

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

Fakulta elektrotechniky
a komunikačních technologií

BAKALÁŘSKÁ PRÁCE



VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA ELEKTROTECHNIKY

A KOMUNIKAČNÍCH TECHNOLOGIÍ

FACULTY OF ELECTRICAL ENGINEERING AND COMMUNICATION

ÚSTAV TELEKOMUNIKACÍ

DEPARTMENT OF TELECOMMUNICATIONS

INTERAKTIVNÍ APLETY PRO PODPORU VÝUKY

EDUCATION-SUPPORTING INTERACTIVE APPLETS

BAKALÁŘSKÁ PRÁCE

BACHELOR'S THESIS

AUTOR PRÁCE

AUTHOR

Jakub Korbel

VEDOUCÍ PRÁCE

SUPERVISOR

doc. Mgr. Pavel Rajmic, Ph.D.

BRNO 2017



Bakalářská práce

bakalářský studijní obor **Teleinformatika**

Ústav telekomunikací

Student: Jakub Korbel

ID: 163969

Ročník: 3

Akademický rok: 2016/17

NÁZEV TÉMATU:

Interaktivní aplety pro podporu výuky

POKYNY PRO VYPRACOVÁNÍ:

Nastudujte problematiku vztahující se k apletům, které budete v rámci bakalářské práce implementovat s použitím HTML5 a JavaScriptu. Pro každý aplet navrhnete vzhled a uživatelské rozhraní a funkcionalitu. Aplety implementujte a otestujte. Aplety budou pokrývat témata gama-korekce obrazu, bitových rovin obrazu, návrhu sazebního obrazce a iterativního chování proximální gradientní metody pro minimalizaci součtu konvexních funkcí.

DOPORUČENÁ LITERATURA:

[1] Beneš, B.; Sochor, J.; Felkel, P.; Žára, J.: Moderní počítačová grafika. Computer Press, Brno, 2005.

[2] Schildt, H. : Java7, výukový kurz, Computer Press, Brno, 2012.

[3] Combettes, P. L.; Pesquet, J.-C. Proximal Splitting Methods in Signal Processing. Springer's Fixed-Point Algorithms for Inverse Problems in Science and Engineering 49. pp. 185–212. 2011

Termín zadání: 1.2.2017

Termín odevzdání: 8.6.2017

Vedoucí práce: doc. Mgr. Pavel Rajmic, Ph.D.

Konzultant:

doc. Ing. Jiří Mišurec, CSc.
předseda oborové rady

UPOZORNĚNÍ:

Autor bakalářské práce nesmí při vytváření bakalářské práce porušit autorská práva třetích osob, zejména nesmí zasahovat nedovoleným způsobem do cizích autorských práv osobnostních a musí si být plně vědom následků porušení ustanovení § 11 a následujících autorského zákona č. 121/2000 Sb., včetně možných trestněprávních důsledků vyplývajících z ustanovení části druhé, hlavy VI. díl 4 Trestního zákoníku č. 40/2009 Sb.

ABSTRAKT

Tato bakalářská práce je zaměřena na podporu výuky v oblasti zpracování obrazu. Teoretická část se zabývá matematickým pohledem na danou problematiku. Praktická část je zaměřena na vypracování čtyř webových apletů (nástrojů pro podporu výuky) - aplet Gama korekce, Sazebního obrazce, Bitových rovin a Numerické minimalizace konvexních funkcí.

KLÍČOVÁ SLOVA

Aplet, HTML5, JavaScript, gama, bitové roviny, obrazec, minimalizace.

ABSTRACT

This bachelor theses is focused to the support of education in branch of image processing. The teoretic part deals with mathematics view to this problematic. The pratical part is focused to creating of web applets (tools for supporting of education) - applets Gamma correction, Rate shape, Bit plane, Numeric minimization convex function.

KEYWORDS

Applet, HTML5, JavaScript, gamma, bit plane, shape, minimizing.

KORBEL, Jakub *Interaktivní aplety pro podporu výuky*: bakalářská práce. Brno: Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií, Ústav telekomunikací, 2017. 46 s. Vedoucí bakalářské práce doc. Mgr. Pavel Rajmic, Ph.D.

PROHLÁŠENÍ

Prohlašuji, že svou bakalářskou práci na téma „Interaktivní aplety pro podporu výuky“ jsem vypracoval samostatně pod vedením vedoucího bakalářské práce a s použitím odborné literatury a dalších informačních zdrojů, které jsou všechny citovány v práci a uvedeny v seznamu literatury na konci práce.

Jako autor uvedené bakalářské práce dále prohlašuji, že v souvislosti s vytvořením této bakalářské práce jsem neporušil autorská práva třetích osob, zejména jsem nezasáhl nedovoleným způsobem do cizích autorských práv osobnostních a/nebo majetkových a jsem si plně vědom následků porušení ustanovení § 11 a následujících autorského zákona č. 121/2000 Sb., o právu autorském, o právech souvisejících s právem autorským a o změně některých zákonů (autorský zákon), ve znění pozdějších předpisů, včetně možných trestněprávních důsledků vyplývajících z ustanovení části druhé, hlavy VI. díl 4 Trestního zákoníku č. 40/2009 Sb.

Brno

.....

podpis autora

PODĚKOVÁNÍ

Rád bych poděkoval vedoucímu bakalářské práce panu doc. Mgr. Pavlovi Rajmicovi, Ph.D. za odborné vedení, konzultace a podnětné návrhy k práci. Dále bych chtěl poděkovat svým rodičům a sourozencům za neustálou podporu při studiu a trpělivost při psaní samotné bakalářské práce.

Brno

.....

podpis autora

PODĚKOVÁNÍ

Výzkum popsáný v této bakalářské práci byl realizován v laboratořích podpořených z projektu SIX; registrační číslo CZ.1.05/2.1.00/03.0072, operační program Výzkum a vývoj pro inovace.

Brno

.....
podpis autora

OBSAH

1	Úvod	10
2	Teoretická část	11
2.1	Graf funkce	11
2.1.1	Vlastnosti funkcí	11
2.1.2	Druhy elementárních funkcí	12
2.2	Rozbor bitmapové fotografie	14
2.3	Rozklad obrazu na bitové roviny	14
2.3.1	Klasický bitový kód	14
2.3.2	Grayův kód	15
2.4	Histogram	19
2.4.1	Čtení histogramu	19
2.5	Sazební obrazec	22
2.5.1	Van de Graafovo úhlopříčné pravidlo	22
2.5.2	Pravidlo s kružnicí	23
2.5.3	Sazební obrazec pomocí empirických pravidel	23
2.5.4	Hrubý a čistý formát	24
2.6	Gama korekce	24
2.7	Proximální gradientní metoda	26
2.7.1	Proximální operátor	26
2.7.2	Proximální gradientní metoda (dopředně-zpětného dělení)	26
2.8	Popis programovacího jazyka a jeho výhody	28
2.8.1	HTML5	28
2.8.2	JavaScript	29
2.8.3	CSS	30
2.8.4	Výběr jazyka	31
2.8.5	Vývojové prostředí Bluefish	32
3	Praktická část	33
3.1	Aplet bitových rovin	33
3.2	Aplet sazebního obrazce	35
3.3	Aplet gama korekce	37
3.4	Numerická minimalizace konvexních funkcí	39
4	Závěr	42
	Literatura	43

SEZNAM OBRÁZKŮ

2.1	Graf konvexní a konkávní funkce	12
2.2	Graf lineární funkce	12
2.3	Graf mocninné funkce	13
2.4	Graf exponenciální funkce	13
2.5	Graf logaritmické funkce	14
2.6	Obrázek přehledu bitových rovin	15
2.7	Rozkladu digitálního obrazu na bitové roviny	16
2.8	Ukázka rozkladu digitálního obrazu na bitové roviny pomocí Grayova kódu	17
2.9	Ukázka histogramu	19
2.10	Jasný obraz	20
2.11	Středotónový obraz	21
2.12	Tmavý obraz	21
2.13	Van de Grafovo úhlopříčné pravidlo	22
2.14	Náčrt konstrukce sazebního obrazce pomocí pravidla s kružnicí	23
2.15	Ukázka gama korekce na změnu fotografie	25
2.16	Šedotónová stupnice kvantování jasu	25
2.17	Graf numerické minimalizace	27
2.18	Vložení kontextů plátna do stránky pomocí Canvas	28
2.19	Ukázka programovacího jazyka JavaScript	30
2.20	Obrázek CSS kódu přímého stylu	30
2.21	Obrázek CSS kódu pomocí stylopisu	31
2.22	Obrázek CSS kódu	31
2.23	Vývojové prostředí Bluefish	32
3.1	Aplet bitových rovin	34
3.2	Aplet sazebního obrazce	36
3.3	Aplet gama korekce	38
3.4	Výsledný graf apletu	40
3.5	Aplet numerické minimalizace	41

1 ÚVOD

V této interaktivní době jsou webové nástroje ideální pomůckou pro podporu výuky, rozšiřují možnosti učitelů a umožňují studentům praktické osvojení dané problematiky. Napomáhají k lepšímu pochopení učiva zvyšují motivaci a přispívají k celkové aktivitě při studiu. Student si prakticky vyzkouší danou problematiku probírané látky, což je základní charakteristikou interaktivních nástrojů.

Tímto způsobem získává student lepší celkový pohled na věc, než jen samotnou teorii z knih. Díky této technice, dokáže vyučující snáze získat pozornost. Tento styl výuky je velice úspěšný i pro studenty s mentálním postižením či jinou specifickou poruchou učení.[13]

Interaktivní nástroje mají do budoucna velké uplatnění na všech stupních výuky a ve většině předmětů. Tento přístup má stále ještě velký potenciál k rozšíření. I nadále se budeme stále více setkávat s interaktivními nástroji a to i mimo samotné vzdělávání.

Tato bakalářská práce se zaměřuje na vytvoření čtyř webových interaktivních, nástrojů, které mají sloužit k výuce předmětů zaměřených na zpracování grafiky a obrazu.

První webový nástroj se zabývá problematikou úpravy gama-korekce a barev RGB fotografií, kde je pro větší přehled úprav na fotografe histogram a graf korekce. Druhý se zabývá bitovou rovinou obrazu, kde monochromatický obrázek rozkládáme a skládáme do bitových rovin pomocí klasického bitového kódu nebo Grayova kódu. Třetí zobrazuje konstrukci sazebního obrazce pomocí *Van de Graafova úhlopříčného pravidla* a dalších pravidel, například pravidlo s kružnicí a empirických pravidel. Dále vypočítává využití a okraje stránky v různých formátech. Čtvrtý aplet vykresluje a počítá numerickou minimalizaci konvexních funkcí pomocí proximální gradientní metody.

2 TEORETICKÁ ČÁST

2.1 Graf funkce

2.1.1 Vlastnosti funkcí

Reálná funkce f s jednou proměnnou x je taková binární relace z množiny R do množiny R , kde pro každé $x \in R$ existuje nejvýše jedno $y \in R$, pro které $[x, y] \in f$. Množina všech x , pro které je pouze jedno y , se nazývá definičním oborem funkce f a značí se D_f . Všechny množiny $y = f(x)$, kde $x \in D_f$, se nazývá oborem hodnot funkce f a značíme jej H_f .

Vlastnosti funkce:

- rostoucí je pro všechna $x_1, x_2 \in R : x_1 < x_2 \Rightarrow f(x_1) < f(x_2)$,
- klesající je pro všechna $x_1, x_2 \in R : x_1 < x_2 \Rightarrow f(x_1) > f(x_2)$,
- neklesající je pro všechna $x_1, x_2 \in R : x_1 < x_2 \Rightarrow f(x_1) \leq f(x_2)$,
- nerostoucí je pro všechna $x_1, x_2 \in R : x_1 < x_2 \Rightarrow f(x_1) \geq f(x_2)$,
- prostá je pro všechna $x_1, x_2 \in R : x_1 \neq x_2 \Rightarrow f(x_1) \neq f(x_2)$. [2]

Pokud se jedná o funkci f , která je prostá na svém definičním oboru, pak k ní existuje inverzní funkce f^{-1} . V tomto případě platí vztah $D_{f^{-1}} = H_f, H_{f^{-1}} = D_f$. Grafy funkcí f a f^{-1} jsou navzájem souměrné dle přímky $x = y$. [2]

Funkce f , pro kterou platí, že x je prvkem $D_f \Leftrightarrow (-x) \in D_f$, se nazývá:

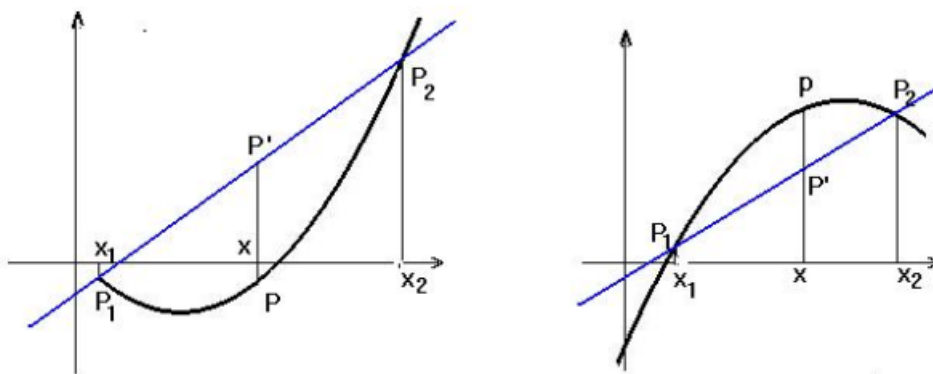
- sudá pro všechna $x \in D_f : f(-x) = f(x)$,
- lichá pro všechna $x \in D_f : f(-x) = -f(x)$. [2]

Další vlastností funkce je konvexnost a konkávnost. Tyto vlastnosti můžeme vidět na obrázku 2.1. Funkce f , která se nazývá konvexní resp. konkávní, definovaná jako $J \subset R$ má tuto vlastnost: Jsou-li $x_1, x, x_2 \in J$ libovolné tři body takové, že $x_1 < x < x_2$, pak bod $P = [x, f(x)]$ leží buď nad, respektive pod přímkou P_1P_2 , kde $P_1 = [x_1, f(x_1)], P_2 = [x_2, f(x_2)]$, jestliže funkce f je spojitá na celém definičním oboru J platí:

- Funkce f je konvexní, když funkce f' roste.
- Funkce f je konkávní, když funkce f' klesá. [3]

Dále platí nerovnost pro libovolnou $\lambda \in [0, 1]$:

- $f(\lambda x_1 + (1 - \lambda)x_2) \leq \lambda f(x_1) + (1 - \lambda)f(x_2)$ pro konvexní funkci,
- $f(\lambda x_1 + (1 - \lambda)x_2) \geq \lambda f(x_1) + (1 - \lambda)f(x_2)$ pro konkávní funkci. [7]



Obr. 2.1: Graf konvexní funkce vlevo, graf konkávní funkce vpravo.[3]

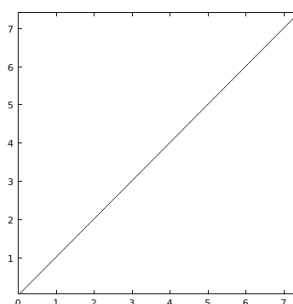
2.1.2 Druhy elementárních funkcí

Základní vlastnosti a přehled elementárních funkcí a jejich reálné proměnné:

- **Lineární funkce** je dána předpisem

$$y = kx + q; \quad k, q \in R.[3] \quad (2.1)$$

Pro $k > 0$ platí, že funkce je rostoucí. Pro $k < 0$ je funkce klesající a pro $k = 0$ je funkce konstantní. Graf lineární funkce je zobrazen na obrázku 2.2.[3]

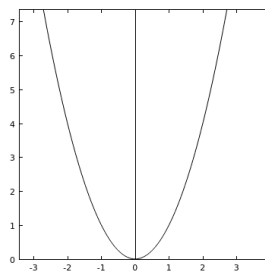


Obr. 2.2: Graf lineární funkce.

- **Mocninná funkce** předpisem

$$f(x) = x^a.[3] \quad (2.2)$$

Jedná se o funkci s přirozeným exponentem, kde $a \in N$ a všechny $x \in R$. Je-li a sudé číslo, je funkce klesající na intervalu $(-\infty, 0)$ a rostoucí v intervalu $(0, \infty)$. Funkce je v celém definičním oboru konvexní viz obr. 2.3. Je-li a liché číslo, je funkce rostoucí v celém definičním oboru. Na intervalu $(-\infty, 0)$ je funkce konkávní a na intervalu $(0, \infty)$ se jedná o funkci konvexní.[3]

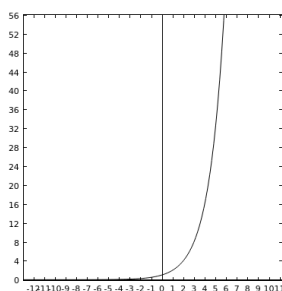


Obr. 2.3: Graf mocninné funkce, hodnota mocniny je číslo 2.

- **Exponenciální funkce** je dána předpisem

$$f(x) = a^x, a > 0. [3] \quad (2.3)$$

Funkce je rostoucí, když $a > 1$ a klesající pro $0 < a < 1$. Funkce je konstantní pro $a = 1$. Graf exponenciální funkce pro $a = 2$ je na obrázku 2.4.[3]



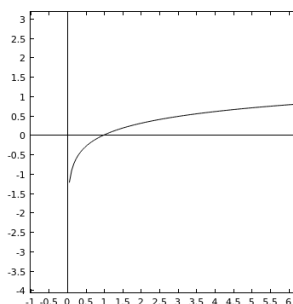
Obr. 2.4: Graf exponenciální funkce o hodnotě 2.

- **Logaritmická funkce** při základu a , kde $0 < a < 1$ nebo $a > 1$ je definována na intervalu $(0, \infty)$. Logaritmická funkce je inverzní funkcí k exponenciální funkci $f(x) = a^x$. [3]

Funkce je dána předpisem:

$$f(x) = \log_a x. [3] \quad (2.4)$$

Funkce je rostoucí pro $a > 1$ a klesající pro $0 < a < 1$.



Obr. 2.5: Graf logaritmické funkce.

2.2 Rozbor bitmapové fotografie

Digitální fotografie vzniká při dopadu fotonů určitého spektrálního rozsahu, který dopadá na digitální snímač fotoaparátu o určité ploše a počtu buněk (měřících bodů). Každá buňka pomocí barevného filtru propustí jen určitou barvu. Foton předá energii každé buňce, a tím vzniknou různé náboje. Tyto náboje jsou speciálním zesilovačem zesíleny a napětí je pomocí A/D převodníku vyjádřeno jako posloupnost bitů, které se ukládají do bodů (pixelů) mřížky. Na tomto principu vzniká fotografie. Digitálních snímačů existuje celá řada např. CDD, CMOS, Foveon X3, 3CDD, atd.

Fotografie má určitou velikost, která má danou výšku a šířku na základě počtu pixelů. Barevná hloubka slouží k popisu barvy RGB v jednom pixelu. Čím více bitů je v pixelu, tím větší množství barev se vyskytuje na fotografii.[6]

2.3 Rozklad obrazu na bitové roviny

2.3.1 Klasický bitový kód

Každý monochromatický digitální obraz se skládá z pixelů (obrazový bod) s N bity s určitou intenzitou, která nabývá hodnot $<0,255>$ resp. 8 bitů, z které vytvoříme celkem 8 bitových rovin obrazu. To lze tedy vyjádřit pomocí posloupnosti:

$$p_o = p_0 2^0 + p_1 2^1 + p_2 2^2 + \dots + p_{N-1} 2^{N-1}. [16] \quad (2.5)$$

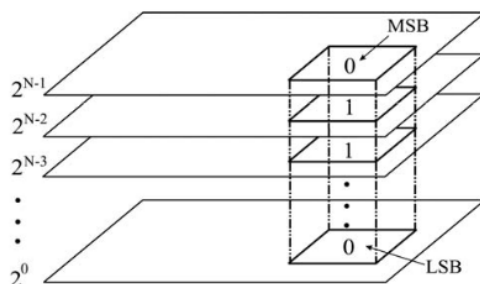
Jak vidíme, tak jednotlivé bity p_i mají určitou váhu, která má velikost:

$$\{i = 0, 1, \dots, N - 1\}. [16] \quad (2.6)$$

Z celého monochromatického digitálního obrazu se z každého pixelu vyberou jen bity s určitou váhou p_o . Nejnižší bitová rovina bude mít váhu 0, též nazvaná zkratkou LSB

(Least Significant Bit). LBS má nejnižší význam na výsledném monochromatickém digitálním obrazu. Významnost váhy v jednotlivých bitových rovinách obrazu roste. Poslední 7. bitová rovina je nejvýznamnější, protože má největší vliv ve výsledném digitálním obrazu. Tato rovina se nazývá zkratkou MSB (Most Significant Bit). Obrázek 2.6 ukazuje přehled vrstev bitových rovin v monochromatickém obrazu. Klasický binární kód má velké bitové změny, tyto změny můžeme vidět v tabulce 2.1, kde v prvním sloupci můžeme vidět bitový kód a ve druhém sloupci jeho bitovou změnu. [16]

Obrázek 2.7 ukazuje klasický rozklad digitálního obrazu na bitové roviny.



Obr. 2.6: Obrázek přehledu bitových rovin. [16]

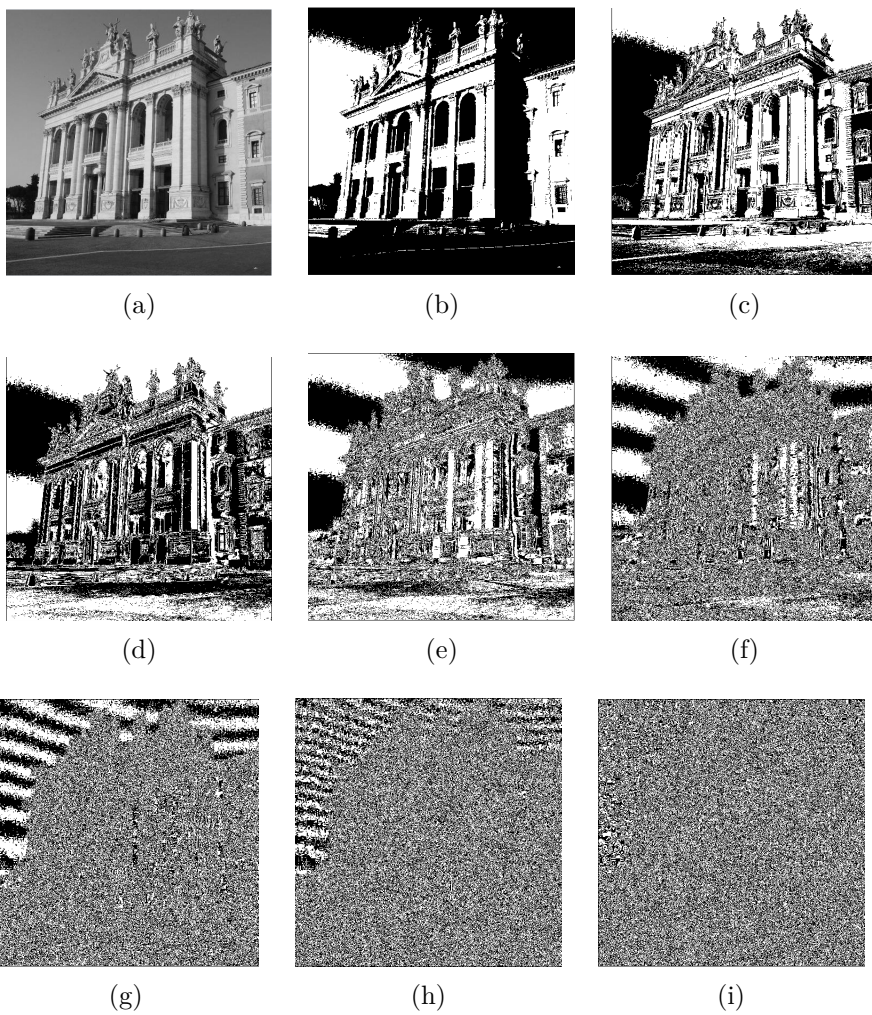
2.3.2 Grayův kód

Zrcadlový binární kód, též nazývaný Grayův kód podle Franka Graye, je binární číselná soustava. Vždy dvě po sobě jdoucí číselné hodnoty se liší pouze v jednom bitu. Při změně u standardního binárního kódu po sobě jdoucími číselnými stavy se mnohdy změní hodnota více než v jednom bitu. Například mezi sedmičkou a osmičkou se změní hodnota 4 bitů současně. Výpočet spočívá v ponechání dvojkové číslice přímého binárního kódu s nejvyšší vahou. V každém kroku se dvojková číslice přímého binárního kódu invertuje před jedničkou vyšší váhy.

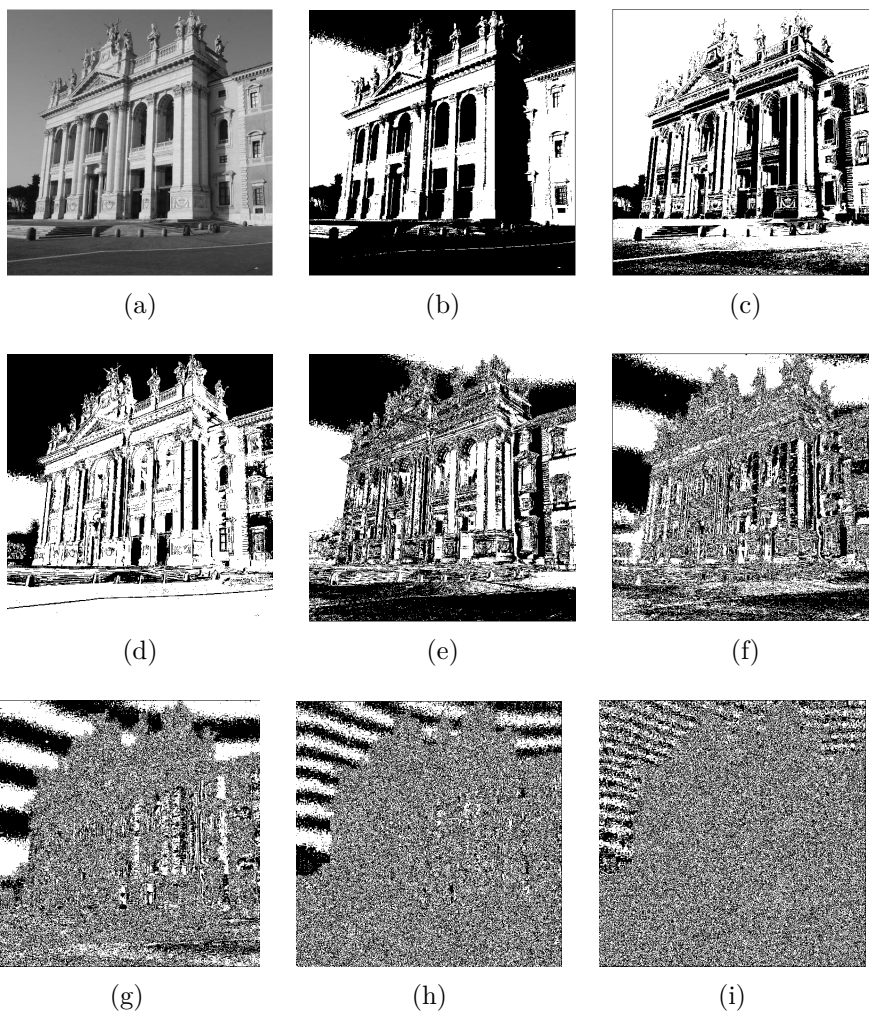
Jednotlivé hodnoty Grayova kódu vidíme v tabulce číslo 2.1 ve čtvrtém sloupci a jejich následná změna je uvedena v pátém sloupci. Grayův kód byl původně vytvořen, aby sloužil pro vyloučení vzniku chybné hodnoty při přechodu mezi dvěma stavy výstupu z elektromechanických prepínačů. Jelikož v reálném systému nikdy nelze současně zaručit změnu více logických hodnot, přečtení a úplné vyhodnocení.

Dnes je Grayův kód používán k usnadnění opravy chyb v digitální komunikaci, například v digitální pozemní či digitální kabelové televizi. Grayův kód se používá i v digitálních modulačních schemech například QAM.[17] [18]

Obrázek 2.8 ukazuje rozklad digitálního obrazu na bitové roviny pomocí Grayova kódu.



Obr. 2.7: Ukázka rozkladu digitálního obrazu na bitové roviny pomocí klasického bitového kódu. a) původní obrázek, b) 0. váha též LSB, c) 1. váha, d) 2. váha, e) 3. váha, f) 4. váha, g) 5. váha, h) 6. váha, i) 7. váha též MSB.



Obr. 2.8: Ukázka rozkladu digitálního obrazu na bitové roviny pomocí Grayova kódu. a) původní obrázek, b) 0. váha též LSB, c) 1. váha, d) 2. váha, e) 3. váha, f) 4. váha, g) 5. váha, h) 6. váha, i) 7. váha též MSB.

Tab. 2.1: Tabulka bitového kódu, Grayova kódu a jejich bitových změn.

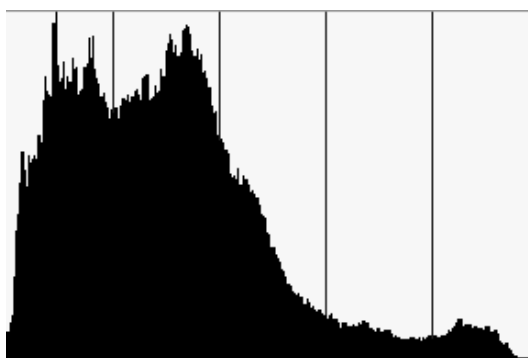
Dekadické číslo	Binární kód	bitová změna	Grayův kód	bitová změna
0	0000	-	0000	-
1	0001	1	0001	1
2	0010	2	0011	1
3	0011	1	0010	1
4	0100	3	0110	1
5	0101	1	0111	1
6	0110	2	0101	1
7	0111	1	0100	1
8	1000	4	1100	1
9	1001	1	1101	1
10	1010	2	1111	1
11	1011	1	1110	1
12	1100	3	1010	1
13	1101	1	1011	1
14	1110	2	1001	1
15	1111	1	1000	1

2.4 Histogram

Statistická technika zhuštěné reprezentace rozložení dat je nazývána jako histogram. Pro vytvoření histogramu jsou vstupní data dělena podle své hodnoty do definovaných buněk s daným rozsahem. Histogram je soubor čísel, které vyjadřují četnosti vstupních dat v jednotlivých buňkách. Histogram často bývá vykreslen ve formě sloupcového grafu. Vztah hodnoty H histogramu s indexem i odpovídá určité barvě šedi. Pro charakterizaci obrazu je histogram zásadní pomůckou ke zjištění množství a frekvence jednotlivých barev obsažených v obraze. Histogram nám tedy říká, kolik pixelů v obraze má určitou intenzitu a s jakou pravděpodobností je zastoupen. V žádném případě nám histogram nenese informaci o poloze daného pixelu v mřížce. Jestliže je obraz jen z jasových složek, histogram bude jednorozměrný vektor. Jestliže se bude vyskytovat z více složek jako je například RGB, bude histogram složen ze tří vektorů. Histogram RGB může být zobrazen hned několika způsoby:

- zobrazení všech barevných kanálů zvlášť,
- zobrazení všech barevných kanálů do jednoho složeného histogramu pomocí průhledností,
- zobrazení pouze jasu obrazu,
- zobrazení průměru všech histogramů,
- zobrazení vícerozměrného histogramu.[6][1]

2.4.1 Čtení histogramu



Obr. 2.9: Ukázka histogramu.

V ukázce obrázku 2.9 je popsáno základní čtení histogramu. Vlevo, u nulové hodnoty vodorovné osy, se nachází absolutní černá barva. S rostoucí hodnotou na vodorovné ose roste intenzita až do maxima, což odpovídá bílé barvě. Svislá osa nám udává četnost pixelů.

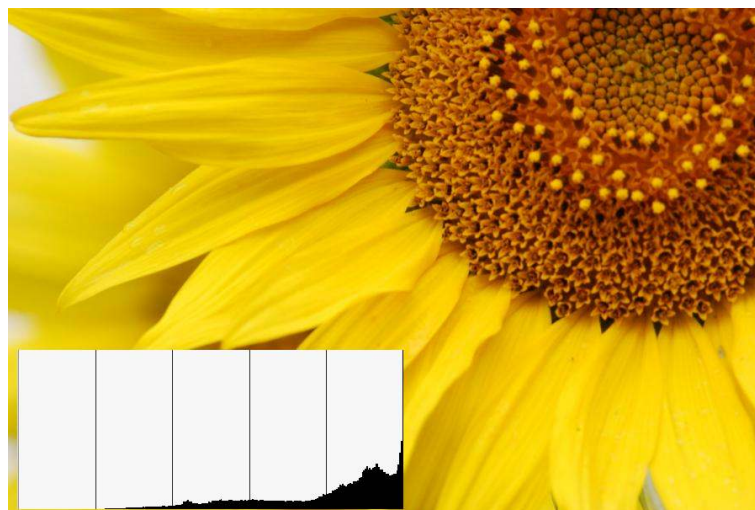
Histogram může být rozdělen podle umístění na tři stejně velké části:

- **stíny** (*shades*) – nachází se v dolní oblasti histogramu,
- **střední tóny** (*mid-tones*) – nachází se ve střední oblasti histogramu,
- **světla** (*lights*) – jsou v nejvyšší a nejjasnější oblasti histogramu.

Pomocí těchto oblastí rozlišujeme čtyři základní druhy obrazců:

- **jasný obraz** – má většinu barev ve světlé oblasti (*high-key*),
- **tmavý obraz** – je opakem jasného obrazu (*low-key*),
- **středotónový obraz** – má většinu barev kolem střední oblasti (*mid-key*), též se nazývá obraz s nízkým kontrastem,
- **bimodální obraz** – má vysoký kontrast, jelikož má ostré vrcholy na pravé a levé straně histogramu.[1]

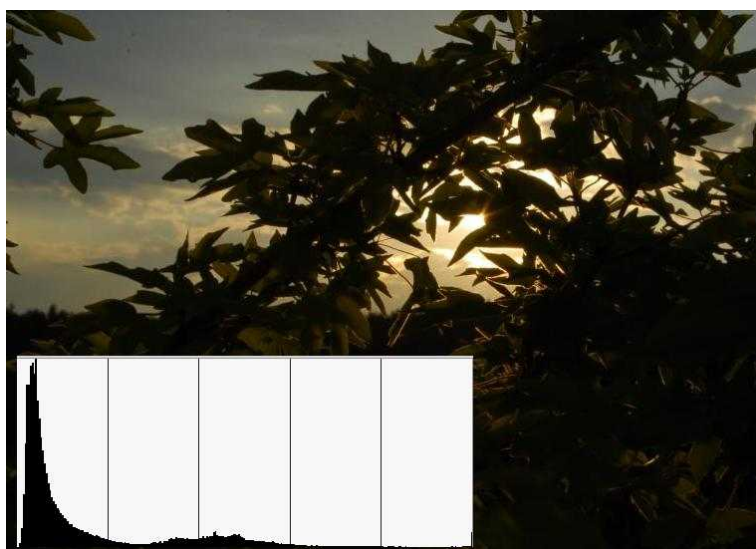
Na obrázku 2.10 lze vidět jasný histogram, u kterého je většina hodnot ve světlé oblasti. Obrázek 2.11 zobrazuje většinu hodnot na histogramu ve střední oblasti. Na následujícím obrázku 2.12 je většina hodnot v tmavé oblasti.



Obr. 2.10: Jasný obraz.



Obr. 2.11: Středotónový obraz.



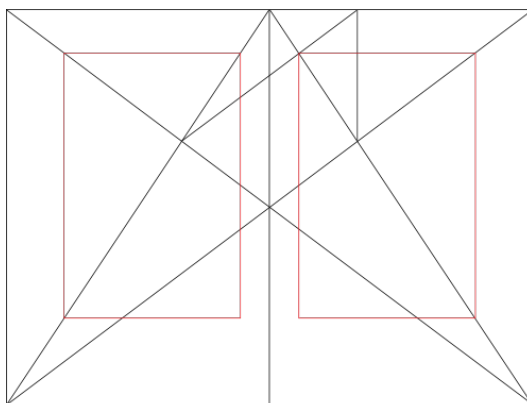
Obr. 2.12: Tmavý obraz.

2.5 Sazební obrazec

Sazební obrazec je nákres pro rozložení textu, obrázků a okrajů na čisté ploše stránky. Pro každý typ a účel tiskoviny se sazební obrazec liší. Například noviny mají jiný sazební obrazec než básnická sbírka. Stanovení správného poměru textu a bílé plochy je náročným úkolem. Je potřeba vzít v úvahu rozsah publikace a její rozměr včetně velikosti písma a účelu. Bohužel v dnešní době se z ekonomických důvodů upouští od správného sazebního obrazce a standardně se volí 2,5 cm od hrany stránky.

- **Jednostránková publikace** – pod tímto pojmem máme na mysli dokument tištěný pouze z jedné strany listu a z druhé strany zůstává čistý. Jednostránkovou publikaci poznáme tak, že se vlevo nachází čistá stránka a vpravo dokument.
- **Dvojstránková publikace** – dokument je tištěn na obě stránky listu. Tento typ publikace poznáme tak, že vpravo i vlevo se nachází vytištěný dokument. Tento dokument má většinou zrcadlové uspořádání.

Postupů pro usazení obrazce na stránku je mnoho a to hlavně díky historickým zkušenostem. Nejuniverzálnějším a nejjednodušším způsobem rozvržení stránky je *Van de Graafovo úhlopříčné pravidlo* viz obrázek 2.13.



Obr. 2.13: Van de Graafovo úhlopříčné pravidlo. [6]

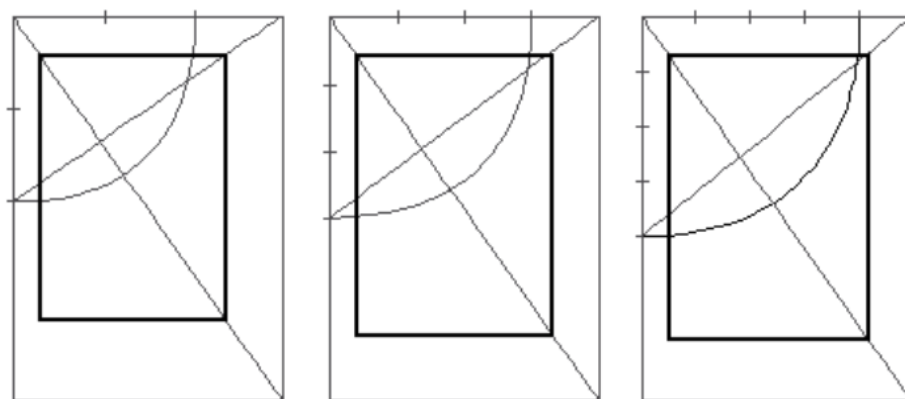
2.5.1 Van de Graafovo úhlopříčné pravidlo

Van de Graafovo úhlopříčné pravidlo s výsledky je poměr okrajů – vnitřní : horní : vnější : dolní $\Rightarrow 1 : r : 2 : 2r$, kde r představuje poměr výšky : šířky stránky. Například u formátu stránky A4 jsou okraje v poměru $1 : \sqrt{2} : 2 : 2\sqrt{2}$. Vnitřní (horní) okraj se rovná $1/9$ šířky nebo výšky stránky, vnější (dolní) okraj je roven

2/9 šířky (výšky). Z konstrukce je zřejmé, že obrazec je určen jen rozměry stránky a dá se spočítat, že zabírá 44 % plochy stránky. [6]

2.5.2 Pravidlo s kružnicí

Konstrukce sazebního obrazce pomocí pravidla s kružnicí má výsledný poměr $1 : \sqrt{2} : \frac{\sqrt{2}}{R} : \frac{2}{R}$, kde $R = \frac{x-1}{x}$ a to stránky s poměrem šířky : šířce okolo $1 : \sqrt{2}$. Toto pravidlo má dva stupně volnosti v konstrukci, které si může zvolit každý sám dle libosti. Tvorba sazebního obrazce vznikne tak, že užší část stránky rozdělíme na x dílů (první stupeň volnost). Dále $x - 1$ na šířce stránky přeneseme po kružnici na hřbetní stránku. Z tohoto bodu se vede úhlopříčka na protější horní roh stránky. Dále narýsujeme úhlopříčku z levého horního rohu na spodní pravý roh. Nad průsečíkem těchto úhlopříček se vynese požadovaná šíře sazby (druhý stupeň volnosti), podle které se stanoví výška obrazce. Častá volba druhého stupně volnosti bývá $x = 3, 4, 5$. Náčrt konstrukce můžeme vidět na obrázku 2.14. [6]



Obr. 2.14: Náčrt konstrukce sazebního obrazce pomocí pravidla s kružnicí. [6]

2.5.3 Sazební obrazec pomocí empirických pravidel

V případech, kdy formát stránky je v jiném poměru než $1 : \sqrt{2}$, lze obecně použít empirická pravidla pro poměry okrajů.

- První empirické pravidlo je v poměru jednotlivých okrajů levý : horní : pravý : dolní = 3 : 4 : 5 : 7 .
- Druhým empirickým pravidlem je v poměru 2 : 3 : 4 : 6. [6]

Dalším pravidlem je úhlopříčné pravidlo 1/12.

2.5.4 Hrubý a čistý formát

Formáty papíru jsou jasně dané šířkou papírenských strojů, které tyto formáty vyrábějí. Tyto formáty se též nazývají hrubé formáty. Formátová soustava je po celém světě stejná z důvodu návaznosti na formáty tiskových a knihařských strojů. Tyto formáty mívají o 5% větší šířku a výšku stránky než konečné čisté formáty oříznutých polygrafických výrobků, které známe z řady A a B. Čistým formátem je tedy formát hotové tiskoviny.[8]

Tiskovina rovněž může být ve vazbě např. brožované, pevné, kroužkové, jednostranné, dvoustranné. K vazbě potřebuje určitou hřbetní mezeru, která změní rozměry výsledných okrajů a tím i výsledný sazební obrazec. [6]

2.6 Gama korekce

Gama korekce je prakticky velmi zjednodušená změna lineárního vstupu na exponenciální výstup.

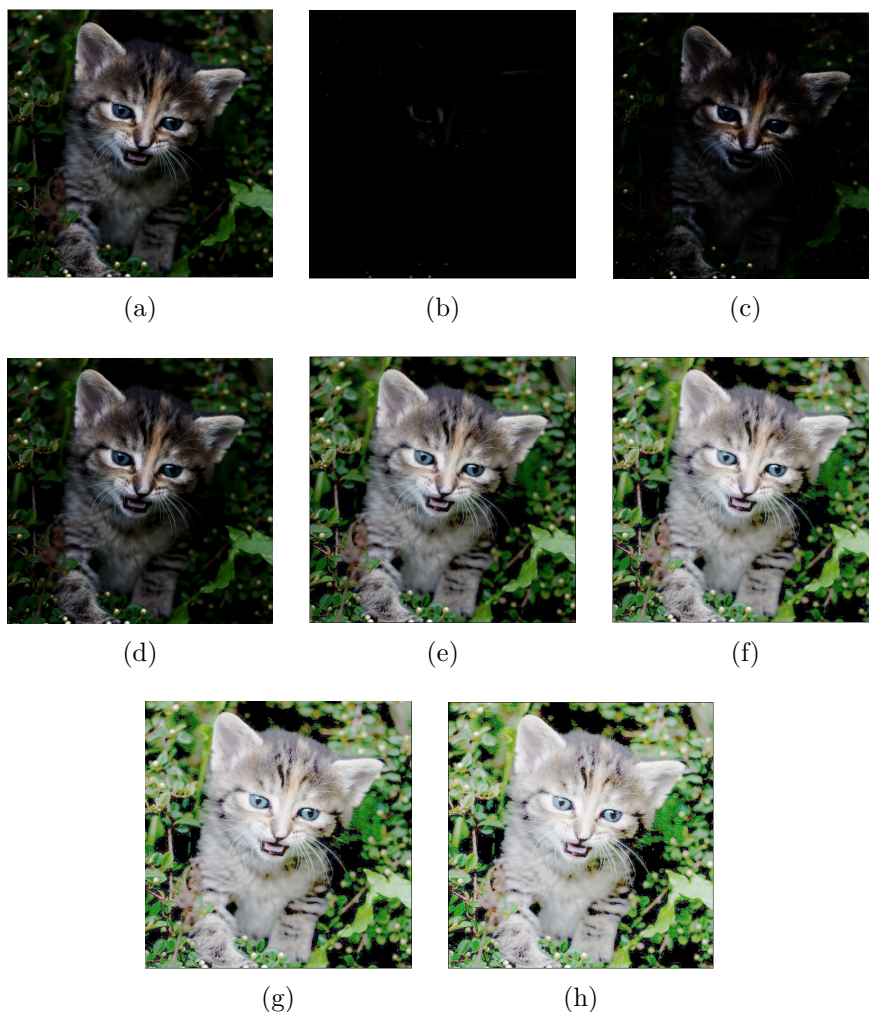
$$Y_{\text{výstup}} = Y_{\text{vstup}}^{\gamma} [6] \quad (2.7)$$

Tato korekce se využívá pro úpravu jasu a barev. Gama korekce má parametr γ , který se rovná exponentu vstupu. Gama korekce u klasických CRT monitorů se pohybuje v rozmezí $2,5 \pm 0,3$. Tato hodnota zajišťuje lepší linearitu pro vnímání obrazu lidským okem, který má logaritmickou charakteristiku. Gama korekce zajišťuje rovnoměrný přechod jasu (světých a tmavých míst na monitoru). Kdyby jas nebyl rozvržen rovnoměrně, došlo by k velkým skokům tmavých odstínů a ve světých by byly příliš malé rozdíly sousedních hodnot. To můžeme vidět na obrázku 2.16.[6]

Gama korekce se používá u CRT obrazovek, ale i k opravě jasu při špatné expozici fotografie. Tuto změnu můžeme vidět na fotografiích 2.15, kde Gama korekce nemění jen nejtmavější a nejsvětější bod fotografie, ale mění celé rozložení jasu ve fotografii. Ve výsledku vzbuzuje dojem zesvětlování nebo ztmavování výsledné fotografie. Jelikož gama korekce nemění krajní body (bílé a černé), nedochází ke ztrátě kresby nebo přepalům bílé ve fotografii.[21][6]

Gama korekce má velký vliv na barvy ve fotografii. Když snímek zesvětlíme pomocí gama korekce, tak se barvě sníží relativní sytost všech barev. Naopak, pokud fotografii ztmavíme, zvýšíme relativní sytost všech barev ve fotografii.[21]

Řada fotoeditorů v dnešní době umí měnit gama korekci ve snímku. Například v fotoeditoru Gimp k tomu slouží nástroj křivka, která umožňuje libovolně měnit tvar gama křivky.[21]



Obr. 2.15: Ukázka gama korekce na změnu fotografie. a) původní fotografie, b) fotografie s gama korekcí 0,1, c) fotografie s gama korekcí 0,5, d) fotografie s gama korekcí 1, e) fotografie s gama korekcí 2, f) fotografie s gama korekcí 3, h) fotografie s gama korekcí 4, i) fotografie s gama korekcí 5.

Logaritmické kvantování	0.0	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9	1.0
Lineární kvantování	0.0	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9	1.0

Obr. 2.16: Šedotónová stupnice kvantování jasů.

2.7 Proximální gradientní metoda

2.7.1 Proximální operátor

Proximální operátor konvexní funkce je přirozeným projekčním operátorem na konvexní množinu. Tento přirozený projekční operátor nalézá bod v konvexní množině, který leží nejbližší výchozímu bodu. Operátor je velmi důležitý v analýze a v numerickém řešení konvexních optimalizačních problémů.[4][5]

Definice:

- Je-li funkce $f(x)$ zdola polospojité na podmnožině I metrického prostoru a platí, že pro libovolné $x_0 \in I$ platí tato nerovnost:

$$\liminf_{x \rightarrow x_0} f(x) \geq f(x_0). [5] \quad (2.8)$$

- Funkce $f : R^n \rightarrow (-\infty, \infty)$ je konvexní funkce zdola spojitá s neprázdným definičním oborem a pro každé $x \in R^n$ má minimalizační problém:

$$\min_{y \in R^n} \frac{1}{2} \|x - y\|_2^2 + f(y), \quad (2.9)$$

má právě jedno řešení. Bod minima účelové funkce označujeme $\text{prox}_f(x)$. Tento operátor prox_f ležící $R^n \rightarrow R^n$ nazýváme proximální operátor f . [5]

Vlastnosti proximálního operátoru

Je-li φ zdola polospojité konvexní funkce, která má neprázdný definiční obor $\varphi : R^n \rightarrow (-\infty, \infty)$ platí následující:

- **Posunutí**- Je-li $\psi = \varphi(\cdot - z)$, kde $z \in R^n$. Potom $\text{prox}_\psi(x) = z + \text{prox}_\varphi(x - z)$.
- **Změna měřítka** - Je-li $\psi = \varphi(\frac{\cdot}{\rho})$, kde $\rho \in R \setminus \{0\}$. Potom $\text{prox}_\psi(x) = \rho \text{prox}_{\frac{\varphi}{\rho^2}}(\frac{x}{\rho})$. [5]

2.7.2 Proximální gradientní metoda (dopředně-zpětného dělení)

Proximální gradientní metoda má mnoho různých algoritmů, pro aplikaci v této bakalářské práci se omezíme na algoritmus tzv. dopředně-zpětného dělení pro výpočty numerické minimalizace konvexních funkcí, protože se jedná o úlohy, kde je jedna z funkcí hladká a také kvadratického datového členu.[4]

Obecný vztah konvexní úlohy:

$$\arg_x \min g(x) + h(x). [5] \quad (2.10)$$

Zatímco h je v $R^n \rightarrow R$ zdola polospojité konvexní funkce a současně g je v $R^n \rightarrow R$ konvexní funkce, která je diferencovatelná s β -lipschizovským spojitým gradientem ∇g . Pro každé $[x,y] \in R^n \rightarrow R$ platí vztah

$$\| \nabla g(x) - \nabla g(y) \| \leq \beta \| (x - y) \|, [4] \quad (2.11)$$

kde $\beta \in (0, \infty)$. Pokud $g + h$ je koercivní má alespoň jedno řešení, tj.

$$\lim_{\|x\| \rightarrow \infty} g(x) + h(x) = \infty. [5] \quad (2.12)$$

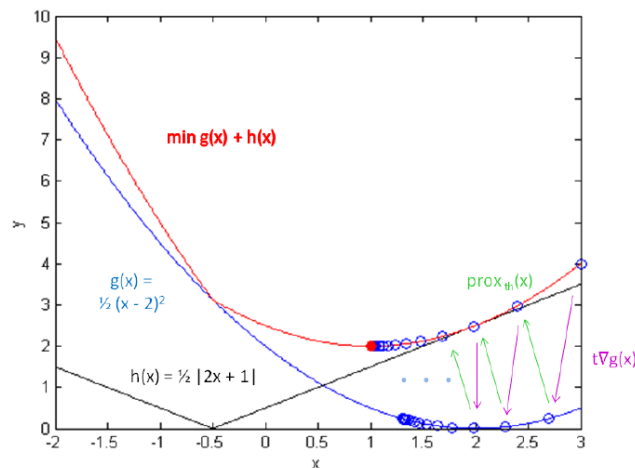
A jestliže $g + h$ je ryze konvexní, je řešení pouze jediné. Když je $x \in R^n$, pak x je řešením úlohy $\Leftrightarrow x = \text{prox}_{t^k h}(x - t^k \times \nabla g(x))$ platí pro všechna $t \in (0, \infty)$. Charakterizace pomocí pevného bodu implikuje možnost numerického řešení iterativně a to:

$$x^{k+1} = \text{prox}_{t^k h}(x^k - t^k \times \nabla g(x^k)) [4] \quad (2.13)$$

s vhodnou hodnotou parametru t^k . Horní index k je číslo iterace. Pokud platí $t^k < \frac{2}{\beta}$, výsledná posloupnost x^k tvořena iteracemi algoritmem 2.13 přibližuje k řešení problému 2.10. Tato iterace se skládá ze dvou kroků:

- Dopředného (explicitního) gradientního kroku - tento krok využívá pouze g na výpočet $x^{k+\frac{1}{2}} = x^k - t^k \nabla g(x^k)$.
- Zpětného (obecně implicitního) proximálního kroku - tento krok využívá pouze h na výpočet $x^{k+1} = \text{prox}_{t^k h}(x^{k+\frac{1}{2}})$. [5]

Graf 2.17 ukazuje příklad průběhu proximální gradientní metody. Výsledné minimum znázorňuje červený bod.



Obr. 2.17: Graf numerické minimalizace. [4]

2.8 Popis programovacího jazyka a jeho výhody

2.8.1 HTML5

Historie HTML (zkratka z anglického *HyperText Markup Language*) sahá až do roku 1991, kdy Tim Berners napsal první definici jazyka HTML, jako součást projektu WWW. Tento projekt se zabýval možností publikace vědeckých výsledků z CERNu do celého světa.

Poté nastoupil rychlý vývoj od verze HTML1 až po dnešní verzi HTML5. HTML je značkovací jazyk používaný pro tvorbu webových stránek. Nejnovější verze HTML5 přinesla oproti starším verzím větší podporu multimediálního obsahu. Dále umožňuje tvorbu interaktivnějších a zajímavějších webových stránek. Samotná koncepce HTML bohužel nezaručuje fakt, že se obsah zobrazí ve všech prohlížečích stejně. [9][10][11]

Canvas

S poslední verzí HTML5 vyšel kromě různých nových tagů také jeden revoluční tag, a to `<canvas>`. Canvas je kreslicí plátno, které poprvé uvedla na svět firma Apple na svém operačním systému OS, kde ho zabudovala přímo do prohlížeče Safari. Postupem byl zabudováván do dalších prohlížečů. V dnešní době je již podporován všemi moderními prohlížeči na trhu.

Pro vytvoření plátna musíme nastavit následující atributy:

- výšku plátna,
- šířku plátna,
- id pro kreslení JavaScriptem.

Na obrázku 2.18 můžeme vidět ukázkou vložení dvou kontextů pláten do stránky. Tyto atributy nelze nastavovat pomocí CSS, protože by nedošlo k nastavení kontextu.

Canvas podporuje vykreslovat kontexty:

- 2d - vykresluje 2D objekty,
- WebGL - ve vývoji, vykresluje 3D objekty.

Na samotné plátno můžeme vykreslit cokoliv pomocí JavaScriptu.[14]

```
<canvas id="canvas1" width="500" height="500"></canvas>
<canvas id="canvas2" width="500" height="500"></canvas>
```

Obr. 2.18: Vložení kontextů plátna do stránky pomocí Canvas.

2.8.2 JavaScript

První JavaScript vytvořil v roce 1995 Erendan Eich. Byl poprvé implementován v prohlížeči Netscape Navigátor 2.0. JavaScript dnes existuje v mnoha implementacích, je to klientský skriptovací jazyk. To znamená, že se program odesílá se stránkou na klientský počítač (prohlížeč), a pak je teprve vykonáván.

JavaScript je:

- interpretovaný (nemusí se kompilovat),
- multiplatformní,
- objektově orientovaný,
- case sensitivní - záleží na velikosti písmen ve skriptu.

JavaScript umožňuje provádění pokročilých interaktivních animací a zajímavých grafických efektů.

Má však stále určité omezení:

- nefunguje bez webového prohlížeče,
- existuje mnoho verzí JavaScriptu což často vede k chybám,
- kromě cookies neumí přistupovat k souborům ani systémovým objektům,
- kromě cookies neumí ukládat žádná data.

JavaScript má podobnou syntaxi s jazyky C a Java. V dnešní době je JavaScript podporován drtivou většinou webových prohlížečů. Jeho výhody jsou přenesení hlavní výpočetní zátěže z webového serveru na klientský systém. Jako hlavní nevýhoda JavaScriptu je nemožnost manipulace se soubory.

Základní tagy v JavaScriptu:

- V HTML se zapisuje JavaScript mezi tagy `<script></script>`. Vše mezi těmito tagy je napsáno v jazyce JavaScript.
- Pro tag deklarace proměnné slouží klíčové slovo `var` za ním následuje jméno proměnné.
- Deklarace funkce se provádí slovem `function`. Za tímto slovem následuje jméno funkce.
- K základnímu větvení a cyklování slouží příkazy `if`, `while` a `for`.^{[12][11][15]}

Na obrázku 2.19 vidíme ukázkou programovacího jazyka JavaScript, konkrétně načtení pláten Canvas, obrázku, změny obrázku a ovládacího panelu.

```
function onLoad() {
    var canvas1 = $('#canvas1');
    var canvas2 = $('#canvas2');
    setupRadio();
    drawPictureOriginal(canvas1);
    redrawPicture(canvas2, 0);
}
```

Obr. 2.19: Ukázka programovacího jazyka JavaScript.

2.8.3 CSS

První návrh CSS byl vytvořen v roce 1994 a to Håkon Wium Lie v standardizační organizaci W3C. Zkratka CSS znamená *Cascading Style Sheets*, česky ji nazýváme kaskádové styly. Význam slova *kaskádové* znamená, že se definice stylů mohou na sebe vrstvit. Hlavním úkolem kaskádových stylů bylo umožnění oddělení formátování vzhledu dokumentu od jeho struktury a obsahu. Dříve se tagy formátu zapisovaly přímo do HTML kódu. Tím vznikl velice složitý a nepřehledný kód. Další výhoda CSS oproti samostatnému HTML je větší možnost formátování, dynamičtější práce a rychlejší načítání.

V dnešní době je již další verze CSS a to CSS 2. Tato vylepšená verze dokáže složitější formy stylů. Bohužel nemá plnou podporu v nejrozšířenějším prohlížeči Internet Exploreru. [11][10]

Styl můžeme nadeklarovat několika způsoby, například:

- Přímý styl – přímo do textu zdroje k elementu, který chceme zformátovat, napíšeme atribut `style="..."`. Viz. obr. 2.20
- Stylopisu (angl. "stylesheet") – který se nachází v hlavičce stránky. Je to takový seznam stylů, který říká, co má být jak zformátováno. Viz. obr. 2.21
- Externí stylopis – je soubor s formátem `.css`. Stránka se na tento soubor odkazuje tagem `<link>`. Tímto způsobem jde použít stejný soubor stylů do více stránek současně. Tento způsob patří mezi nejčastěji používaný v programování webových stránek. Viz. obr. 2.22

```
<p style="color: red">Tento odstavec bude červený.</p>
```

Obr. 2.20: Obrázek CSS kódu přímého stylu.

```
<style>
p {color: red}
</style>
```

Obr. 2.21: Obrázek CSS kódu pomocí stylopisu.

```
<link href="css/style.css" rel="stylesheet" type="text/css">
```

Obr. 2.22: Obrázek CSS kódu linkem pro načtení stylu v externím souboru.

Stručný přehled co CSS dovede:

- nastavit libovolnou a přesnou velikost písma,
- prokládání písma,
- kapitálky,
- odsazování prvního řádku od odstavce,
- zvětšit řádkování,
- automaticky formátovat nadpisy,
- zvýrazňovat www odkazy,
- nastavení pozadí stránky.

CSS má mnoho možností a s JavaScriptem dokáže opravdu na stránkách mnoho.

2.8.4 Výběr jazyka

Zvažoval jsem dvě možnosti při výběru vhodného jazyka pro tvorbu webového interaktivního apletu. První možností bylo vytvoření Java apletu. Tuto možnost jsem hned na začátku zavrhl a to z důvodu:

- nutná instalace Java Runtime (JRE) na operační systém počítače,
- úplné stáhnutí apletu do klientského počítače,
- vysoké požadavky na rychlost připojení k internetu,
- nutnost aktualizace nových verzí JRE,
- nemožnost spuštění apletu z důvodu přísného výchozího nastavení zabezpečení u klienta (Firewall),
- nulová podpora v prohlížečích na mobilních telefonech a tabletech.[10]

Druhou možností bylo vytvoření stránek pomocí metody tzv. „Moderního webu“, která se skládá z následujících tří programovacích jazyků: HTML5, CSS a JavaScriptu. Výhody této metody jsou:

- podpora prohlížečů na mobilních telefonech a PC,
- nenáročnost,
- funkčnost i s pomalejším připojením k internetu,

- čistý a přehledný kód pomocí schématických značek,
- není potřeba instalace Runtime,
- žádné potíže se zabezpečením u klienta.[12][10]

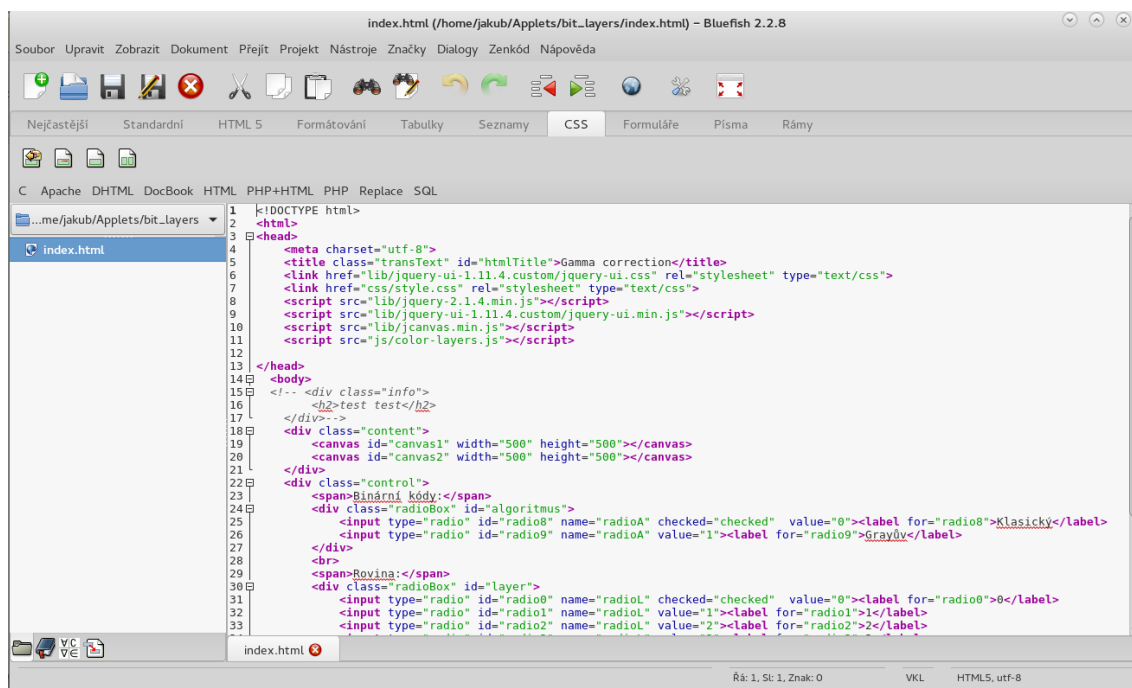
2.8.5 Vývojové prostředí Bluefish

Bluefish vytvořil Oliver Sessink a Chris Mazuc, kteří nazvali aplikaci pracovně a to Thtml editor v roce 1997. Po čase se k této dvojici připojil Neil Millar, který vytvořil aktuální logo editoru s modrou rybou a změnil jeho název na Bluefish. Bluefish byl původně vytvořen jako HTML editor dokumentů, což ovlivňuje jeho vzhled do teď.

Bluefish je nejvyspělejším editorem WWW stránek co se týče funkčnosti. Je založený na knihovně GTK(XFce, Gnome). A je primárně vyvíjen pro Linux. Má podporu pro Windows i Mac OSX, kde zatím nejsou k dispozici všechny jeho funkce. Bluefish je velice rychlý, čistý a dokáže otevřít současně přes 500 dokumentů. Podporuje práci na vzdálených souborech s podporou FTP, SFTP, HTTP, HTTPS, WebDAV, CIFS.

Bluefish dokáže editovat a pracovat mnoho programovacích jazyků, například C/C++, HTML, CSS, JavaScript and JQuery a Python. Ještě dokáže pracovat s mnoha dalšími typy kódování včetně UTF-8.[19][20][22]

Při vývoji webových apletů jsem použil Bluefish verzi 2.2.8.



Obr. 2.23: Vývojové prostředí Bluefish.

3 PRAKTICKÁ ČÁST

Praktická část mé bakalářské práce je zaměřena na vytvoření webového interaktivního apletu, který bude sloužit pro studijní účely studentům VUT. Tyto aplety jsou menší aplikace zakomponované přímo do webových stránek. Aplety lze spustit pouze na webových prohlížečích a jsou odzkoušeny na nejznámějších prohlížečích Mozilla Firefox, Google Chrome, Internet Explorer a Opera.

Tato bakalářská práce popisuje čtyři webové aplety:

- aplet na znázornění bitových rovin ve fotografii,
- aplet na znázornění konstrukce sazebního obrazce,
- aplet na úpravu fotografie pomocí gama korekce,
- aplet na znázornění numerické minimalizace konvexních funkcí.

Tyto aplety jsou tvořeny pomocí open source vývojového prostředí BlueFish 2.2.8 a pomocí jazyků HTML5, JavaScript a CSS.

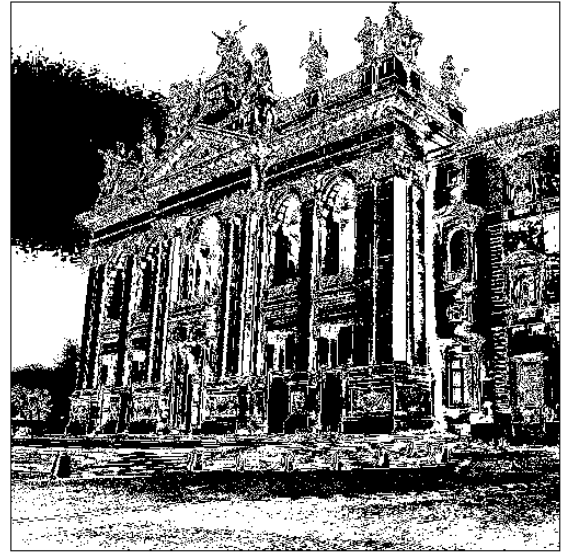
3.1 Aplet bitových rovin

Složení apletu bitových rovin:

- vstupní fotografie,
- výstupní fotografie,
- dvě tlačítka výběru klasického bitového kódu nebo Grayova kódu,
- osm tlačítek pro volbu požadované bitové roviny.

Tento aplet rozkládá původní monochromatickou fotografii na jednotlivé bitové roviny. Rozklad probíhá od nejnižší nulté bitové roviny až do nejdůležitější sedmé bitové roviny. Po spuštění webové stránky s apletem jsou všechny hodnoty ve výchozím stavu. Výchozí stav se rovná nulté rovině s klasickým bitovým kódem. Na levé straně je pevně vložena originální fotografie a na pravé straně je výsledná fotografie v dané bitové úrovni. Překliknutím tlačítka ve výběru kódu na Grayův kód se okamžitě změní výsledná fotografie. Překliknutím tlačítek pro bitovou rovinu se původní výstupní fotografie mění podle požadované bitové roviny.

Na obrázku 3.1 můžeme vidět zvolený klasický bitový kód ve druhé bitové rovině fotografie.



Binární kódy:

Klasický

Grayův

Rovina:

0

1

2

3

4

5

6

7

Obr. 3.1: Aplet bitových rovin.

3.2 Aplet sazebního obrazce

Složení apletu sazebního obrazce:

- výstupní obrazec,
- menu s výběrem formátu stránky,
- buňka s volbou výšky a šířky stránky,
- buňka s volbou šířky hřbetní mezery,
- buňky s volbou ořezu okrajů,
- tlačítko s volbou *Van de Graafovo úhlopříčné pravidlo*,
- tlačítko s volbou kružnicového pravidla s dvěma buňkami pro volbu dvou volností,
- tlačítko s volbou univerzálního pravidla 3:4:5:7 s volbou vnitřního okraje x,
- tlačítko s volbou univerzálního pravidla 2:3:4:6 s volbou vnitřního okraje x,
- tlačítko s volbou úhlopříčného pravidla 1/12.

Na horní části levé strany vidíme výstupní obrazec, který se při změně zadaných hodnot okamžitě překreslí. Obrazec obsahuje dvojstránku zvoleného formátu tiskoviny a výsledný vypočítaný sazební obrazec. Pod výsledným obrazcem je menu s volbou formátu stránky, kde máme na výběr nejčastější formáty a to:

- A3 (297 mm x 420 mm)
- A4 (210 mm x 297 mm)
- A5 (148 mm x 210 mm)
- B3 (353 mm x 500 mm)
- B4 (250 mm x 353 mm)
- B5 (176 mm x 250 mm)

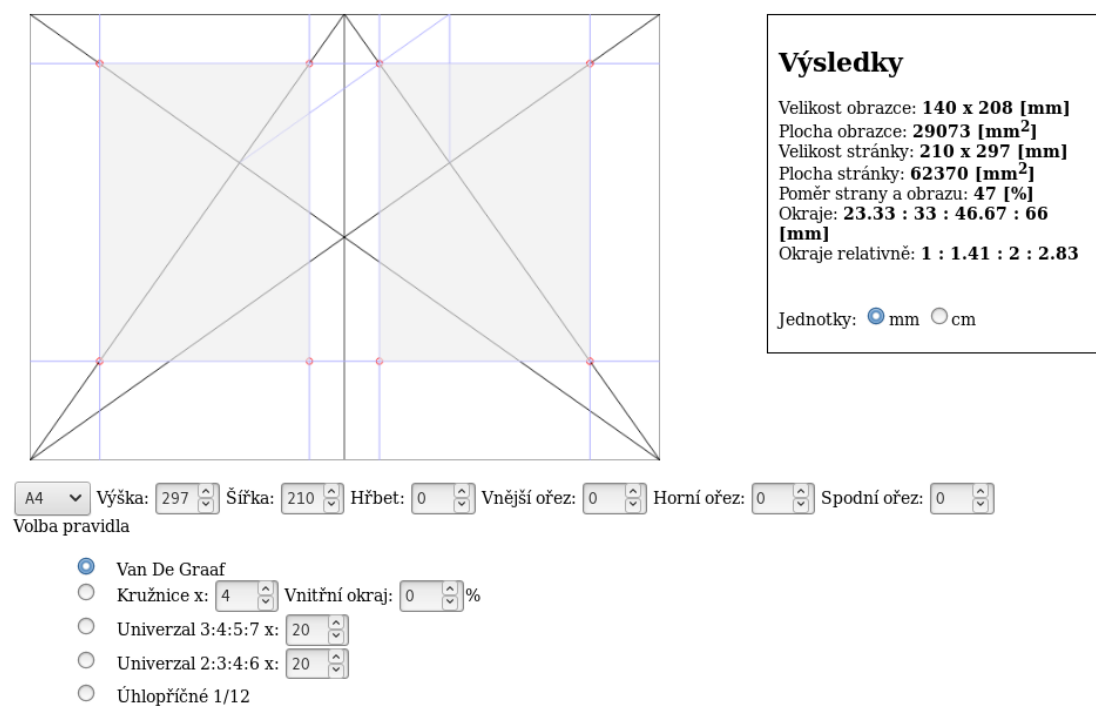
Vpravo od menu s formáty jsou buňky s volbou šířky a výšky stránky. Do nich se automaticky zapisují rozměry předdefinovaného formátu z menu nebo můžeme zapsat libovolné rozměry výšky a šířky stránky. Po zápisu hodnot se změna okamžitě projeví na obraze a ve výsledku.

Ve střední části stránky vlevo můžeme vidět buňku s volbou šířky hřbetní mezery. Hřbetní mezeru můžeme libovolně volit, ale nejčastěji se pohybuje od 0 cm do 2 cm. Dále vpravo od této volby můžeme vidět buňky s volbou ořezu. Zde si můžeme libovolně nastavit horní, spodní a vnější ořez papíru. Tím zajistíme, že sazební obrazec pracuje s čistým formátem papíru. Při zadání jakékoliv hodnoty do těchto buněk se okamžitě změní a vykreslí tato změna do obrazce. V levé horní části jsou vypsány výsledné plochy jedné stránky a to:

- velikost obrazce v čistém formátu,
- plocha sazebního obrazce,
- velikost stránky,
- celková plocha stránky,

- výsledný poměr plochy stránky k ploše sazebního obrazce.

Dále tento aplet vykresluje výsledné okraje jedné stránky a to v absolutním a relativním poměru vnitřní : horní : vnější : dolní okraje. Také máme možnost zvolit jednotky a to cm^2 nebo mm^2 .



Obr. 3.2: Aplet sazebního obrazce.

3.3 Aplet gama korekce

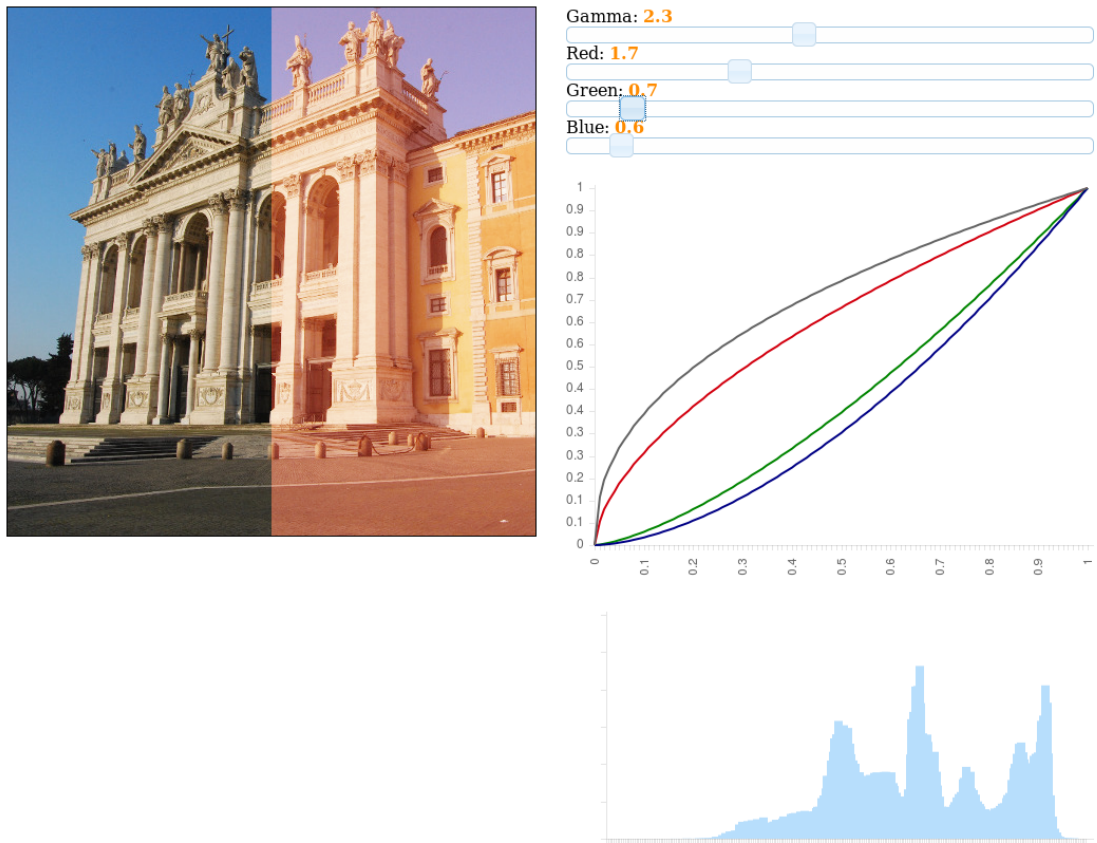
Aplet se skládá z následujících částí:

- vstupní a výstupní fotografie,
- posuvník Gamma,
- posuvník Red,
- posuvník Green,
- posuvník Blue,
- graf korekce,
- histogram.

Obrázek je rozdělen na poloviny, kde v levé části se nachází původní fotografie, která se při nastavení gama korekce nemění. Pravá část fotografie se mění podle nastavení gama korekce, kterou nastavuje uživatel pomocí posuvníku Gamma na levé části stránky. Uživatel může nastavovat celkovou gamu pomocí posuvníku Gamma, nebo jednotlivou gama korekci barev RGB (red, green, blue) a to pomocí posuvníků Red, Green, Blue. Nad každým posuvníkem je číslo, které značí aktuální hodnotu úpravy gama korekce. Rozsah úpravy se pohybuje od 0,1 do 5 a to postupným krokem 0,1.

Pod posuvníky se nachází graf gama korekce. V grafu se nacházejí čtyři křivky gama korekce a jeho jednotlivých barev red, green, blue. Rozsah x je od 0 do 1 a y od 0 do 1. Každá změna nastavení posuvníků automaticky změní hodnoty dané křivky.

Pod grafem se nachází histogram, který zobrazuje aktuální hodnoty jasu ve výsledné pravé části fotografie. Výsledný obrázek apletu gama korekce 3.3 vidíme ve světlé části histogramu.



Obr. 3.3: Aplet gama korekce.

3.4 Numerická minimalizace konvexních funkcí

Složení apletu numerické minimalizace konvexních funkcí:

- výstupní graf s vykreslenými funkcemi,
- první roller s konvexní funkcí $g(x)$,
- druhý roller s konvexní funkcí $h(x)$,
- buňka pro zadání počátečního bodu x^0 ,
- buňka pro zadání kroku t ,
- tabulka s výsledky,
- tlačítko pro postupnou iteraci po jednom kroku vpřed - $>$,
- tlačítko pro postupnou iteraci po jednom kroku zpět - $<$,
- tlačítko pro vykreslení kompletní iterace - \gg ,
- tlačítko pro smazání výsledných bodů na grafu - C.

Tento aplet interaktivně vykresluje definované funkce a body pomocí numerické minimalizace.

Definované funkce $g(x)$:

1. $\frac{1}{2}(x - 2)^2$,
2. x^2 ,
3. $\frac{1}{3}(2x + 2)^2$,
4. $\frac{1}{5}x^2$.

Definované funkce $h(x)$:

1. $\frac{1}{2}|2x + 1|$,
2. $|x|$,
3. $|-3 - x|$,
4. $\frac{1}{3}|3x + 2|$.

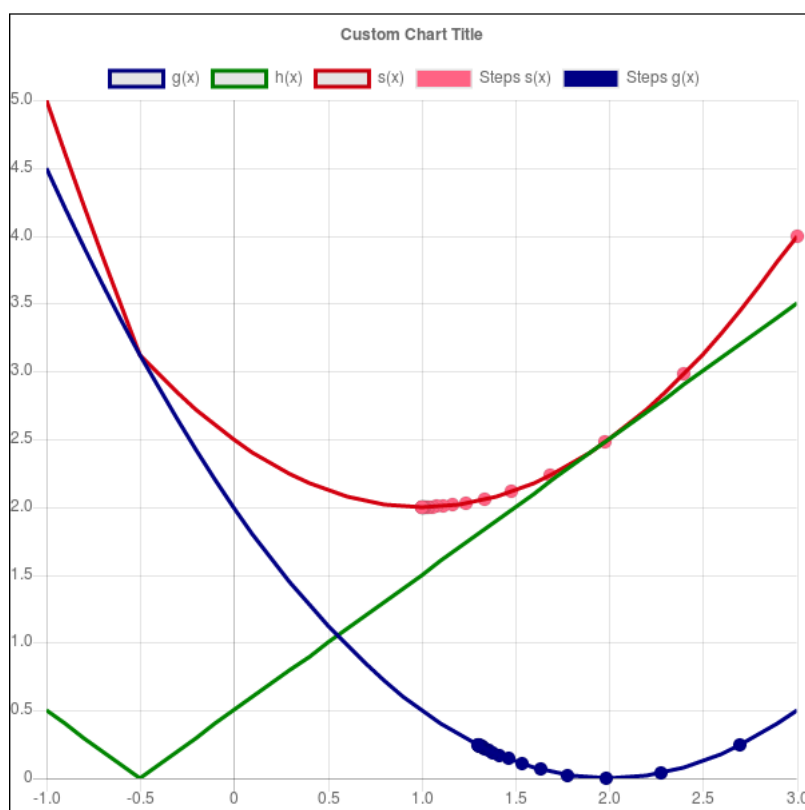
V levé části stránky vidíme výsledný graf s vykreslenými křivkami a body. Ukázka na obrázku 3.4.

V pravé horní části se nachází ovládání apletu, kde máme na výběr ze dvou rollerů $g(x)$ a $h(x)$ s předdefinovanými konvexními funkcemi. Dále se tu nachází buňka pro zadání libovolného počátečního bodu x^0 a buňka pro libovolné zadání kroku t . Buňka kroku t při zadání vyšší nebo rovna $\frac{2}{\beta}$ upozorní na nekonvergenci řešení zčervenáním. V této části se také nachází tlačítka $<$, C, $>$, \gg . Tlačítko $<$ umožňuje zpětnou iteraci. Kliknutím na tlačítko $<$ dojde k posunutí ukazatele ve výsledné tabulce výše a následné smazání výsledného bodu v grafu. Tlačítko C smaže všechny výsledné body v grafu. Tlačítko $>$ vykresluje výsledné body po jednom kroku v grafu a posune ukazatele ve výsledné tabulce o řádek níž. Tlačítko \gg umožňuje kompletní

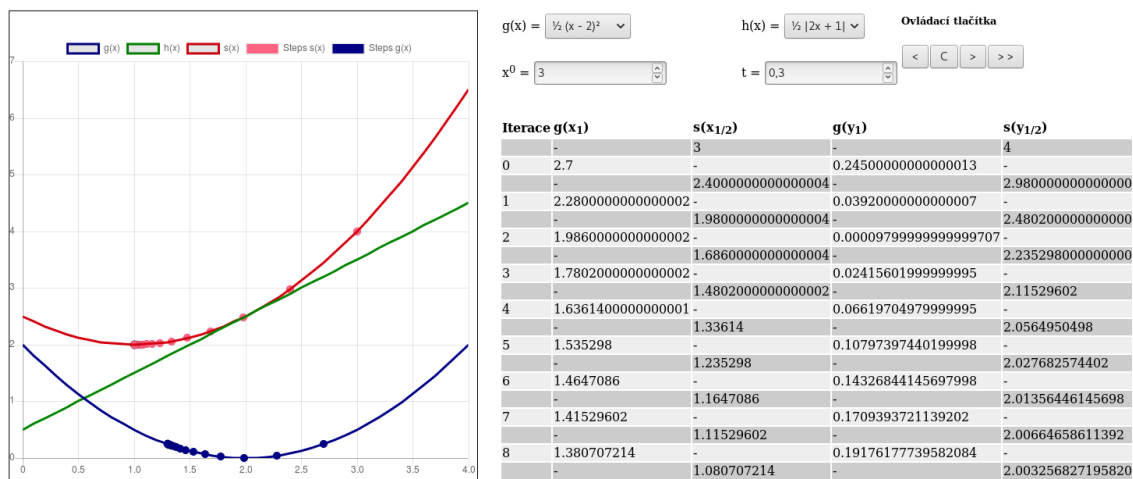
vykreslení výsledných bodů do grafu.

Ve spodní pravé části se nachází výsledná tabulka. První sloupec je číslo iterace. Druhý sloupec $g(x_1)$ vypisuje výsledky zpětného algoritmu. Třetí sloupec $s(x_{1/2})$ vypisuje výsledky dopředného gradientního kroku. Čtvrtý sloupec $g(y_1)$ vypisuje výsledky zpětného gradientního kroku. Pátý sloupec $s(y_{1/2})$ vypisuje výsledky dopředného kroku. Na obrázku 3.4 vidíme výsledný graf apletu. Modrá křivka znázorňuje konvexní funkci $g(x)$, zelená křivka znázorňuje konvexní funkci $h(x)$ a červená křivka znázorňuje součet konvexních funkcí $s(x) = h(x) + g(x)$. Červené body jsou výsledky dopředného gradientního kroku, modré body jsou výsledky zpětného gradientního kroku.

Celý aplet se jakoukoliv změnou hodnot automaticky přepočítá. Na obrázku 3.5 můžeme vidět celý aplet numerické minimalizace.



Obr. 3.4: Výsledný graf apletu.



Obr. 3.5: Aplet numerické minimalizace.

4 ZÁVĚR

Ve své bakalářské práci jsem vytvořil aplety pro podporu výuky zpracování obrazu. Jedná se o programy, které je možné spouštět ve většině prohlížečů, mobilních telefonů a počítačů. Zvolil jsem HTML5, CSS a JavaScript jako nejvhodnější programovací jazyk pro tvorbu tohoto výukového apletu. Všechny aplety jsou vytvořeny ve vývojovém prostředí Bluefish, které je open source distribucí. Bylo zapotřebí, abych si nastudoval teorii ke tvorbě apletů a syntaxe HTML5, CSS a JavaScript.

První kapitola teoretické části definuje potřebné matematické elementární funkce a jejich vlastnosti ke zpracování obrazu a vnímání barevné informace člověkem. Dále jsem se zabýval rozbořením vzniku fotografie, zpracováním, čtením jejího histogramu, tvorbou sazebního obrazce, gama korekcí a proximální gradientní metodou.

Aplet pro korekci gama a jednotlivých složek RGB rozděluje vybranou fotografii na poloviny, kde v levé části je originál obrazu a v pravé námi provedená korekce. Všechny změny se promítají do histogramu. Dále jsem vytvořil aplet sazebního obrazce, který počítá okraje sazby na stránce. Též ji vykresluje na dvojstránce s možností výběru předdefinovaného formátu či zadání rozměrů stránky. Tento aplet umožňuje ořez vnější, spodní, horní a také hřbetní mezery. V tomto apletu lze volit z pěti pravidel sazebního obrazce. Též například u kružnicového sazebního obrace můžeme volit dva typy volnosti. Výsledek můžeme převést z milimetrů na centimetry. Aplet bitových rovin libovolně ukazuje osmi bitové roviny monochromatické fotografie. V tomto apletu máme dvě volby kódování a to klasického kódu a Grayova kódu. Dalším apletem je numerická minimalizace konvexních funkcí, která vykresluje předdefinované konvexní funkce v grafu a počítá minimum pomocí proximální gradientní metody. Tento aplet umožňuje zvolit libovolný krok pro počáteční hodnotu začátku iterace. Pomocí tlačítek můžeme postupně vykreslovat iterační kroky na grafu.

LITERATURA

- [1] Jiří Žára, Bedřich Beneš, Jiří Sochor, Petr Felkel. *Moderní počítačová grafika*. 2004 [cit. 2015-12-08].
- [2] ČVUT. Fakulta elektrotechniky katedra Matematiky ČVUT. <http://math.feld.cvut.cz/>. [online]. 24.6.2010 [cit. 2015-12-07]. Dostupné z: <<http://math.feld.cvut.cz/0educ/predpokl/msu2.pdf>>.
- [3] Vlasta Krupková, Petr Fuchs *Matematika 1*. <http://www.umat.feec.vutbr.cz/>. [online]. 2014 [cit. 2015-12-07]. Dostupné z: <<https://www.vutbr.cz/studium/ects-katalog/detail-predmetu?apid=148700>>.
- [4] Marie Daňková, *Komprimované snímání v perfuzním zobrazování pomocí magnetického rezonance*. 2014 [cit. 2015-12-07].
- [5] Pavel Rajmic, *Řídké a nízkohodnostní reprezentace signálů s aplikacemi*. 2014 [cit. 2015-12-07].
- [6] Pavel Rajmic, *Základy počítačové sazby a grafiky*. 2012 [cit. 2015-12-07].
- [7] Lenka Dekanová. *Konverzní funkce*. [online]. 2007 [cit. 2015-12-10]. Dostupné z: <http://is.muni.cz/th/150978/prif_b/Novy3.pdf>.
- [8] Střední škola mediální grafiky a tisku Praha. <http://www.sspp.cz/praxe/zrcadlo.pdf>. [online]. [cit. 2015-12-10]. Dostupné z: <<http://www.sspp.cz/praxe/zrcadlo.pdf>>.
- [9] Jiří Kosek, <http://htmlguru.cz>. [online]. [cit. 2015-12-10]. Dostupné z: <<http://htmlguru.cz/uvod-historie.html>>.
- [10] UPC Česká republika s.r.o. www.dostupnyinternet.cz. [online]. [cit. 2015-12-10]. Dostupné z: <<http://www.dostupnyinternet.cz/blog/html-5>>.
- [11] <http://www.garth.cz/>. [online]. [cit. 2015-12-10]. Dostupné z: <<http://www.garth.cz/>>.
- [12] Steve Champeon, <http://archive.oreilly.com/javascript/>. [online]. [cit. 2015-12-10]. Dostupné z: <http://archive.oreilly.com/pub/a/javascript/2001/04/06/js_history.html>.
- [13] o. s. Ephata. www.ephata.cz. [online]. 15.12.2015 [cit. 2015-12-15]. Dostupné z: <<http://ephata.cz/vyuziti-novych-audiovizualnich-pomucek-v-rozvoji-komunikacnich-schopnosti-handicapovanych-deti/>>.

- [14] Drahomír Hanák. *Canvas aneb grafika JavaScriptem*. www.itnetwork.cz. [online]. [cit. 2016-05-01]. Dostupné z: <<http://www.itnetwork.cz/javascript/zaklady/canvas-aneb-grafika-javascriptem/>>.
- [15] Dušan Janovský. *Jak psát web*. <http://www.jakpsatweb.cz/>. [online]. [cit. 2016-05-01]. Dostupné z: <<http://www.jakpsatweb.cz//>>.
- [16] Iveta Gladišová, Ján Mihalík, Jozef Zavacký. *Bezstratová kompresia obrazu pomocou stavového binárneho aritmetického kódu vania jeho bitových rovín*. [online]. [cit. 2016-05-01]. Dostupné z: <http://web.tuke.sk/fei-ldipv/publications/Journal%20papers/Slab_obzor_bezstrat_kodovanie_final.pdf>.
- [17] Radomil Matoušek. *METODY KÓDOVÁNÍ* [online]. [cit. 2016-05-01]. Dostupné z: <<http://www.uai.fme.vutbr.cz/~matousek/TIK/TIKv19.pdf>>.
- [18] Milan Solil. *Vývoj a metody automatizace*. [online]. [cit. 2016-05-01]. Dostupné z: <https://www.google.cz/url?sa=t&rct=j&q=&esrc=s&source=web&cd=17&ved=0ahUKEwj9h_2W4LnMAhWVOsAKHfSMB2oQFghvMBA&url=http%3A%2F%2Fwww.isstvm.cz%2Fsites%2Fdefault%2Ffiles%2Fstranka%2F188%2Fvy_32_inovace_atm_3s_sl_10_11_grayuv_kod.pptx&usg=AFQjCNGSHWe-Erwnmf5pPgCIbmsQumOTNg>.
- [19] Olivier Sessink. *Bluefish Features*. . [online]. [cit. 2016-05-01]. Dostupné z: <<http://bluefish.openoffice.nl/features.html>>.
- [20] Marek Olšavský . *Bluefish a Screem – Gnome (GTK) web editory*. . [online]. [cit. 2016-05-01]. Dostupné z: <http://www.linuxsoft.cz/article.php?id_article=867>.
- [21] Roman Pihan. *Gamma korekce*. . [online]. [cit. 2016-05-01]. Dostupné z: <http://fotoroman.cz/glossary2/3_gamma.html>.
- [22] Miroslav Hrončok. *Bluefish: HTML editor pro GNOME, Xfce a pro vás*. . [online]. [cit. 2016-05-01]. Dostupné z: <<http://www.linuxexpres.cz/software/textove-editory/bluefish-html-editor-pro-gnome-xfce-a-pro-vas>>.
- [23] Prashant Dwivedi. *www.github.com*. [online]. [cit. 2015-12-10]. Dostupné z: <<https://github.com/hay-wire/Chart.js/commit/df9751102850840442516aad7eb693922c1826d1>>.
- [24] Fernando Perez *www.github.com*. [online]. [cit. 2015-12-10]. Dostupné z: <www.plot.ly/javascript/histograms/>.

- [25] Scott González, Jörn Zaefferer, Felix Nagel, Mike Sherov, Rafael Xavier de Souza *www.jqueryui.com* [online]. [cit. 2015-12-10]. Dostupné z: <<http://jqueryui.com/slider/#default>>.
- [26] Caleb Evans. *www.projects.calebevans.me*. [online]. [cit. 2015-12-10]. Dostupné z: <<http://projects.calebevans.me/jcanvas/docs/introduction/>>.
- [27] Juhana, Simo Endre, Rutwick Gangurde, *www.stackoverflow.com*. [online]. [cit. 2015-12-10]. Dostupné z: <<http://stackoverflow.com/questions/14012221/gamma-adjustment-on-the-html5-canvas>>.
- [28] Nick Downie. *www.github.com*. [online]. [cit. 2015-12-10]. Dostupné z: <<https://github.com/nnnick/Chart.js/tree/master/samples>>.
- [29] Nick Downie. *www.chartjs.org*. [online]. [cit. 2015-12-10]. Dostupné z: <<http://www.chartjs.org/docs/#line-chart-prototype-methods>>.

5 OBSAH PŘILOŽENÉHO CD

Přiložené CD obsahuje elektronickou verzi této práce ve formátu PDF, dále obsahuje složky se zdrojovými kódy k jednotlivým apletům. Aplety lze jednoduše spustit otevřením HTML souboru v některém z webových prohlížečů.

Obsah CD:

- xkorbe02.pdf – elektronická verze této bakalářské práce,
- bit_roviny – složka se soubory k Apletu bitových rovin,
- gama_korekce – složka se soubory k Apletu gama korekce,
- numikof – složka se soubory k Apletu numerické minimalizace konvexních funkcí,
- sazba – složka se soubory k Apletu sazebního obrazce.