná hodnota je ve stupních

Při zadání hodnot do polí je třeba změny potvrdit stisknutím klávesy `Enter`.

Pro použití mřížky je potřeba zaškrknout příslušené zaškrtávací pole. Poté je přes obrázek vykreslena sada bodů, která znázorňuje deformační mřížku. Bod je možné přesunout po kliknutí na požadovaný bod a následně na místo, kam se má přesunout. Kliknutím pravým tlačítkem myši se změny potvrdí.

V jednu chvíli lze používat pouze deformační modifikace bez mřížky, anebo deformace podle mřížky, nicméně obě možnosti zároveň nelze kombinovat. Po kliknutí na tlačítko vybízející ke zpracování obrázku se provedou požadované změny na načteném obraze a výstup je pro usnadnění opět načten na zobrazovací plochu. Program je poté uzamčen a nelze již provádět žádné další změny.

6.6 resizeModifier

Tento modifikátor je zodpovědný za nastavení velikosti výstupního obrazu, který je ve výchozím stavu nastaven na stejnou velikost jako obrázek vstupní. Pokud byla nastavena uživatelem hodnota pro změnu měřítka pro libovoulnou osu souřadnicového systému, tak se rozměry jednoduše přepočítají násobením.

6.7 transformModifier

Zde se vytvoří jednotlivé geometrické transformační matice podle uživatelem zadaných parametrů a jejich hodnot. Jednotlivé matice se vzájemně vynásobí a vznikne jedna transformační matice. Násobení matic probihá v tomto pořadí:

- Rotace
- Změna měřítka
- Zkosení na osách x, y
- Posun

Poté se pro každý pixel ve výstupním obraze vypočítají souřadnice v obraze vstupním. Ty se získají vynásobením pozice ve výstupním obraze s transformační maticí. Pokud byla uživatelem zvolena možnost přetečení, všechny pozice odkazující se na pixel mimo vstupní obraz se přepočítají tak, aby byla tato pozice v obraze.

6.8 interpolateModifier

V tomto modifikátoru se určí koncové hodnoty pixelu výsledného obrazu. Pokud pixel odhazoval na neceločíselnou pozici, pak je zapotřebí tuto hodnotu získat za pomoci interpolací.

Podle zvolené interpolace se vytvoří konvoluční filtr. Hodnoty konvolučního filtru jsou vypočítány dle patřičného konvolučního jádra, které dostane vzdálenost pixelu od hledaného bodu a vrátí váhu pixelu. Po aplikování konvoluce již známe potřebnou hodnotu.

6.9 warpModifier

Tento modifikátor vždy dostane z GUI dvě instance `std::vector` souřadnic bodů, reprezentující vstupní a výstupní mřížku. Implementován je dvouprůchodový síťový warping, úprava vstupního obrázku tedy proběhne ve dvou průbězích. Pro první je potřeba vytvořit dočasnou mřížku. Ta obsahuje vodorovné souřadnice z výstupní mřížky a souřadnice horizontální souřadnice z mřížky vstupní. Odtud je volán horizontální warping s obrazem vstupním a dočasným, dále s mřížkou dočasnou a zdrojovou.

Pro každý řádek jsou vypočítány průsečíky pomyslných horizontálních přímek mezi body mřížky dočasné a vstupní. Tím se rozdělí každý řádek na segmenty, které je možné velikostně porovnat. Pro každý bod v dočasném výstupním obraze je vypočítaná pozice v obraze vstupním. Pokud je každý bod přemapován, je zavolána interpolace k určení hodnot pixelů. Obdobně se provede vertikální warping, avšak namísto obrazu vstupního je použit obraz dočasného výstupu a místo obrazu dočasného výstupu obraz výstupní. Záměna je provedena i s mřížkami, místo dočasné mřížky je použita mřížka výstupní a místo mřížky vstupní je použita mřížka dočasná.

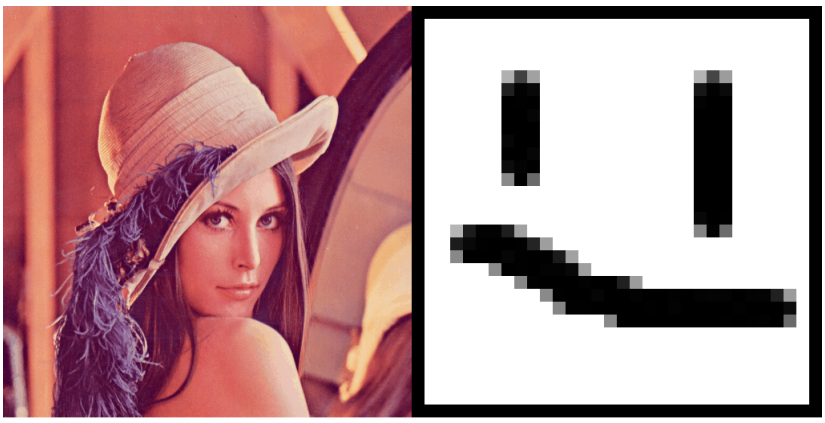
Kapitola 7

Testování

Obsah této kapitoly se zaměřuje na testování funkcionality vytvořeného programu. Popsány jsou testy všech implementovaných modifikací. Nejprve je popsána datová sada zvolená pro účely testování. Poté následují modifikace, kde pro každou modifikaci je zmíněn její očekávaný výsledek a připomenut princip. Všechny testy jsou doplněny obrázky pro ilustraci postupu testu. Celkové zhodnocení je na samotném konci této kapitoly. Všechny obrázky testů, včetně zde zobrazených, lze najít v plné velikosti na přiloženém paměťovém médiu.

7.1 Datová sada

Pro demonstraci funkcionality implementovaného programu byly za vstup zvoleny dva obrázky. První obrázek o čtvercovém rozlišení 32 pixelů byl vytvořen v programu Malování na systému Windows 7. Jako druhý obrázek byl zvolen již velmi známý testovací obrázek Lenna, vyobrazující Lenu Söderbergovou, při čtvercovém rozlišení 512 pixelů.



Testovací obrázky smile and Lena

7.2 Zvětšení

Modifikace zvětšení je ve své podstatě velmi jednoduchá. Výsledný obrázek se zvětší při zadání hodnoty větší než 1, zmenší při hodnotě mezi 0 a 1 a žádná efektivní operace se neprovede při zadání hodnoty 1.

Při zvětšení by ve výsledném obraze vznikala prázdná místa, která se doplní podle druhu interpolace ze vstupního obrazu. Naopak při zmenšení je zapotřebí sloučit více pixelů

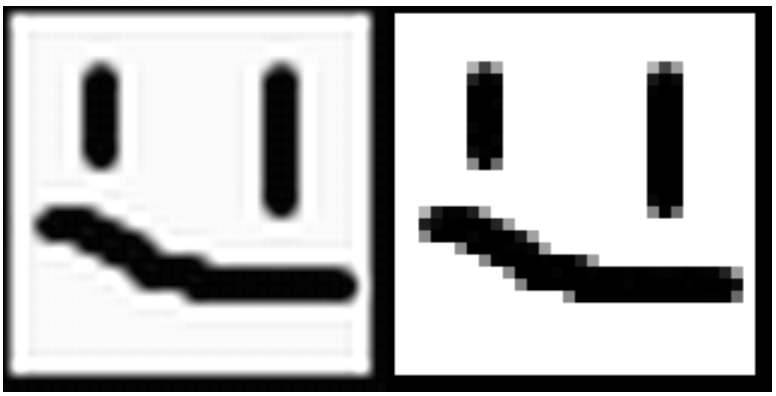
dohromady. K tomuto úkonu je využita interpolace. Podle druhu použité interpolace pak lze očekávat odlišný výstup.

Interpolace podle nejbližšího souseda pouze jednoduše zvětší vstupní obrázek a informace, které je potřeba doplnit vezme podle předem spočteného pixelu, jehož hodnotu vykopíruje. Ostatní implementované interpolace výslednou hodnotu vypočítají prokládáním okolních pixelů.

První test zvětšení zvětšil celý obrázek na obou osách pětkrát pro všechny podporované interpolace. Pro interpolaci nejbližšího souseda lze očekávat velké množství ostrých přechodů. Při interpolaci lineární se ostré hrany vytrácí a dochází ke zjemnění přechodů. Větší míry zjemnění detailů se pak dosáhne při použití kubické interpolace. Interpolace Lanczos by měla oproti kubické interpolaci lehce zvýraznit přechody, nicméně stále zachovat určitou míru jemnosti.

Druhý test naopak vstupní soubory zmenšil na faktor 0,60. Tato hodnota byla zvolena záměrně, aby se při zpětném mapování funkce nedotazovala na každý druhý celý pixel, ale byly vynuceny desetinné hodnoty.

Obdobně jako u zvětšení, zmenšení podle nejbližšího souseda vyprodukuje obraz s ostrými přechody. Lineární interpolace produkuje stejný výsledek při méně ostrých přechodech. Kubická interpolace zjemní tyto přechody ještě o trochu více a interpolace Lanczos se pak liší oproti kubické v podstatě minimálně.



Zvětšení obrázku smile,interpolace: lanczos a nejbližší soused

7.3 Zkosení

Zkosení postupně přesunuje pixely ze vstupního obrázku na posunuté souřadnice ve výstupním obrázku.

7.4 Posun

Modifikace posunu je velmi triviální. Souřadnice ve výsledném obrázku odpovídají souřadnicím ve zdrojovém obrázku s přičtením zadaného ofsetu. Standardně jsou hodnoty posunu pro obě osy rovny nule. Z této podstaty je možné nechat obrázek při posunu přetéct na dané ose zpět na druhou stranu.

7.5 Rotace

Rotační modifikace je příkladem složené transformace. Aby bylo možné zrotovat obrázek kolem libovolného bodu, je potřeba nejprve bod středu souřadnic obrázku posunout na požadovaný bod, provést rotaci, a poté jej opět posunout na své původní místo. V této práci se k rotaci používá fixní bod, který vždy odpovídá středu vstupního obrázku.

Při rotaci o 0 či 360 stupňů lze očekávat výstup shodný se vstupem. Rotace o 180 stupňů obrázek převrátí. Jiné hodnoty rotace, které nejsou násobkem 90 se větší či menší mírou dostanou za svoje hranice a výsledný obrázek je tedy zaříznut, aby se vešel do svých původních rozměrů.

Rotace obrázku smile,interpolace: lanczos a nejbližší soused

7.6 Deformace podle mřížky

Test deformace podle mřížky je proveden na mrizce 3×3. Při přesnutí prostředního bodu do jedné třetiny výšky i šířky obrázku dojde k posunutí bodu deformace. Tím je docíleno, že pravá strana obrázku je rozšířena ve směru posunutí obrázku, zatímco levá strana je o toto rozšíření zúžena.

Kapitola 8

Závěr

Cílem této bakalářské práce bylo zrealizovat systém pro provádění deformací nad obrázky. Implementovaný systém tuto činnost vykováná s pomocí specificky nastavených matic a interpolací pro doplnění chybějcích informací.

Různé deformace jsou implementovány jako prosté funkce, které projíždí každý pixel zdrojového obrazu a podle druhu deformace provádí různé matematické operace. Tyto funkce jsou vloženy do fronty, ze které jsou postupně brány a volány, čimž se docílilo určité modularity.

Program vykováná požadované operace s minimální odchylkou od očekávaného výstupu. Jedinou výjimkou je deformace podle mřížky, která při mřížce s rozlišením o hodnotě 5 a výše vytvoří špatný výstup i při implicitním nastavení mřížky.

Možné vylepšení či rozšíření může být plná implementace interaktivity s frontou modifikátorů, aby si uživatel mohl zvolit různé modifikace v různém pořadí tak, jako je tomu například v programu pro 3D modelování Blender.

V další fázi rozšíření by program mohl pracovat jako jakýsi interpret a požadované modifikace by byly psané v interpretovaném jazyce, který by program vykonával v momentě přečtení jejich názvu z fronty. Takto by bylo možné přidávat nové modifikace i za běhu programu, což by mohlo výrazně urychlit jejich vývoj a ladění.

Literatura

[1] *About Adobe Photoshop*. [Online; navštíveno 15.05.2019].
URL https://www.techopedia.com/definition/32364/adobe-photoshop

[2] *About Corel PHOTO-PAINT*. [Online; navštíveno 15.05.2019].
URL https://www.coreldraw.com/cz/pages/photo-paint/

[3] *About GIMP*. [Online; navštíveno 15.05.2019].
URL https://www.gimp.org/about/

[4] *Corel PHOTO-PAINT photos*. [Online; navštíveno 15.05.2019].
URL https://9to5mac.com/2019/03/12/coreldraw-graphics-suite-mac/

[5] *Interpolace a převzorkování*. [Online; navštíveno 15.05.2019].
URL http://www.fotoroman.cz/glossary/3_interpolace.htm

[6] *Lanczos resampling*. [Online; navštíveno 15.05.2019].
URL https://en.wikipedia.org/wiki/Lanczos_resampling

[7] *Mesh warping*. [Online; navštíveno 15.05.2019].
URL https://affinity.help/photo/en-US.lproj/index.html?page=pages/SizeTransform/meshWarping.html?title=Mesh%20warping

[8] *Mesh warping transformation*. [Online; navštíveno 15.05.2019].
URL https://i.stack.imgur.com/IqB90.png

[9] *Raster vs Vector*. [Online; navštíveno 15.05.2019].
URL https://vector-conversions.com/vectorizing/raster_vs_vector.html

[10] *Resampling*. [Online; navštíveno 15.05.2019].
URL https://www.ldv.ei.tum.de/fileadmin/w00bfa/www/content/_uploads/Vorlesung/_3.4/_Resampling.pdf

[11] *Sampling and Quantization*. [Online; navštíveno 15.05.2019].
URL https://sisu.ut.ee/sites/default/files/imageprocessing/files/digitizn.pdf

[12] Cornelisse, D.: *An intuitive guide to Convolutional Neural Networks*. [Online; navštíveno 15.05.2019].
URL https://medium.freecodecamp.org/an-intuitive-guide-to-convolutional-neural-networks-260c2de0a050

[13] Dove, J.: *Adobe Photoshop 2018 review*. [Online; navštíveno 15.05.2019].
URL https://www.macworld.com/article/3236722/adobe-photoshop-cc-2018-review-photo-editor-gets-into-the-ai-spirit-with-a-solid-grip-on-emerging-t.html

[14] Hlaváč, V.: *Digitální obraz, základní pojmy*. [Online; navštíveno 15.05.2019].
URL http://people.ciirc.cvut.cz/~hlavac/TeachPresCz/11DigZprObr/014DigitalImageCz.pdf?fbclid=IwAR2xxrolhFugbPFn52WKHserqZvgYWJphDDnogRPjdFJk7OtEBDYlw8-1KI

[15] Kheng, L. W.: *Image Warping and Morphing*. [Online; navštíveno 15.05.2019].
URL https://www.comp.nus.edu.sg/~cs4340/lecture/imorph.pdf

[16] Mansurov, N.: *What is Distortion?* [Online; navštíveno 15.05.2019].
URL https://photographylife.com/what-is-distortion

[17] Pagès, J.: *GIMP 2.10 Release Notes*. [Online; navštíveno 15.05.2019].
URL https://www.gimp.org/release-notes/gimp-2.10.html

[18] Paiva, A. R. C.: *Image representation, sampling and quantization*. [Online; navštíveno 15.05.2019].
URL http://www.sci.utah.edu/~arpaiva/classes/UT_ece6962/image_representation_and_discretization.pdf

[19] Phillips, D.: *Image Processing in C*. [Online; navštíveno 15.05.2019].
URL http://homepages.inf.ed.ac.uk/rbf/BOOKS/PHILLIPS/cips2ed.pdf?fbclid=IwAR0CqGUusiuKBfSKuBEMuuRMotscse1OAsvu-z4-5iBmdUkxRu8K-lA_CzQ

[20] Rybář, J.: *Škálovací algoritmy 2D obrazu*. [Online; navštíveno 15.05.2019].
URL https://is.muni.cz/th/onjsd/text.pdf?fbclid=IwAR1xWd8MddfsGXXJQh4dkpTYpPV7JHMv-qf9ZftvM-kGnJLIOEldT91bYi0

[21] Súkupová, L.: *Vzorkování signálu a aliasing*. [Online; navštíveno 15.05.2019].
URL http://www.sukupova.cz/vzorkovani-signalu-a-aliasing/