

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA ELEKTROTECHNIKY A KOMUNIKAČNÍCH TECHNOLOGIÍ
ÚSTAV TELEKOMUNIKACÍ

FACULTY OF ELECTRICAL ENGINEERING AND COMMUNICATION
DEPARTMENT OF TELECOMMUNICATIONS

SOUBOR PROGRAMŮ PRO PODPORU VÝUKY ZPRACOVÁNÍ
SIGNÁLŮ A OBRAZŮ

DIPLOMOVÁ PRÁCE
MASTER'S THESIS

AUTOR PRÁCE
AUTHOR

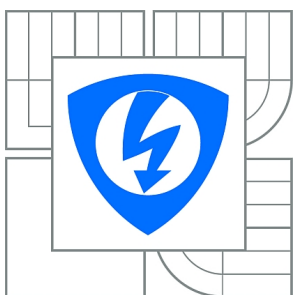
Bc. MIROSLAV RIŠKO

BRNO 2014



VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

BRNO UNIVERSITY OF TECHNOLOGY



FAKULTA ELEKTROTECHNIKY A KOMUNIKAČNÍCH
TECHNOLOGIÍ

ÚSTAV TELEKOMUNIKACÍ

FACULTY OF ELECTRICAL ENGINEERING AND COMMUNICATION
DEPARTMENT OF TELECOMMUNICATIONS

SOUBOR PROGRAMŮ PRO PODPORU VÝUKY ZPRACOVÁNÍ SIGNÁLŮ A OBRAZŮ

BUNDLE OF PROGRAMS FOR TEACHING SIGNAL AND IMAGE PROCESSING

DIPLOMOVÁ PRÁCE

MASTER'S THESIS

AUTOR PRÁCE

AUTHOR

Bc. MIROSLAV RIŠKO

VEDOUCÍ PRÁCE

SUPERVISOR

Mgr. PAVEL RAJMIC, Ph.D.

BRNO 2014



VYSOKÉ UČENÍ
TECHNICKÉ V BRNĚ

Fakulta elektrotechniky
a komunikačních technologií

Ústav telekomunikací

Diplomová práce

magisterský navazující studijní obor
Telekomunikační a informační technika

Student: Bc. Miroslav Riško

ID: 115267

Ročník: 2

Akademický rok: 2013/2014

NÁZEV TÉMATU:

Soubor programů pro podporu výuky zpracování signálů a obrazů

POKYNY PRO VYPRACOVÁNÍ:

Cílem práce je vytvořit programy pro interaktivní podporu výuky zejm. v kurzech BZSG, MGMP, BASS. Budou mít podobu JAVA apletů, které budou tématicky zaměřeny na: 1/lineární kombinaci signálů, 2/ aritmetické kódování, 3/ gama korekci, 4/ použití palety pro reprezentaci bitmapového obrazu. Student aplety navrhne, implementuje a ověří jejich funkčnost.

DOPORUČENÁ LITERATURA:

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

[2] R. C. Gonzales, R. E. Woods, Digital Image Processing, Third Edition, Prentice Hall, 2008

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

Termín zadání: 10.2.2014

Termín odevzdání: 28.5.2014

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

Konzultanti diplomové práce:

prof. Ing. Kamil Vrba, CSc.

Předseda oborové rady

UPOZORNĚNÍ:

Autor diplomové práce nesmí při vytváření diplomové 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

Cieľom mojej diplomovej práce je vytvorenie programov, ktoré budú slúžiť ako podpora výuky. Práca je rozdelená na dve časti - teoretickú a praktickú. V teoretickej časti sa budem zaoberať teóriou potrebnou k vypracovaniu tejto práce. V praktickej časti bude opísaná samotná funkcionálna vytvorených appletov.

KLÍČOVÉ SLOVÁ

Signál, applet, java, gama, aritmetické kódovanie

ABSTRACT

The aim of my diploma project is to create a program that will serve as a support for the teaching. Project is divided into two parts - theoretical and practical. In the theoretical part I will discuss the theory necessary for development of this work. The practical part will describe functionality of created applets.

KEYWORDS

Signal, applet, java, gamma, arithmetic coding

RIŠKO, Miroslav *Soubor programů pro podporu výuky zpracování signálů a obrazů*: diplomová práce. Brno: Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií, Ústav telekomunikací, 2014. 49 s. Vedúci práce bol Mgr. Pavel Rajmic, Ph.D.

PREHLÁSENIE

Prehlasujem, že som svoju diplomovou prácu na tému „Soubor programů pro podporu výuky zpracování signálů a obrazů“ vypracoval samostatne pod vedením vedúceho diplomovej práce, využitím odbornej literatúry a ďalších informačných zdrojov, ktoré sú všetky citované v práci a uvedené v zozname literatúry na konci práce.

Ako autor uvedenej diplomovej práce ďalej prehlasujem, že v súvislosti s vytvorením tejto diplomovej práce som neporušil autorské práva tretích osôb, najmä som nezasiahol nedovoleným spôsobom do cudzích autorských práv osobnostných a som si plne vedomý následkov porušenia ustanovenia § 11 a nasledujúcich autorského zákona č. 121/2000 Sb., vrátane možných trestnoprávných dôsledkov vyplývajúcich z ustanovenia § 152 trestného zákona č. 140/1961 Sb.

Brno

.....

(podpis autora)



Faculty of Electrical Engineering
and Communication
Brno University of Technology
Technicka 12, CZ-61600 Brno
Czech Republic
<http://www.six.feec.vutbr.cz>

PODĚKOVÁNÍ

Výzkum popsáný v této diplomové 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)



EVROPSKÁ UNIE
EVROPSKÝ FOND PRO REGIONÁLNÍ ROZVOJ
INVESTICE DO VAŠÍ BUDOUCNOSTI



**OP Výzkum a vývoj
pro inovace**

OBSAH

1	Úvod	11
2	Teoretický úvod	12
2.1	Typy signálov	12
2.1.1	Sínus	12
2.1.2	Kosínus	12
2.1.3	Gaussov šum	13
2.1.4	Obdĺžnikový signál	13
2.1.5	Píla	13
2.1.6	Impulz	14
2.2	Lineárna kombinácia vektorov	14
2.3	Konvexná kombinácia vektorov	15
2.4	Barycentrické súradnice	16
2.5	Úprava obrazu pomocou gama korekcie	17
2.6	Aritmetické kódovanie	18
2.6.1	Pseudokód	19
2.6.2	Výstupné slovo	19
3	Popis použitého softwaru	20
3.1	Java	20
3.2	Java Applet	21
3.3	Štruktúra appletu	22
3.4	Knižnica JFreeChart	24
3.5	NetBeans	24
4	Praktická časť	26
4.1	Applet: Lineárna a konvexná kombinácia	26
4.1.1	Popis appletu	27
4.2	Applet: Gama korekcia	32
4.2.1	Popis appletu	32
4.3	Applet: Aritmetické kódovanie	35
4.3.1	Popis appletu	35
4.4	Applet: Indexácia farby	37
4.4.1	Popis appletu	37
5	Záver	40
	Literatúra	42

A Prílohy	44
A.1 Možné problémy	44
A.1.1 Pod operačným systémom	44
A.1.2 Nastavenie Java security	45
B Prílohy	49
B.1 obsah CD	49

ZOZNAM OBRÁZKOV

2.1	Graf signálu sínus	12
2.2	Graf signálu kosínus	13
2.3	Graf gaussovho šumu	13
2.4	Graf obdĺžnikového signálu	14
2.5	Graf pílového signálu	14
2.6	Graf impulzného signálu	14
2.7	Lineárna kombinácia vektorov	15
2.8	Barycentrické súradnice	16
2.9	Graf gamy korekcie	17
2.10	Graf gamy korekcie	19
4.1	Lineárna kombinácia signálov	26
4.2	Konvexná kombinácia signálov	27
4.3	Voľba kombinácie	28
4.4	Výber signálu	28
4.5	Graf vybraného signálu	29
4.6	Posuvník pre lineárnu kombináciu	30
4.7	Posuvník pre konvexnú kombináciu	30
4.8	Barycentrický trojuholník	30
4.9	Graf výsledného signálu	31
4.10	Graf chybovej hlášky	31
4.11	Applet gama korekcia	32
4.12	Výber obrázku	33
4.13	Výber obrázku	33
4.14	Graf gamy korekcie	34
4.15	Tlačidlo na zobrazenie výsledku	34
4.16	Applet aritmetické kódovanie	35
4.17	Štatistické údaje	35
4.18	Štatistické údaje	36
4.19	Applet aritmetické kódovanie	36
4.20	Applet aritmetické kódovanie	36
4.21	Applet Indexácie farieb	37
4.22	Výber veľkosti matíc	38
4.23	Výber farieb	38
4.24	Ovládacie tlačidlá	38
4.25	Defaultne zobrazené matice	39
A.1	Chybné vykreslenie appletu 1	44
A.2	Chybné vykreslenie appletu 2	45

A.3	Blokovanie appletu 1	46
A.4	Blokovanie appletu 2	46
A.5	Nastavanie JAVA security 1	47
A.6	Nastavanie JAVA security 2	47
A.7	Nastavanie JAVA security 3	48

1 ÚVOD

V dnešnej dobe sa čoraz častejšie vo výučbe začína využívať aktívnejšie zapojenie študenta do výučby rôznymi interaktívnymi pomôckami. Vďaka týmto pomôckam vytvára učiteľ priestor na aktívnu prácu žiaka. Základnou myšlienkou je, aby si žiak sám vytvoril obraz o celku na základe vlastných prežitých skúsenosti pri riešení daného problému a informácií poskytnutými učiteľom.

V porovnaní s minulosťou má dnes učiteľ v rukách silný nástroj, ktorý mu môže prácu zefektívniť alebo ju posunúť na novú kvalitatívnu úroveň. Sú to informačno-komunikačné technológie a na nich založené moderné vyučovacie prostriedky.[2]

Táto diplomová práca sa bude zaoberať vytvorením webových JAVA appletov, ktoré majú pomôcť pri výučbe predmetov zameraných na spracovanie signálu a obrazu. Budú vytvorené celkovo 4 applety.

Zameranie appletov bude nasledovné. V prvom applete sa budem zaoberať kombináciou rôznych signálov, kde si užívateľ bude môcť zvoliť z dvoch typov kombinácií a šiestich rôznych signálov pri ktorých môže nastavovať ich váhu. V druhom applete sa budem zaoberať úpravou obrazu pomocou gama korekcií. Užívateľ si bude môcť vyskúšať ako sa zmení zvolený obrázok pri zmene hodnoty nastavenej gamy. V predposlednom applete si môže užívateľ vyskúšať aritmetické kódovanie, kde môže zadať ľubovoľné vstupné slovo a sledovať ako sa po kroku zmení na binárne výstupné slovo. V poslednom applete bude môcť užívateľ vykresliť maticu rôznymi farbami pomocou indexácie farieb.

2 TEORETICKÝ ÚVOD

V tejto časti práce sa oboznámime s teóriou potrebnou na pochopenie tejto diplomovej práce. Oboznámime sa s:

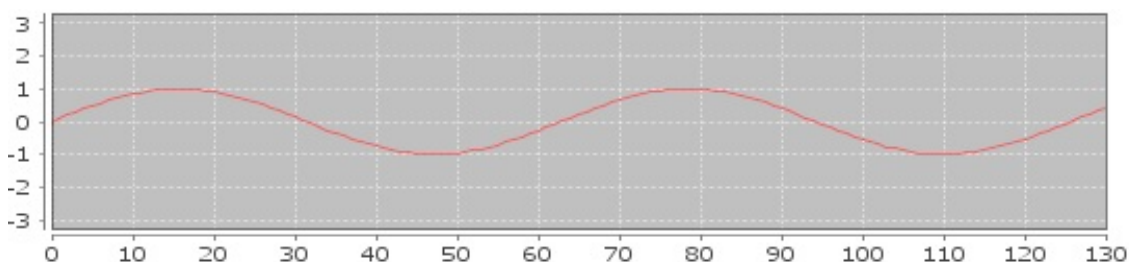
- použitými signálmi v práci
- typmi kombinácií
- barycentrickými súradnicami
- tým ako funguje gama korekcia
- s princípom aritmetického kódovania
- s použitým softwarom v tejto práci

2.1 Typy signálov

V tejto časti práce sa budem zaoberať diskretnými signálmi (definovanými v diskretných časových okamihoch, tvorených postupnosťou funkčných hodnôt), s ktorými ďalej pracujem v tejto práci.

2.1.1 Sínus

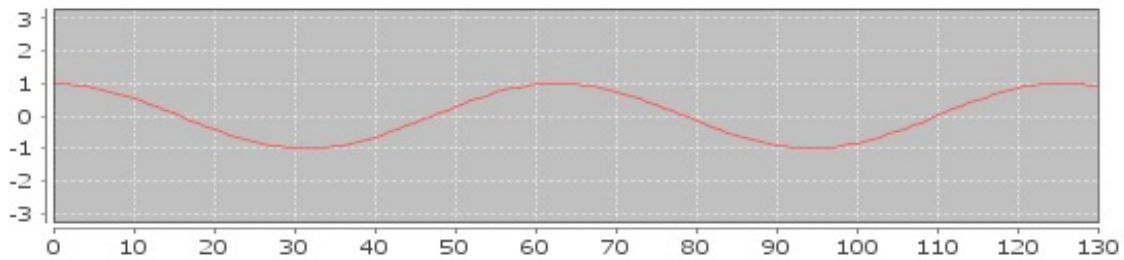
Sínus (vid' obr. 2.1) patrí medzi goniometrické funkcie. V pravouhlom trojuholníku je definovaný ako pomer dĺžky protiľahlej odvesny k uhlu a dĺžky prepony trojuholníka. Graf funkcie sínus sa nazýva sínusoida alebo sínusovka. V našom prípade zobrazujeme graf v 2 periódach.



Obr. 2.1: Graf signálu sínus

2.1.2 Kosínus

Kosínus (vid' obr. 2.2) patrí medzi goniometrické funkcie. V pravouhlom trojuholníku je definovaný ako pomer dĺžky priľahlej odvesny a dĺžky prepony trojuholníka. Graf funkcie kosínus sa nazýva kosínusoida. Táto funkcia sa zvyčajne označuje skratkou \cos . V našom prípade zobrazujem graf v 2 periódach.



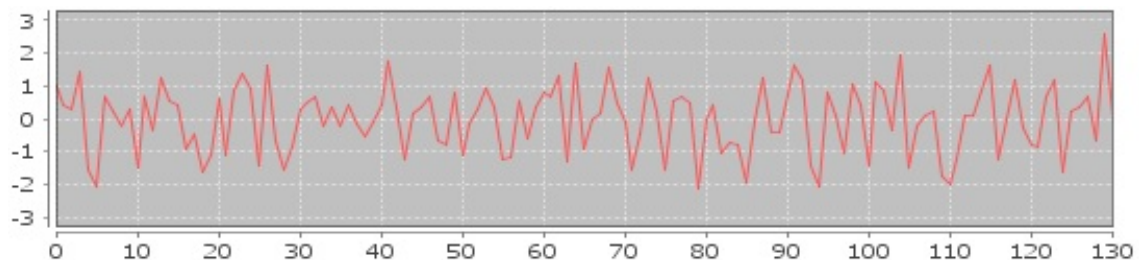
Obr. 2.2: Graf signálu kosínus

2.1.3 Gaussov šum

Gaussov šum (vid' obr. 2.3) predstavuje štatistický šum, ktorý má *funkciu hustoty pravdepodobnosti normálneho rozdelenia* (známeho tiež ako *Gaussovo rozdelenie*).

Hodnoty ktoré šum nadobúdať sú Gaussovo-distribúované. Funkcia hustoty pravdepodobnosti P nahodnej veličiny Z z normálneho rozdelenia je tvaru: [4]

$$P_G(z) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{(z-\mu)^2}{2\sigma^2}}$$



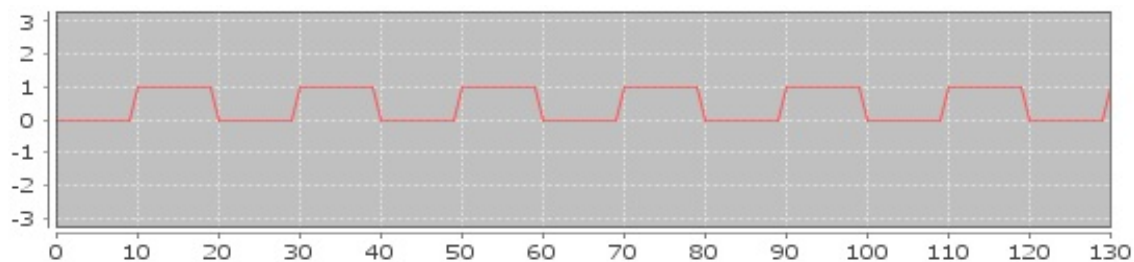
Obr. 2.3: Graf gaussovho šumu

2.1.4 Obdĺžnikový signál

Obdĺžnikový signál (vid' obr. 2.4) patrí medzi neharmonické periodické signály. Jeho signál môžeme rozdeliť na dve časti. V prvej polovici periódy je hodnota signálu nulová a v druhej polovici nadobudá zvolenú hodnotu. Táto zmena signálu nie je skoková, ale je postupná.

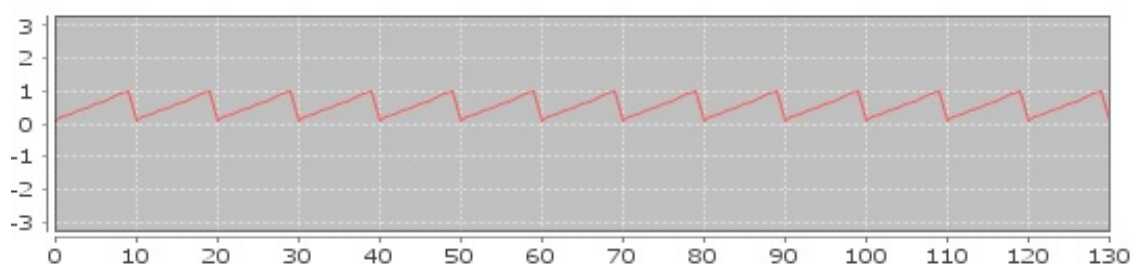
2.1.5 Píla

Pílový signál (vid' obr. 2.5) patrí medzi neharmonické periodické signály, ktorého veľkosť lineárne stúpa dokým nedosiahne maximálnej hodnoty, aby následné mohla



Obr. 2.4: Graf obdĺžnikového signálu

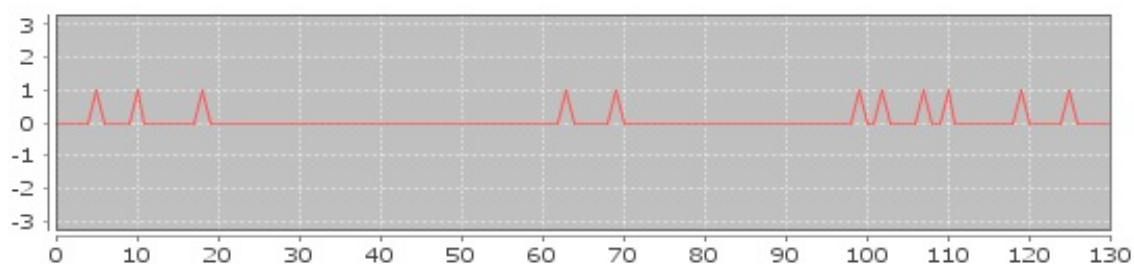
jeho hodnota padnúť na počiatočnú.



Obr. 2.5: Graf pílového signálu

2.1.6 Impulz

Impulzný signál (viď obr. 2.6) je neharmonický neperiodický signál, ktorý je náhodne generovaný v čase t o náhodnom počte n signálov. V tejto práci je možné zvoliť si veľkosť vygenerovaných signálov.



Obr. 2.6: Graf impulzného signálu

2.2 Lineárna kombinácia vektorov

Definícia lineárnej kombinácie n vektorov znie:

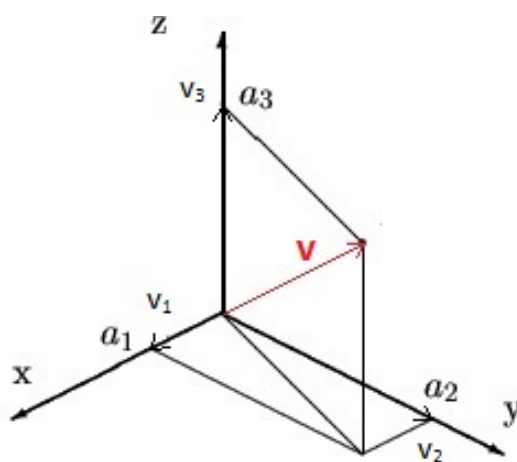
Nech je daných n ľubovoľných vektorov $\vec{v}_1, \vec{v}_2, \dots, \vec{v}_n$.

Každý vektor v tvaru

$$\vec{v} = a_1\vec{v}_1 + a_2\vec{v}_2 + \dots + a_n\vec{v}_n$$

nazývame *lineárnou kombináciou vektorov* (viď obr. 2.7) $\vec{v}_1, \vec{v}_2, \dots, \vec{v}_n$, kde reálne čísla a_1, a_2, \dots, a_n nazývame koeficienty lineárnej kombinácie.

Vieme povedať, že lineárna kombinácia vektorov je jednoduché násobenie a sčítanie, kde vektory násobíme ľubovoľným reálnym číslom a následne tieto násobky sčítame. [13]



Obr. 2.7: Lineárna kombinácia vektorov

2.3 Konvexná kombinácia vektorov

Definícia pre konvexnú kombináciu n vektorov znie:

Nech je daných n ľubovoľných vektorov $\vec{v}_1, \vec{v}_2, \dots, \vec{v}_n$

a reálne čísla $a_1, a_2, \dots, a_n \in R$. Potom výsledný vektor $\vec{v} = a_1\vec{v}_1 + a_2\vec{v}_2 + \dots + a_n\vec{v}_n$, kde $a_1 + a_2 + \dots + a_n = 1$ a navyše platí:

$$a_i \geq 0 \quad i = 1, 2, \dots, n$$

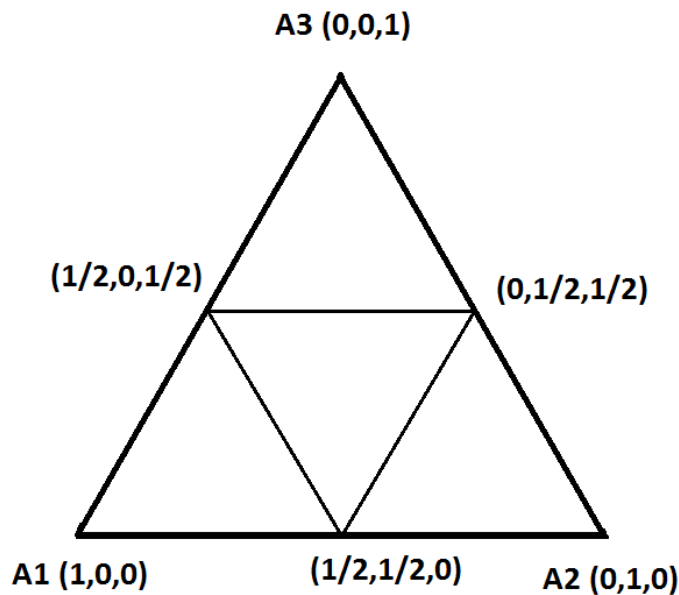
nazývame *konvexnou kombináciou vektorov* $\vec{v}_1, \vec{v}_2, \dots, \vec{v}_n$, kde reálne čísla a_1, a_2, \dots, a_n nazývame koeficienty konvexnej kombinácie.

Vieme povedať, že konvexná kombinácia vektorov je jednoduché násobenie a sčítanie, kedy výsledný súčet všetkých koeficientov vektorov je rovný 1, ale iba za podmienky, že všetky koeficienty vektorov sú reálne čísla z intervalu $[0, 1]$. [3]

2.4 Barycentrické súradnice

Barycentrické súradnice boli objavené v roku 1827 Augustom Ferdinandom Möbiusom. Barycentrické súradnice vyjadruje trojica čísel (t_1, t_2, t_3) reprezentujúci umiestnenie vrchola referenčného trojuholníka A_1, A_2, A_3 .

Pomocou týchto súradníc môžeme určiť bod P , ktorý je geometrickým ťažiskom troch hodnôt a je identifikovaný súradnicami (t_1, t_2, t_3) . Vrcholy trojuholníka sú dané $(1, 0, 0), (0, 1, 0), (0, 0, 1)$ (viď obr. 2.8).[7]



Obr. 2.8: Barycentrické súradnice

V 2D systémoch máme trojuholník definovaný 3 bodmi: $p_1(x_1, y_1), p_2(x_2, y_2), p_3(x_3, y_3)$ a jedným bodom $P(x, y)$. Barycentrické súradnice dovoľujú vyjadriť nové súradnice pre bod P ako lineárnu kombináciu vektorov p_1, p_2, p_3 . Presnejšie, súradnice bodu P si vyjadríme pomocou 3 reálnych skalárov a, b, c nasledujúcim spôsobom:

$$x = ax_1 + bx_2 + cx_3$$

$$y = ay_1 + by_2 + cy_3$$

$$1 = a + b + c$$

Spôsob výpočtu pre a, b, c je nasledujúci:

$$a = \frac{(y_2 - y_3)(x - x_3) + (x_3 - x_2)(y - y_3)}{(y_2 - y_3)(x_1 - x_3) + (x_3 - x_2)(y_1 - y_3)}$$

$$b = \frac{(y_3 - y_1)(x - x_3) + (x_1 - x_3)(y - y_3)}{(y_2 - y_3)(x_1 - x_3) + (x_3 - x_2)(y_1 - y_3)}$$

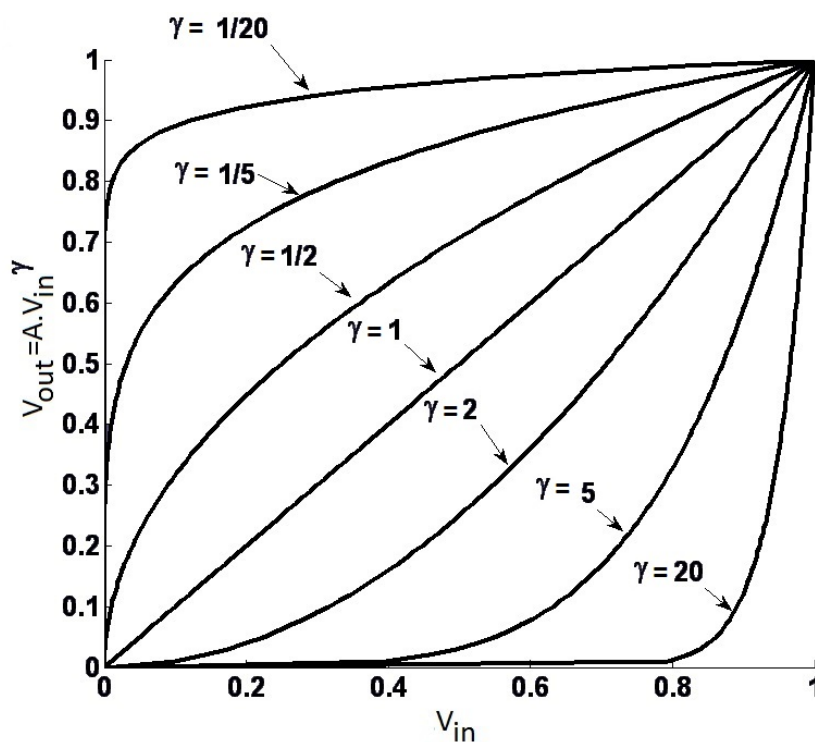
$$c = 1 - a - b$$

Bod P leží v trojuholníku, len ak súčasne platia všetky nasledujúce podmienky:[14]

1. $0 \leq a \leq 1$
2. $0 \leq b \leq 1$
3. $0 \leq c \leq 1$
4. $a + b + c = 1$

2.5 Úprava obrazu pomocou gama korekcie

Gama korekcia charakterizuje reprodukciu tónu stupnice v zobrazovaní systému. Gama zahŕňa jednočíselný parameter, nelineárny vzťah medzi hodnotou kódu (od 0 do 255 v 8-bitovom systéme) a jasom. Takmer všetky kódovacie systémy obrazu sú nelineárne, a preto sa hodnoty gama v jednotlivých systémoch líšia.



Obr. 2.9: Graf gamy korekcie

Hlavným účelom Gamy korekcie vo videu, či obraze je zmena kódu jasú (v pomere k intenzite) na vnemovo-jednotnú doménu, tak ako optimalizovať vnímanie výkonu obmedzením počtu bitov v každom RGB (alebo CMYK) komponentu.

Vzťah medzi vstupným signálom a výstupným signálom (viď obr. 2.9) je definovaný rovnicou[11]:

$$V_{out} = AV_{in}^{\gamma}$$

Kde A je konštanta a vo väčšine prípadov napodúba hodnotu $A = 1$. Hodnoty V_{out} a V_{in} sú typický z rozsahu 0–1 a γ je hodnota samotnej gamy korekcie.

2.6 Aritmetické kódovanie

Aritmetické kódovanie je alternatívou k Huffmanovmu kódovaniu. Odstraňuje niektoré jeho nedostatky – oddeľuje pravdepodobnostný model zdroja od procesu kódovania (preto sa dá ľahšie upraviť na adaptívnu verziu) a nevyžaduje celý počet bitov na kódovanie každého znaku.

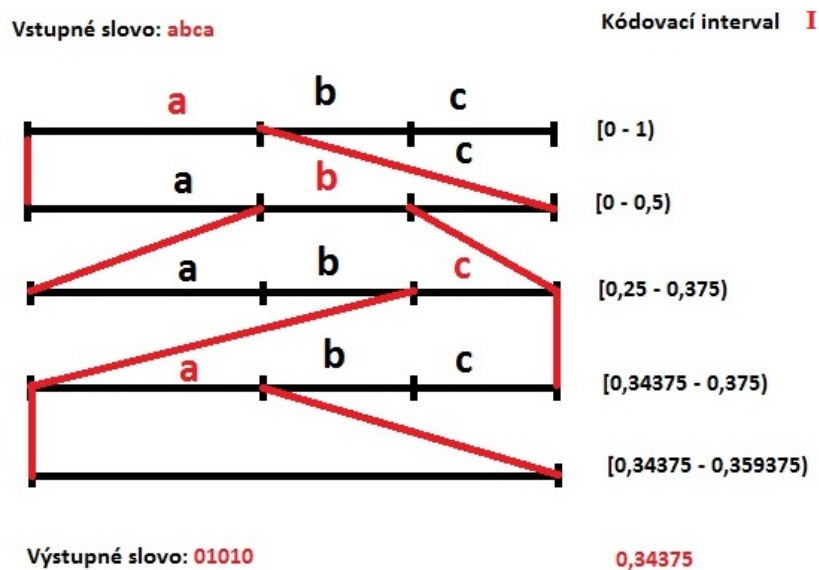
Spoločnou črtou aritmetického a Huffmanovho kódovania je rovnaký model zdroja – pravdepodobnosti výskytov znakov zdrojovej abecedy. Keďže sa pri kódovaní nevyžívajú žiadne kontextové informácie (pozičné závislosti znakov), zvyknú sa označovať ako „zero-order coders“. Do tejto skupiny patrí aj Shannonov-Fanov kód.[15]

Aritmetické kódovanie je špeciálne kódovanie, ktoré nekóduje jednotlivé symboly, ale vytvára kódové slovo pre celú správu. Vstupom do aritmetického kódovania je vstupný text z abecedy zdrojových jednotiek, výstupom je reálne číslo z intervalu 0 (vrátane) až 1.

Pretože všetky čísla zo zadaného intervalu začínajú 0, nie je nutné túto nulu vkladať do kódového slova. U aritmetického kódovania je nutné ďalej uložiť informáciu o počte zakódovaných čísel, aby dekodovací algoritmus včas skončil s delením intervalu. To je možné vyriešiť napríklad dodaním špeciálneho symbolu označujúceho koniec vstupu alebo skorším explicitným uvedením počtu zakódovaných symbolov.[5]

2.6.1 Pseudokód

1. Do výstupu ulož pravdepodobnosť alebo početnosť symbolov.
2. Nastav interval $I = 0$ (vrátane) až 1.
3. Pokiaľ nie je zakódovaný celý text, opakuj:
 - Načítaj ďalší symbol C .
 - Rozdeľ interval I na pod intervaly, ich veľkosti sú úmerné pravdepodobnosti symbolov.
 - Do I ulož pod interval odpovedajúci symbolu C .
4. Výstupom je ľubovoľné číslo z intervalu I bez počiatočnej nuly (viď obr. 2.10).



Obr. 2.10: Graf gamy korekcie

2.6.2 Výstupné slovo

Výstupné slovo aritmetického kódovania sa zvolí z výsledneho intervalu (viď obr. 2.10). Následne sa prevádza do binárneho tvaru, ktorý však môže obsahovať veľké množstvo znakov. Preto sa výsledne slovo orezáva a to podľa nasledujúceho vzťahu:

$$\log_2 \left[\frac{1}{\text{konec_interval} - \text{start_interval}} \right] + 1 = X$$

Kde X označuje dekadické číslo určujúce na koľko znakov sa oreže binárne číslo.

3 POPIS POUŽITÉHO SOFTWARE

3.1 Java

Java je programovací jazyk vyvinutý firmou *Sun Microsystems*, neskôr kúpený firmou *Oracle*. Je to objektovo orientovaný jazyk vychádzajúci z C++, ku ktorému má taktiež syntakticky najbližšie, ak teda nepočítame C#, ktorý vznikol až po príchode Javy.

Oproti svojmu predchodcovi, Java neobsahuje niektoré konštrukcie, ktoré spôsobovali pri programovaní v C++ najväčšie problémy a navyše pridáva mnoho užitočných vlastností, napr.:

- Pridelovanie a uvoľňovanie pamätí je tu obstarané automaticky (pomocou garbage collectoru). Objekt sa neruší pomocou *delete* alebo *free* (), iba sa „ponúkne“ ku zrušeniu (napr. priradením neplatnej referencii null).
- Klasický problém ukazovateľa z C/C++, je tu celkom odstránený, pretože ten sa tu jednoducho nenachádza (resp. nahradený referenciami). Dereferencovanie má samozrejme na starosť runtime. Programátor je tak ušetrený notorickej chyby zápisu pointeru mimo dátovú oblasť.
- Je implementovaný mechanizmus vlákien (*threads*) a je teda možné spúšťať viac úloh v rámci jedného programu. Nezabudlo sa samozrejme ani na ich synchronizáciu pomocou tzv. *monitorov*.
- Vytvorené objekty sa dajú automaticky serializovať, tj. ukladať do súboru, zasielať po sieti a pod.
- Je možné vykonávať reflexiu, čiže zisťovanie informácií o objekte (aké má premenné, metódy a pod.).
- Implementovaný mechanizmus výnimiek, takže všetky runtime chyby je možné odchytiť a spracovať. Výnimky sú samozrejme objektové, takže je možné zachytiť aj celú hierarchiu výnimiek v jednom pozorovanom bloku.
- Značným uľahčením práce pre programátorov je obsiahlosť štandardne dodávaných knižníc, s ktorými sa nemôže zrovnávať pravdepodobne žiaden bežne používaný programovací jazyk.
- Java podporuje tvorbu dynamicky rozšíriteľných aplikácií (plug-in a pod.).
- Java kladie značný dôraz na bezpečnosť najmä vďaka tomu, že je prekladaná do bytcodeu a implementovaná bezpečnostným mechanizmom (podpisovanie kódov a pridelovanie práv pre rôzne akcie). Je možné zaistiť, že program v Jave, ktorý si užívateľ stiahne zo siete, mu nezformuje disk, nebude komunikovať so žiadnym iným počítačom, ale iba s počítačom, z ktorého pochádza atď.

- Všetky tieto výhody je možné získať veľmi rýchlo. Java je jazyk veľmi jednoduchý, prehľadný a zrozumiteľný. Pri osvojovaní základov Javy, spraví začiatočník podstatne menej chýb ako pri štúdiu C/C++.
- Veľkou výhodou Javy je taktiež jej hardwarová nezávislosť, pretože je prekladaná do špeciálneho medzikódu (bytecod). Ten je na konkrétnom počítači alebo zariadení (PC, handheld, mobilný telefón a pod.) interpretovaný, prípadne za behu prekladaný do natívneho kódu (tzv. JIT - Just-In-Time compiler).
- Programátor môže napísať javovský program, napr. na PC pod Windowsom a spustiť ho aj na PC s Linuxom, na Macu, proste všade, kde je k dispozícii Java runtime.

K dispozícii sú napr. tieto knižnice pre tvorbu:

1. grafického užívateľského rozhrania (GUI)
2. vstup/výstup
3. prácu s textom
4. komunikáciu s SQL
5. prácu s komprimovanými súborami
6. a iné

Pre písanie programov je k dispozícii celá rada vývojových prostredí. Najčastejšie sú v praxi používané tieto:[12]

1. Eclipse - opensource - zdarma
2. IDEA - základná verzia zdarma
3. NetBeans - opensource, zdarma

3.2 Java Applet

Java applet je applet napísaný v programovacom jazyku Java. Sťahuje sa zo servera vo forme Bytecodu. Java applety sa spúšťajú buď vo webovom prehliadači alebo v aplikácii AppletViever od spoločnosti Sun Microsystem.

Pri návšteve webovej stránky, ktorá obsahuje applet, sa applet natiahne do internetového prehliadača. Následne je applet spúšťaný v programe Virtual Java Machine, ktorý sa do počítača inštaluje spolu s prehliadačom.

U niektorých prehliadačov (napr. Microsoft Internet Explorer) nie je inštalácia Virtual Java Machine (terminológia Microsoftu uvádza Microsoft Virtual Machine) obsiahnutá v typickej inštalácii.

Applet môže mať sprístupnené niektoré svoje vlastnosti a metódy ako verejné, čo umožňuje externým programátorom manipuláciu s nimi. [9]

Z bezpečnostných dôvodov platia pre applet niektoré obmedzenia a naopak má applet niektoré funkcie rozšírené:

- Applet nemôže nahrávať knižnice ani definovať natívne metódy.
- Applet nemôže nadväzovať sieťové spojenia na iný ako domovský server.
- Applet nemôže zapisovať do súboru na strane klienta (prehliadača).
- Applet nemôže spúšťať programy na domovskom serveri.
- Applet nemôže čítať niektoré systémové premenné.
(napr. metóda `System.getProperty()`)
- Applet môže prehrávať zvuky.
- Applet môže požiadať browser o zobrazenie ľubovoľnej WWW stránky.
- Applet môže volať verejné metódy appletov umiestnených na tej istej WWW stránke.

Niektoré prehliadače umožňujú uvedené obmedzenia nastaviť individuálne pre vybrané applety.[8]

3.3 Štruktúra appletu

Základnú štruktúru appletu tvorí trieda `java.applet.Applet`, ktorá definuje základné metódy tvoriace rozhranie medzi prehliadačom a appletom. Program, ktorý má fungovať ako applet, musí byť potomkom tejto triedy. Applet je spustený v grafickom kontexte a je úzko spojený s „oknovou“ knižnicou AWT.

Trieda `java.applet.Applet` je potomkom triedy `java.awt.Panel`, ktorá umožňuje appletu vlastniť komponenty užívateľského rozhrania, vykonávať grafický výstup a zachytávať udalosti z klávesnice a myši.

Životný cyklus appletu závisí na prehliadači, ktorý v priebehu svojej činnosti volá tieto metódy appletu:

- `public void init ()` - pri inicializácii
- `public void start ()` - pri spustení
- `public void paint (java.awt.Graphics g)` - pri prekresľovaní
- `public void stop ()` - pri zastavení
- `public void destroy ()` - pri ukončení

Tieto metódy sú v triede `java.applet.Applet` definované ako prázdne a pre zmysluplnú činnosť je potrebné v potomkovi prekryť aspoň jednu z nich. Kostra appletu, ktorý implementuje všetky základne metódy, vyzerá takto:

```
public class NovyApplet extends java.applet.Applet

    public void init() {
// kód prevedený pri inicializácii
    }

    public void start()
{ // kód prevedený pri spustení
    }

    public void stop() {
// kód prevedený pri zastavení
    }

    public void destroy() {
// kód prevedený pri ukončení
    }

    public void paint(java.awt.Graphics g) {
// kód prevedený pri prekresľovaní
    }
}
```

Rozdiel medzi zastavením a zrušením spočíva v tom, že pri zrušení appletu dochádza navyše k ukončeniu jeho vlákien a odstráneniu appletu z pamäti. Pri zastavení appletu môžu jeho vlákna pokračovať v činnosti.

U niektorých prehliadačov môže po zastavení appletu dôjsť automaticky k jeho zrušeniu (po metóde `stop ()` sa bezprostredne volá metóda `destroy ()`). [8]

3.4 Knižnica JFreeChart

Javovská knižnica JFreeChart je zadarmo a slúži na zobrazovanie grafov v profesionálnej kvalite v java aplikáciách. Zahŕňa rozsiahlu sadu funkcií, napr.:

- konzistentné a dobre zdokumentované API, podporujúce širokú škálu typov grafov
- flexibilný dizajn, ktorý je ľahko rozširiteľný ako na strane servera, tak na strane klienta
- podporuje mnoho typov výstupu, napr.:
 1. swing komponenty
 2. obrazové súbory (PNG, JPEG ...)
 3. formáty vektorovej grafiky (PDS, EPS a SVG)
- JFreeChart je „open source“ alebo lepšie povedané „free software“. Je distribuovaný pod podmienkami GNU Lesser General Public License (LGPL), čo umožňuje použitie vo vlastných aplikáciách.[6]

3.5 NetBeans

NetBeans je Open Source projekt s veľmi rozsiahlou užívateľskou základňou, rastúcou komunitou vývojárov a takmer 100 partnerov po celom svete. V roku 2000 sa stal hlavný sponzor firma Sun Microsystems.

Dnes existujú dva produkty:

1. Vývojové prostredie NetBeans (NetBeans IDE)
2. Vývojová platforma NetBeans (NetBeans Platform)

Vývojové prostredie NetBeans IDE je nástroj, pomocou ktorého programátori môžu písať, prekladať, ladiť a distribuovať aplikácie. Samotné vývojové prostredie je vytvárané v jazyku Java - avšak podporuje prakticky akýkoľvek programátorský jazyk.

Existuje tak isto veľké množstvo modulov, ktoré toto vývojové prostredie rozširujú. Vývojové prostredie NetBeans je bezplatne šíriteľný produkt a jeho užívanie nie je nijak obmedzené.

Okrem vývojového prostredia je taktiež dostupná vývojová platforma Net-Beans Platform, čo je modulárny a rozširiteľný základ pre vytváranie rozsiahlych počítačových aplikácií. Nezávislí dodávatelia softwaru ponúkajú dodatočné moduly, ktoré je možné ľahko integrovať a taktiež môžu byť použité k vývoji ich vlastných nástrojov a riešení.[10]

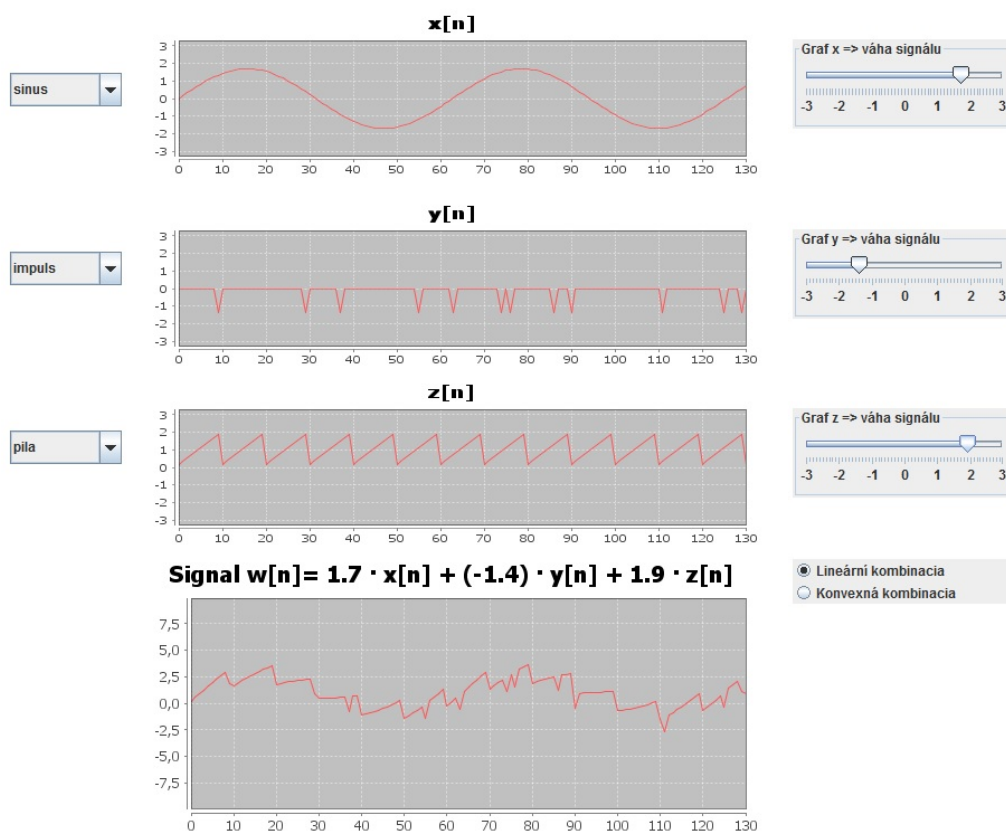
4 PRAKTICKÁ ČASŤ

V praktickej časti mojej diplomovej práci som sa zaoberal vytvorením Java appletov, ktoré majú slúžiť predovšetkým na študijné účely. Táto práca popisuje nasledujúce applety:

1. applet na simuláciu lineárnej a konvexnej kombinácie signálov
2. applet na úpravu obrazu pomocou gama korekcie
3. applet znázorňujúci aritmetické kódovanie
4. applet s indexáciou farieb

Všetky applety som napísal vo vývojovom prostredí NetBeans. Pri tvorení tejto práce som narazil na niekoľko problémov, ktoré som zaznamenal a ich popis sa nachádza v sekcii s prílohami.

4.1 Applet: Lineárna a konvexná kombinácia



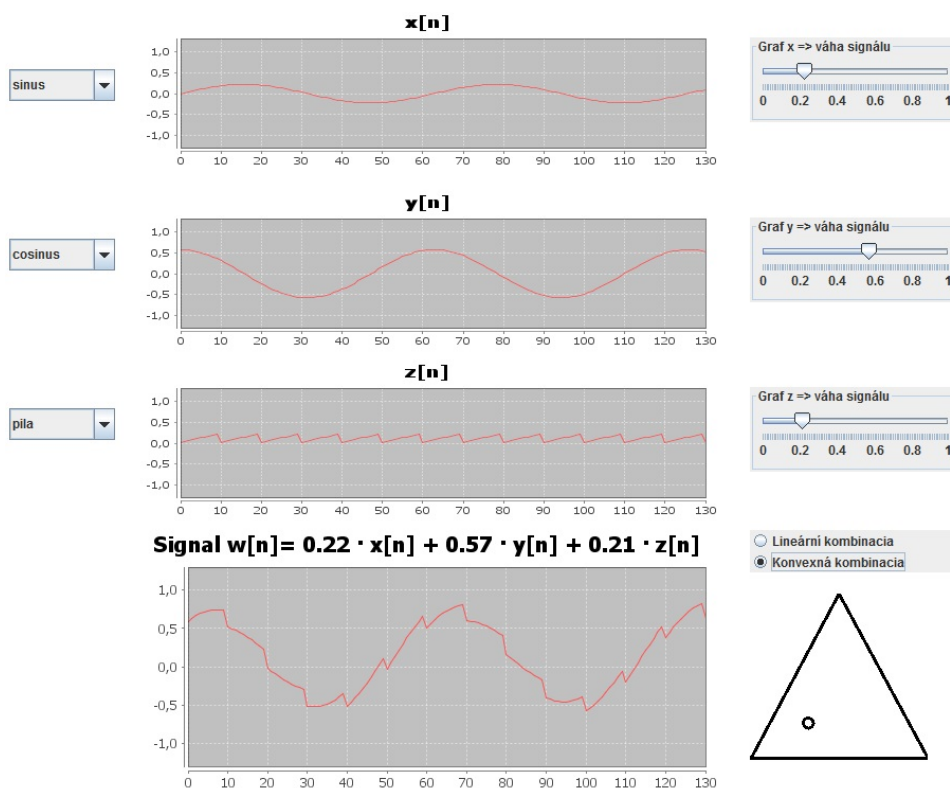
Obr. 4.1: Lineárna kombinácia signálov

V applete je možné simulovať lineárnu kombináciu signálov a konvexnú kombináciu signálov. Užívateľ má možnosť výberu z viacerých druhov signálov, pri ktorých môže nastavovať silu jednotlivých signálov.

4.1.1 Popis appletu

Samotný applet môžeme rozdeliť na 2 hlavné časti:

1. Lineárnu kombináciu (viď obr. 4.1)
2. Konvexnú kombináciu (viď obr. 4.2)



Obr. 4.2: Konvexná kombinácia signálov

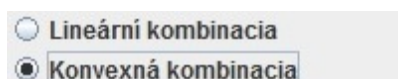
Celý applet by sme si mohli rozdeliť na niekoľko častí:

1. Možnosť voľby typu kombinácie
2. Výber signálu
3. Graf vybraného signálu
4. Posuvník na zmenu veľkosti signálu
5. Barycentrický trojuholník
6. Graf výslednej kombinácie signálov

Možnosť voľby typu kombinácie

Pri spustení appletu je pôvodne nastavené, aby sa signály spočítavali pomocou lineárnej kombinácie.

Pokiaľ chceme prepnúť na konvexnú kombináciu, stačí zaškrtnúť túto voľbu ako je znázornené na obr. 4.3.

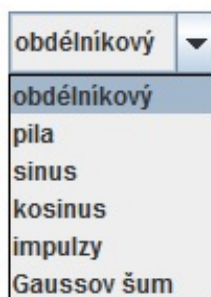


Obr. 4.3: Voľba kombinácie

Pri prepínaní medzi jednotlivými kombináciami sa nastavené hodnoty ne-strácajú, ale uchovávajú sa v pamäti PC.

Výber signálu

V applete sú 3 okná slúžiace na výber signálu. Pri samotnom štarte appletu majú prednastavené signály. V prvom okne je navolený sínusový signál, v druhom kosínus a v poslednom okne je zvolený signál píla. Každé okno pre výber signálu je pridružené práve k jednému grafu a jednému posuvníku.



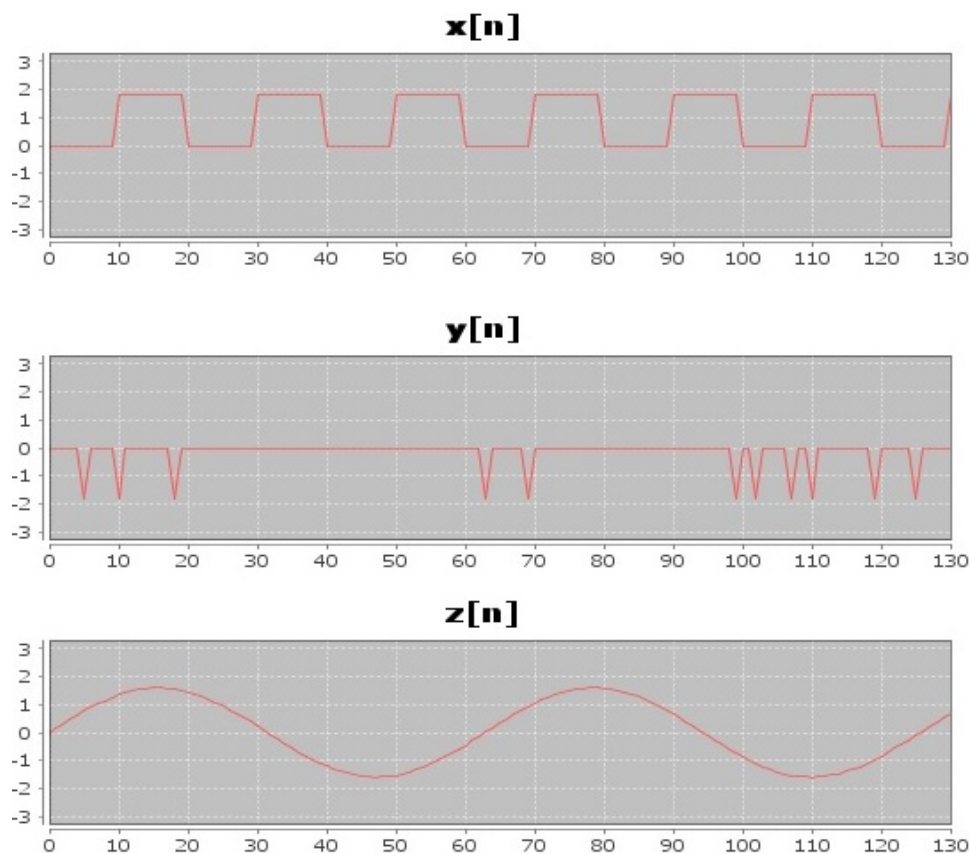
Obr. 4.4: Výber signálu

Celkovo je na výber 6 rôznych signálov (viď obr. 4.4):

1. obdĺžnikový
2. pílový
3. sínusový
4. kosínusový
5. impulsový
6. Gaussov šum

Graf vybraného signálu

Po vybraní daného typu signálu sa ten následne zobrazí v grafe (viď obr. 4.5). Každý graf vykresľuje 131 hodnôt, ktorými je určený daný signál pri rozsahu od -3 do +3. Pre vykresľovanie v Jave bola použitá knižnica *JFreeChart*.

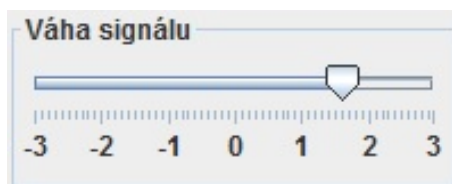


Obr. 4.5: Graf vybraného signálu

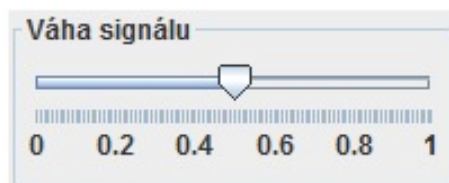
Posuvník na zmenu veľkosti signálu

U každého vybraného signálu je možné meniť jeho váhu. Pri lineárnej kombinácii môžeme upravovať daný signál v rozsahu od -3 do +3 (viď obr. 4.6). Pôvodne je každý posuvník pre všetky signály nastavený na hodnotu 1.

Pri konvexnej kombinácii môžeme upravovať daný signál v rozsahu 0 až 1 (viď obr. 4.7). Prvé dva posuvníky sú prednastavené na hodnotu 0.5 a posledný na hodnotu 0.



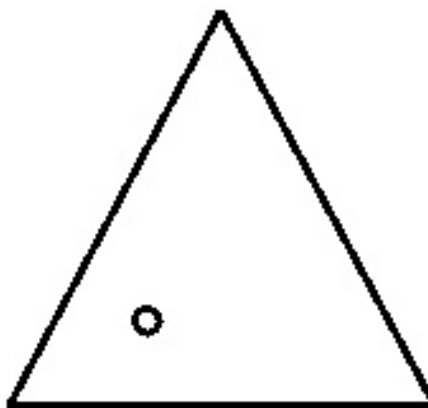
Obr. 4.6: Posuvník pre lineárnu kombináciu



Obr. 4.7: Posuvník pre konvexnú kombináciu

Barycentrický trojuholník

Pri konvexnej kombinácii je aj druhá možnosť ako nastavovať veľkosť váh signálov. Môžeme na to využiť práve barycentrický trojuholník (vid' obr. 4.8), kde sa nastavujú váhy pomocou barycentrických súradníc.



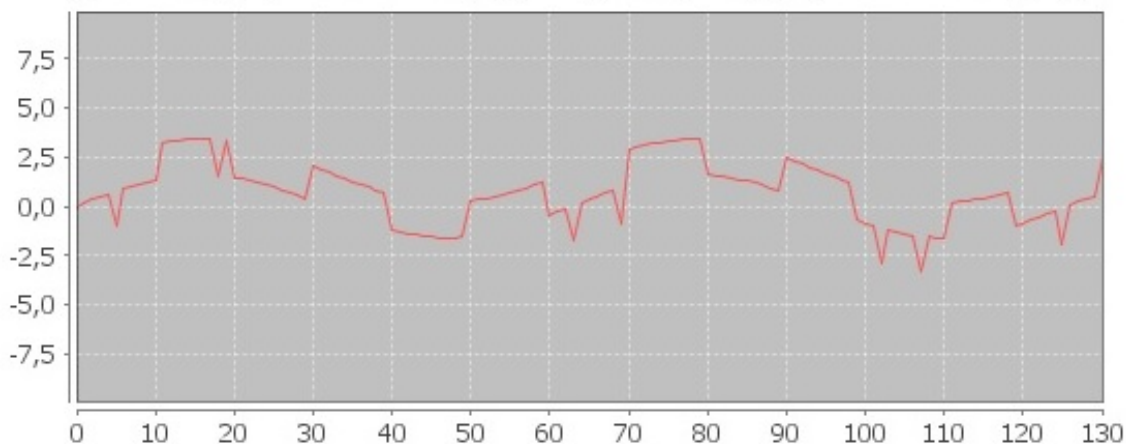
Obr. 4.8: Barycentrický trojuholník

Samotný trojuholník zobrazuje len ohraničenie, kde je možné „kliknúť“ aby sa nastavili požadované váhy signálov (signálom x , y a z odpovedajú skaláre a , b , c z kapitoly 2.4). Po kliknutí do trojuholníka sa táto zmena automaticky zobrazí aj na posuvníkoch a v trojuholníku sa znázorni kruh, ktorý označuje miesto kliku (vid' obr. 4.8).

Grafy výsledných kombinácií signálov

Zvolené signály ($x[n]$, $y[n]$, $z[n]$) sú vynásobené koeficientmi určenými posuvníkmi. Potom sa tieto signály sčítajú. Výsledný graf kombinácie je možné vidieť na obr. 4.9.

$$\text{Signál } w[n] = 1.8 \cdot x[n] + (-1.8) \cdot y[n] + 1.6 \cdot z[n]$$



Obr. 4.9: Graf výsledného signálu

Pri vykresľovaní grafu konvexnej kombinácii musí byť najskôr splnená podmienka, že súčet všetkých koeficientov signálov je rovný 1. V opačnom prípade sa zobrazí na mieste grafu upozorňujúce hlásenie (vid obr. 4.10).

$$\text{Signál } w[n] = 0.5 \cdot x[n] + 0.5 \cdot y[n] + 0.03 \cdot z[n]$$

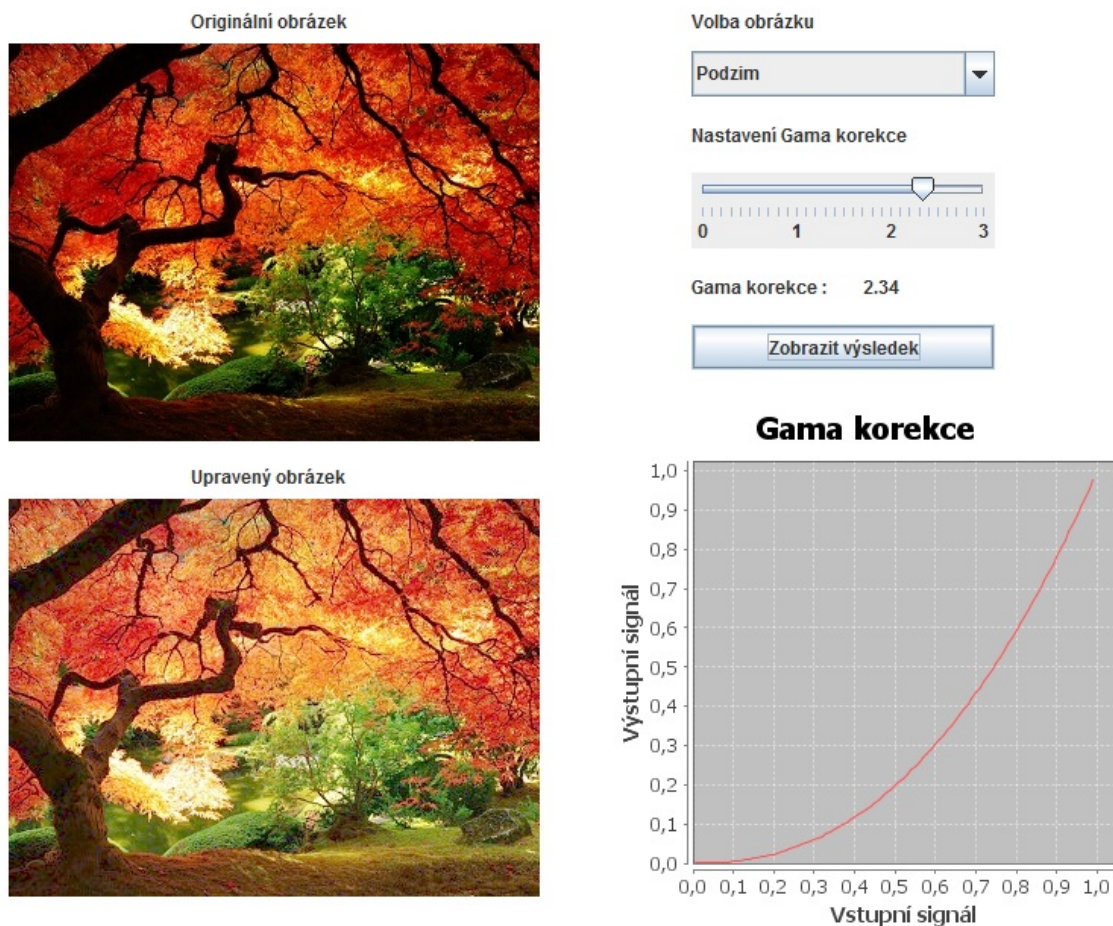
Není splněna podmínka součtu koeficientů 1

Součet zesílení x,y a z: 1.03

Obr. 4.10: Graf chybovej hlášky

4.2 Applet: Gama korekcia

V applete má užívateľ možnosť vybrať si z viacerých obrázkov, pri ktorých môže meniť úroveň gama korekcie. Samotný applet je možné vidieť na obr. 4.11



Obr. 4.11: Applet gama korekcia

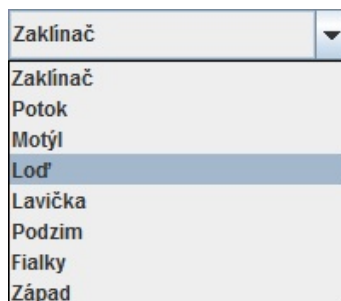
4.2.1 Popis appletu

Applet si môžeme rozdeliť na nasledujúce časti:

1. Vstupný obrázok
2. Výstupný obrázok
3. Možnosť výberu obrázku
4. Nastavenie veľkosti gamy
5. Graf korekcie
6. Tlačidlo na zobrazenie výsledku

Možnosť výberu obrázku

Na výber máme 8 rôznych obrázkov (viď obr. 4.12)



Obr. 4.12: Výber obrázku

Po vybratí obrázku sa automatický zobrazí nový a teda zmení predchádzajúci nastavený obraz.

Nastavenie veľkosti gamy

Pri zapnutí appletu je prednastavená hodnota gama korekcie na hodnotu 2, 2. Veľkosť gamy korekcie nastavujeme pomocou posuvníka (viď obr. 4.13), ktorý ma rozsah od 0 do 3 pri citlivosti 0,01.

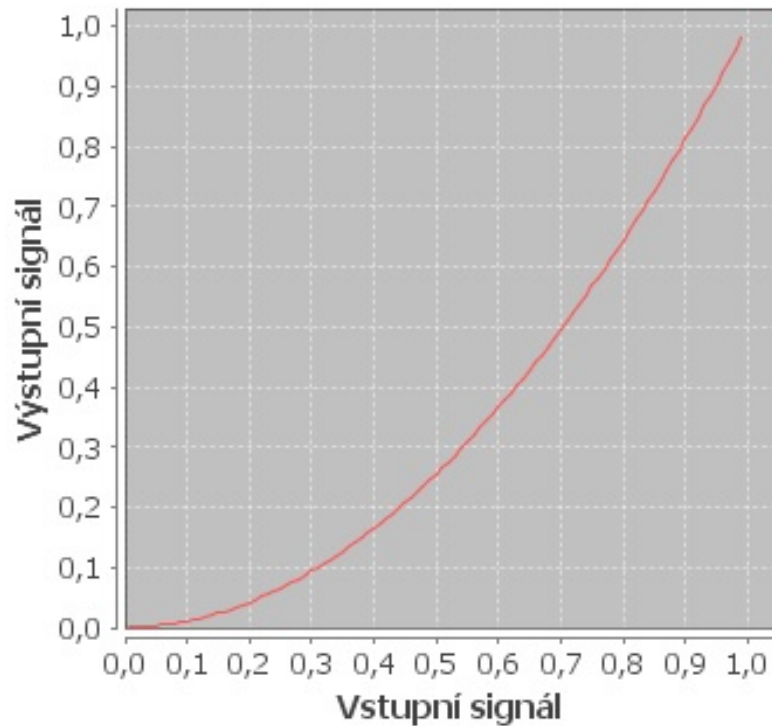


Obr. 4.13: Výber obrázku

Graf korekcie

Na obr. 4.14 je možné vidieť graf gamy korekcie, ktorý zobrazuje zmenu výstupného obrazu podľa veľkosti zvolenej hodnoty gamy (nastavuje sa pomocou posuvníka).

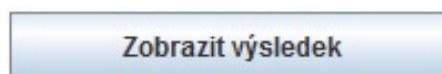
Gama korekce



Obr. 4.14: Graf gamy korekcie

Tlačidlo na zobrazenie výsledku

Po zvolení veľkosti gammy korekcií si pomocou tohto tlačidla zobrazíme výslednú úpravu obrázku a zároveň aj graf korekcie. Tlačidlo na zobrazenie výsledku je možné vidieť na obr. 4.15.



Obr. 4.15: Tlačidlo na zobrazenie výsledku

4.3 Applet: Aritmetické kódovanie

V applete je možné vyskúšať si aritmetické kódovanie zadaním vstupného slova a sledovať ako sa mení výsledný interval každým krokom vypočítavania.

Spustiť Krok Reset

Vstupní slovo
abcaab

Kódovaný znak

Znak	Procento	Start	Konec

Krokovacie slovo

Výstupní slovo

Obr. 4.16: Applet aritmetické kódovanie

4.3.1 Popis appletu

Applet si môžeme rozdeliť na 4 hlavné časti:

1. Ovládacie tlačidlá
2. Vstupné pole
3. Štatistické údaje
4. Výstupné pole

Znak	Procento	Start	Konec
a	50.0		
b	33.33333333333333		
c	16.666666666666664		

Obr. 4.17: Štatistické údaje

Po štarte appletu je vo vstupnom poli označenom ako „Vstupní slovo“ prednastavené „abcaab“ (vid' obr. 4.16). Tlačidlom „Spustit“ spúšťame celý applet. Automaticky sa vypíše štatistická tabuľka (vid' obr. 4.17) obsahujúca informácie o početnosti každého znaku. Po kliknutí na tlačidlo „Spustit“ sa toto tlačidlo deaktivuje, ale aktivuje sa tlačidlo „Krok“ a „Reset“.

Kódovaný znak **abcaab**

Znak	Procento	Start	Konec
a	50.0	0.3888888889	0.3958333334
b	33.33333333333333	0.3958333334	0.400462963
c	16.666666666666664	0.400462963	0.4027777778

Obr. 4.18: Štatistické údaje

Pomocou tlačidla Krok následne posúvame celý výpočet po jednom kroku. Pri každom kliknutí na tlačidlo „Krok“ sa menia štatistické údaje v tabuľke teraz už obsahujúcej aj informácie o rozsahu intervalu a zároveň sa krokovacie slovo blíži k hodnote výstupného slova. Zároveň sa zobrazuje aktuálne kódovaný znak (na obrázku znázornený červene vid' obr. 4.18).

Krokovacie slovo

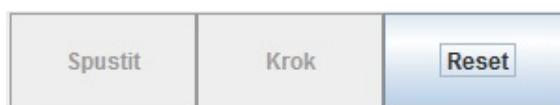
011001000111000111000111000111001101001110000011000110

Výstupní slovo

011001000

Obr. 4.19: Applet aritmetické kódovanie

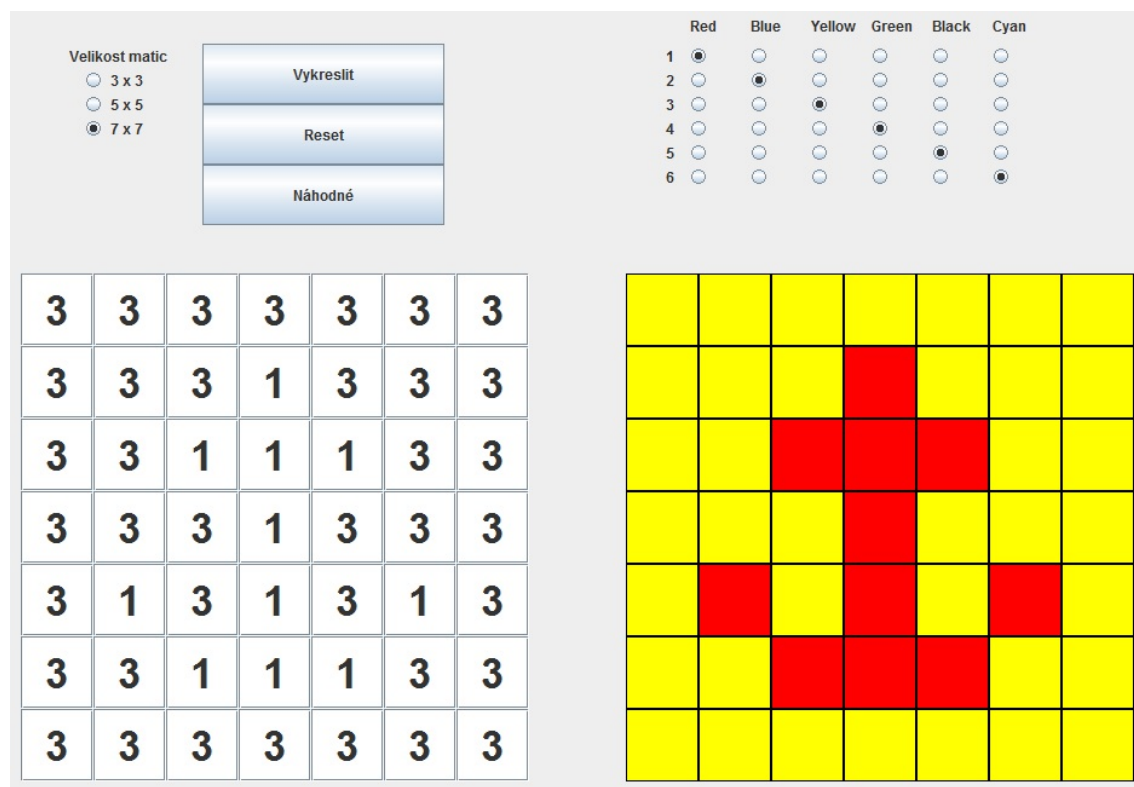
Ak dôjde kódovací proces na koniec, tlačidlo „Krok“ sa deaktivuje. Výsledné slovo a krokovacie slovo nemá vždy totožnú hodnotu. Výsledné slovo sa orezáva na zmenšený počet bitov tak, aby každé číslo z výsledneho intervalu mohlo byť vyjadrené rovnakou bitovou postupnosťou (vid' obr. 4.19). Nakoniec môžeme všetko vyresetovať pomocou tlačidla „Reset“.



Obr. 4.20: Applet aritmetické kódovanie

4.4 Applet: Indexácia farby

V tomto applete si môžeme vyskúšať indexáciu farieb, kde je možné priradiť rôznym číslam rôzne farby a následne to vykresliť v matici. Samotný applet je možné vidieť na obr. 4.21.



Obr. 4.21: Applet Indexácie farieb

4.4.1 Popis appletu

Applet môžeme rozdeliť na nasledujúce časti:

1. výber veľkosti matice
2. indexovanie farieb
3. ovládacie tlačidlá
4. vykresľovacie matice

Výber veľkosti matice

Na obr. 4.22 je možné vidieť, že na výber matice máme tri možnosti. Pôvodne je nastavená matica 7x7.



Obr. 4.22: Výber veľkosti matic

Indexovanie farieb

Užívateľ má možnosť prideliť 6 rôznym číslam 6 rôznych farieb. Každému číslu môže byť pridelená maximálne 1 farba, ale 1 farba môže byť pridelená viacerým číslam. Pri štarte appletu je nastavená indexácia farieb pomocou diagonály, ako je možné vidieť na obr. 4.23

	Red	Blue	Yellow	Green	Black	Cyan
1	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
2	<input type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
3	<input type="radio"/>	<input type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
4	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>
5	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>
6	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input checked="" type="radio"/>

Obr. 4.23: Výber farieb

Ovládacie tlačidlá

Pri štarte appletu si užívateľ môže zvoliť veľkosť matice, prideliť číslam určité farby a následne ich môže zapísať do editovateľnej matice. Ak užívateľ chce vykresliť maticu je nutné stlačiť tlačidlo „Vykresliť“. Ak sa mu obrazec nepozdáva, môže ho celý vymazať pomocou tlačidla „Reset“. Ak užívateľ nemá jasnú predstavu vyplnenia matice, má možnosť kliknúť na tlačidlo „Náhodné“ (viď obr. 4.24) a matica sa vyplní náhodnými číslami (farby musí nastaviť sám).

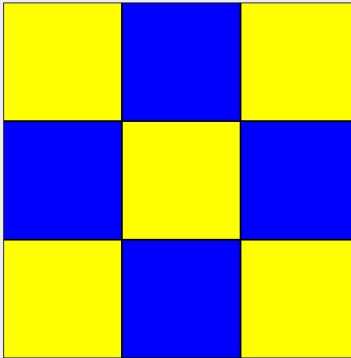


Obr. 4.24: Ovládacie tlačidlá

Vykresľovacie matice

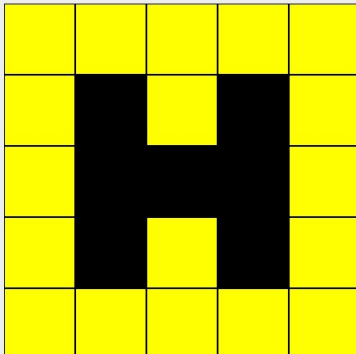
Pre každú veľkosť matice je nastavené prednastavené vykreslenie (vid' obr. 4.22). Matica obsahujúca čísla je editovateľná. Užívateľ má možnosť sám zmeniť čísla podľa jeho vlastnej predstavy.

3	2	3
2	3	2
3	2	3



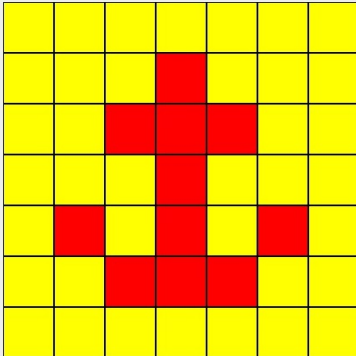
Veľkosť matice 3x3

3	3	3	3	3
3	5	3	5	3
3	5	5	5	3
3	5	3	5	3
3	3	3	3	3



Veľkosť matice 5x5

3	3	3	3	3	3	3
3	3	3	1	3	3	3
3	3	1	1	1	3	3
3	3	3	1	3	3	3
3	1	3	1	3	1	3
3	3	1	1	1	3	3
3	3	3	3	3	3	3



Veľkosť matice 7x7

Obr. 4.25: Defaultne zobrazené matice

5 ZÁVER

Výsledkom mojej diplomovej práce sú 4 funkčné webové applety napísané v programe NetBeans, ktoré v budúcnosti budú slúžiť ako podpora vo výuke. Vytvorené sú nasledujúce applety:

Applet znázorňujúci kombinácie signálov

V applete je možné simulovať lineárnu a konvexnú kombináciu signálov. Popis jednotlivých kombinácií je možné vidieť v kapitole 3. Užívateľ si môže vybrať zo 6 typov signálov, pri ktorých je možné meniť veľkosť zvolených signálov pomocou posuvníkov. Pri lineárnej kombinácii je daný rozsah od -3 do +3 a pri konvexnej kombinácii je rozsah od 0 do 1. Bližší popis fungovania appletu sa nachádza v kapitole 4.1.

Applet na úpravu obrazu pomocou gama korekcie

V applete si môže užívateľ zvoliť 1 z 8 rôznych obrázkov, na ktorých je možné vyskúšať zmenu zobrazenia obrázka pri rôznych úrovniach gama korekcie. Veľkosť gama korekcie je možné regulovať na posuvníku, ktorý má rozsah od 0 do 3. Užívateľ má možnosť porovnávať upravený obrázok s originálnym obrázkom. V grafe je možné vidieť krivku gama korekcie. Bližší popis fungovania appletu sa nachádza v kapitole 4.2.

Applet znázorňujúci aritmetické kódovanie

Ako už názov napovedá v tomto applete je možné si vyskúšať aritmetické kódovanie. Po zadaní vstupného slova (možné použiť celú abecedu aj číselné znaky) môže užívateľ sledovať po 1 kroku ako prebieha samotné kódovanie a to v tabuľke zobrazujúcej interval pre každý použitý znak, v ktorom sa momentálne nachádza. Výstupné slovo je zobrazené v spodnej časti appletu v binárnej podobe. Bližší popis fungovania appletu sa nachádza v kapitole 4.3.

Applet s indexáciou farieb

Pri tomto applete má užívateľ možnosť zvoliť si jednu z 3 veľkostí vykresľovanej matice. V editovateľnej matici je možné zadávať čísla od 1 do 6. Ku každému číslu je možné prideliť iba 1 farbu z 6 rôznych farieb, ale zároveň platí, že 1 farbu môže prideliť ku všetkým číslam. Bližší popis fungovania appletu sa nachádza v kapitole 4.4.

Všetky applety sú umiestnené na stránke vedúceho práce:
<http://www.utko.feec.vutbr.cz/~rajmic/applets/>

LITERATÚRA

- [1] *Aritmetické kódování* [online]. [cit. 2014-05-26]. Dostupné z: <<http://martin.lipinsky.cz/skola/pt/HTML/54/default.htm>>
- [2] BOBOT, V a JAKUBEKOVÁ, M. *Interaktívne vyučovanie v školských vzdelávacích programoch* [online]. [cit. 2014-05-26]. Dostupné z: <http://www.mpc-edu.sk/library/files/publik_ciaivvsvp_finalversion_na_web-29_10_2012.pdf>
- [3] BUSA, J. *Mini minimalizácia* [online]. 2011 [cit. 31. 12. 2013]. Dostupné z URL: <<http://web.tuke.sk/fei-km/old/OM/OMa4.pdf>>
- [4] CATTIN, P.: *Image Restoration: Introduction to Signal and Image Processing* [online]. 2008 [cit. 31. 12. 2013]. Dostupné z URL: <<http://miac.unibas.ch/SIP/06-Restoration.html>>
- [5] HORDĚJČUK, V. *Aritmetické kódování* [online]. [cit. 2014-05-21]. Dostupné z: <<http://old.voho.cz/wiki/informatika/kodovani/aritmeticke/>>
- [6] *JFreeChart: Welcome To JFreeChart!* [online]. 2005 [cit. 31. 12. 2013]. Dostupné z URL: <<http://www.jfree.org/jfreechart/>>
- [7] JULES, C. *Accurate point in triangle test* [online]. 25. január 2014 [cit. 2014-05-21]. Dostupné z: <<http://totologic.blogspot.fr/2014/01/accurate-point-in-triangle-test.html>>
- [8] KOTALA, Z. a P. TOMAN. *Applet* [online]. 04. február 2001 [cit. 31. 12. 2013]. Dostupné z URL: <<http://v1.dione.zcu.cz/java/sbornik/17.html>>
- [9] MOROKES, D. *Java Applet krok za krokem* [online]. 27. september 2002 [cit. 31. 12. 2013]. Dostupné z URL: <<http://interval.cz/clanky/java-applet-krok-za-krokem/>>
- [10] *NetBeans* [online]. 2013 [cit. 31. 12. 2013]. Dostupné z URL: <https://netbeans.org/index_cs.html>
- [11] POYNTON, Charles. *The rehabilitation of gamma* [online]. [cit. 2014-05-21]. Dostupné z: <http://www.poynton.com/PDFs/Rehabilitation_of_gamma.pdf>
- [12] *Programovací jazyk JavaTM* [online]. 2013 [cit. 31. 12. 2013]. Dostupné z URL: <<http://v1.dione.zcu.cz/java/uvod.html>>

- [13] VODIČKOVÁ, V.: *Lineárna kombinácia vektorov* [online]. 4. apríl 2013
[cit. 31. 12. 2013]. Dostupné z URL:
<<http://www.sportgymke.sk/mvd/AnalytickaGeometria/LinearnaKombinacia-Vektorov.pdf>>
- [14] WEISSTEIN, E. *Barycentric Coordinates* [online]. 16. máj 2014
[cit. 2014-05-21]. Dostupné z:
<<http://mathworld.wolfram.com/BarycentricCoordinates.html>>
- [15] STANEK, M. *Aritmetické kodovanie a bwt: Kódovanie a kryptológia*. 2002.

A PRÍLOHY

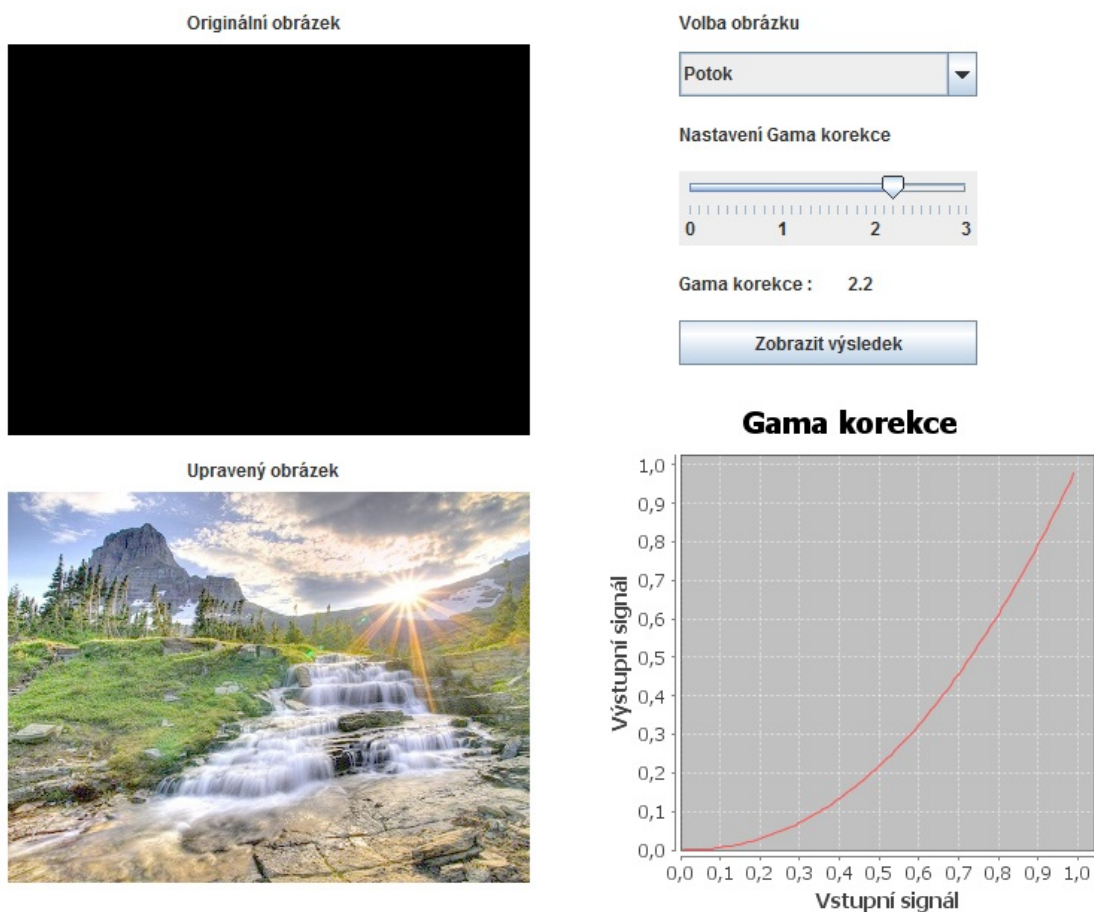
A.1 Možné problémy

Pri vytváraní appletu znázorňujúceho gama korekciu som narazil na niekoľko chýb.

1. Pod operačným systémom
 - Windows 8
 - Windows 8.1
 - Windows Vista
2. Nastavenie Java security

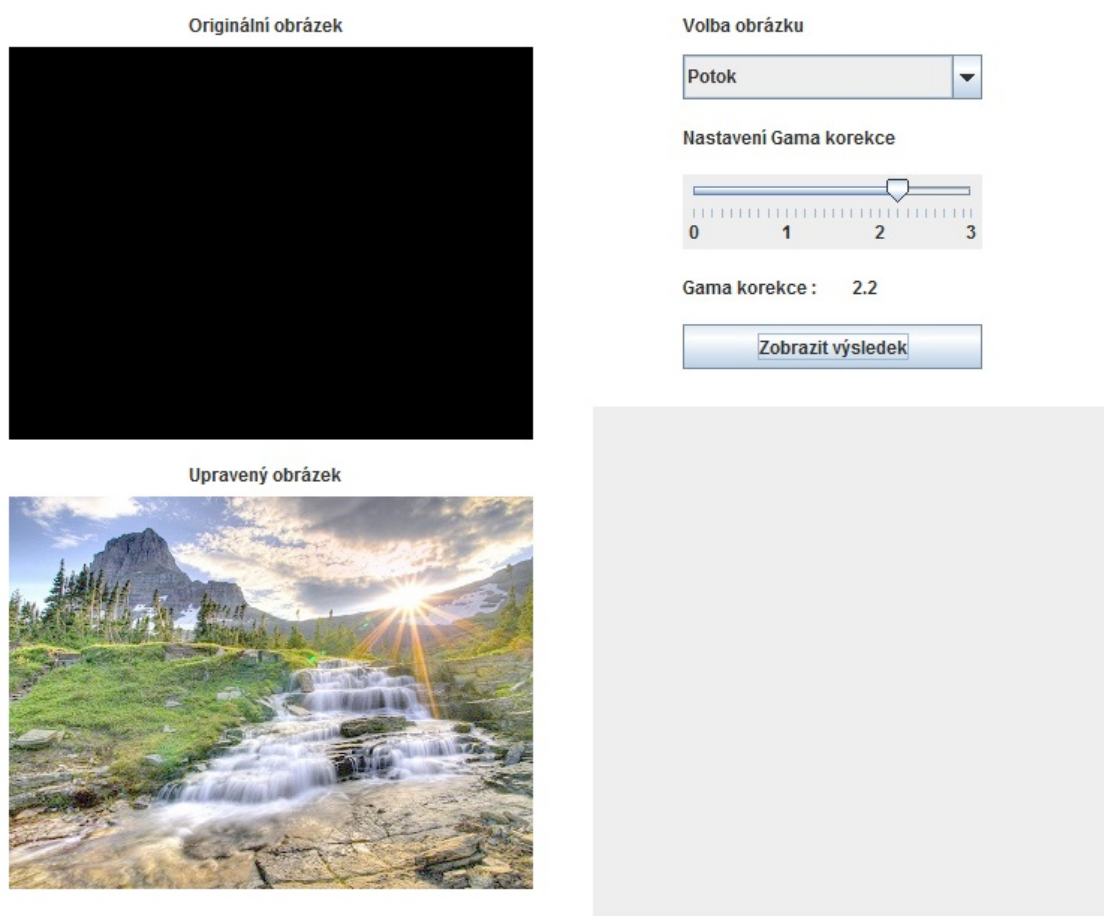
A.1.1 Pod operačným systémom

Je chyba vo vykresľovaní appletu po spustení, resp. pri zmene z pôvodného nastaveného obrázka na vybraný obrázok (viď obr. A.1).



Obr. A.1: Chybné vykreslenie appletu 1

Zároveň pri kliknutí na tlačidlo „zobraziť výsledok“, opätovne nevykreslí všetky obrázky korektne (vid' obr. A.2).



Obr. A.2: Chybné vykreslenie appletu 2

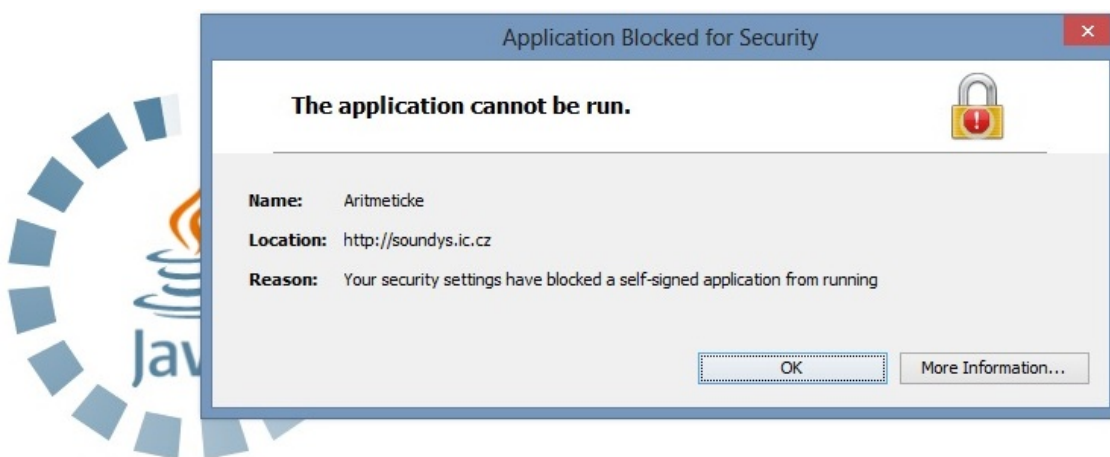
A.1.2 Nastavenie Java security

Pokiaľ je na PC nainštalovaná najnovšia verzia JAVY (dnešným dňom je vydaná verzia 7u51), nastáva problém s bezpečnostnými pravidlami JAVY, ktoré sú implementované s najnovšou verziou.

Po spustení akéhokoľvek appletu Vám vyskočia 2 po sebe vyskakovacie okná znemožňujúce spustenie appletu (vid' obr. A.3 a obr. A.4).



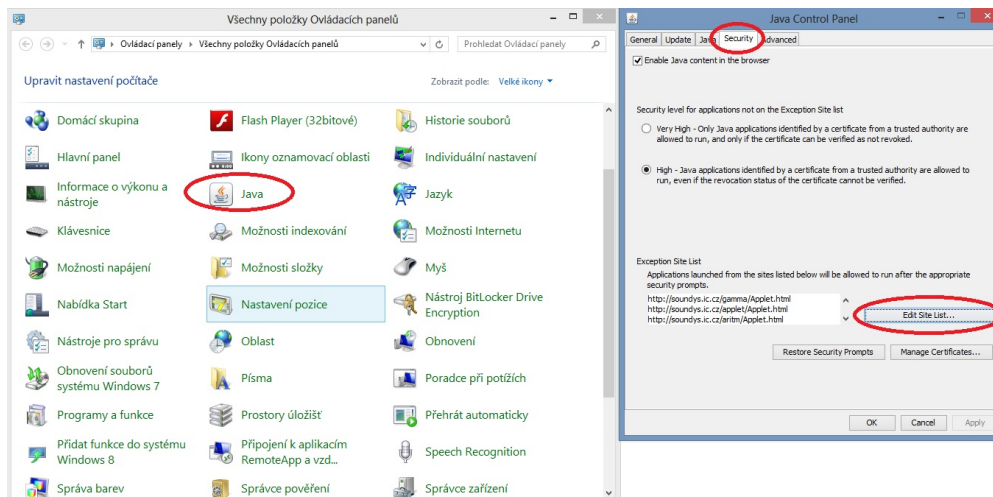
Obr. A.3: Blokovanie appletu 1



Obr. A.4: Blokovanie appletu 2

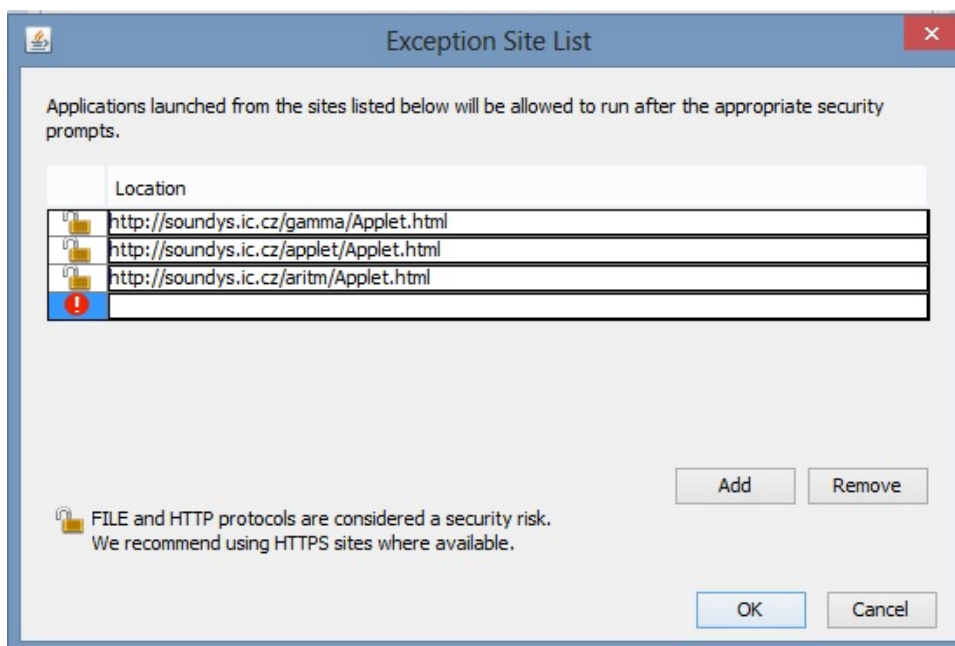
Na zobrazenie appletu je nutné spraviť nasledujúce kroky.

1. Otvoriť ovládacie panely → Java → Security → Edit Side List (viď obr. A.5)



Obr. A.5: Nastavovanie JAVA security 1

2. Po kliknutí na Edit Side List sa objaví nižšie uvedené okno, v ktorom je potrebné kliknúť na tlačítko „Add“ a do voľného riadku zadať URL adresu daného Java Appletu a potvrdiť tlačidlom OK (viď obr. A.6).



Obr. A.6: Nastavovanie JAVA security 2

3. Po potvrzení vyskočí oznamovacie okno, kde posledný krát potvrdíte dôveryhodnosť daného appletu (viď obr. A.7).



Obr. A.7: Nastavanie JAVA security 3

B PRÍLOHY

B.1 obsah CD

Priložené CD obsahuje zdrojové kódy ku všetkým 4 appletom a obrázky použité v jednotlivých appletoch.

Cesta ku zdrojovým kódom a knižniciam pre jednotlivé applety je nasledujúca:

1. applet na simuláciu lineárnej a konvexnej kombinácie signálov
 - ./applety/konvoluce/Applet3/dist/Applet4.jar
 - ./applety/konvoluce/lib
2. applet na úpravu obrazu pomocou gama korekcie
 - ./applety/gamma/test/test.jar
 - ./applety/gamma/lib
3. applet znázorňujúci aritmetické kódovanie
 - ./applety/AppletAritmeticke/dist/AppletAritmeticke.jar
4. applet s indexáciou farieb
 - ./applety/IndexovaneBarvy/IndexovaneBarvy/dist/IndexovaneBarvy.jar

Pri nahrávaní appletu gama korekcie na FTP server je nutné skopírovať obrázky do adresára pomenovaného „test“, ktorý sa musí nachádzať v koreňovom adresári obsahujúcom zdrojový súbor „test.jar“. Zároveň je nutné skopírovať do koreňového adresára aj zložku „lib“ obsahujúcu dodatočné knižnice. Skopírovať zložku „lib“ do koreňového adresára je nutné aj pri applete na simuláciu lineárnej a konvexnej kombinácie signálov.