VŠB – Technická univerzita Ostrava

Fakulta elektrotechniky a informatiky

DIPLOMOVÁ PRÁCE

2018 Bc. Patrik Mik

VŠB – Technická univerzita Ostrava
Fakulta elektrotechniky a informatiky
Katedra kybernetiky a biomedicínského inženýrství

Systém pro rozpoznání předmětů na základě obrazového signálu

System for Subject Detection based on Image Signal Recognition

2018 Bc. Patrik Mik

VŠB - Technická univerzita Ostrava
Fakulta elektrotechniky a informatiky
Katedra kybernetiky a biomedicínského inženýrství

Zadání diplomové práce

Student: **Bc. Patrik Mik**

Studijní program: N2649 Elektrotechnika

Studijní obor: 2612T041 Řídicí a informační systémy

Téma: Systém pro rozpoznání předmětů na základě obrazového signálu
System for Subject Detection based on Image Signal Recognition

Jazyk vypracování: čeština

Zásady pro vypracování:

1. Rozbor problematiky obrazové detekce s popisem metod.
2. Návrh systému pro rozpoznání předmětů na základě obrazového signálu.
3. Realizace systému pro rozpoznání předmětů na základě obrazového signálu.
4. Měření a analýza s vyhodnocením skutečných parametrů systému.
5. Zhodnocení dosažených výsledků závěrečné práce.

Seznam doporučené odborné literatury:

[1] SOJKA, Eduard. *Digitální zpracování a analýza obrazů.* 1. vyd. Ostrava: VŠB-Technická univerzita, 2000. ISBN 80-707-8746-5.
[2] HLAVÁČ, Václav a Miloš SEDLÁČEK. *Zpracování signálů a obrazů.* Vyd. 2., přeprac. Praha: Vydavatelství ČVUT, 2005. ISBN 80-010-3110-1.
[3] CYGANEK, Bogusław. *Methods and algorithms of object recognition in digital images.* Krakow: AGH University of Science and Technology Press, 2009. ISSN 0867-6631.

Formální náležitosti a rozsah diplomové práce stanoví pokyny pro vypracování zveřejněné na webových stránkách fakulty.

Vedoucí diplomové práce: **Ing. Zdeněk Macháček, Ph.D.**

Datum zadání: 01.09.2016

Datum odevzdání: 30.04.2018

VYSOKÁ ŠKOLA BÁŇSKÁ-TECHNICKÁ UNIVERZITA Fakulta elektrotechniky a informatiky 1D16 OSTRAVA

doc. Ing. Jiří Koziorek, Ph.D.
vedoucí katedry

prof. Ing. Pavel Brandštetter, CSc.
děkan fakulty

Prohlášení

Prohlašuji, že jsem tuto diplomovou práci vypracoval samostatně. Uvedl jsem všechny literární prameny a publikace, ze kterých jsem čerpal.

V Ostravě dne 30. 4. 2018

Podpis:.....

Poděkování

Rád bych poděkoval Ing. Zdeňku Macháčkovi, PhD., za cenné rady, věcné připomínky a vstřícnost při konzultacích a při tvorbě diplomové práce.

Abstrakt

Tato diplomová práce se zabývá zpracováním obrazu, konkrétně rozpoznáváním předmětů na základě obrazového signálu. V práci jsou popsány základní vlastnosti digitálního obrazu, teoretické znalosti předzpracování digitálního obrazu a metody segmentace objektů v obraze. Praktická část práce se zabývá realizací knihovny pro zpracování obrazu v jazyce C++ ve vývojovém prostředí Microsoft Visual Studio a následné využití knihovny pro implementaci systému vizuálního rozhodčího stolní hry Qwirkle. Systém vizuálního rozhodčího využívá metody segmentace na principu prahování barev herních kamenů s následným rozpoznáním tvarů herních kamenů založeném na podobnosti se vzory. Vytvořený systém je doplněn o vhodné uživatelské rozhraní. Výsledky testování algoritmu jsou součástí práce.

Klíčová slova

Digitální obraz, segmentace, rozpoznání objektu, korelace, rozpoznání tvaru, rozpoznání barvy, C++, Qwirkle

Abstract

This diploma thesis deals with image processing, particularly about the subject detection based on image signal recognition. The basic features of digital image, the theoretical knowledge of digital image preprocessing and methods of segmentation of objects in the picture are described. The practical part deals with the implementation of the C ++ image processing library in the Microsoft Visual Studio development environment and the subsequent use of the library for the implementation of visual referee system in the Qwirkle board game. The Visual Referee System uses image segmentation based on the thresholding colour method of the playing tiles with subsequent recognition of the shape of the playing tiles based on similarity to the patterns. The created system is complemented by an appropriate user interface. The outcome of an algorithm test is a part of this diploma thesis as well.

Keywords

Digital image, Image segmentation, Subject Detection, Shape Recognition, Colour Recognition, C++, Qwirkle

Obsah

1. Úvod 12
2. Digitální obraz 13
2.1 Reprezentace obrazu 13
2.2 Barevné modely 14
3. Předzpracování obrazu 17
3.1 Bodové jasové transformace 17
3.2 Lokální jasové transformace - Obrazové filtry 18
3.3 Matematická morfologie 21
4. Segmentace obrazu 24
4.1 Prahování 24
4.2 Detekce hran 25
4.3 Houghova transformace 27
4.4 Spojování oblastí 29
4.5 Split & Merge - štěpení a spojování oblastí 29
5. Popis a klasifikace objektů 30
5.1 Základní radiometrické vlastnosti 30
6. Metody detekce 32
6.1 Voila-Jones detektor 32
6.2 HoG detektor 34
7. Návrh systému vizuálního rozhodčího 36
7.1 Hra Qwirkle 36
7.2 Návrh systému 37
7.3 Knihovna pro zpracování obrazu 37
8. Systém vizuálního rozhodčího 44
8.1 Detekce provedení tahu 44
8.2 Předzpracování obrazu 44
8.3 Rozpoznání barvy 49
8.4 Rozpoznání tvaru 50
8.5 Herní analýza 51
8.6 Grafické uživatelské rozhraní 58

8.7 Konstrukce 62
9. Testování systému 65
9.1 Detekce a rozpoznání herních kamenů 65
9.2 Výpočetní čas systému 66
10. Závěr 69
Literatura 71
Seznam příloh 73

Seznam použitých zkratek a symbolů

C	–	Počet sloupců
CMYK	–	Barevný model (Cyan – azurová, Magenta – purpurová, Yellow – žlutá)
C++	–	Počet sloupců
$f(x, y)$	–	Dvourozměrná obrazové funkce
GUI	–	Graphical User Interface – Grafické uživatelské rozhraní
h	–	Výška binární oblasti
$h(i, j)$	–	Dvourozměrná funkce konvolučního jádra
HOG	–	Histogram of Oriented Gradients – Histogram orientovaných gradientů
HSV	–	Barevný model (Hue – odstín barvy, Saturation – sytost barvy, Value – jas)
$H(q)$	–	Histogram s počtem q úrovní
M	–	Výška obrazu
N	–	Šířka obrazu
px	–	Pixel
R	–	Počet řádků
RGB	–	Barevný model (Red – červená, Green – zelená, Blue – modrá)
S_x, S_y	–	Souřadnice středu
t	-	Čas [sekunda]
T	–	Velikost prahu
w	–	Šířka binární oblasti
x	–	Souřadnice v rovině - index sloupce obrazové matice
y	–	Souřadnice v rovině - index řádku obrazové matice
σ	–	Směrodatná odchylka Gaussova filtru

Seznam ilustrací

Obr. 2-1 Konvence souřadnic digitálního obrazu [10] ... 13
Obr. 2-2 a) Aditivní míchání barev, b) Subtraktivní míchání barev [17] ... 14
Obr. 2-3 Barevný model RGB [15] ... 15
Obr. 2-4 Barevný model CMY [15] ... 16
Obr. 3-1 Lineární zpracování obrazu [1] ... 19
Obr. 3-2 Strukturální element a) disk, b) čtverec, c) linie (0°), d) linie (90°), e) diamant ... 21
Obr. 3-3 Dilatace – Výchozí binární obraz ... 22
Obr. 3-4 Dilatace – Aplikace strukturálního elementu a) čtverec (3), b) disk (2), c) linie (3, 0°) ... 22
Obr. 3-5 Eroze – Výchozí binární obraz ... 23
Obr. 3-6 Eroze – Aplikace strukturálního elementu a) čtverec (3), b) disk (2), c) linie (3, 0°) ... 23
Obr. 4-1 Jasové profily nejběžnějších hran [1] ... 25
Obr. 4-2 Odezvy skokové 1D funkce na její 1. a 2. derivaci [1] ... 26
Obr. 4-3 Houghova transformace pro hledání přímky ... 28
Obr. 4-4 Split & Merge – Quad-tree [9] ... 29
Obr. 6-1 Základní Haarovy příznaky ... 32
Obr. 6-2 Integrální obraz ... 33
Obr. 7-1 Qwirkle - Herní kameny ... 36
Obr. 7-2 Blokový návrh systému vizuálního rozhodčího ... 37
Obr. 7-3 Detekce uzavřených obastí - Vstupní binární obraz ... 38
Obr. 7-4 Detekce uzavřených oblastí - Maska kontroly sousedních pixelů ... 39
Obr. 7-5 Detekce uzavřených oblastí – 1. fáze ... 39
Obr. 7-6 Detekce uzavřených oblastí - Princip korekce ... 40
Obr. 7-7 Detekce uzavřených oblastí - Výsledný obraz ... 40
Obr. 7-8 Směry procházení obrazu ... 41
Obr. 7-9 Parametry oblastí ... 42
Obr. 7-10 Výpočet vzdálenosti a úhlu dvou oblastí ... 43
Obr. 8-1 Binární obraz po prahování ... 45
Obr. 8-2 Binární obraz po extrakci extrémů ... 46
Obr. 8-3 Detekce uzavřených oblastí ... 46
Obr. 8-4 Natočení obrazu podle oblastí ... 47
Obr. 8-5 Detekce špatně umístěného herního kamene do mřížky ... 48

Obr. 8-6 Vzdálenosti herních kamenů podle jejich velikosti 49
Obr. 8-7 Natočený obraz s nalezenými oblastmi 49
Obr. 8-8 Sada vytvořených vzorů 50
Obr. 8-9 Princip rozpoznání symbolů herních kamenů 51
Obr. 8-10 Princip vytváření herního pole 52
Obr. 8-11 Fáze herního pole a) Vytvořené herní pole, b) Vyříznuté herní pole, c) Dekódované herní pole 53
Obr. 8-12 Princip průchodu řady ve vertikálním směru 56
Obr. 8-13 Skládání kamenů 57
Obr. 8-14 Grafické uživatelské rozhraní vizuálního rozhodčího 58
Obr. 8-15 Dialogové okno pro výběr hráče začínající hru 59
Obr. 8-16 Dialogové okno pro manuální změnu herního kamene v poli 60
Obr. 8-17 GUI - parametrizace 61
Obr. 8-18 Chybová hláška 61
Obr. 8-19 Návrh modelu konstrukce 62
Obr. 8-20 Stavebnicový systém hliníkových profilů 63
Obr. 8-21 Skutečný stav konstrukce 64
Obr. 9-1 Graf závislosti času zpracování obrazu na počtu detekovaných herních kamenů 67
Obr. 9-2 Graf závislosti času analýzy hry na počtu detekovaných herních kamenů 67
Obr. 9-3 Graf závislosti času vykreslení herního pole na počtu detekovaných herních kamenů 68
Obr. 9-4 Graf závislosti celkového času algoritmu na počtu detekovaných oblastí 68

Seznam tabulek

Tab. 1 Výsledky testování algoritmu 65
Tab. 2 Výsledné průměrné časy algoritmu..... 66

1. ÚVOD

Tématem diplomové práce je systém pro rozpoznání předmětů na základě obrazového signálu. Cílem této práce je seznámení se s principy obrazové detekce s popisem metod, jehož nedílnou součástí je celý řetězec zpracování obrazové informace. Součástí této práce je také návrh a realizace systému vizuálního rozhodčího pro deskovou hru Qwirkle v jazyce C++, bez možnosti využít některou z dostupných knihoven pro zpracování obrazu.

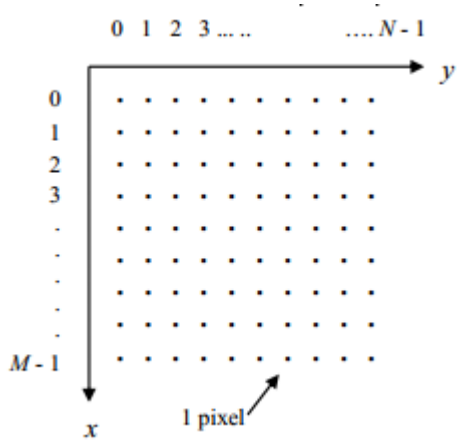
Tato problematika spadá do širší oblasti počítačového vidění, která se snaží technickými prostředky alespoň částečně napodobit lidské vidění a vnímání. Počítačové vidění je v dnešní době nezbytné v oblastech medicíny (detekce nádorů, nebo jiných maligních změn tkáně), v oblasti výroby (kontrola kvality výrobků), až po oblast armádní (navádění autonomních vozidel, bezpilotních letounů, apod.).

Práce má za úkol popsat jednotlivé kroky řetězce zpracování obrazu, od snímá obrazu, až po samotnou klasifikaci objektů, které jsou nutné pro detekci a rozpoznání. Práce je proto rozdělena do deseti kapitol, včetně úvodní kapitoly. Ve druhé a třetí kapitole získá čtenář teoretické znalosti o samotné reprezentaci digitálního obrazu, barevných modelech, základních jasových transformacích obrazu využívaných pro předzpracování obrazu a morfologických operacích. Čtvrtá kapitola se věnuje metodám segmentace obrazu. Pátá kapitola se zabývá popisem objektů v obraze a jejich následnou klasifikací. V šesté kapitole má čtenář možnost se seznámit s nejrozšířenějšími metodami pro detekci objektů v obraze. Sedmá kapitola se zabývá návrhem systému vizuálního rozhodčího společně s realizací knihovny pro zpracování obrazu v jazyce C++ v programovacím prostředí Microsoft Visual Studio. Osmá kapitola pak obsahuje samotnou implementaci vizuálního rozhodčího stolní hry Qwirkle s využitím vytvořené knihovny. Tato kapitola je rozdělena do čtyř větších podkapitol, přičemž první podkapitola popisuje realizaci algoritmu pro nalezení a rozpoznání herních kamenů, druhá podkapitola popisuje metody pro práci s herním polem, ve třetí je popsáno grafické uživatelské rozhraní a čtvrtá kapitola obsahuje návrh konstrukce. V posledních dvou kapitolách je pak algoritmus testován a zhodnocen.

2. DIGITÁLNÍ OBRAZ

2.1 Reprezentace obrazu

Obraz je definován jako dvourozměrná funkce $f(x, y)$, kde x a y jsou souřadnice v rovině a amplituda f pro každou dvojici souřadnic (x, y) nazývaná jas, intenzita, nebo úroveň šedi obrazu v tomto bodě. Jestliže x, y a hodnoty f jsou konečné a diskrétní, potom obraz nazýváme digitální obraz.



Obr. 2-1 Konvence souřadnic digitálního obrazu [10]

Je to tedy množina bodů uspořádaných do matice $M \times N$, kde M představuje celočíselný počet řádků a N počet sloupců matice. Každý bod této matice je nazýván pixel, nebo také obrazový prvek a je nejmenší jednotkou digitálního obrazu charakterizovaný svou intenzitou. Každý pixel si „nese“ svou zakódovanou barevnou informaci. Na základě barevné informace lze rozlišit tři základní druhy digitálního obrazu:

- **Binární obraz**

Binární obraz je definován jako matice $M \times N$, kde prvky této matice mohou nabývat pouze hodnot 1 nebo 0, popřípadě 255 nebo 0, reprezentující bílou a černou barvu. Bitová hloubka u těchto obrazů je rovna jedné. Binární obrazy se zpravidla používají pro definici obrazových masek.

- **Monochromatický obraz**

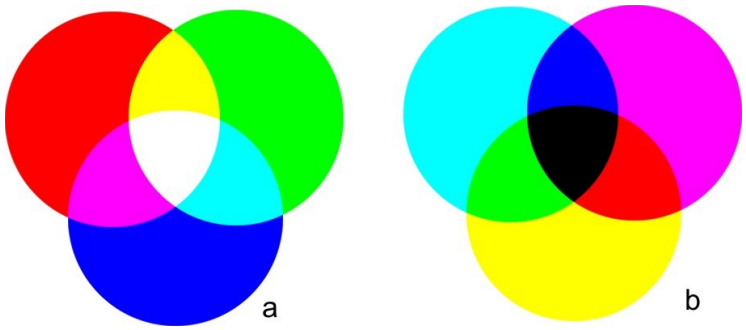
Obraz monochromatický (šedo-tónový), je definován jako matice $M \times N$, kde prvky této matice mohou nabývat hodnot v rozsahu 0-255. Kde bílá je reprezentována hodnotou 255 a černá hodnotou 0.

- **Barevný obraz**

V případě barevného obrazu se jedná o diskrétní funkci, kde každé dvojici plošných souřadnic (x, y) odpovídá vektor hodnot pro jednotlivé komponenty obrazu. Takovýto obraz je možné reprezentovat trojrozměrnou maticí $M \times N \times L$, kde L je počet komponent obrazu v závislosti na využívaném barevném modelu obrazu. V případě RGB barevného obrazu je tento obraz reprezentován maticí o velikosti $M \times N \times 3$. Výsledná barva daného pixelu je dána kombinací tří složek obrazu, pomocí kterých je možné vytvořit celou škálu barevných odstínů. Způsob míchání, určení základních barev, ze kterých se budou ostatní barvy skládat a poměr jednotlivých barev určují barevné modely.

Jsou rozlišovány dva základní typy míchání barev:

- **Aditivní míchání**: Výsledná barva je dána součtem tří základních barev. Pokud jsou všechny tři barvy zastoupeny s maximální intenzitou, je získána bílá barva, naopak pokud všechny barvy mají nulovou intenzitu, výslednou barvou je barva černá. (*Obr. 2-2a*)
- **Subtraktivní míchání**: Výsledná barva je dána rozdílem tří doplňkových barev. Pokud jsou všechny tři barvy zastoupeny s maximální intenzitou, je získána černá barva, naopak pokud všechny mají nulovou intenzitu, výslednou barvou je barva bílá. (*Obr. 2-2b*)



Obr. 2-2 a) Aditivní míchání barev, b) Subtraktivní míchání barev [17]

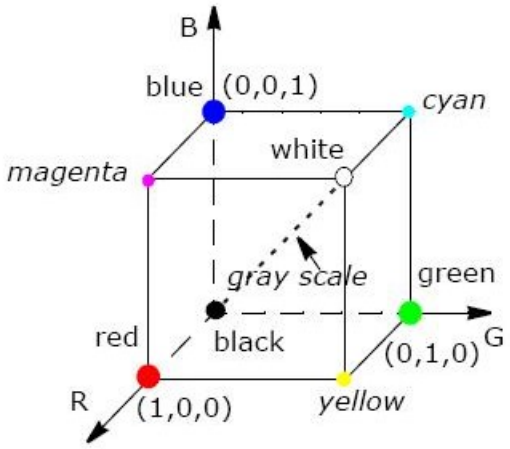
2.2 Barevné modely

Všechny barvy barevného obrazu jsou tvořeny kombinací několika veličin, v závislosti na barevném modelu. V následující kapitole jsou popsány tři základní barevné modely a jejich využití při míchání základních barev.

2.2.1 RGB model

Jedná se o barevný model skládající se ze tří komponent a je založen na aditivním míchání barev: červené - R, zelené – G a modré – B. RGB model je možné reprezentovat krychlí v kartézské soustavě souřadnic, kde každá z os udává intenzitu jedné ze tří barevných komponent modelu (Obr. 2-3). Při míchání dochází k vytvoření vektoru o třech složkách v intervalu $< 0{,}1 >$, vyjadřující podíl velikosti dané barvy. Nulová intenzita všech tří barevných komponent reprezentuje černou barvu, která je

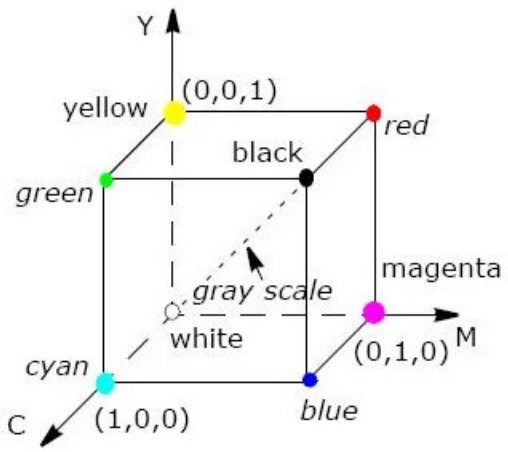
soustředěna v počátku souřadnicového systému krychle. Při plné intenzitě barevných komponent dochází ke vzniku bílé barvy. Diagonální křivka spojující vrcholy černé a bílé barvy reprezentuje barvy v odstínech šedi. V počítačové grafice dochází k reprezentaci jednotlivých komponent pomocí 8-bitového celočíselného datového typu v rozsahu $< 0{,}255 >$. Výsledným vektorem je takto následně možné zobrazit až $(2^8)^3 = 16\,777\,216$ odstínů barev.



Obr. 2-3 Barevný model RGB [15]

2.2.2 CMY, CMYK model

Barevný model CMY je model skládající se ze tří komponent, obdobně jako u modelu RGB. Je založen na subtraktivním míchání doplňkových barev: azurové – C, purpurové – M a žluté – Y. Tím, že se jedná o model založený na subtraktivním míchání barev, je možné model reprezentovat krychlí inverzní ke krychli reprezentující model RGB, kde hlavní osy představují intenzitu CMY komponent. Nulová intenzita všech tří barevných komponent reprezentuje bílou barvu, která je soustředěna v počátku souřadnicového systému krychle. Teoreticky při plné intenzitě barevných komponent dochází ke vzniku černé barvy. Prakticky však tomu tak není, proto se v praxi přidává černá barva a vzniká model CMYK, který je používán u tiskáren.



Obr. 2-4 Barevný model CMY [15]

2.2.3 HSV model

Barevný model HSV se skládá ze tří komponent, které však nereprezentují konkrétní barevné složky jako v předchozích případech, ale jejich vlastnosti: Hue, Saturation, Value. Občasně je tento model označován také jako model HSB: Hue, Saturation, Brightness. Model HSV je možné reprezentovat šestihranným jehlanem, kde počátek souřadnicového systému je situován do vrcholu jehlanu a jeho hlavní osa je shodná se svislou osou systému.

- **Hue** – Barevný odstín, který je definován jako velikost úhlu v rozsahu od 0° do 360°.
- **Saturation** – Sytost barvy, která je dána v procentech od 0% (šedá) do 100% (plně sytá barva).
- **Brightness** – Jas barvy, který určuje jak bude barva světlá nebo tmavá a kolik světla je barva schopna odrazit.

3. PŘEDZPRACOVÁNÍ OBRAZU

3.1 Bodové jasové transformace

Bodové jasové transformace obrazu jsou takové transformace, u kterých je hodnota výstupního obrazového bodu s danými souřadnicemi závislá pouze na obrazovém bodu se stejnými souřadnicemi ve vstupním obraze. Pro realizace takových transformací je někdy potřebné získat informace o celkovém rozložení jasu v obraze, které zprostředkovává histogram obrazu. Mezi nejčastěji používané bodové jasové transformace patří inverze, úprava jasu, kontrastu, prahování nebo ekvalizace histogramu.

3.1.1 Histogram

Histogram je znázornění vektoru četnosti jednotlivých jasových úrovní obrazu. Je popsán jako diskrétní funkce:

$$H(q) = n_q, \quad q_0 \leq q \leq q_k \tag{3.1}$$

, kde q – úroveň jasu

q_k – nejvyšší možná jasová úroveň obrazu

q_0 – nejnižší možná jasová úroveň obrazu

n – celkový počet bodů v obraze s úrovní jasu q.

Graficky je histogram nejčastěji reprezentován sloupcovým grafem, kde horizontální osa znázorňuje jednotlivé jasové úrovně obrazu. V případě obrazu v odstínech šedi s 8-bit hloubkou, budou horizontální osu tvořit hodnoty v rozsahu $< 0{,}255 >$. Vertikální osa pak znázorňuje četnost bodů dané úrovně. V případě barevného obrazu je histogram zobrazován pro každou barevnou složku zvlášť.

V ideálním případě má histogram rovnoměrně rozložené množství všech jasových úrovní, v reálném případě při vhodných světelných podmínkách má histogram vrchol ve středu a je dynamicky rozložen přes celý rozsah jasových úrovní. Vlivem špatné expozice snímku může docházet ke stavu, kdy jsou četně zastoupeny pouze krajní jasové úrovně. Je-li histogram četně zastoupen v oblasti blížící se 0, jedná se o podexponovaný obraz (tmavý obraz), v opačném případě se jedná o přeexponovaný obraz (světlý obraz).

3.1.2 Ekvalizace histogramu

Ekvalizace histogramu je algoritmus umožňující změnu kontrastu vstupního obrazu s využitím jeho histogramu. Při ekvalizaci histogramu dochází k rovnoměrnému rozložení jasových úrovní v celé šíři

histogramu s přibližně stejnou četností jednotlivých jasových úrovní. Dochází tedy ke zvýšení kontrastu úrovní jasu v blízkosti maxim histogramu a naopak ke snížení kontrastu v blízkosti jeho minim.

Ekvalizace histogramu je provedena ve 4 krocích. V prvním kroku je získán histogram původního obrazu $H(q)$, jehož jasová stupnice $q = \langle q_0, q_k \rangle$. Ve druhém kroku je nutné vypočítat kumulovaný histogram obrazu. Kumulovaný histogram je vypočítán pomocí rovnice (3.2), jehož prvek q je dán součtem jeho prvku $q - 1$ a prvku q původního histogramu.

$$H_k(q) = \sum_{i=q_0}^{q_k} H(i) = H_k(q-1) + H(q) \tag{3.2}$$

V následujících dvou krocích je kumulovaný histogram normalizován na převodní charakteristiku podle zvoleného rozsahu jasových hodnot výstupního obrazu a provedena samotná transformace jasové stupnice.

$$p = T(q) = \left(\frac{p_k - p_0}{N \cdot M}\right) \cdot H_k(q) + p_0 \tag{3.3}$$

, kde M, N – rozměry obrazu

p – jasová stupnice výstupního obrazu v rozsahu $\langle p_0, p_k \rangle$.

3.1.3 Úprava jasu a kontrastu

Ke změně jasu dochází při přičtení nebo odečtení konstanty od všech bodů v obraze. Při změně jasu je histogram posouván doprava, nebo doleva v závislosti na přičtení nebo odečtení jasové konstanty. Změnu jasové úrovně je možné vyjádřit pomocí rovnice:

$$p = q + c_j \tag{3.4}$$

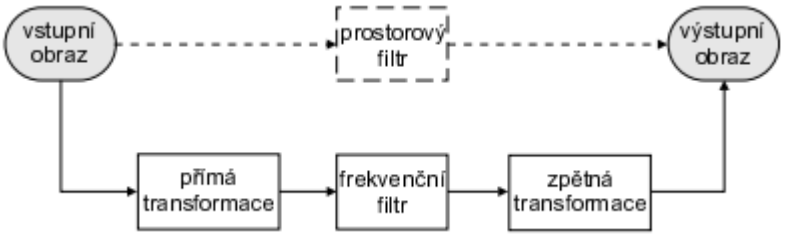
, kde c_j – konstanta změny jasové úrovně,

q – jasová úroveň původního obrazu.

V případě změny kontrastu dochází ke smršťování, nebo roztahování histogramu obrazu.

3.2 Lokální jasové transformace - Obrazové filtry

Filtrace obrazu je část předzpracování obrazu, jehož vstupem i výstupem je dvourozměrný signál (obraz). Jak naznačuje Obr. 3-1, lineární filtraci lze provést dvěma cestami: filtrací v prostorové oblasti, filtrací ve frekvenční oblasti.



Obr. 3-1 Lineární zpracování obrazu [1]

Filtrace v prostorové oblasti – zpracovává obraz jako lineární kombinaci vstupního obrazu s koeficienty filtru. Základním matematickým nástrojem je konvoluce.

Filtrace ve frekvenční oblasti – nejdříve obraz převede lineární integrální transformací do frekvenční oblasti, kde se filtruje a výsledek filtrace se inverzní lineární integrální transformací převede opět na obraz. Příkladem je převedení obrazu pomocí Fourierovy transformace. Tyto operace jsou však velmi výpočetně a časově náročné, proto nebudou v práci popisovány.

3.2.1 Prostorová filtrace

Konvoluce je matematický operátor zpracovávající dvě funkce. Jelikož je obraz reprezentován jako dvourozměrná diskrétní veličina, je v algoritmech zpracování obrazu využita konvoluce diskrétní. Vzorec diskrétní konvoluce má následující tvar:

$$f(x,y) * h(x,y) = \sum_{i=-S-1/2}^{S-1/2} \sum_{j=-R-1/2}^{R-1/2} f(x-i, y-j) \cdot h(i,j) \qquad \textbf{(3.5)}$$

,kde: $f(x,y)$ – intenzita vstupní obrazové funkce v souřadnicích (x,y)

$h(i,j)$ – intenzita konvolučního jádra o rozměrech $R \times S$ v souřadnicích (i,j), kde R a S jsou lichá čísla

x – index sloupce obrazové matice

y – index řádku obrazové matice.

Diskrétní konvoluce může být popsána jako operace ovlivňující matici pomocí jiné matice zvané konvoluční jádro (maska). Jako první matice je použita vstupní obrazová matice určená k úpravě. Použité konvoluční jádro závisí na požadovaném efektu. Konvoluce postupně prochází jednotlivé pixely obrazu a v závislosti na velikosti konvolučního jádra zpracovává i okolí daného pixelu.

3.2.2 Nízkofrekvenční filtry

Nízkofrekvenční filtry propouští nízké frekvence a potlačují vysoké frekvence. Výsledný obraz je oproti originálu vyhlazený, proto neobsahuje ostré hrany. Použitím větší masky, dojde k většímu vyhlazení obrazu. Nízkofrekvenční filtry se používají na „rozmazání“ obrazu a na odstranění šumu z obrazu. Mezi tyto filtry patří například průměrový, Gaussův filtr, nebo medián filtr. Medián filtr ovšem nepatří mezi filtry lineární.

- **Průměrový filtr**

Jeho princip spočívá v náhradě každé intenzity bodu v obraze vypočtením aritmetického průměru jeho okolí a bodu samotného. Aplikací takovéhoto filtru dochází k rozmazávání hran v obraze. Velikost rozmazání závisí na velikosti a typu konvoluční masky filtru. Rovnice (3.6) zobrazuje nejčastěji používanou konvoluční masku h velikosti 3×3, využívající okolí všech osmi bodů. Jeho modifikací však mohou vznikat i konvoluční masky využívající pouze okolních čtyř bodů, nebo se zvyšuje váha středového bodu masky. Větší masky se vytvářejí analogicky.

S ohledem na nežádoucí rozmazávání se obyčejné průměrování většinou používá jako pomocná metoda pro výpočet střední hodnoty jasu. Tento mezivýsledek je potom použit v důmyslnějších nelineárních vyhlazovacích metodách.[1]

$$h = \frac{1}{9}\begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix} \tag{3.6}$$

$$h = \frac{1}{5}\begin{bmatrix} 0 & 1 & 0 \\ 1 & 1 & 1 \\ 0 & 1 & 0 \end{bmatrix}, \qquad h = \frac{1}{10}\begin{bmatrix} 1 & 1 & 1 \\ 1 & 2 & 1 \\ 1 & 1 & 1 \end{bmatrix} \tag{3.7}$$

- **Medián filtr**

Jedná o nelineární metodu vyhlazování obrazu. Tato metoda vyhlazuje z obrazu lokální extrémy okolí, tím, že hledá medián určený z hodnot jasu bodů v lokálním okolí vstupního obrazu. Tato hodnota následně reprezentuje jasovou hodnotu bodu výsledného obrazu. Proto je tato metoda vhodná pro potlačení impulsního šumu (například Salt & Pepper) v obraze. Výsledný efekt filtru závisí na hustotě šumu ve vstupním obraze a na velikosti prozkoumávaného lokálního okolí.

Hlavní nevýhodou filtrace mediánem v obdélníkovém okolí je to, že porušuje tenké čáry a ostré rohy v obraze. Tento nedostatek se dá obejít použitím jiného tvaru okolí. [1]

3.2.3 Vysokofrekvenční filtry

Vysokofrekvenční filtry propouští vysoké frekvence a potlačují nízké frekvence. Vysokofrekvenční filtry se využívají pro detekci hran v obraze a s jejich využitím je možné docílit ostření obrazu

zvýrazněním vysokých frekvencí. Detekce hran je blíže popsána v kapitole 4.2, proto je v této kapitole popsáno pouze ostření obrazu.

Pro obraz g, který je výsledkem ostření f lze napsat:

$$g(x,y) = f(x,y) - C \cdot S(x,y) \tag{3.8}$$

, kde $S(x,y)$ – operátor udávající strmost obrazové funkce

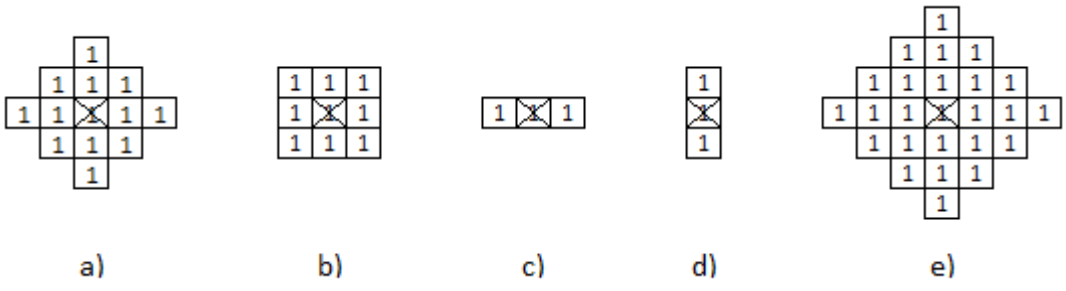
C – kladný součinitel síly ostření.

K zostření obrazu lze také docílit pomocí konvoluce. Při ostření je nutné zanechat jas obrazu nezměněn – suma vah v konvoluční masce se musí rovnat 1 a musí dojít ke zvýraznění hodnoty aktuálního pixelu a potlačení hodnot okolí. Příkladem takových konvolučních masek jsou následující rovnice:

$$h = \begin{bmatrix} -1 & -1 & -1 \\ -1 & 9 & -1 \\ -1 & -1 & -1 \end{bmatrix}, \qquad h = \begin{bmatrix} 0 & -0.5 & 0 \\ -0.5 & 3 & -0.5 \\ 0 & -0.5 & 0 \end{bmatrix} \tag{3.9}$$

3.3 Matematická morfologie

Při zpracování obrazu je matematická morfologie využita ke zkoumání interakce mezi obrazem a jinou určitou bodovou množinou nazývanou strukturální element. Nejčastěji je využívana pro zpracování binárních obrazů, ale je možné ji zobecnit pro šedo-tónové obrazy. Metody matematické morfologie jsou využívány pro předzpracování obrazů, zdůraznění struktur objektů nebo pro jejich popis. Principem morfologických operací je systematické posouvání strukturálního elementu po obraze s vyhodnocením odezev podle dané operace. Jednotlivé strukturální elementy jsou definovány svým tvarem a velikostí a reprezentativním bodem. Mezi základní tvary strukturálních elementů patří disk, čtverec, obdélník, linie nebo diamant. Základními morfologickými operacemi jsou dilatace a eroze, z nichž je možné odvodit další operace, jako jsou otevření, uzavření nebo získání obrysu objektů.

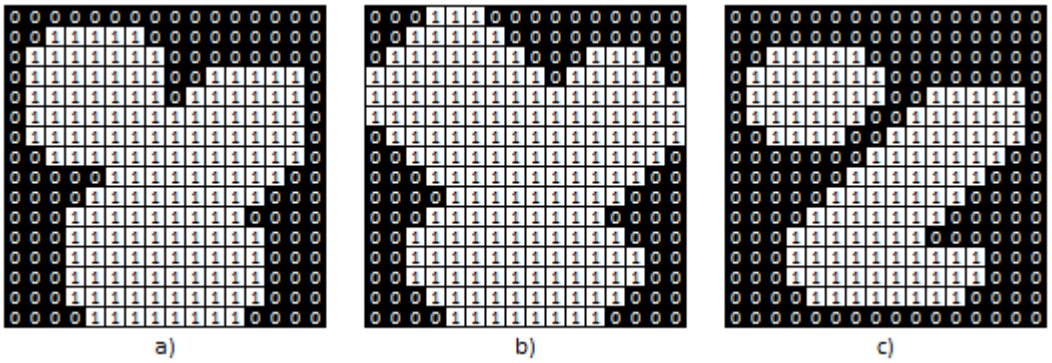


Obr. 3-2 Strukturální element a) disk, b) čtverec, c) linie (0°), d) linie (90°), e) diamant

3.3.1 Dilatace

V případě dilatace binárního obrazu dochází ke zhušťování oblastí zájmu vstupního obrazu. Způsob a míra zahuštění oblastí závisí na tvaru a velikosti aplikovaného strukturálního elementu. Tato metoda se využívá k zaplnění děr oblastí binárního obrazu. Principem dilatace je posouvání strukturálního elementu obrazem a v případě shody bodu obrazu s reprezentativním bodem strukturálního elementu, dochází k replikaci strukturálního elementu na okolní body obrazu. Tento princip demonstruje binární obraz Obr. 3-3, na který je aplikována dilatace tří různých strukturálních elementů. Výsledky je možné vidět na obrázku Obr. 3-4.

Obr. 3-3 Dilatace – Výchozí binární obraz

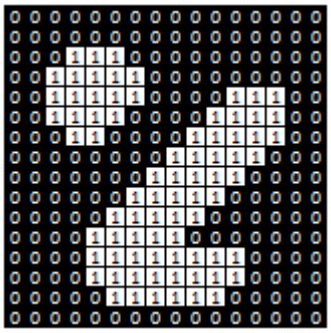


Obr. 3-4 Dilatace – Aplikace strukturálního elementu a) čtverec (3), b) disk (2), c) linie (3, 0°)

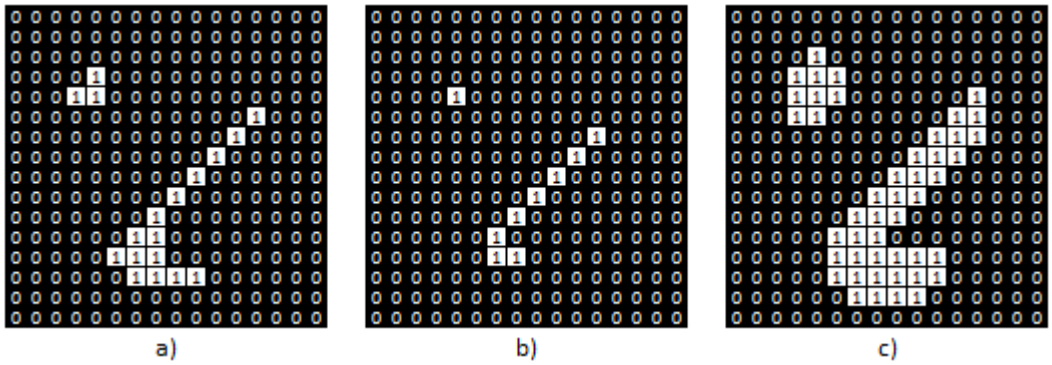
3.3.2 Eroze

Metoda eroze je duální operací k operaci dilatace. V případě eroze binárního obrazu dochází ke ztenčování oblastí zájmu vstupního obrazu. Stejně jako u dilatace je způsob a míra ztenčení oblastí závislá na tvaru a velikosti strukturálního elementu. Tato metoda se využívá k oddělení, popřípadě ke zjednodušení oblastí binárního obrazu. Principem eroze je posouvání strukturálního elementu obrazem a v případě shodnosti obrazu se strukturálním elementem ve všech jeho bodech, je výstupní obraz v bodě překrývajícím reprezentativním bodem strukturálního elementu reprezentován hodnotou jedna. Tento

princip demonstruje binární obraz Obr. 3-5, na který je aplikována eroze tří různých strukturálních elementů. Výsledky\eroze lze vidět na obrázku Obr. 3-6.



Obr. 3-5 Eroze – Výchozí binární obraz



Obr. 3-6 Eroze – Aplikace strukturálního elementu a) čtverec (3), b) disk (2), c) linie (3, 0°)

4. SEGMENTACE OBRAZU

Základní myšlenkou segmentace obrazu je seskupení jednotlivých pixelů do společných oblastí na základě jejich podobnosti dané některým z kritérií. Takovými kritérii mohou být například jejich intenzita, barva, nebo tvoří texturu, či vytvářejí tvar. Existuje mnoho metod pro segmentaci obrazu, které se liší ve složitosti a oblasti jejich použití. Metody segmentace lze rozdělit do následujících kategorií.

Segmentace prahováním – Jedná se o základní metodu segmentace obrazu vycházející ze znalosti, že objekt a pozadí jsou charakterizovány rozdílnou barvou či jasem. Prahování může být rozděleno na prahování prosté, s více prahy, částečné nebo adaptivní.

Metody vycházející z detekce hran – Metody orientované na detekci významných hran v obraze. Lokální hrany jsou detekovány pomocí hranových detektorů na základě rozdílu hodnot okolních pixelů. Mezi metody vycházející z detekce hran lze zařadit prahování obrazu hran, sledování hranice, aktivní kontury, nebo Houghovu transformaci.

Metody orientované na regiony v obraze – principiálně jsou stejné jako edge-based metody. Pokud lze identifikovat hrany, měly by teoreticky ohraničovat regiony nalezené region-based segmentací. Kontury regionů však mohou být porušené, nemusí ohraničovat celý region. Mezi tyto metody patří rozrůstání oblastí, spojování oblastí, štěpení oblastí, štěpení a spojování oblastí, watershed nebo shluková analýza.

Znalostní metody – Metody patřící do této kategorie využívají atlas předloh či modelů segmentovaných objektů. Atlas je generován automaticky ze souboru trénovacích dat, nebo jsou do něj informace vloženy ručně, na základě lidské zkušenosti. V průběhu segmentace algoritmus hledá transformaci známých objektů, šablon v atlasu, na objekty nalezené v obraze. Tento proces se obvykle nazývá atlas-warping a nejčastěji využívá lineární transformace.

Hybridní metody – Některé segmentační techniky je těžké zařadit do jedné z předchozích čtyř kategorií, protože obsahují prvky každé z nich. Mezi hybridní metody se řadí také například metody založené na matematické morfologii, nebo metoda využívající matematických charakteristik obrazu, jako je průběh gradientu.

4.1 Prahování

Prahování je jedna ze základních metod segmentace, při které dochází k vytvoření binárního obrazu. Prahování lze aplikovat na obrazy s více úrovněmi jasu, dochází tedy k převodu obrazu s více úrovněmi jasu na obraz binární. Hodnoty 1 výsledného binárního obrazu, reprezentují oblasti zájmu, zatímco hodnoty 0 reprezentují pozadí. Při prahování se vychází z předpokladu, že body hledaných oblastí mají stejný nebo dosti podobný jas. Body oblasti zájmů jsou přiřazovány na základě prahu, který je definován buď jednou jasovou úrovní, nebo rozsahem úrovní jasů.

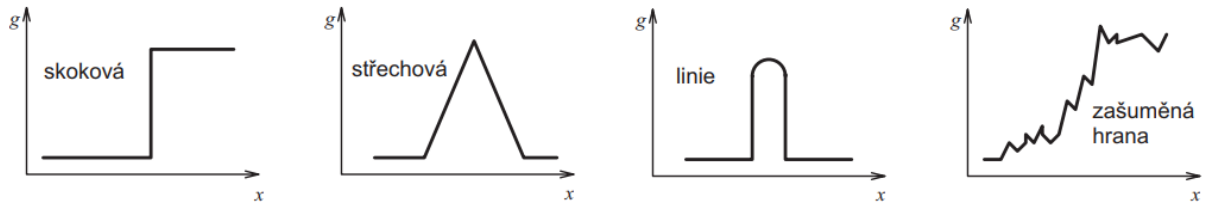
$$g(x,y) = \begin{cases} 1, & f(x,y) \geq T \\ 0, & jinak \end{cases} \tag{4.1}$$

$$g(x,y) = \begin{cases} 1, & f(x,y) \in \langle a,b \rangle \\ 0, & jinak \end{cases} \tag{4.2}$$

U tohoto globálního prahování celého obrazu může při nerovnoměrném nasvícení snímaného objektu docházet k chybám. Proto se využívá prahování lokální, kdy je obraz rozdělen do několika částí a práh je určen pro každou část zvlášť. Úspěšnost prahování závisí na znalosti správné hodnoty prahu. Prah je možné určit experimentálně, z histogramu, statisticky, procentuálně nebo z globální znalosti.

4.2 Detekce hran

Hranu v obraze lze definovat jako místo, kde dochází k prudké změně jasové úrovně sousedních pixelů. V ideálním případě se jedná o změnu skokovou. Hrana je tedy vysokofrekvenční informací obrazu. Na obrázku Obr. 4-1 jsou zobrazeny nejběžnější jasové profily hran. První tři profily jsou profily idealizované, v reálných obrazech se však setkáváme s hranou zašuměnou.



Obr. 4-1 Jasové profily nejběžnějších hran [1]

Nejčastěji používané metody pro detekci hran jsou metody gradientní. Tyto metody využívající skutečnost, že v místě hrany má absolutní hodnota první derivace průběhu jasu vysokou hodnotu. Za velikost hrany lze vzít velikost gradientu a směr hrany je pak kolmý ke směru gradientu. Pro spojitou funkci $f(x,y)$ jsou velikost gradientu $|\nabla f(x,y)|$ a směr gradientu Θ dány vztahy:

$$|\nabla f(x,y)| = \sqrt{\left(\frac{\partial f}{\partial x}\right)^2 + \left(\frac{\partial f}{\partial y}\right)^2} \tag{4.3}$$

$$\Theta = \arg\left(\frac{\partial f}{\partial x}, \frac{\partial f}{\partial y}\right) \tag{4.4}$$

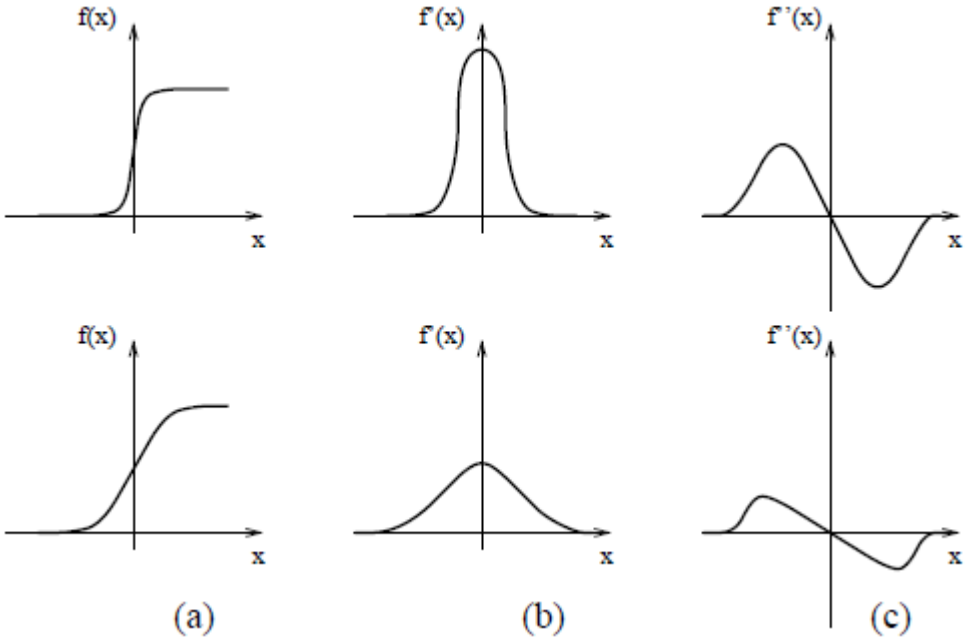
Jelikož však při zpracování obrazu je využíván obraz diskrétní, je nutné parciální derivace nahradit diferencemi. Ty jsou vypočteny jako rozdíl dvou jasových úrovní sousedních pixelů. Řádkový gradient ve směru z leva doprava je definován jako:

$$f_x(x,y) = f(x,y) - f(x-1,y) \tag{4.5}$$

Sloupcový gradient ve směru od spodu nahoru je definován jako:

$$f_y(x,y) = f(x,y) - f(x,y-1) \tag{4.6}$$

V případech, kdy není nutné znát směr a velikost hran a je možné si vystačit pouze s informací, kde se dané hrany nacházejí, je možné použít metody využívající druhé derivace. Druhá derivace protíná v místě hrany nulovou hodnotu. Výše zmíněné fakty, že v místě hrany nabývá první derivace obrazové funkce jejího maxima a druhá derivace protíná nulovou hodnotu, demonstruje obrázek Obr. 4-2 pro jednorozměrnou funkci.



Obr. 4-2 Odezvy skokové 1D funkce na její 1. a 2. derivaci [1]

Gradientní metody detekce hran mohou být aproximovány operátory pomocí konvoluční masky. Základními operátory aproximující první derivaci jsou Robertsův operátor, operátor Prewittové, Sobelův operátor, Robinsonův operátor, Kirschův operátor. Mezi operátory aproximující druhou derivaci patří například Laplacián nebo Laplacian of Gaussian.

Robertsův operátor

Jedná se o jedinou aproximaci využívající okolí 2×2, ze které vychází jeho vysoká citlivost na šum.

$$h_1 = \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix}, \quad h_2 = \begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix} \tag{4.7}$$

Zbylé operátory aproximující první derivaci využívají okolí 3×3 pro osm směrů. Jednotlivé operátory se liší hodnotami konvolučních masek. Následující rovnice (4.8) až (4.11), ukazují příklady konvolučních masek jednotlivých operátorů. Konvoluční masky h_1 slouží pro výpočet ve směru gradientu -180°, h_2 ve směru gradientu -135° a h_3 ve směru gradientu -90°.

Operátor Prewittové

$$h_1 = \begin{bmatrix} 1 & 1 & 1 \\ 0 & 0 & 0 \\ -1 & -1 & -1 \end{bmatrix}, \; h_2 = \begin{bmatrix} 0 & 1 & 1 \\ -1 & 0 & 1 \\ -1 & -1 & 0 \end{bmatrix}, \; h_3 = \begin{bmatrix} -1 & 0 & 1 \\ -1 & 0 & 1 \\ -1 & 0 & 1 \end{bmatrix} \tag{4.8}$$

Sobelův operátor

$$h_1 = \begin{bmatrix} 1 & 2 & 1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{bmatrix}, \; h_2 = \begin{bmatrix} 0 & 1 & 2 \\ -1 & 0 & 1 \\ -2 & -1 & 0 \end{bmatrix}, \; h_3 = \begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix} \tag{4.9}$$

Robinsonův operátor

$$h_1 = \begin{bmatrix} 1 & 1 & 1 \\ 1 & -2 & 1 \\ -1 & -1 & -1 \end{bmatrix}, \; h_2 = \begin{bmatrix} 1 & 1 & 1 \\ -1 & -2 & 1 \\ -1 & -1 & 1 \end{bmatrix}, \; h_3 = \begin{bmatrix} -1 & 1 & 1 \\ -1 & -2 & 1 \\ -1 & 1 & 1 \end{bmatrix} \tag{4.10}$$

Kirschův operátor

$$h_1 = \begin{bmatrix} 3 & 3 & 3 \\ 3 & 0 & 3 \\ -5 & -5 & -5 \end{bmatrix}, \; h_2 = \begin{bmatrix} 3 & 3 & 3 \\ -5 & 0 & 3 \\ -5 & -5 & 3 \end{bmatrix}, \; h_3 = \begin{bmatrix} -5 & 3 & 3 \\ -5 & 0 & 3 \\ -5 & 3 & 3 \end{bmatrix} \tag{4.11}$$

Laplaceův operátor

Na rozdíl od gradientu, Laplaceův operátor pracuje se druhými derivacemi obrazové funkce. Laplaceův operátor využívá konvoluční masky pro 8-okolí nebo 4-okolí velikosti 3×3 a jsou invariantní vůči otočení.

$$h_1 = \begin{bmatrix} -1 & -1 & -1 \\ -1 & 8 & -1 \\ -1 & -1 & -1 \end{bmatrix} \qquad h_2 = \begin{bmatrix} 0 & -1 & 0 \\ -1 & 4 & -1 \\ 0 & -1 & 0 \end{bmatrix} \tag{4.12}$$

4.3 Houghova transformace

Houghova transformace je technika, která může být použita k izolaci vlastností určitého tvaru v rámci obrazu. Protože vyžaduje, aby požadované vlastnosti byly specifikovány v nějaké parametrické formě, se klasická Houghova transformace nejčastěji používá k detekci pravidelných křivek, jako jsou přímky, kruhy, elipsy apod. Existuje však i zobecněná Houghova transformace, která může být použita

v aplikacích, kde není možný analytický popis objektů. Z důvodu její výpočetní náročnosti nebude zobecněná Houghova transformace dále popisována.

Vstupním parametrem pro Houghovu transformaci je nejčastěji binární obraz vzniklý aplikací hranového detektoru s následným prahováním. Jde o transformaci z Kartézského souřadnicového systému do polárního [4].

Největší výhodou této metody je možnost spolehlivě nalézat pozice struktur, které jsou narušeny šumem, popřípadě přerušeny neideální detekcí hran. Naopak nevýhodou této metody je, že nemá informace ohledně počátku a konci křivek, proto se například z úseček stávají přímky.

4.3.1 Detekce přímek

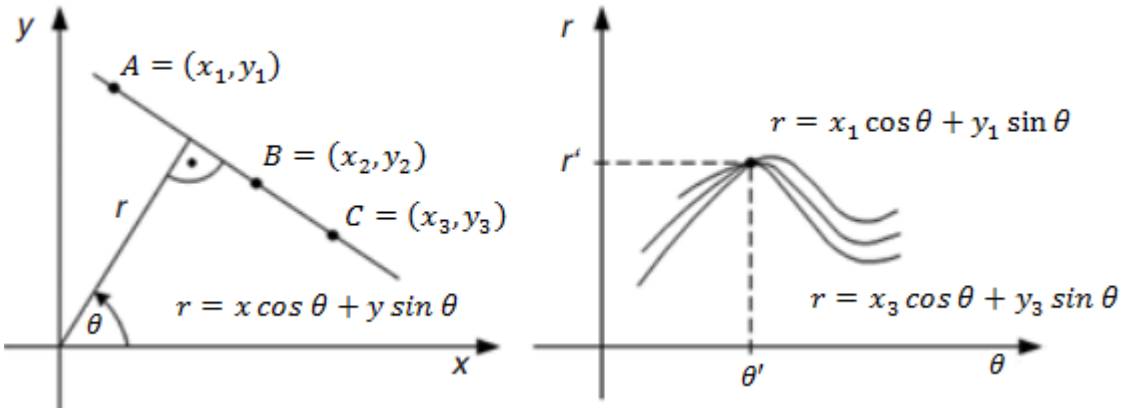
Detekce přímek pomocí Houghovy transformace vychází z normálového tvaru přímky, který je definován jako:

$$r = x \cdot \cos\theta + y \cdot \sin\theta \tag{4.13}$$

, kde r – délka normály přímky od počátku

θ – úhel mezi normálou a osou x.

Je uvažován parametrický prostor H (akumulátor) o rozměrech $360 \times \sqrt{M^2 + N^2}$, kde M a N jsou rozměry vstupního obrazu. Každým bodem vstupního obrazu nesoucí informaci je vedena přímka pod úhlem θ, pro kterou je dopočítána délka normály přímky. Množina všech možných řešení (r, θ) vytvoří v parametrickém prostoru spojitou křivku. Promítnutím všech bodů, které leží na stejné přímce do parametrického prostoru, budou získány spojité křivky, které se protnou v jediném bodě. Nalezením maxim akumulátoru, reprezentující průsečíky spojitých křivek, lze následně určit hodnoty r a θ pro stanovení popisu přímky vstupního obrazu. Celý tento princip demonstruje obrázek pro tři body A, B, C.



Obr. 4-3 Houghova transformace pro hledání přímky

4.4 Spojování oblastí

Spojování oblastí je proces sdružování sousedních pixelů nebo sbírky pixelů (regionů) podobných vlastností do větších oblastí. Princip spojování oblastí spočívá v umístění tzv. semínek do obrazu, které se rozrůstají na základě porovnávání se sousedními pixely. Sousední pixely, které splňují kritérium homogenity, jsou přiřazeny do oblasti vytvářeného semínka. Tento krok porovnávání sousedních pixelů probíhá do té doby, dokud oblast není zcela ohraničena okrajem obrazu, nebo pixely, které nesplňují kritérium homogenity. Obdobně jsou vytvářeny další oblasti, dokud všechny pixely nespadají do jedné z oblastí. Kritériem homogenity mohou být úrovně šedi, barvy nebo textury, příkladem takového kritéria pro obraz v odstínech šedi definuje následující rovnice:

$$|f - f_S| \leq T \tag{4.14}$$

, kde f_S – jasová úroveň semínka

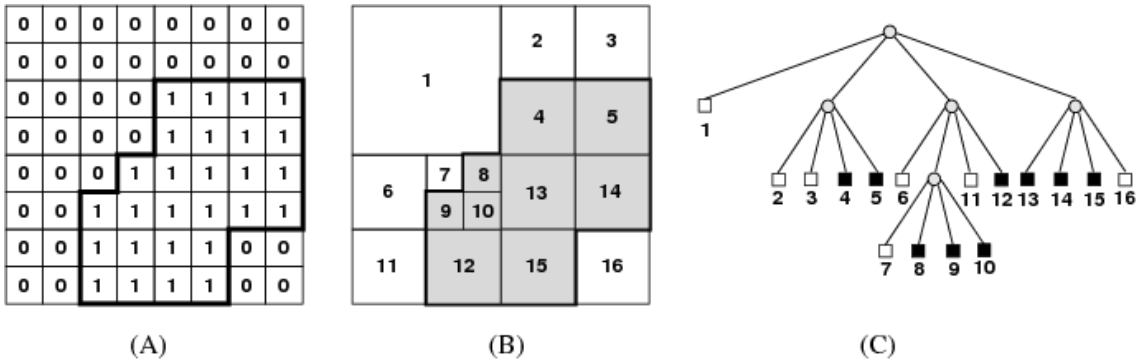
f – jasová úroveň zkoumaného pixelu

T – práh.

4.5 Split & Merge - štěpení a spojování oblastí

Metoda dělení a spojování oblastí je založena na reprezentaci dat kvadratického stromu (quadtree). V případě jsou-li atributy obrazového segmentu nehomogenní, je obrazový segment rozdělen na čtyři kvadranty. Jsou-li atributy některých ze sousedních kvadrantů homogenní, dochází k jejich sloučení. [2]

Obrázek Obr. 4-4 znázorňuje reprezentaci dvourozměrné binární oblasti (A) s kvadranty (C). Oblast je rozdělována na kvadranty, subkvadranty (B), dokud bloky neobsahují pouze jedničky nebo nuly.



Obr. 4-4 Split & Merge – Quad-tree [9]

5. POPIS A KLASIFIKACE OBJEKTŮ

Popis a klasifikace jsou posledními kroky posloupnosti zpracování obrazu. Tyto dva kroky byly spojeny do jedné kapitoly, jelikož spolu úzce souvisí. Úkolem popisu segmentovaných objektů je reprezentace jednotlivých objektů matematicky, buď souborem číselných charakteristik, nebo symboly zvolené abecedy tak, aby co nejlépe vystihovaly charakteristické rysy objektů. Z obrazu segmentovaných objektů jsou vypočítány příznaky x_i, ze kterých je sestaven příznakový vektor X. Příznakový vektor může v obecném případě obsahovat vysoký počet zcela odlišných charakteristických rysů objektů. Existuje mnoho možných popisů objektů, které je možné dělit do několika skupin. Mezi dvě základní dělení patří:

- **Podle domény popisované vlastnosti objektu**

Tato skupina rozděluje vlastnosti popisu objektu na fotometrické a radiometrické. Mezi fotometrické vlastnosti patří například průměrná jasová úroveň, diference průměrné jasové úrovně objektu od průměrné jasové úrovně blízkého okolí, rozptyl jasových hodnot objektu a odráží optické vlastnosti objektu. Mezi vlastnosti radiometrické patří například velikost, délka hlavní a vedlejší osy, obvod a odráží geometrické vlastnosti objektu.

- **Podle oblasti výpočtu**

Podle oblasti výpočtu je možné vlastnosti objektů dělit na ty, které jsou založeny na regionech nebo na hranicích. V případě popisu založeném na regionech je hodnota příznaku vypočtena na základě hodnot pixelů objektu. Pro jeho výpočet je nutná znalost jasových hodnot pixelů objektu. V případě popisu založeném na hranici je hodnota příznaku vypočtena na základě hraniční reprezentaci objektu.

Úkolem klasifikace je pak přiřazení objektů do jednotlivých tříd. Jedná se tedy o algoritmus, rozhodující o příslušnosti objektů k dané třídě, na základě jeho popisu. Podle způsobu popisu, který tvoří vstup do klasifikátoru, pak hovoříme o příznakovém nebo syntaktickém rozpoznávání. U příznakového rozpoznávání rozhodujeme podle množiny číselných charakteristik objektů, které říkáme vektor příznaků. Druhou možností klasifikace je rozhodování na základě struktury objektu. [18]

5.1 Základní radiometrické vlastnosti

Velikost – Jedná se o plochu objektu reprezentovanou součtem pixelů spojité oblasti.

Obvod – Hodnota počtu hraničních pixelů.

Konvexnost – Udává míru podobnosti objektu se svým konvexním obalem. Jedná se tedy o vyjádření poměru velikosti konvexního obalu objektu k jeho ploše. Konvexnost může nabývat hodnot z intervalu $\langle 0; 1 \rangle$, kde hodnota 1 reprezentuje objekty konvexní, v opačném případě se jedná o objekt konkávní.

Nekompaktnost – Udává míru podobnosti objektu k ideálnímu kruhu, který má nekompaktnost o hodnotě 4π. Jedná se o poměr kvadrátu obvodu k velikosti objektu.

Freemanův kód – Jedná se o hraniční radiometrický popis objektu, popisující hranici objektu pomocí řetězce čísel. Tento řetězec reprezentuje směr postupného průchodu obvodových bodů hranice objektu. Směry kódu mohou být určeny pro 4-okolí nebo pro 8-okolí obvodových pixelů.

6. METODY DETEKCE

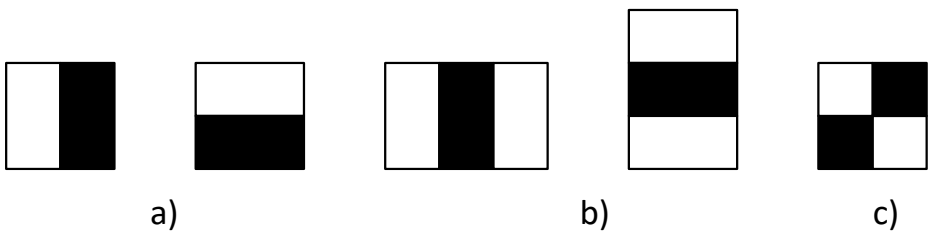
Existuje spousta metod detekce, pro které neexistuje zobecněné řešení. Ve většině případů záleží na tom, co je potřeba detekovat, v jakém prostředí apod., proto se ke každé takové úloze přistupuje individuálně. Všechny principy však vycházejí z řetězce zpracování obrazu. Následující kapitola má za úkol popsat dvě nejrozšířenější metody pro detekci objektů v obraze, které byly primárně vytvořeny pro detekci obličeje, popřípadě celé osoby v obraze. Obě tyto metody jsou založeny na detekci příznaků s následující klasifikací příznaků pomocí učících se algoritmů.

6.1 Voila-Jones detektor

Objektový detektor Viola-Jones byl poprvé představen Paulem Violou a Michaelem Jonesem v roce 2001. Jedná se o detektor objektů pracující s šedo-tónovými obrazy. Snahou detektoru Voila-Jones je získat velkou řadu jednoduchých příznaků s minimálními výpočetními nároky. Takovým typem příznaků jsou příznaky založené na principu podobnému definici Haarově vlnky. Pro natrénování detektoru využívá klasifikační algoritmus AdaBoost, který skládá jednoduché klasifikátory do kaskády na základě vstupní množiny negativních a pozitivních vzorků (snímků). Výhodou tohoto detektoru je rychlost, dostatečná spolehlivost a značná nezávislost na osvětlení a velikosti sledovaného objektu. Z těchto důvodů je v praxi tento detektor často používán například při detekci obličejů a zároveň vzniká velká řada jeho modifikací. [10][12]

6.1.1 Haarovy příznaky

Haarovy příznaky jsou obdélníkové oblasti, které jsou využity jako vstup do klasifikačního algoritmu. Jednotlivé příznaky jsou použity na celý vstupní obraz, přičemž postupně dochází ke změně velikosti příznaků. Tyto vlnky mohou být tvořeny dvěma (hranový příznak), třemi (čárkový příznak) či čtyřmi (diagonální příznak) obdélníkovými oblastmi. Hodnota hranového příznaku je vypočtena jako rozdíl sum intenzit pixelů těchto dvou stejně velkých oblastí. V případě čárkového příznaku je jeho hodnota vypočtena jako součet v rámci dvou vnějších obdélníků odečtených od součtu ve středovém obdélníku.



Obr. 6-1 Základní Haarovy příznaky

6.1.2 Integrální obraz

Pro urychlení výpočtů odezev Haarový příznaků na vstupní obraz, je tento obraz převeden na obraz integrální. Integrální obraz v bodě x, y obsahuje součet všech pixelů nad a vlevo od daného bodu, včetně bodu samotného. Hodnota integrálního obrazu v bodě (x, y), lze vyjádřit pomocí rovnice:

$$ii(x,y) = \sum_{x' \leq x, y' \leq y} i(x', y') \tag{6.1}$$

, kde $ii(x,y)$ je integrální obraz,

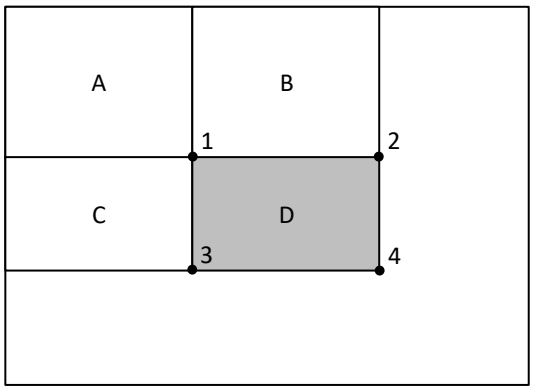
$i(x,y)$ je originální obraz.

S využitím následující dvojice rovnic, lze integrální obraz vypočítat jediným průchodem původního obrazu.

$$s(x,y) = s(x,y-1) + i(x,y) \tag{6.2}$$

$$ii(x,y) = ii(x-1,y) + s(x,y) \tag{6.3}$$

, kde $s(x,y)$ je kumulovaný součet hodnot pixelů v řádku, při splnění podmínek $s(x,-1) = 0$ a $ii(-1,y) = 0$.



Obr. 6-2 Integrální obraz

6.1.3 AdaBoost klasifikace

Klasifikační algoritmus Adaboost (Adaptive Boosting) je základní metodou pro trénování kaskády klasifikátorů. Byl publikován v roce 1995 Yoavem Freundem a Robertem Schapirem. Základním principem algoritmu je výběr slabých lineárních klasifikátorů $h_t(x)$ z množiny všech klasifikátorů H tak, aby jejich lineární kombinací vznikl silný nelineární klasifikátor $H(x)$, který reprezentuje stupeň kaskády. Jeho vstupem je trénovací množina dat $(x_1, y_1; x_2, y_2; \ldots; x_j, y_j)$, kde x_j je n-rozměrný vektor

příznaků a y_j značí, jedná-li se o pozadí $y_j = -1$, či o objekt $y_j = 1$. Jedná se o takzvaný diskrétní AdaBoost, jehož výstupem je klasifikátor zařazující data do dvou tříd.

6.2 HoG detektor

Metoda histogramu orientovaných gradientů byla poprvé představena Navneetem Dalalem a Billem Triggsem v roce 2005. Tato metoda patří do skupiny parametrických metod popisu a byla vyvinuta za účelem detekce lidských postav v obraze. Metoda HOG pracuje s šedo-tónovými, tak i barevnými obrazy. Principem metody je popis objektů pomocí histogramu gradientů, reprezentovanými svou velikostí a směrem, s výslednou klasifikací pomocí Support Vector Mechine algoritmu.

6.2.1 Konstrukce deskriptoru

Deskriptor metody HOG je získán tak, že dojde k rovnoměrnému rozdělení vstupního obrazu do čtvercových bloků o hraně k pixelů. Tyto bloky jsou následně dále rozčleněny na buňky o hraně l pixelů, kde platí $l < k$, pro které jsou vypočteny normované histogramy orientací gradientů. Výsledné histogramy buněk jsou zprůměrovány v jeden, který pak tvoří HOG popis bloku, po jehož získání se blok posune o zvolený počet pixelů a celý proces se opakuje. Finální HOG deskriptor je zřetězením všech popisů jednotlivých bloků uvnitř vstupního obrazu.[13]

6.2.2 Výpočet gradientu

Autory metody byly testovány různé varianty výpočtu gradientu v obraze, jenž vysoce ovlivňuje výsledný deskriptor. Nejlepších výsledků bylo dosaženo kombinací Gaussova filtru při směrodatné odchylce $\sigma = 0$ a aplikací konvolučních masek $[-1, 0, 1]$, respektive $[-1, 0, 1]^T$ pro výpočet diskrétního gradientu. Výpočet gradientu je tedy možno vyjádřit pomocí dvojice rovnic:

$$I_x = I * [-1,0,1] \tag{6.4}$$

$$I_y = I * [-1,0,1]^T \tag{6.5}$$

, kde I – vstupní obraz,

I_x, I_y – výstupní obrazy po aplikaci konvolučních masek pro výpočet gradientů ve vertikálním a horizontálním směru.

6.2.3 Histogram orientací gradientů

Poté co jsou získány obrazy I_x a I_y, je pro každý pixel vypočtena velikost gradientu $m(x, y)$ a směr gradientu $\Theta(x, y)$:

$$m(x,y) = \sqrt{I_x^2 + I_y^2} \tag{6.6}$$

$$\Theta(x,y) = \tan^{-1}\left(\frac{I_y}{I_x}\right) \tag{6.7}$$

Z vypočtených gradientů je zkonstruován histogram orientací, jehož počet binů je určen rozdělením směru gradientu $\Theta(x,y)$ do počtu sektorů s v intervalu i. Počet sektorů s je číslo definující počet stejně velkých výsečí, do kterých je interval i rozdělen. Interval i bývá nejčastěji v rozsahu $i = \langle 0,\pi\rangle$, $i = \langle 0,2\pi\rangle$. Robustnost výsledného deskriptoru zvyšuje fakt, že velikost gradientu $m(x,y)$ je poměrově rozdělena mezi sousední biny. To znamená, že gradient s velikostí 8 a směru 15° - leží mezi biny pro 0 ° a 20°, je tato velikost gradientu proporcionálně rozdělena mezi oba biny – 2 pro bin 0, 6 pro bin 20.

6.2.4 Normalizace histogramu

Z důvodu invariance výsledného deskriptoru vůči změně jasu a kontrastu dochází k normalizování získaných histogramů orientovaných gradientů, pomocí některého z normalizačních schémat $L^1 - norm$, $L^1 - sqrt$ nebo $L^2 - norm$. Předpisy těchto schémat jsou reprezentovány následujícími rovnicemi převzatými z [16].

$$L^1 - norm = \frac{v}{(\|v\|_1 + e)} \tag{6.8}$$

$$L^1 - sqrt = \sqrt{\frac{v}{(\|v\|_1 + e)}} \tag{6.9}$$

$$L^2 - norm = \frac{v}{\sqrt{(\|v\|_1^2 + e^2)}} \tag{6.10}$$

, kde v – nenormalizovaný vektor,

$\|v\|_k$ – k-tá norma nenormalizovaného vektoru,

e – malá konstanta, tzv. regularizace.

6.2.5 Klasifikace

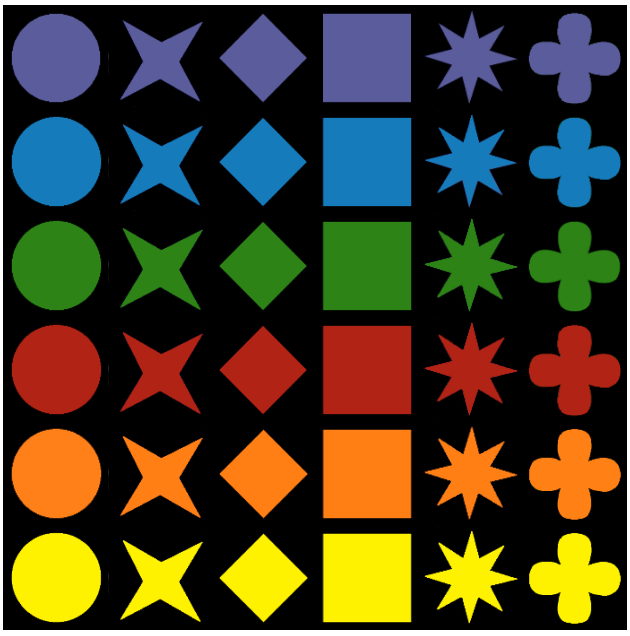
V metodě HOG je pro klasifikaci použito nástroje SVM, konkrétněji lineární SVM – lineární Support Vector Mechine algoritmus. Jedná se o metodu strojového učení, což je nauka o získávání a zpracování znalostí. Lineární SVM klasifikátor je jednoduchá klasifikační metoda založena na rozdělení prostoru příznaků na lineární úseky.

7. NÁVRH SYSTÉMU VIZUÁLNÍHO ROZHODČÍHO

Praktickým úkolem této práce je návrh a realizace aplikace vizuálního rozhodčího pro stolní hru QWIRKLE, bez možnosti využití některé z dostupných knihoven pro zpracování obrazu. Z důvodu časové náročnosti vytvoření vlastní knihovny pro zpracování obrazu, bylo při návrhu aplikace vizuálního rozhodčího využito jednoduchých metod pro detekci a následné rozpoznání objektů. V následujících podkapitolách je popsána hra Qwirkle, společně s návrhem řetězce zpracování obrazu pro aplikaci vizuálního rozhodčího. Podkapitoly dále obsahují návrh knihovny pro zpracování obrazu a popis metod využívaných při realizaci aplikace vizuálního rozhodčího, na které se bude odkazovat při realizaci systému v kapitole 8.

7.1 Hra Qwirkle

Hra Qwirkle je stolní logická hra určena pro 2 až 6 hráčů skládající se ze 108 herních kamenů. Tři sady herních kamenů šesti různých barev a tvarů zobrazených na obrázku Obr. 7-1. Princip hry spočívá ve skládání řad z herních kamenů stejných barev nebo tvarů. Kameny přidané do řady musí sdílet stejnou vlastnost jako kameny tvořící řadu. Řada tvořena tvary může obsahovat pouze jeden kámen od každé ze šesti barev. Řada tvořena barvami může obsahovat pouze jeden kámen od každého ze šesti tvaru. To znamená, že jedna řada může být složena maximálně z šesti kamenů a žádný z kamenů se v řadě nesmí opakovat. Kompletní pravidla a bodování hry jsou dostupná v příloze práce.



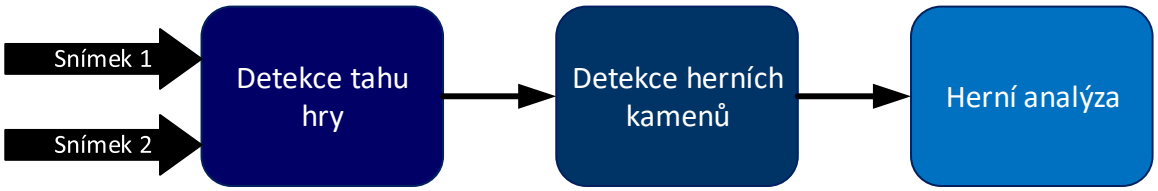
Obr. 7-1 Qwirkle - Herní kameny

7.2 Návrh systému

Při návrhu systému se vychází ze znalosti barev a tvarů jednotlivých herních kamenů. Pro jednotlivé barvy herních kamenů budou stanoveny prahy pro barevné RGB složky obrazu. Na základě těchto barev bude docházet k prahování obrazu a extrakci potencionálních symbolů herních kamenů. Výsledný binární obraz bude dále segmentován pro rozlišení a označení jednotlivých uzavřených binárních oblastí. Jednotlivé oblasti bude možné popsat svými radiometrickými vlastnostmi (výška, šířka, obsah) a na základě těchto vlastností budou oblasti klasifikovány. Oblasti budou klasifikovány do dvou tříd, do třídy reprezentující symboly herních kamenů a třídy reprezentující ostatní oblasti. U oblastí, jenž budou klasifikovány jako symbol herního kamene, bude docházet k rozpoznání jejich barvy a tvaru. Pro rozpoznání barvy oblastí bude docházet k jejich vyříznutí z originálního RGB obrazu a k určení barvy na základě prahování barevného obrazu pomocí prahu reprezentující jednotlivé barvy herních kamenů. Binární obraz s největší četnosti zastoupení bílých pixelů bude určovat výslednou barvu herního kamene. Pro určení tvaru oblastí bude využita metoda založená na porovnávání se vzory vytvořených symbolů, pomocí korelace.

Na základě souřadnic středů oblastí v obraze bude sestavena replika herního pole a bude docházet k porovnání herních polí aktuálního tahu hry s tahem předchozím. Porovnáním herních polí dojde k nalezení nových herních kamenů a následné analýze herních pravidel.

Pro zajištění automatičnosti systému, musí být systém doplněn o detekci provedení tahu hry. Tato detekce bude vycházet z porovnání sekvence snímků na základě jejich rozdílů.



Obr. 7-2 Blokový návrh systému vizuálního rozhodčího

Pro snímání scény hry bude využita obyčejná webkamera A4tech PK-910H, která bude směřovat kolmo na scénu hry.

7.3 Knihovna pro zpracování obrazu

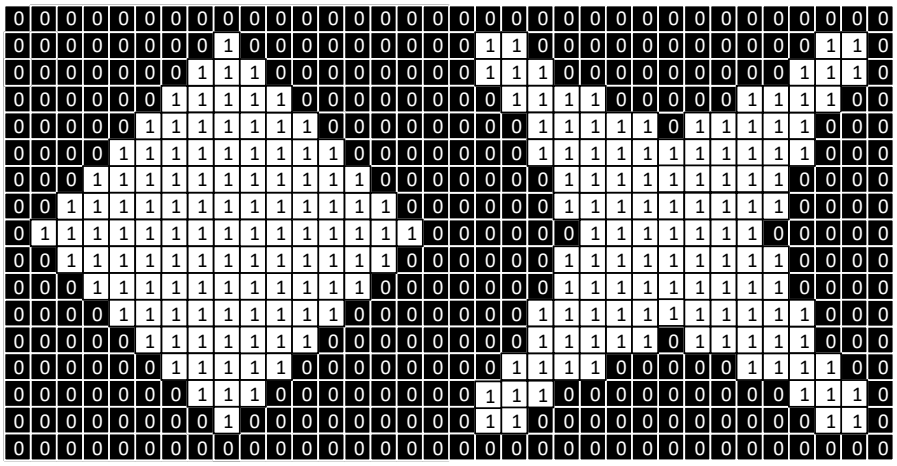
Na základě teoretických znalostí popsaných v předchozích kapitolách byla navržena knihovna pro zpracování obrazu. Knihovna je realizována ve vývojovém prostředí Microsoft Visual Stuido Community 2017 v jazyce C++. Samotná knihovna je strukturovaná do šesti tříd, z nichž jedna reprezentuje obraz obecně. Jelikož však jasové úrovně barevného, monochromatického a binárního obrazu jsou reprezentovány rozdílnými datovými typy, nelze definovat obraz pouze obecně, ale jsou odvozeny další tři podtřídy reprezentující tři druhy obrazů: barevný, monochromatický a binární obraz.

Další dvě třídy jsou vytvořeny pro práci s regiony binárních obrazů. Knihovna neobsahuje metody pro čtení dat ze souboru, či jakéhokoliv jiného zařízení, obsahuje pouze metody pro jeho zpracovávání. Pro čtení snímku je využita otevřená multiplatformní knihovna OpenCV. V následujících podkapitolách jsou popsány často využívané metody při zpracování obrazu pro realizaci vizuálního rozhodčího.

7.3.1 Detekování uzavřených oblastí

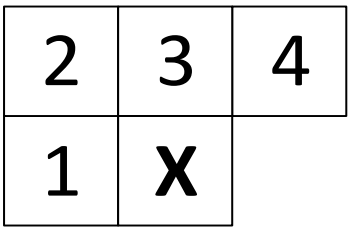
Vstupním parametrem této metody je binární obraz, kde hodnota `false` reprezentuje pozadí a hodnota `true` oblast zájmu (Obr. 7-3). Na výstupu této metody je obraz, kde každá uzavřená oblast je označena jiným číslem, zároveň s hodnotou počtu uzavřených oblastí. Metoda je volána přes instanci třídy, pracující s uzavřenými oblastmi.

```
grayImage* BlobLogic::blobDetect(bool** inputImage)
```

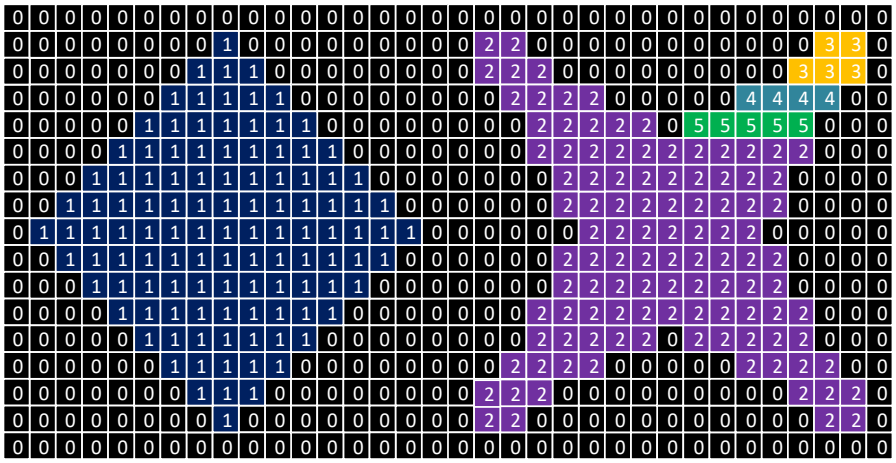


Obr. 7-3 Detekce uzavřených obastí - Vstupní binární obraz

Samotná metoda pracuje v několika krocích. V prvním kroku dojde k vytvoření dočasného obrazu, který je o 1 pixelový lem větší, než obraz původní. Tento krok je proveden z důvodu kontroly krajních pixelů. V dalším kroku je obraz procházen pomocí dvou smyček `for`, pro průchod řádků a sloupců obrazu. Je-li aktuální pixel nenulový (nesoucí informaci) dojde ke kontrole sousedních pixelů. Maska kontroly sousedních pixelů je zobrazena na Obr. 7-4, kde čísla v masce reprezentují pořadí, v jakém jsou sousední pixely kontrolovány. Je-li jeden z jeho sousedů již označen číslem, dojde k označení aktuálního pixelu číslem stejným. V opačném případě dojde k označení číslem novým. Takto je procházen celý obraz.

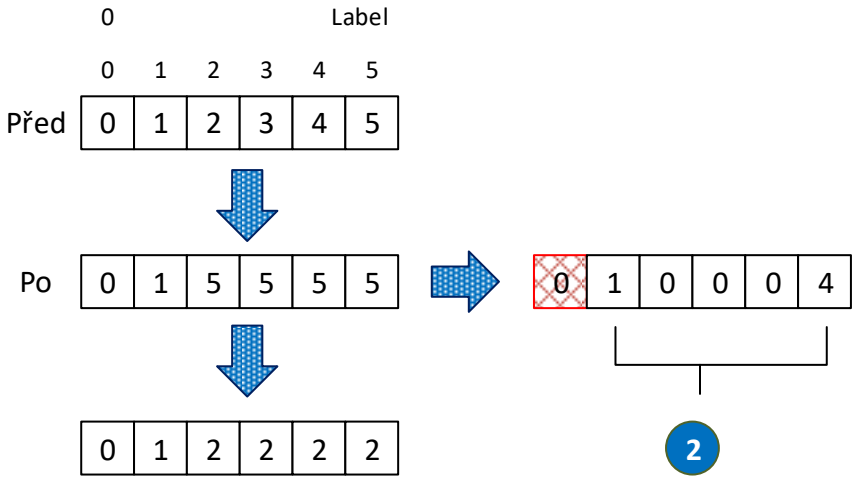


Obr. 7-4 Detekce uzavřených oblastí - Maska kontroly sousedních pixelů



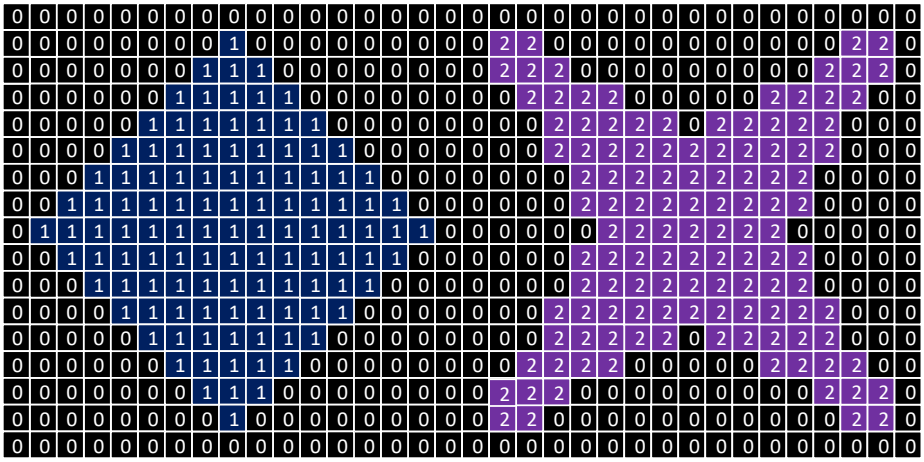
Obr. 7-5 Detekce uzavřených oblastí – 1. fáze

Na příkladu obrázku Obr. 7-5 lze vidět situace, kde jedna uzavřená oblast je označena vícero hodnotami. Proto musí dojít ke korekci. V první fázi korekce dojde k vytvoření 1D pole označených oblastí, kde index pole reprezentuje čísla oblastí. Toto pole je naplněno stejnými hodnotami podle indexu. V další fázi je procházen vytvořený obraz s označenými oblastmi a dochází ke kontrole, zda aktuální pixel je nenulový a zároveň, zda jeden z jeho sousedů není také nenulový a jeho hodnota je jiná než hodnota aktuálního pixelu. V případě, že ano, dojde k přepsání hodnoty v poli, pro následné sloučení sousedních oblastí. Na výstupu této části je upravené pole, jejichž hodnoty však nejsou seřazeny postupným číslováním (některé číslice jsou vynechány a některé jsou zastoupeny vícenásobně).



Obr. 7-6 Detekce uzavřených oblastí - Princip korekce

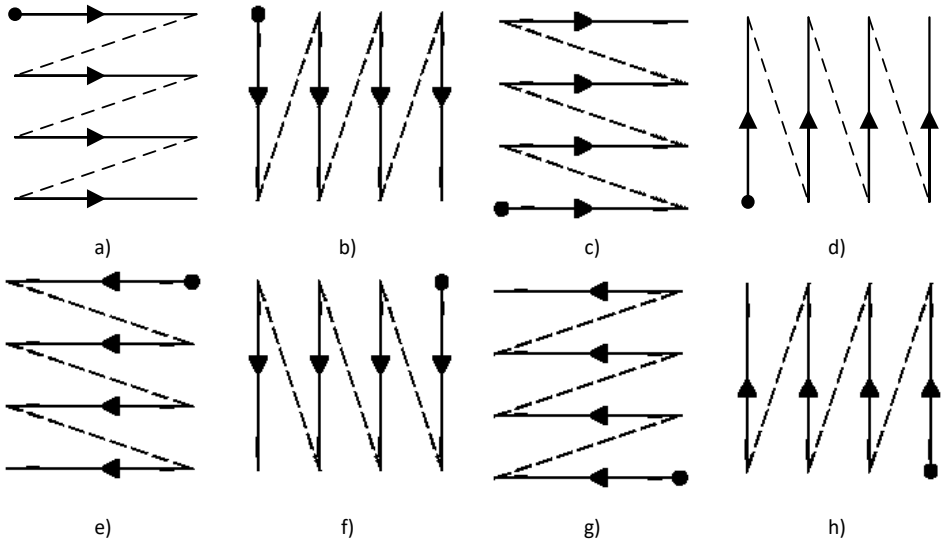
Seřazení hodnot předchází zjištění počtu oblastí po korekci, tzn. počet uzavřených oblastí obrazu. Smyčkou `for` je procházeno pole po korekci a je počítáno kolikrát se daná hodnota v poli nachází, tyto hodnoty jsou ukládány do pole 2, výjimkou je hodnota 0 reprezentující pozadí. Následně jsou v poli 2 spočítány nenulové hodnoty. Výsledek reprezentuje počet uzavřených oblastí vstupního obrazu. Na základě seřazených hodnot v poli oblastí je upraven obraz, tento výstupní obraz je zobrazen na Obr. 7-7.



Obr. 7-7 Detekce uzavřených oblastí - Výsledný obraz

7.3.2 Zjištění parametrů oblastí

Metodu pro zjištění parametrů uzavřených oblastí, lze použít až po samotné detekci uzavřených oblastí. Výstupem této metody je pole o velikosti [5][n+1], kde *n* je počet uzavřených oblastí obrazu. V poli jsou uloženy parametry jednotlivých uzavřených oblastí (*x* a *y* souřadnice středu, šířka, výška a velikost oblastí).



Obr. 7-8 Směry procházení obrazu

Pro zjištění parametrů je obraz procházen všemi osmi směry, jak ukazuje Obr. 7-8. V každém směru jsou hledány krajní body jednotlivých oblastí. Nalezené krajní body jsou seřazeny od nejmenší po největší pro zisk nejmenší a největší souřadnice dané oblasti, na základě kterých jsou spočteny parametry oblasti podle následujících rovnic:

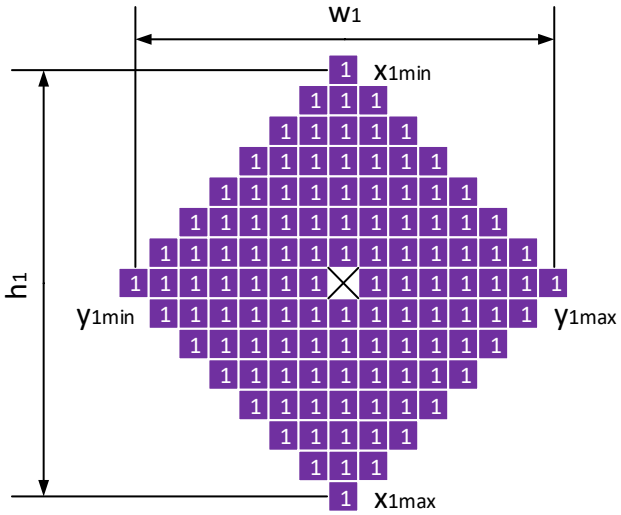
$$h_N = y_{Nmax} - y_{Nmin} \tag{7.1}$$

$$w_N = x_{Nmax} - x_{Nmin} \tag{7.2}$$

$$S_{XN} = \frac{w_N}{2} + x_{Nmin} \tag{7.3}$$

$$S_{YN} = \frac{h_N}{2} + y_{Nmin} \tag{7.4}$$

, kde h – výška oblasti, w – šířka oblasti, N – číslo oblasti, y_{max} – největší y souřadnice oblasti, y_{min} – nejmenší y souřadnice oblasti, x_{max} – největší x souřadnice oblasti, x_{min} – nejmenší x souřadnice oblasti, S_X – x souřadnice středu, S_Y – y souřadnice středu.



Obr. 7-9 Parametry oblastí

7.3.3 Úhel a vzdálenost dvou oblastí

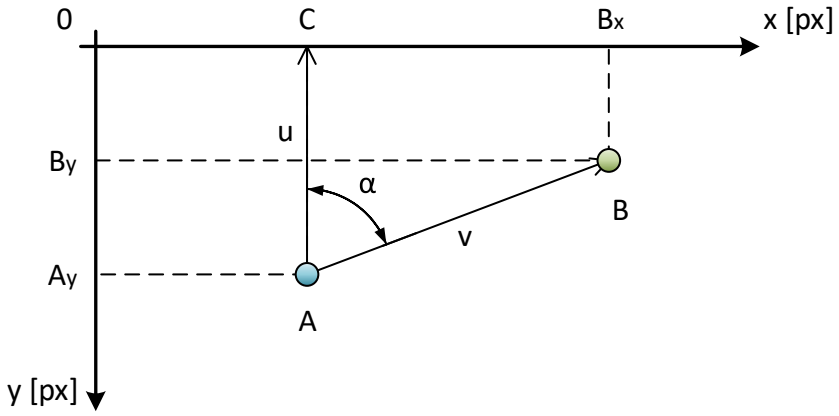
Nalezené binární oblasti v obraze si lze představit jako dva body v rovině, jejichž pozice jsou definovány středovými souřadnicemi jednotlivých oblastí. Bod, jehož x souřadnice je menší, je nazván bodem A, druhý bod, bodem B. Vzdálenost mezi těmito body je vypočtena jako velikost vektoru v, definována těmito dvěma body.

$$\vec{v} = (v_1, v_2) = B - A = (B_x - A_x; B_y - A_y) \tag{7.5}$$

$$|\vec{v}| = \sqrt{v_1^2 + v_2^2} \tag{7.6}$$

Pro výpočet úhlu mezi dvěma oblastmi je z bodu *A* vztyčena kolmice do počátku osy x. Takto vznikne vektor *u* a úhel je počítán jako úhel dvou vektorů:

$$\cos\alpha = \frac{\vec{u} \cdot \vec{v}}{|\vec{u}| \cdot |v|} = \frac{u_1 v_1 + u_2 v_2}{\sqrt{u_1^2 + u_2^2} + \sqrt{v_1^2 + v_2^2}} \tag{7.7}$$



Obr. 7-10 Výpočet vzdálenosti a úhlu dvou oblastí

8. SYSTÉM VIZUÁLNÍHO ROZHODČÍHO

Podle návrhu a s využitím vytvořené knihovny pro zpracování obrazu byl realizován systém vizuálního rozhodčího pro stolní hru Qwirkle. Systém byl realizován v jazyce C++ v programovacím prostředí Microsoft Visual Studio 2017. Systém vizuálního rozhodčího lze rozdělit do několika částí: detekce provedení tahu hry, zpracování obrazu a herní analýza. Následující podkapitoly mají za úkol popsat jednotlivé zmíněné části, společně s grafickým uživatelským rozhraním a návrhem konstrukce.

8.1 Detekce provedení tahu

Před samotným zpracováním snímaného obrazu musí dojít ke zjištění stavu, kdy došlo k provedení tahu hry. Z teoretických předpokladů by k tahu mělo dojít, jakmile dojde ke změně scény a následně pak k jejímu ustálení. Pro zjišťování těchto změn scény dochází ke snímání scény v pravidelných časových intervalech 750 milisekund a k porovnávání sekvence snímků, na základě rozdílové metody analýzy obrazu. V nasnímaném obraze se však díky záznamovému zařízení nachází hodně šumu, který je proměnlivý v čase a způsobuje velké změny jasových úrovní barevného obrazu. V obraze jsou proto hledány oblasti zájmu – herní kameny, vůči kterým je změna scény hledána. Hledání herních kamenů je popsáno v následující kapitole, jejichž výsledkem je binární obraz, kde oblasti zájmu (herní kameny) nabývají hodnotu 1, zatímco pozadí 0. Ze dvou po sobě jdoucích snímků je vytvořen binární rozdílový obraz, na základě kterého je vypočtena procentuální změna scény podle následující rovnice:

$$z = \frac{\sum f_w(x, y)}{R \times C} \cdot 100\ [\%] \tag{8.1}$$

, kde $\sum f_w(x, y)$ – součet bílých pixelů rozdílového obrazu,

R – počet řádků obrazu,

C – počet sloupců obrazu.

V případě, kdy procentuální změna scény je menší než $5\ \%$, je scéna považovaná za ustálenou. V opačném případě se jedná o scénu, ve které dochází k pohybu.

8.2 Předzpracování obrazu

Po získání barevného snímku z kamery dojde k jeho zmenšení na rozměr 640×480 pixelů. Z testování je tento rozměr dostačující pro analýzu hry a zároveň se snižuje výpočetní náročnost v případě použití kamery s vysokým rozlišením. Platí totiž pravidlo, že čím méně dat je dostupných, tím rychlejší je jejich samotné zpracování.

Na základě kritérií barev herních kamenů daných rozsahy prahů pro každou barevnou složku obrazu zvlášť, je obraz segmentován. Rozsahy prahových hodnot pro jednotlivé barvy jsou dány následujícími rovnicemi. Tyto prahy se však při různých světelných podmínkách mohou lišit, proto je v grafickém uživatelském rozhraní umožněna korekce těchto barev a prahy mohou být libovolně změněny.

- **Fialová barva:**

$$(1 \leq R \leq 50) \cap (0 \leq G \leq 61) \cap (100 \leq B \leq 235) \quad \textbf{(8.2)}$$

- **Zelená barva:**

$$(0 \leq R \leq 22) \cap (45 \leq G \leq 132) \cap (0 \leq B \leq 30) \quad \textbf{(8.3)}$$

- **Modrá barva:**

$$(0 \leq R \leq 20) \cap (16 \leq G \leq 140) \cap (150 \leq B \leq 255) \quad \textbf{(8.4)}$$

- **Oranžová barva:**

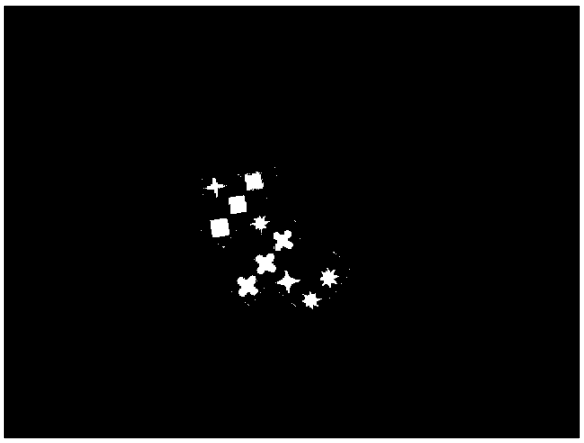
$$(100 \leq R \leq 255) \cap (16 \leq G \leq 82) \cap (0 \leq B \leq 61) \quad \textbf{(8.5)}$$

- **Žlutá barva:**

$$(115 \leq R \leq 255) \cap (137 \leq G \leq 255) \cap (0 \leq B \leq 110) \quad \textbf{(8.6)}$$

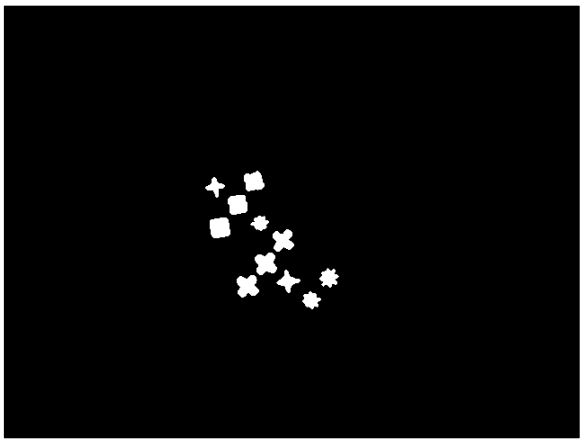
- **Červená barva:**

$$(65 \leq R \leq 215) \cap (0 \leq G \leq 20) \cap (0 \leq B \leq 20) \quad \textbf{(8.7)}$$



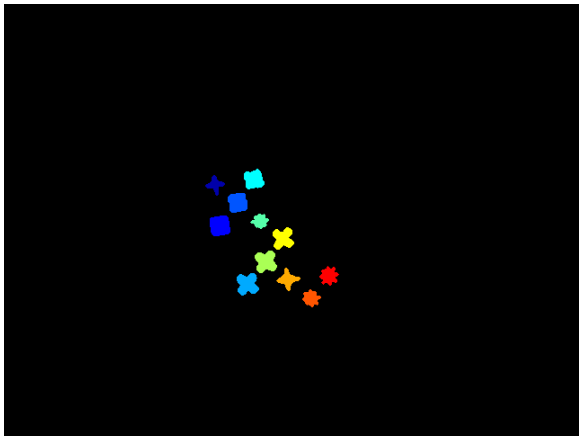
Obr. 8-1 Binární obraz po prahování

Takto vytvořený binární obraz je vyhlazen pomocí medián filtru. Při tomto vyhlazení dochází k eliminaci šumu přeneseného z barevného obrazu. Tento šum vytváří oblasti velikosti v jednotkách pixelů, proto postačí tento filtr aplikovat pro okolí 3×3. Aplikací morfologických operací dojde následně i k eliminaci malých shluků, které nepředstavuji oblast zájmu.



Obr. 8-2 Binární obraz po extrakci extrémů

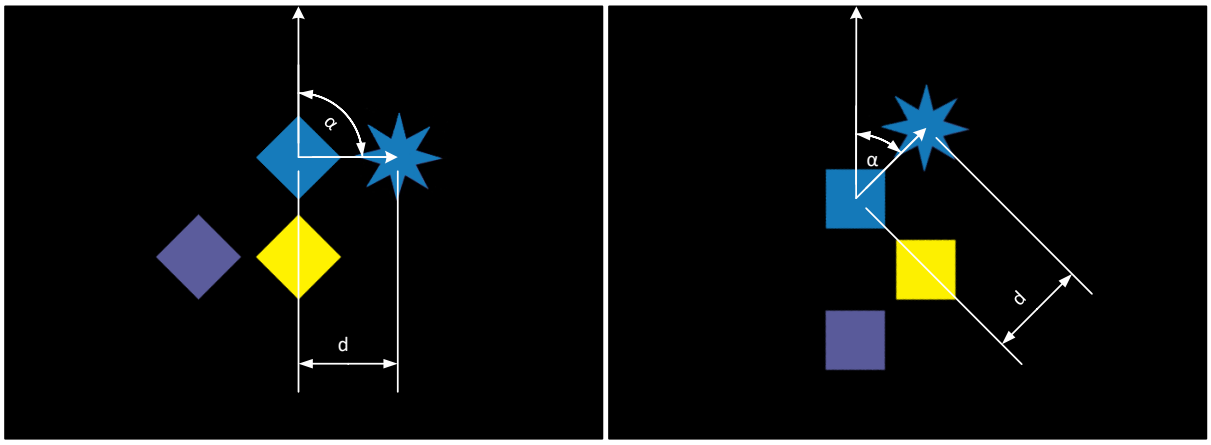
Vlivem špatně zvoleného prahu barev, může při segmentaci dojít ke stavu, že všechny tyto oblasti neodpovídají oblastem reprezentující herní kámen. Proto dojde k eliminaci oblastí, které jsou menší než 50% průměrné velikosti všech nalezených oblastí. Výsledky takto aplikovaných metod, lze vidět na obrázku Obr. 8-2. Před zjištěním velikostí jednotlivých oblastí, musí dojít k detekci těchto uzavřených oblastí. To je provedeno voláním metody `blobDetect`, která označí každou uzavřenou oblast jiným číslem tak, aby se dalo s každou oblastí pracovat zvlášť. Zároveň dojde k určení počtu uzavřených oblastí. Celý princip této metody je popsán v kapitole 7.3.1. Po detekci uzavřených oblastí jsou metodou `getLabelParameters` nalezeny parametry jednotlivých oblastí a je spočtena průměrná velikosti nalezených oblastí.



Obr. 8-3 Detekce uzavřených oblastí

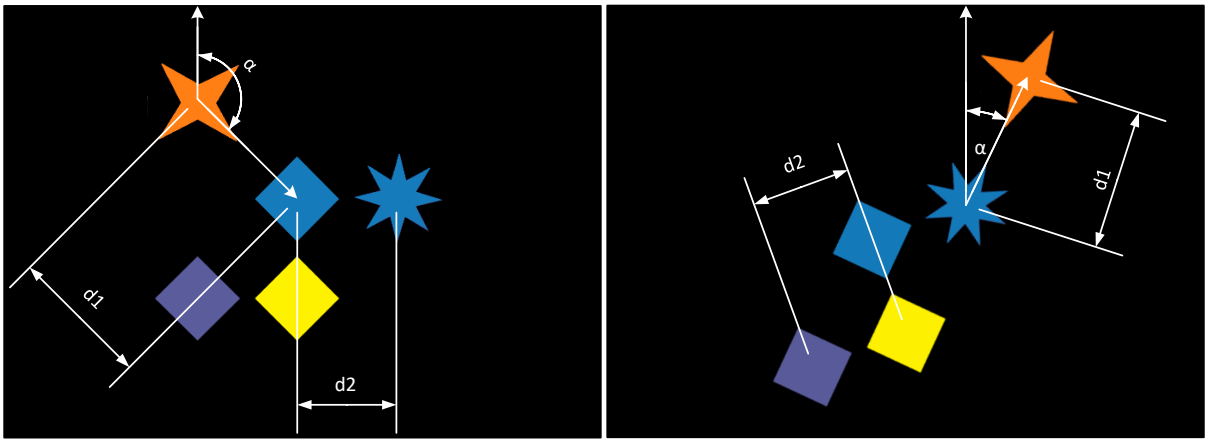
Celý tento proces zajistí nalezení uzavřených oblastí reprezentující herní kameny. Pro rozpoznání tvarů herních kamenů a následné vytvoření herního pole musí být zajištěno, aby kameny byly rovnoběžně s okrajem obrazu. Vždy tomu tak nemusí být, proto musí dojít k natočení obrazu. Obraz je otáčen na základě postavení uzavřených oblastí v obraze.

Podle pravidel jsou herní kameny přidávány do mřížky tak, aby nestály osamoceně a dotýkaly se vzájemně svou stranou. V ideálním případě by měly svírat úhel 0°, 90°,180° nebo 270°. Proto je nutné docílit toho, aby uzavřené oblasti reprezentující herní kameny právě tento úhel svíraly. Pro řešení tohoto problému je nápomocna implementovaná metoda `getAngle()` třídy `Parametrs`. Tato metoda s využitím metody `getNearestBlob()` nalezne oblast nacházející se nejblíže k první detekované oblasti a dojde k určení úhlu, jenž svírají. Tyto metody jsou detailněji popsány v kapitole 7.3.3. Natočením obrazu o tento úhel dojde k zarovnání herních kamenů.



Obr. 8-4 Natočení obrazu podle oblastí

Můžou však nastat situace, kdy dojde ke špatnému umístění herního kamene a kameny mohou stát osamoceně. Příklady této situace jsou zobrazeny na obrázku Obr. 8-5. Oproti předchozí situaci je nutné tento obraz otočit o dalších 45°. Při této situaci dochází k tomu, že vzdálenost mezi středy první detekované oblasti a ji nejbližší oblasti je přibližně rovna úhlopříčce vzdálenosti oblastí při správném umístění kamenů. Této skutečnosti se využije pro rozlišení těchto dvou situací. Je nutné však určit vzdálenost oblastí při správném umístění herních kamenů.



Obr. 8-5 Detekce špatně umístěného herního kamene do mřížky

U žádné z dvojic oblastí nemůže být se 100% jistotou tvrzeno, že vzdálenost mezi jejími středy je rovna vzdálenosti při správném umístění herních kamenů. Tuto vzdálenost lze však určit pomocí šířky a výšky symbolů na herních kamenech, které jsou v obraze reprezentovány binárními oblastmi. Šířka a výška obrazců je vždy stejná a jsou si rovny, proto platí následující odvozená rovnice:

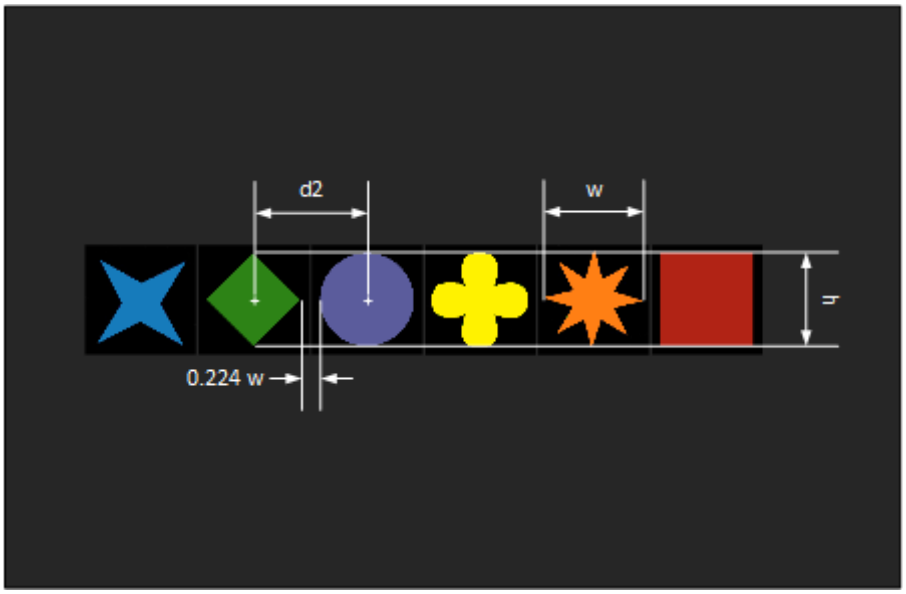
$$w = h \tag{8.8}$$

$$d_2 = w + w * 0{,}224 = h + h * 0{,}224\ [px] \tag{8.9}$$

, kde w – šířka obrazce herního kamene,

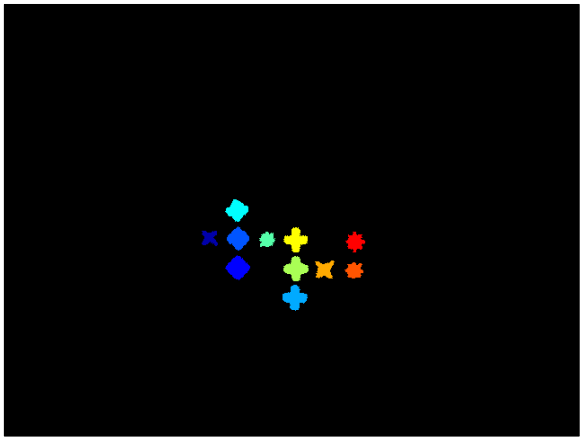
h – výška obrazce herního kamene.

Z teoretické znalosti velikostí herních kamenů a jejich obrazců je odvozena rovnice (8.9) pro výpočet vzdáleností středů oblastí při správném umístění herních kamenů. Šířka a výška nalezených oblastí však závisí na segmentaci barevného obrazu. Může tedy docházet k odchylkám, proto je vzdálenost počítána z aritmetického průměru rozměrů všech nalezených oblastí.



Obr. 8-6 Vzdálenosti herních kamenů podle jejich velikosti

Obraz je tedy otočen na základě těchto kritérií: Je-li $d_1 \approx d_2$, dojde k otočení obrazu o úhel α. V případě, kdy $d_1 \approx d_2\sqrt{2}$ dojde k otočení obrazu o úhel $(\alpha - 45°)$. V ostatních případech se jedná o chybu, kdy došlo ke špatnému umístění herního kamene.



Obr. 8-7 Natočený obraz s nalezenými oblastmi

8.3 Rozpoznání barvy

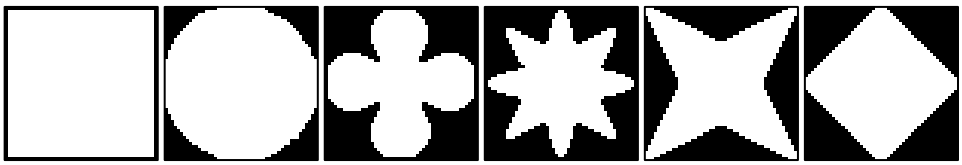
Na základě parametrů nalezených oblastí jsou zkoumané oblasti postupně vyříznuty z originálního RGB obrazu, tak, aby byly zachovány všechny barvy. Každá oblast je dále procházena pixel po pixelu a je zkoumáno, kolik bodů oblasti spadá do předem definovaných prahů barev. Tyto hodnoty jsou ukládány do pole o velikosti 1×6, kde index reprezentuje jednotlivou barvu (1 – fialová, 2 – zelená, 3 – modrá,

4 – oranžová, 5 – žlutá, 6 – červená). Nalezením maxima v poli je určena výsledná barva oblasti, tedy i herního kamene. Barva kamene je následně reprezentována číslem indexu pole.

8.4 Rozpoznání tvaru

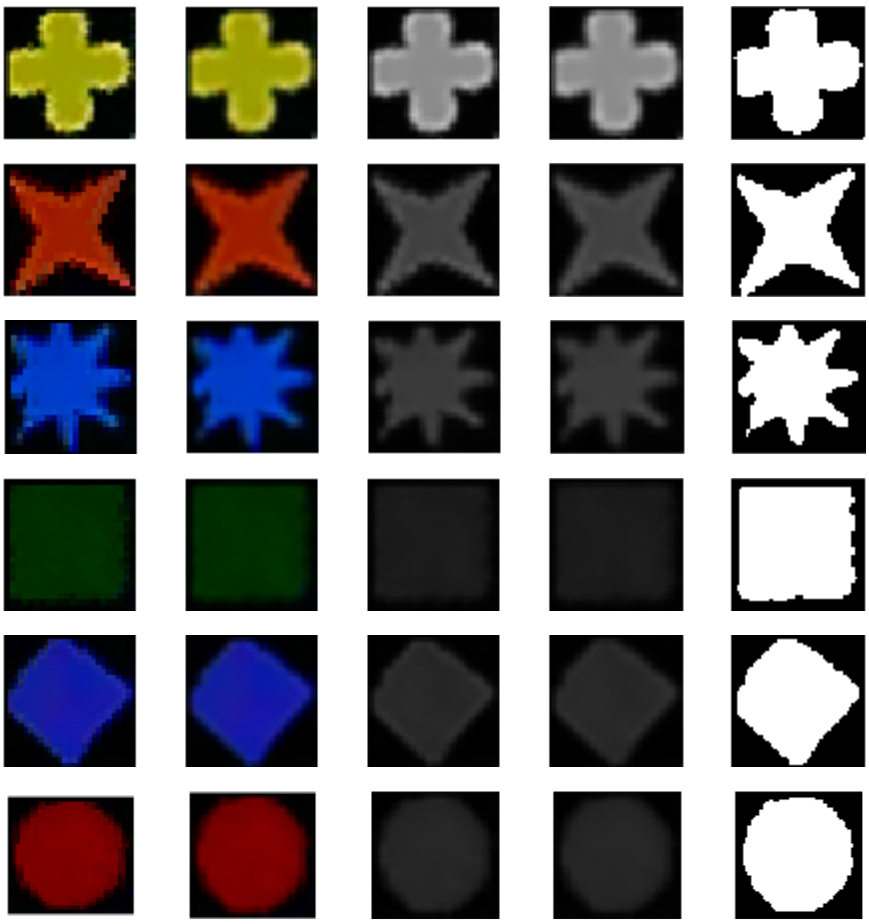
Při rozpoznávání tvarů se vychází ze znalosti objevujících se tvarů ve hře – kruh, čtverec, kosočtverec, čtyřcípá hvězda, osmicípá hvězda, a květina. Na základě této znalosti byla vytvořena sada 6 vzorů těchto tvarů o normalizované velikosti 50×50 pixelů. Samotné rozpoznávání tvarů nalezených oblastí probíhá v následujících krocích:

- Vyříznutí oblasti z RGB obrazu
- Normalizace velikosti
- Převedení obrazu na obraz v odstínech šedi
- Nalezení prahu
- Převedení obrazu v odstínech šedi na obraz binární
- Vyhodnocení – určení tvaru



Obr. 8-8 Sada vytvořených vzorů

Obdobně jako při rozpoznávání barvy oblasti je zkoumaná oblast nejdříve vyříznuta z originálního RGB obrazu na základě parametrů oblasti. Následně je vybraná oblast normalizována, tak, aby její velikost korespondovala s velikostí vytvořených vzorů, 50×50. Tento obraz je převeden na obraz v odstínech šedi. Z histogramu obrazu v odstínech šedi je pomocí Otsuovy metody zjištěna úroveň prahu pro vytvoření binárního obrazu. Tato metoda spočívá v nalezení optimálního prahu na základě předpokladu, že histogram obrazu je bimodální. Binární obraz oblasti je porovnáván se všemi vzory pomocí korelace. Při shodě daného pixelu zkoumané oblasti s pixelem vzoru je inkrementována určitá pozice vytvořeného pole. Velikost pole je 1×6, kde index pole reprezentuje jednotlivý tvar (1 – kruh, 2 – čtverec, 3 – kosočtverec, 4 – čtyřcípá hvězda, 5 – osmicípá hvězda, 6 – květina). Po průchodu celého obrazu jsou jednotlivé hodnoty přepočteny na procenta a pole pak uchovává hodnoty pravděpodobnosti zkoumané oblasti k určitému tvaru v procentech. Nalezením maxima v poli je určen výsledný tvar oblasti. Výsledný tvar je následně reprezentován číslem indexu pole.



Obr. 8-9 Princip rozpoznání symbolů herních kamenů

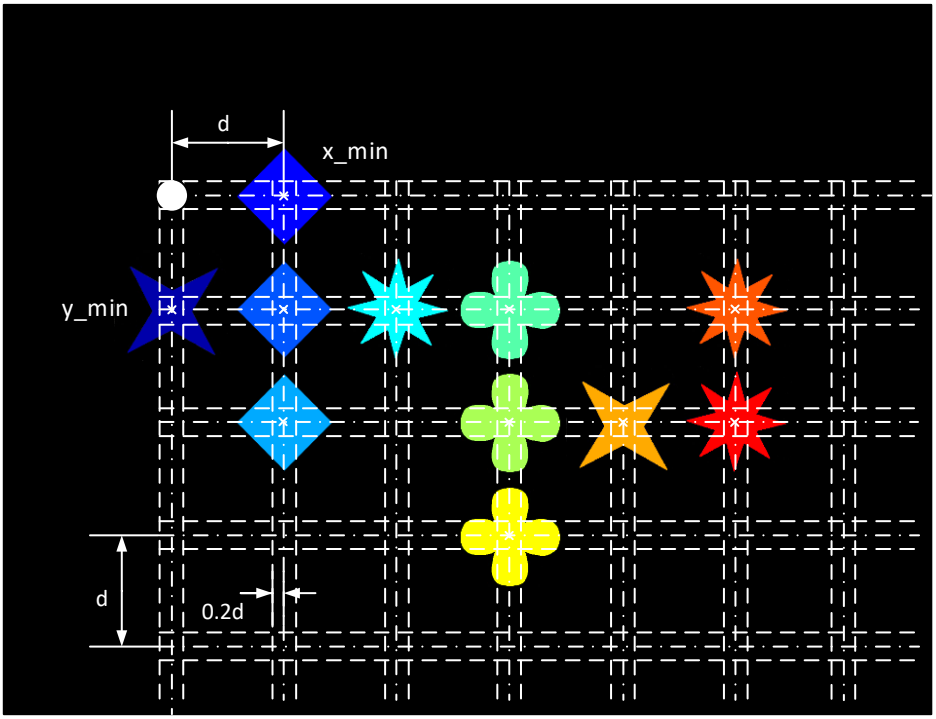
8.5 Herní analýza

Na základě nalezených oblastí a jejich parametrů dochází k analýze samotné hry. Pro analýzu hry jsou vytvořeny dvě třídy. Třída herního pole `gameLayout` a třída rozhodčího pro bodové ohodnocení `gameLogic`. V aplikaci je třída `gameLayout` zastoupena dvěma instancemi. Objektem `ActualGame`, reprezentující herní pole vytvořené na základě aktuálně nasnímaného obrazu. A objektem `FullGame`, který reprezentuje celkové herní pole, jenž je průběžně během hry aktualizováno na základě aktuálního herního pole. V následujících podkapitolách je popsán princip vytváření herního pole s následným výpočtem bodů.

8.5.1 Vytváření herního pole

Při vytváření herního pole je jeho velikost pevně stanovena na velikost 100×100. Princip naplnění herního pole je nastíněn na obrázku Obr. 8-10, kde však místo číselného označení oblastí, jsou pro názornost oblasti rozlišeny barevně. Nalezením minim x a y souřadnic středů všech oblastí, jsou získány

krajní souřadnice, které slouží jako referenční poloha a určují pomyslný počátek. Pomocí tří vnořených smyček *for* je vytvořeno herní pole. První z nich slouží pro procházení řádků a s každou iterací je navyšována hodnota souřadnice y o hodnotu vzdálenosti středů sousedních hracích kostek. Druhá smyčka slouží pro procházení sloupců a obdobně jako u smyčky první dochází s iterací k navyšování x souřadnice. Ve třetí smyčce jsou procházeny parametry jednotlivých oblastí a dochází ke kontrole, zda střed některé z oblastí nezapadá do meze vzdálenosti středů $\pm 30\%$ vůči novým x, y souřadnicím. Když střed některé z oblastí spadá do meze, je na pozici pole zapsán index reprezentující danou oblast, v opačném případě je na pozici pole zapsána nula reprezentující pozadí.



Obr. 8-10 Princip vytváření herního pole

Takto vytvořené herní pole (Obr. 8-11a) je třeba upravit. U objektu herního pole z aktuálně nasnímaného obrazu dochází k vykopírování části herního pole obsahující pouze nenulové hodnoty, a k dekódování těchto oblastí na základě rozpoznaných barev a tvarů oblastí. Je tedy vytvořeno trojrozměrné pole o velikosti $R \times C \times 2$, kde počet řádků a sloupců pole je zjištěn na základě souřadnic krajních nenulových hodnot původního herního pole. Výpočet řádků a sloupců nového pole je definován rovnicemi (8.10) a (8.11).

$$R = y_{max} - y_{min} + 1 = \mathrm{y}_{max} + 1 \tag{8.10}$$

$$C = x_{max} - x_{min} + 1 = x_{max} + 1 \tag{8.11}$$

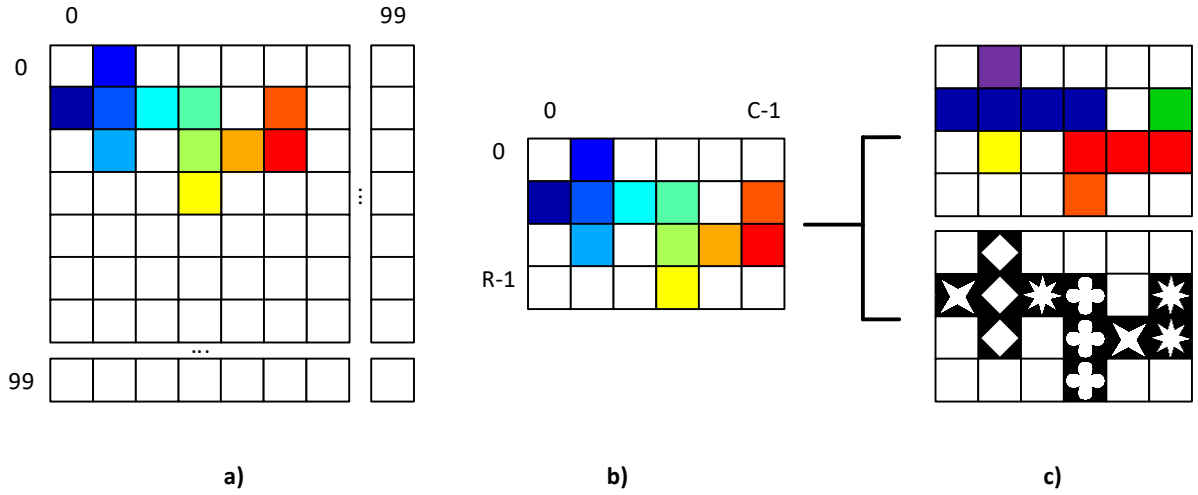
, kde R – počet řádků pole

C – počet sloupců pole

y_{max}, y_{min} – největší a nejmenší y souřadnice

x_{max}, x_{min} – největší a nejmenší x souřadnice.

Krajní souřadnice jsou hledány obdobně jako u metody pro zjištění parametrů uzavřených oblastí (7.3.2). Dochází k procházení pole, pouze však odzadu, jelikož jeho nejmenší souřadnice jsou nulové. Z nalezených krajních bodů jsou vybrány maxima, které určují největší x a y souřadnice.



Obr. 8-11 Fáze herního pole a) Vytvořené herní pole, b) Vyříznuté herní pole, c) Dekódované herní pole

Fáze vytvoření herního pole aktuálně nasnímaného obrazu jsou zobrazeny na obrázku Obr. 8-11. Pro názornost jsou na obrazech prvky polí označeny barevně, popřípadě obrázkem tvarů. Ve skutečnosti se ve všech případech jedná o pole celočíselného datového typu. V případě pole zobrazeného na obrázku Obr. 8-11c platí, že jednotlivé barvy a tvary jsou reprezentovány čísly zmíněných v kapitolách 8.3 a 8.4.

K vytvoření celkového herního pole hry dochází při prvním bodovaném tahu hry, jsou-li splněny podmínky pro nové kameny. To je vytvořeno vložením aktuálního dekódovaného herního pole do středu matice o velikosti $100 \times 100 \times 2$. V případě ostatních bodovaných tahů hry, kdy je již celkové herní pole vytvořeno, dochází k aktualizaci tohoto pole na základě přidání nově nalezených herních kamenů.

8.5.2 Porovnání herních polí

Metoda slouží pro porovnání výřezu hracího pole s herním polem v plné velikosti (100×100). Tuto metodu je možné využít při porovnání celkového herního pole, které bude systematicky vytvářeno během hry a oříznutého herního pole vytvořeného na základě aktuálně nasnímané části herní plochy.

Po nasnímání dalšího tahu, nemusí kamera snímat celou hrací plochu, ale pouze její část. Proto je část nasnímaného pole porovnávána s postupně vytvořeným hracím polem, aby se zjistilo, kam zapadá. Hrací pole vytvořené na základě aktuálně nasnímané části hrací plochy je porovnáváno ve čtyřech iteracích, kde každá iterace reprezentuje otočení herního pole o 90°, v případě, kdyby došlo k natočení kamery. Celkové herní pole je při každém natočení procházeno od počátku, pouze však do pozice $100 - výška/šířka$ aktuálně nasnímaného herního pole. Celkové herní pole nemůže být procházeno do jeho konečné pozice 100, jelikož by došlo k porovnávání hodnot mimo rozsah pole. Výsledkem je trojrozměrná matice velikosti $100 \times 100 \times 4$ s hodnotami součtu shodných prvků momentálně nasnímaného a celkového hracího pole. Nalezením maxima této matice dojde k získání souřadnic, které definují natočení a pozici pro umístění momentálně nasnímaného herního pole do herního pole celkového. Vkládáním aktuálního hracího pole na vypočtené pozice, dojde ke zjištění počtu nových kamenů v daném tahu hry a jejich pozice v poli jsou zaznamenány do matice pro jejich následnou kontrolu.

8.5.3 Kontrola podmínek nových kamenů

Pro nalezené nové kameny jsou definovány podmínky vyplývající z pravidel hry.

- **Kontrola, zda existuje nový kámen.**

Toto pravidlo platí nad rámec všech ostatních pravidel. Není-li splněna podmínka, že počet nových kamenů je větší než nula, není důvod kontrolovat další definované podmínky.

- **Kontrola, zda nové kameny nestojí osamoceně.**

Hra neumožňuje stav, kdy by kámen stál osamoceně. Proto pro každý nový kámen dochází ke kontrole sousedních kamenů v rámci jeho čtyř-okolí. Podmínka je splněna, v případě, zda existuje alespoň jeden sousední kámen.

- **Kontrola, zda nové kameny jsou v jedné řadě.**

V jednom tahu hry, je možné přidávat kameny pouze do jedné z vytvořených řad. Proto dochází ke kontrole, zda nové kameny opravdu spadají do jedné řady. K této kontrole dochází při stavu, kdy je více nových kamenů než jen jeden. Dochází ke kontrole souřadnic herních kamenů v rámci vytvořeného herního pole. Podmínka je splněna, mají-li všechny nové kameny jednu ze souřadnic stejnou.

- **Kontrola, zda nové kameny mají společnou vlastnost (stejný tvar nebo barvu).**

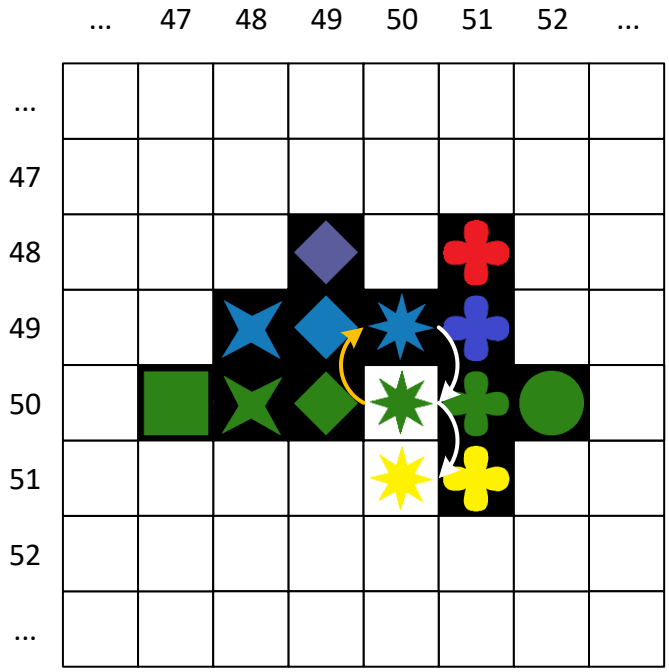
Jelikož nové kameny musí spadat do jedné řady, která je buď složena z jedné barvy, nebo z jednoho tvaru, jsou nové kameny kontrolovány, zda mají jednu ze společných vlastností. Každý kámen je v herním poli zastoupen číslem reprezentující barvu a tvar daného kamene. Při kontrole splnění této podmínky dochází k porovnávání jednotlivých vlastností nových kamenů. Podmínka je splněna, v případě shodnosti jedné vlastnosti pro všechny nové kameny.

- **Kontrola, zda některé z nových kamenů nejsou stejné.**

Řada tvořená tvary může obsahovat pouze jeden kámen od každé ze šesti barev. Například řada čtverců může obsahovat pouze jeden modrý čtverec. Řada tvořená barvami může obsahovat pouze jeden kámen od každého ze šesti tvarů. Například žlutá řada může obsahovat pouze jeden žlutý kruh. Proto dochází ke kontrole, zda některé z dvojic nových herních kamenů nemají obě společné vlastnosti (barvu a tvar).

8.5.4 Bodové hodnocení

Jsou-li splněny všechny podmínky nových kamenů, dochází ke spočtení bodů. Počítání bodů je rozděleno do dvou částí: počítání bodů v horizontálním směru a počítání bodů ve směru vertikálním. Algoritmus pro počítání bodů pracuje následovně: Souřadnice všech nových kamenů jsou zkopírovány do pomocné matice, která se bude během počítání upravovat tak, aby nedošlo ke spočtení bodů v jedné řadě vícekrát vůči jinému novému kameni. Pro každý nový kámen jsou zkoumány jeho souřadnice. V případě, že jeho souřadnice jsou nulové ($x = 0, y = 0$), znamená to, že tento kámen byl již započten do bodování v řadě vůči předchozímu novému kameni a přejde se na další nový kámen. V případě, kdy souřadnice nového kamene jsou nenulové, dojde k nalezení prvního kamene v řadě. Pro horizontální řadu je to kámen nejvíce vlevo, pro řadu vertikální je to kámen nejvíce nahoře od nového kamene. Od nalezeného prvního kamene se prochází řada opačným směrem až na její konec. Tímto průchodem dojde ke kontrole souřadnic kamenů v řadě. Jsou-li souřadnice některého z kamene v řadě rovny souřadnicím některého z nových kamenů, jsou souřadnice daného nového kamene vynulovány, aby stejná řada nebyla započtena vícekrát vůči jinému novému kameni. Průchodem řady od prvního kamene po poslední dochází také ke zjištění počtu kamenů v řadě a k uložení hodnot kamenů do matice, pro následnou kontrolu řady, zda splňuje pravidla hry. Princip průchodu řady je naznačen na obrázku Obr. 8-12, kde kameny s bílým pozadím značí kameny nové. Řada je procházena v rámci prvního nového kamene (zelená osmicípá hvězda) a je hledán její začátek. Po nalezení začátku řady, je řada procházena opačným směrem až na její konec. Součástí této vertikální řady je i druhý nový kámen (žlutá osmicípá hvězda), proto již nedochází ke stejnému šetření řady v rámci druhého nového kamene.



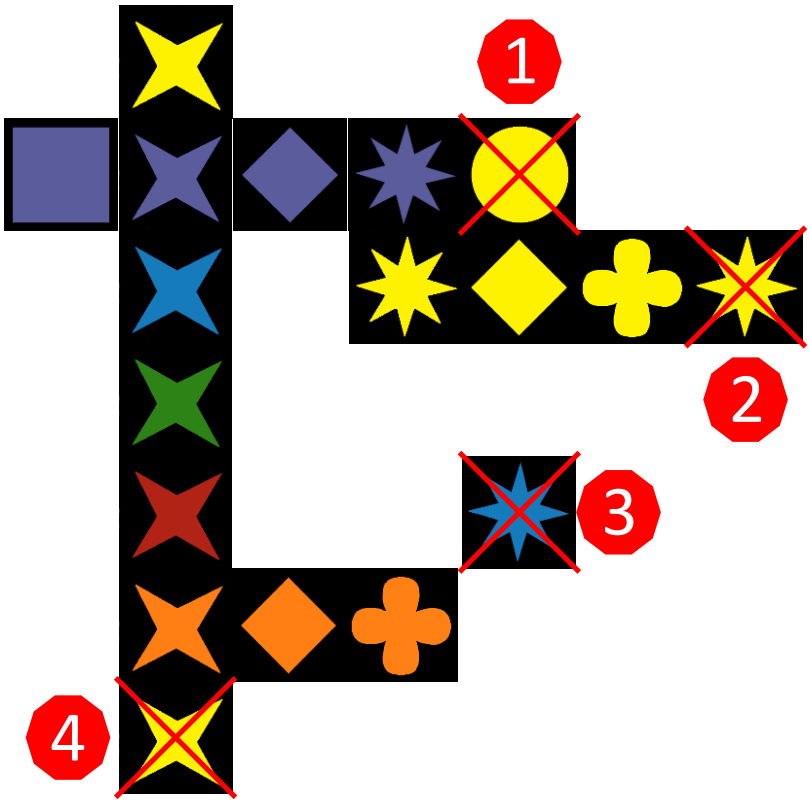
Obr. 8-12 Princip průchodu řady ve vertikálním směru

- **Kontrola, zda počet kamenů v řadě není více než 6**

Není-li splněna podmínka, že počet kamenů v řadě je menší než 6, není důvod kontrolovat další definované podmínky a dojde k vyvolání chybného stavu. Je-li podmínka splněna, dojde ke kontrole dalších podmínek s výjimkou stavu, kdy je řada tvořena pouze jedním kamenem. Při tomto stavu dojde k analýze následujícího nového kamene, bez bodového ohodnocení řady.

- **Kontrola, zda některé kameny v jedné řadě nejsou stejné**
- **Kontrola, zda kameny v řadě mají společnou vlastnost (stejný tvar nebo barvu)**

Jsou-li splněny i výše uvedené podmínky dojde k přičtení tolika bodů, kolika kameny je řada tvořena. Výjimkou je řada tvořena šesti kameny tvořící QWIRKLE, zde dojde k přičtení 12 bodů.

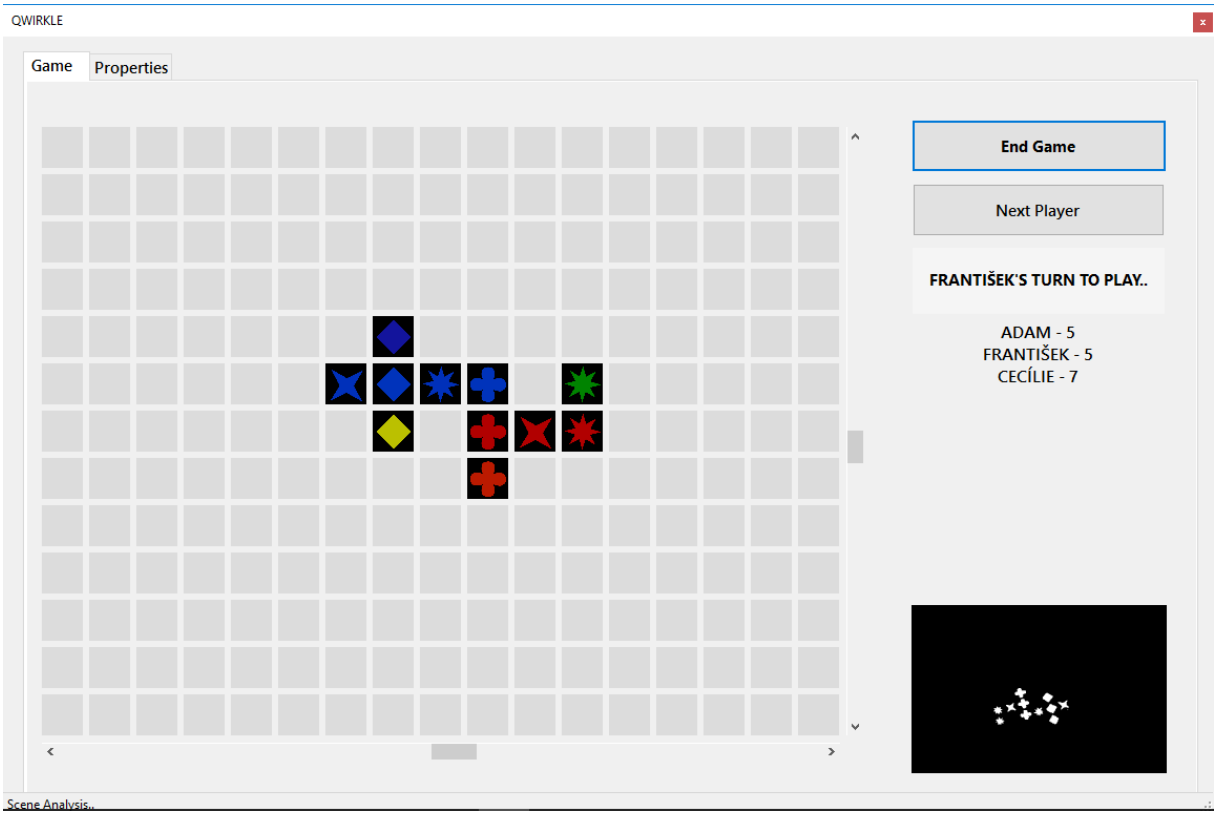


Obr. 8-13 Skládání kamenů

Obrázek Obr. 8-13 ukazuje příklad možnosti skládání herních kamenů do pole. Obrázek taky obsahuje čtyři ukázky špatného skládání herních kamenů. Kámen číslo 1 sice zapadá do vertikální řady se žlutým kosočtvercem i do horizontální řady svým tvarem, ale nezapadá barevně. Kámen číslo 2 zapadá do horizontální řady barvou, jeho tvar se však v řadě již vyskytuje. Kámen číslo 3 se žádnou stranou nedotýká ostatních kamenů, stojí tedy samostatně. Kámen číslo 4 je sedmý kámen v řadě.

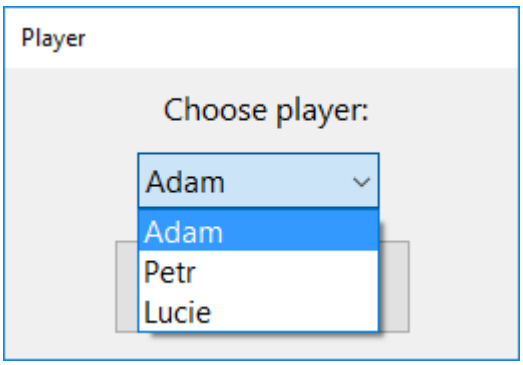
8.6 Grafické uživatelské rozhraní

Pro grafické uživatelské rozhraní je vytvořená formulářová aplikace pro platformu .NET Framework 4.6, která je psaná v jazyce C++/CLI.



Obr. 8-14 Grafické uživatelské rozhraní vizuálního rozhodčího

Formulář uživatelského rozhraní je pomocí ovládacího prvku `TabControl` rozdělen do dvou záložek. První ze záložek obsahuje ovládací prvky pro monitorování hry. Jsou zde umístěny pouze dvě tlačítka. Tlačítko „START“ sloužící pro spuštění a zároveň pro vypnutí vizuálního rozhodčího hry. V případě spuštění hry pomocí tohoto tlačítka dochází k zakázání přepínání mezi záložkami a zároveň dochází k zpřístupnění druhého z tlačítek „NEXT PLAYER“. Toto tlačítko slouží pro volání události změny hráče, který je právě na tahu. Této události se využívá v případě, kdy hráč není schopen odehrát tah a dochází k jeho přeskočení. Při spuštění hry dochází k zobrazení dialogového okna, obsahující `ComboBox` komponentu s předem nastaveným jmenným sezamem hráčů. Tento seznam slouží pro výběr hráče, jenž započne samotnou hru. Podle pravidel je to ten hráč, který je schopen ze svých prvotních šesti herních kamenů složit nejdelší řadu.



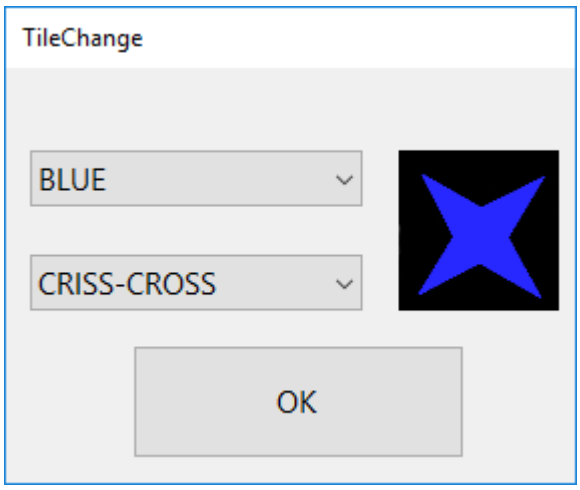
Obr. 8-15 Dialogové okno pro výběr hráče začínající hru

Další komponentou této záložky „Game“ je `TextBox` sloužící k zobrazení skóre hráčů. Vstupním parametrem `TextBoxu` je pole datového typu `String`, proto umožňuje zobrazení skóre všech hráčů. Toto pole je aktualizováno po každém úspěšném tahu hry.

Dominantní částí záložky je panel obsahující 221 komponent typu `PictureBox` rozdělených do 17 sloupců a 13 řádků, vytvářející pomyslnou matici. Každý `PictureBox` má svůj specifický název podle pozice, na které se v rámci matice nachází. Indexy řádků a sloupců pomyslné matice jsou číslovány od nuly, proto `PictureBox` v pravém dolním rohu má název `picBox12_16`. S využitím těchto vlastností a vlastností prvku `ImageList`, který uchovává obrazy jednotlivých herních kamenů, je možné tyto obrazy vykreslovat a vizualizovat tak část herního pole. Obrazy v `ImageListu` jsou číselně pojmenovány na základě označení barev a tvarů jednotlivých herních kamenů podle kapitol 8.3 a 8.4. Součástí panelu jsou také dva posuvníky pro procházení herního pole ve vertikálním a horizontálním směru. Pro vykreslení je procházena část herního pole pomocí dvou cyklů `for`. Tato část herního pole je definována rozsahem matice `PictureBoxů`. V jednotlivých iteracích cyklů je složen název obrazu herního kamene z `ImageListu`, na základě barvy a tvaru v herním poli. Pozice hledaného kamene v herním poli je definována pozicemi posuvníků v součtu s indexy jednotlivých cyklů. Dochází také ke složení názvů `PictureBoxů` na základě indexů cyklů. K jednotlivým `PictureBox` komponentám je pak přistupováno dynamicky podle složeného názvu.

```
safe_cast<PictureBox^>(Controls[picBoxName])->Image = imageList->Images[imageName];
```

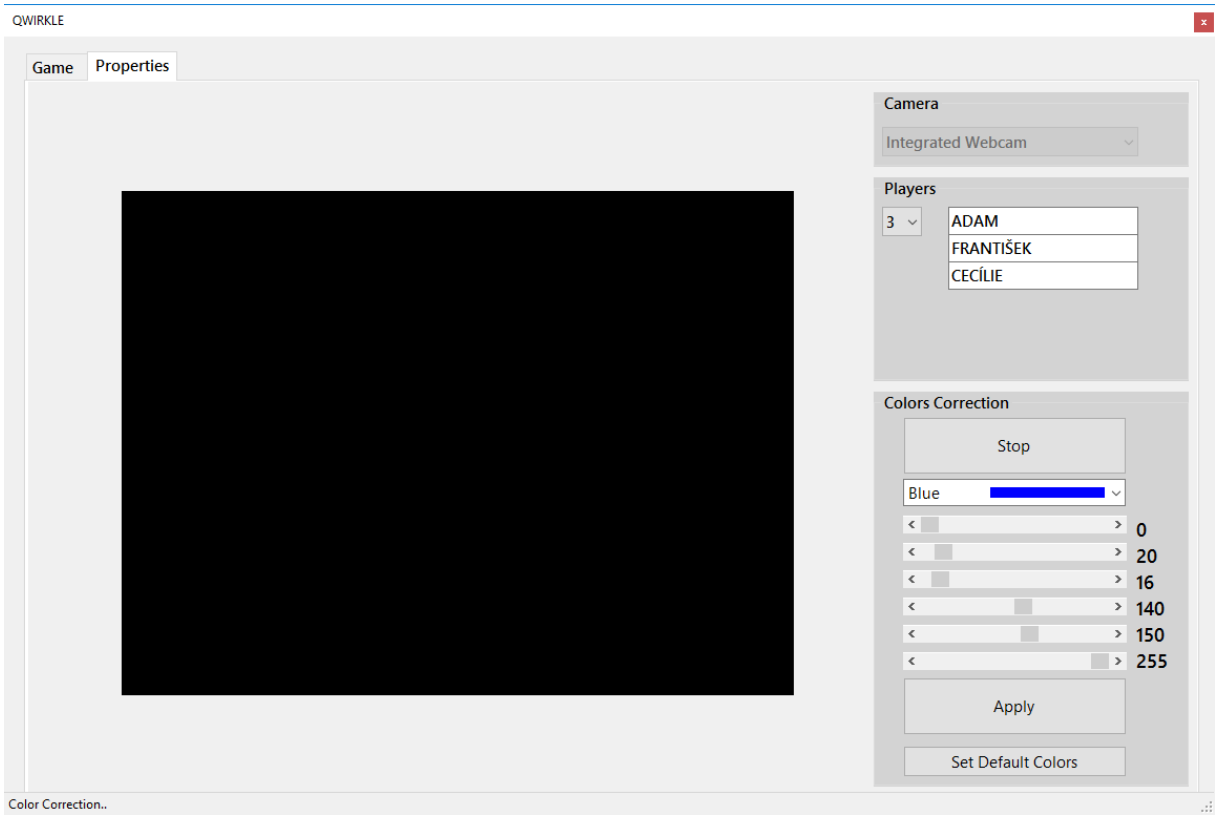
V případě špatného rozpoznání tvaru, či barvy herního kamene, je uživateli umožněna jeho manuální změna v herním poli. Dvojklikem na kterýkoliv z `PictureBoxů` je vyvolána událost pro zobrazení dialogového okna umožňující výběr herního kamene (Obr. 8-16). Výběrem herního kamene dochází ke zjištění názvu komponenty, která tuto událost vyvolala, a tak i ke zjištění pozice v rámci matice `PictureBoxů`. Společně s aktuálními pozicemi posuvníků je určena pozice v celkovém herním poli a hodnota herního kamene změněna dle výběru.



Obr. 8-16 Dialogové okno pro manuální změnu herního kamene v poli

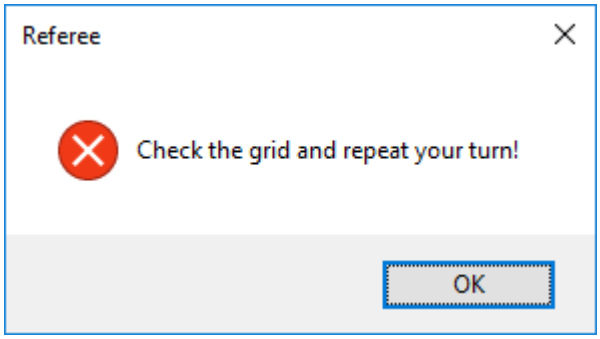
Posledním prvkem první záložky je stavový řádek informující o aktuální fázi zpracování hry. Stavový řádek může nabývat tří stavů. První stav je nazván „Scene Analysis“, informující uživatele o právě probíhající detekci tahu hry. Detekce tahu hry je rozdělena do dvou částí, kdy je nejdříve detekována změna scény a následně její ustálení. Tyto dvě části jsou od sebe rozlišeny přidáním teček k nápisu do stavového řádku. V případě detekce změny scény je ve stavovém řádku zobrazen název „Scene Analysis..“. V případě detekce ustálení scény je zobrazen název „Scene Analysis…“. Druhým informačním stavem je stav „Image Processing“, informující o právě probíhajícím zpracování obrazu. Posledním informačním stavem je stav „Game Analysis“, jenž informuje o právě probíhajícím vyhodnocování pravidel hry.

Druhá záložka obsahuje ovládací prvky pro nastavení parametrů a konfiguraci hry. Je rozdělena do tří skupin pomocí komponenty `GroupBox`. První skupina „Color Correction“ slouží pro korekci prahů barev, které jsou využity pro segmentaci obrazu a nalezení jednotlivých herních kamenů v obraze. Výběrem jedné z šesti barev je možné měnit prahy barev pomocí posuvníků. Skupina obsahuje šest posuvníků, přičemž každá dvojice posuvníků slouží pro nastavení minimální a maximální hodnoty prahu každé z barevných složek RGB obrazu. Stisknutím tlačítka „Apply“ dochází k nastavení prahů pro právě zvolenou barvu. Pro případnou aplikaci výchozích prahů barev slouží tlačítko „Set default“. Druhá skupina ovládacích prvků obsahuje komponentu `ComboBox` pro výběr zařízení pro snímání obrazu. Seznam dostupných zařízení je vytvořen při spuštění aplikace. V průběhu chodu aplikace, již nedochází k aktualizaci tohoto seznamu, proto v případě připojení nového zařízení je nutné mít aplikaci vypnutou, případně musí dojít k jejímu restartu. Poslední skupina slouží pro parametrizaci počtu hráčů společně s jejich jmény. `TextBox` komponenty pro vložení jmen hráčů jsou zobrazovány v závislosti na zvoleném počtu hráčů.



Obr. 8-17 GUI - parametrizace

V průběhu hry a v případě chybně zahraného tahu jsou hráči upozornění na danou chybu formou informačního prvku `MessageBox`. Celkem může ve hře nastat devět chybných stavů vycházejících z pravidel hry. Možné chybové stavy jsou popsány v kapitolách 8.5.3 a 8.5.4.



Obr. 8-18 Chybová hláška

8.7 Konstrukce

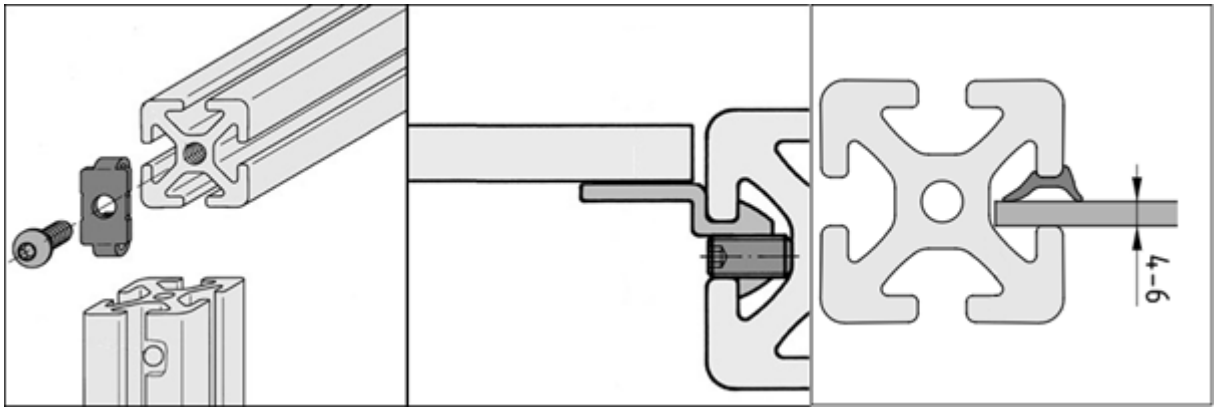
Základní koncepce návrhu konstrukce spočívá v tom, aby tato konstrukce nebyla jednoúčelová, ale aby ji bylo možné použít pro jiné systémy, než jen pro systém vizuálního rozhodčího hry Qwirkle. Při návrhu konstrukce byly hlavní požadavky kladeny na přenositelnost konstrukce s možnosti její rychlé demontáže. K vytvoření rámu konstrukce pro tento účel nejlépe vyhovuje stavebnicový systém hliníkových profilů. Existuje mnoho firem zabývajících se výrobou těchto stavebnicových systémů. Jako příklad uvedu firmy – Industrietechnik und Maschinenbau GmbH, a MayTec GmbH. S každým výrobcem se liší parametry, vlastnosti a dostupnost jednotlivých součástí, proto bylo nutné vytvořit návrh konstrukce a specifikovat součásti pro jednotlivé výrobce zvlášť. Návrhy jednotlivých konstrukcí byly navrženy v softwaru Autodesk Inventor. Na základě cenové kalkulace byl vybrán stavebnicový systém firmy Item, dodávaný firmou Habekorn. Výrobce Item nabízí profily v pěti rozměrových řadách, pro tento případ byla vybrána řada Profile 8.

Obr. 8-19 Návrh modelu konstrukce

Návrh konstrukce spočívá ve vytvoření kvádru, který lze rozdělit na tři části: dolní podstava, horní podstava a čtyři stojné nohy, spojující tyto podstavy. Tento návrh vychází z požadavku možnosti jednoduché demontáže konstrukce. Jednotlivé profily vytvářející kvádr jsou spojeny šroubovým spojem Standard-Fastening Set 8, zobrazeném na obrázku Obr. 8-20a).

Hra Qwirkle neobsahuje žádnou hrací desku, jako většina stolních her tohoto typu. Herní kameny se mohou rozkládat různě do prostoru a mohou tak zabírat různou plochu. S ohledem na tento fakt, jsou vnější rozměry podstav konstrukce zvoleny na základě herní praxe 1000×800 milimetrů. Součástí

dolní podstavy je i kompozitní bílá deska sloužící jako podklad pro hraní samotné hry. Deska je ke konstrukci ukotvena pomocí hliníkové úhlové lišty Rebate Profile 8.

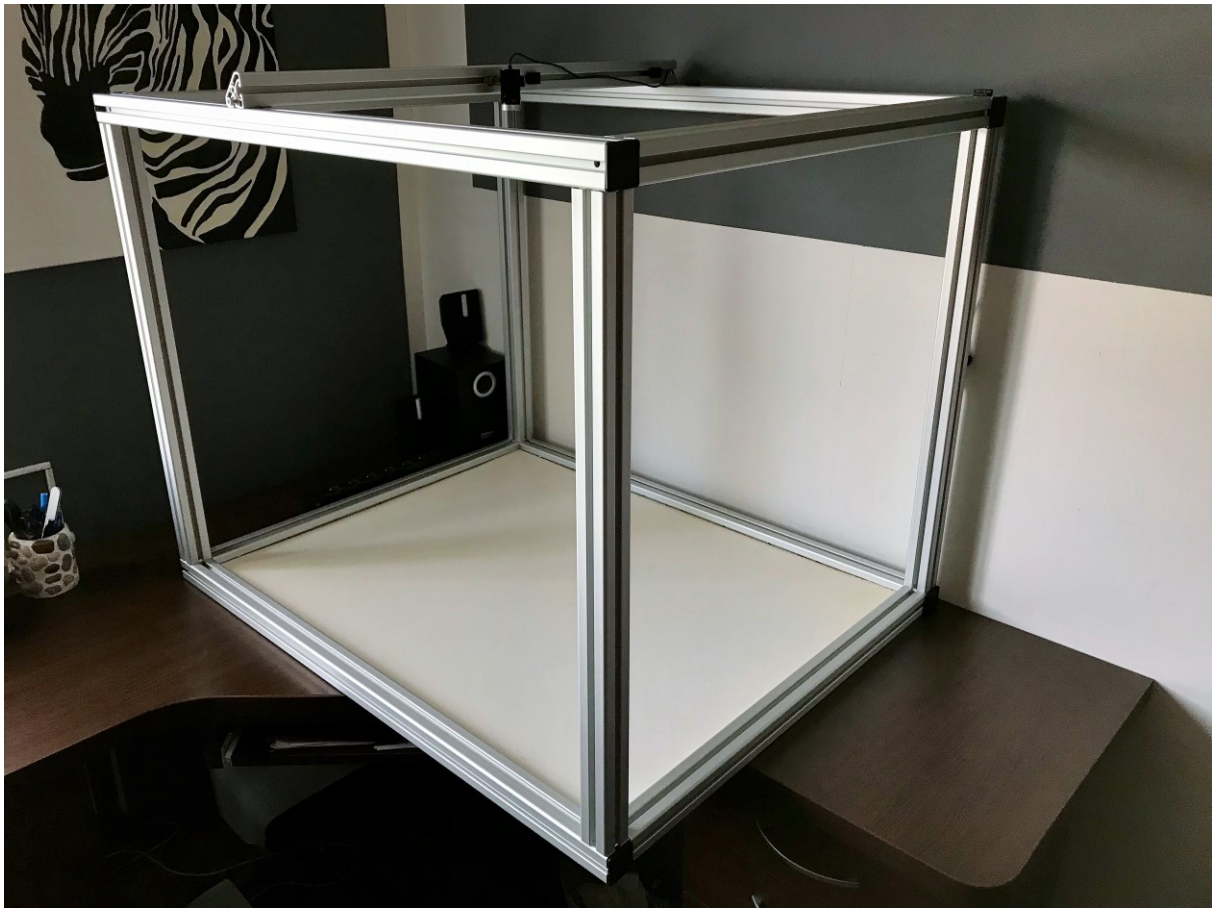


a) b) c)

Obr. 8-20 Stavebnicový systém hliníkových profilů

a) Šroubový spoj Standard-Fastening Set 8, b) Hliníková úhlová lišta Rebate Profile 8, c) Těsnění pro výplně

Rám horní podstavy je rozměrově stejný jako u dolní podstavy. Navíc zde přibyla i příčka rozdělující delší stranu, pro uchycení kamery. Součástí návrhu horní podstavy je osvětlení vytvořené z extrudované desky, jenž se používá pro světelné ultratenké reklamní panely. Ta je nasvícena hranou pomocí led pásku, který je přilepen v drážce profilu. Takovéto světlo se rozvádí po celé ploše panelu a rovnoměrně nasvěcuje scénu. Z principu využití extrudované desky, není umožněno do této desky vrtat jakékoliv díry při jejím okraji. Proto je deska vložena do drážek profilů a pomocí těsnění pro výplně uchycena. V jejím samotném středu je vyvrtán otvor pro objektiv kamery. Výška stojných noh byla zvolena na základě parametrů webové kamery a plochy, jenž tato kamera musí snímat.



Obr. 8-21 Skutečný stav konstrukce

9. TESTOVÁNÍ SYSTÉMU

Tato kapitola se zabývá testováním vytvořeného systému. Realizovaný algoritmus byl testován na správnost detekce a rozpoznání jednotlivých herních kamenů, společně s měřením výpočetního času algoritmu. Pro testování byl využit notebook s procesorem Intel Core i5 s taktovací frekvencí 2,40GHz a pracovní pamětí 8GB. V následujících podkapitolách jsou vysvětleny principy testování a výsledky jednotlivých testů.

9.1 Detekce a rozpoznání herních kamenů

Testování bylo prováděno vložením celé sady 36 herních kamenů, do snímané scény s následným vykreslením herních kamenů do matice `PictureBoxů` v grafickém uživatelském rozhraní. Testování probíhalo za dvou odlišných světelných podmínek, a to při přirozeném denním světle a při umělém osvětlení. Při testování byly pro obě tyto světelné podmínky použity stejné prahy jednotlivých barev pro segmentaci herních kamenů. Toto testování probíhalo dvacetkrát pro každou ze světelných podmínek a byly zaznamenány hodnoty nesprávného rozpoznání herních kamenů. Z výsledných hodnot byla vypočtena průměrná úspěšnost systémů. Kompletní tabulka výsledků testování je součástí tištěné přílohy číslo I a byla shrnuta do následující tabulky Tab. 1.

Tab. 1 Výsledky testování algoritmu

Typ osvětlení	Přirozené	Umělé
Chybná označení	38/720	68/720
Správná označení	682/720	652/720
Procentuální úspěšnost	94,72%	90,56%
Průměrná úspěšnost	**92,64%**	

Při přirozeném denním světle bylo z celkového počtu 720 herních kamenů, špatně rozpoznáno 38 herních kamenů. Procentuální úspěšnost algoritmu při přirozeném denním světle je 94,72 %. Při umělém osvětlení a zachování nastavení prahů pro segmentaci jednotlivých barev bylo špatně rozpoznáno 68 herních kamenů, procentuální úspěšnost při tomto osvětlení je 90,56%.

9.2 Výpočetní čas systému

Od samého počátku bylo patrné, že výpočetní čas algoritmu je z velké části odvozen od počtu detekovaných herních kamenů hry. Proto bylo měření prováděno postupným přidáváním herních kamenů do snímané scény. Takto byly měřeny data třikrát pro celou sadu 36 herních kamenů. Pro měření časové náročnosti algoritmu byla využita knihovna definovaná hlavičkou `<chrono>`. Čas algoritmu je měřen ve třech jeho částech. První měřená část je definovaná počátkem detekce tahu hry, až po rozpoznání všech herních kamenů aktuálně nasnímaného snímku. Druhou měřenou částí algoritmu je herní analýza, která zahrnuje vytvoření herního pole, porovnání herních polí aktuálního a minulého kola hry, kontrola pravidel hry a bodové hodnocení. Třetí měřenou částí je vykreslení herního pole do matice `PictureBoxů` v grafickém uživatelském rozhraní. Počátek měření času je definován v úseku kódu po detekci ustálení herní scény a jednotlivé úseky kódu ohraničeny příkazem:

```
auto start = std::chrono::steady_clock::now();
```

Rozdílem získaných časů jsou vypočteny časy jednotlivých úseků.

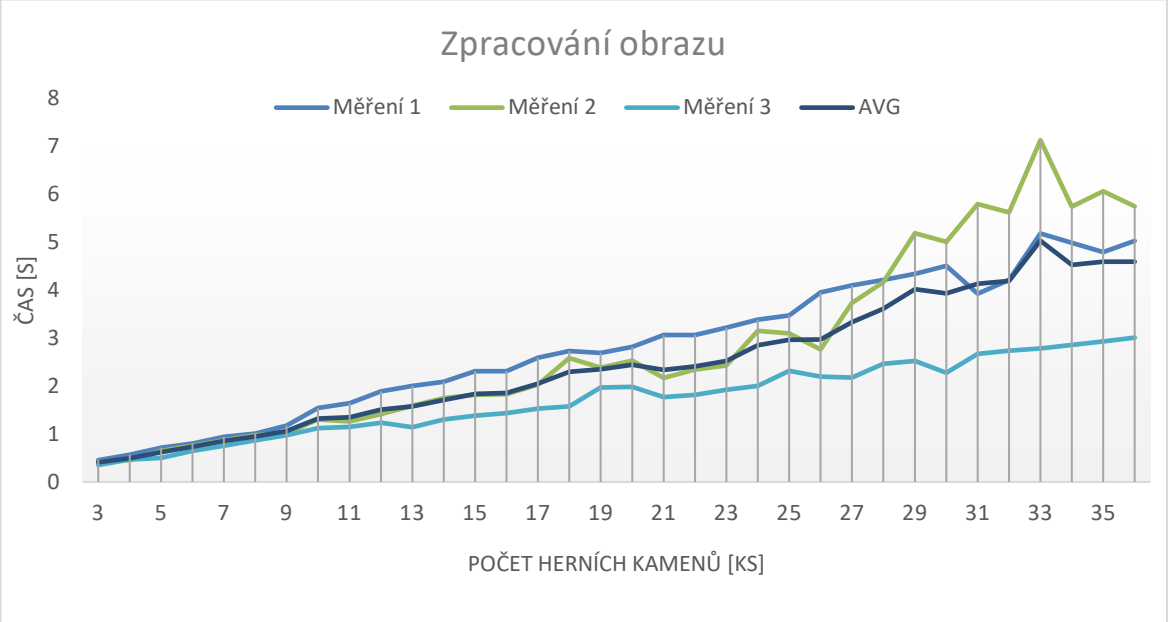
```
double elapsed_time = double(duration_cast <seconds>(end - start).count());
```

Z naměřených dat byly vytvořeny čtyři spojnicové grafy závislostí času zpracování na počtu detekovaných kamenů tří úseků kódu. Z naměřených dat byly vypočteny průměrné časy pro zpracování jednoho herního kamene v jednotlivých úsecích kódu. Tyto výsledky reprezentuje tabulka Tab. 2.

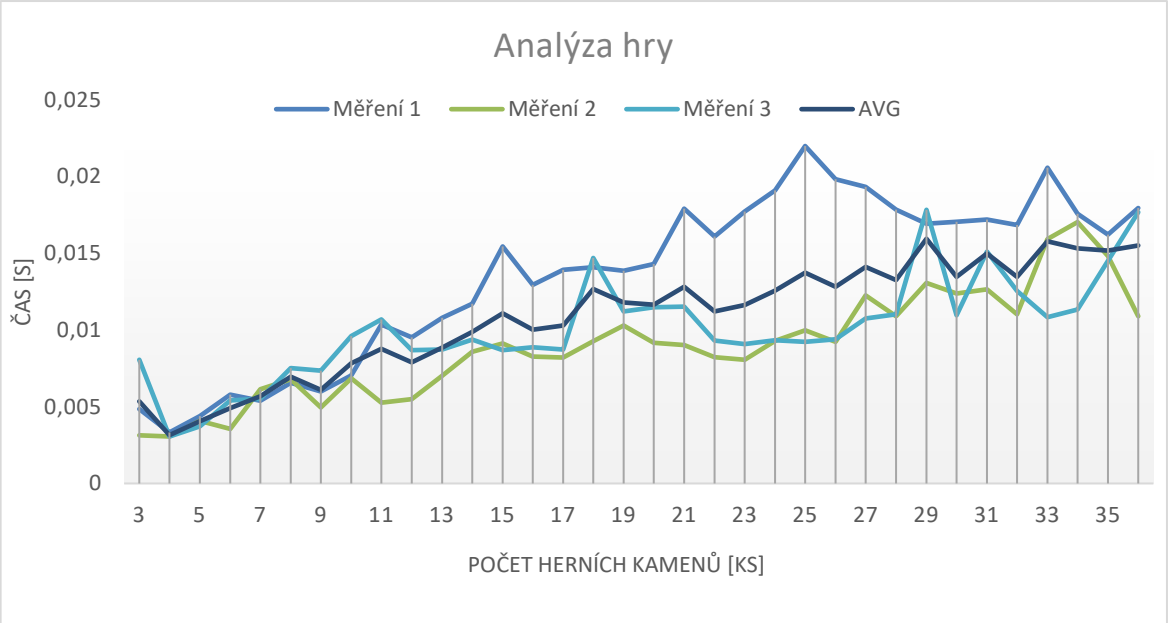
Tab. 2 Výsledné průměrné časy algoritmu

$\bar{t}_{ZO}$ [s]	0,1257
$\bar{t}_{HA}$ [s]	0,0006
$\bar{t}_{V}$ [s]	0,0595

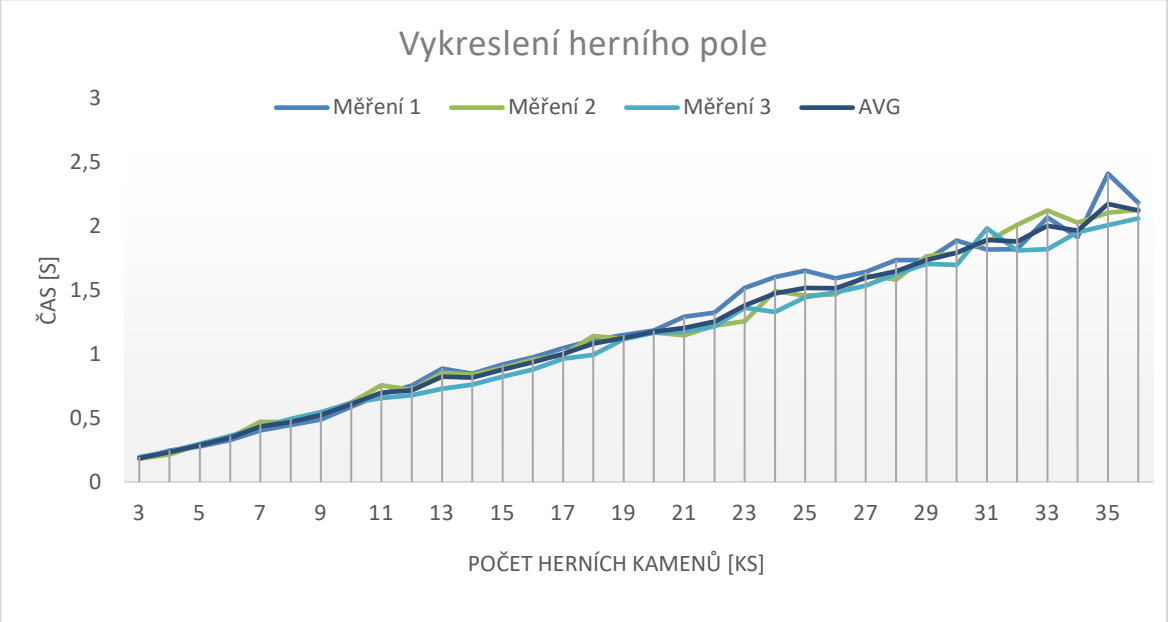
Z naměřených dat reprezentovanými grafy lze stanovit, že výpočetní čas algoritmu je lineárně závislý na počtu detekovaných herních kamenů v obraze ve všech měřených úsecích kódu. Průměrný čas úseku zpracování obrazu při detekci jednoho herního kamene je 0,1257 sekund, což představuje 67,7% celkového výpočetního času systému. Průměrný čas vykreslení herního pole při detekci jednoho herního kamene je 0,0595 sekund, což představuje 32% celkového výpočetního času systému. Průměrný čas herní analýzy je oproti dvěma předchozím časům zanedbatelný, jelikož představuje 0,3% celkového výpočetního času při detekci jednoho herního kamene.



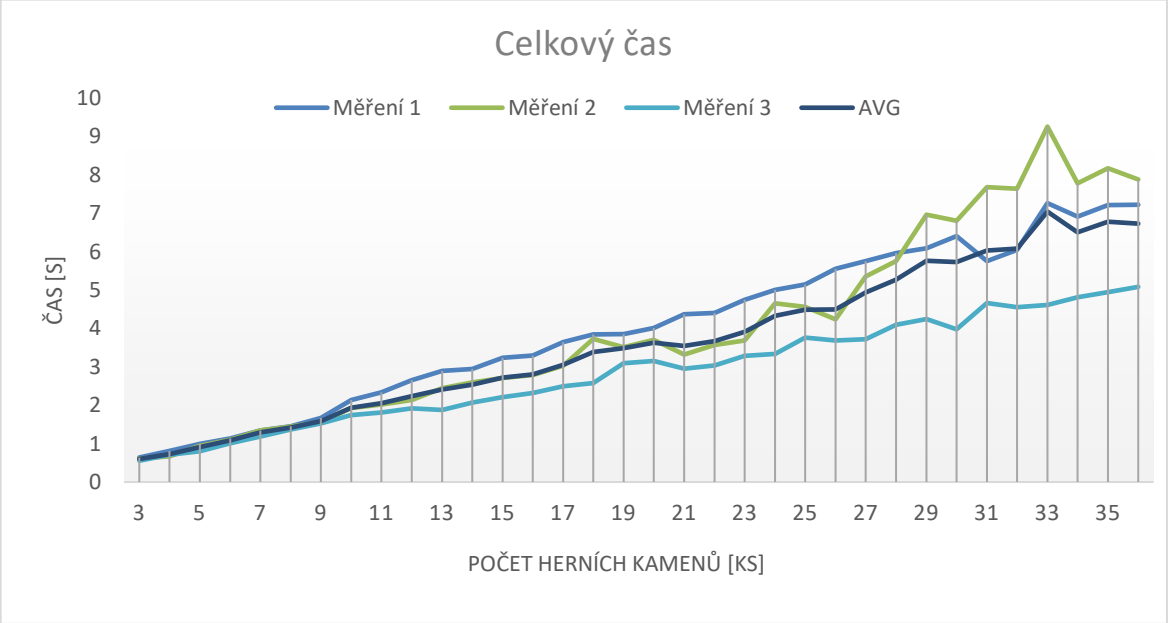
Obr. 9-1 Graf závislosti času zpracování obrazu na počtu detekovaných herních kamenů



Obr. 9-2 Graf závislosti času analýzy hry na počtu detekovaných herních kamenů



Obr. 9-3 Graf závislosti času vykreslení herního pole na počtu detekovaných herních kamenů



Obr. 9-4 Graf závislosti celkového času algoritmu na počtu detekovaných oblastí

10. ZÁVĚR

Úkolem této diplomové práce bylo navrhnout a realizovat systém pro rozpoznání předmětů na základě obrazového signálu, provést měření a analýzu s vyhodnocením skutečných parametrů systému. Toto zadání bylo splněno vytvořením aplikace vizuálního rozhodčího pro stolní hru Qwirkle, která má sloužit jako prezentační nástroj pro účely fakulty. Pro realizaci této aplikace byli specifikované další dva požadavky, naprogramování aplikace v jazyce C++, bez možnosti využít některé z dostupných knihoven pro zpracování obrazu.

Z důvodu časové náročnosti vytvoření vlastní knihovny pro zpracování obrazu, bylo při návrhu a realizaci aplikace vizuálního rozhodčího využito v principu jednoduchých metod pro detekci a následné rozpoznání objektů. Všechny využité metody vycházejí ze znalosti barev, tvarů a velikosti herních kamenů. Detekce herních kamenů byla realizována na základě segmentace vstupního obrazu prahováním podle barev obrazců vyskytujících se na herních kamenech, s následným popisem jednotlivých oblastí jejími radiometrickými parametry a jejich klasifikací podle velikosti, zda se jedná o herní kámen, či nikoliv. Pro rozpoznání herního kamene je využita metoda založená na porovnávání se vzory vytvořených symbolů, pomocí korelace. Pro zcela plně automatický systém, byl algoritmus doplněn o detekci provedení tahu hry, realizovanou rozdílovou metodou sekvencí snímků a o vyšší logiku zajišťující analýzu samotné hry. Systém byl také doplněn o přívětivé grafické uživatelské rozhraní pro vizualizaci stavu hry a o konstrukci z hliníkových profilů.

U systému byl testován výpočetní čas algoritmu. Pro porovnání herních polí předchozího tahu hry s aktuálním, musí v každém tahu hry docházet k detekci a rozpoznání všech herních kamenů v obraze. To sice zajišťuje invariantnost vůči otočení nebo posunu kamery během hry, na druhou stranu to prodlužuje výpočetní čas algoritmu. Z naměřených výsledků lze vidět, že algoritmus je lineárně závislý na počtu detekovaných herních kamenů v obraze a každým následujícím tahem hry roste jeho výpočetní čas. Průměrný čas detekce a rozpoznání s následným vykreslením jednoho herního kamene je 0,186 sekund. Výpočetní čas algoritmu při následujícím tahu hry by se dal zkrátit detekcí pouze nových herních kamenů v obraze, například průnikem obrazů reprezentující aktuální a předchozí tah hry. U tohoto řešení by však muselo být zajištěno, aby během hry nedocházelo k žádné manipulaci se snímacím zařízením, popřípadě musí být zajištěno, aby nedocházelo k natočení pomyslné mřížky, do které jsou herní kameny skládány.

Systém byl také testován na úspěšnost detekce a rozpoznání herních kamenů. Testování probíhalo za dvou odlišných světelných podmínek, při zachování prahů pro segmentaci herních kamenů. Procentuální úspěšnost vytvořeného systému při přirozeném osvětlení je 94,72%, při umělém osvětlení úspěšnost klesla na 90,56%. Tento pokles představuje zachování prahů jednotlivých barev, které se projevilo ztrátou detailů a špatným rozpoznáním u nekompaktních tvarů obrazců herních kamenů. Pro tyto případy je v grafickém uživatelském rozhraní umožněna korekce prahů jednotlivých barev. Tento problém by se při zachování metod zpracování obrazu dal vyřešit automatickým určováním prahu barev v závislosti na intenzitě okolního osvětlení, popřípadě vytvořením módu pro umělé osvětlení, při kterém

by docházelo ke změně prahu barev. Proto by systém šel rozšířit právě o některou z těchto možností. Dalším rozšířením systému by mohlo být vytvoření aplikace pro tablet, která je mnohem uživatelsky přívětivější a z hlediska velikosti lepším řešením, než deskopová aplikace.

LITERATURA

[1] HLAVÁČ, Václav a Miloš SEDLÁČEK. *Zpracování signálů a obrazů*. Vyd. 2., přeprac. Praha: Vydavatelství ČVUT, 2005. ISBN 80-010-3110-1.

[2] PRATT, William K. Digital image processing: PIKS Scientific inside. 4th ed., Newly updated and rev. ed. Hoboken, N.J.: Wiley-Interscience, c2007. ISBN 978-0-471-76777-0.

[3] ZHOU, Huiyu, Jiahua WU a Jianguo ZHANG. *Digital Image Processing: Part II* [online]. Ventus Publishing ApS, 2010 [cit. 2017-12-06]. ISBN 978-87-7681-542-4. Dostupné z: http://zums.ac.ir/files/research/site/ebooks/electrical-electronic-engineering/digital-image-processing-part-two.pdf

[4] ŠPANĚL, Michal a Vítězslav BERAN. *Obrazové segmentační techniky: Přehled existujících metod* [online]. 2006 [cit. 2017-12-12]. Dostupné z: http://www.fit.vutbr.cz/~spanel/segmentace/

[5] SOJKA, Eduard. *Digitální zpracování a analýza obrazů*. Ostrava: VŠB-Technická univerzita, 2000. ISBN 80-707-8746-5.

[6] PHILLIPS, Dwayne. *Image processing in C*. Second Edition. Lawrence, Kansas: R & D Publications, 1994. ISBN 0-13-104548-2.

[7] BALLARD, Dana Harry. a Christopher M. BROWN. *Computer vision*. II. Title. . Englewood Cliffs, N.J.: Prentice-Hall, c1982. ISBN 01-316-5316-4.

[8] IKONOMATAKIS, N., K.N. PLATANIOTIS, M. ZERVAKIS a A.N. VENETSANOPOULOS. Region growing and region merging image segmentation. In: *Proceedings of 13th International Conference on Digital Signal Processing* [online]. IEEE, 1997, s. 299-302 [cit. 2017-12-18]. DOI: 10.1109/ICDSP.1997.628077. ISBN 0-7803-4137-6. Dostupné z: http://ieeexplore.ieee.org/document/628077/

[9] AUTOR NEUVEDEN. *Global Constraint Catalog* [online]. [cit. 19.12.2017]. Dostupný na WWW: http://web.emn.fr/x-info/sdemasse/ctrs/quadtree.png

[10] GONZALEZ, Rafael C., Richard E. WOODS a Steven L. EDDINS. *Digital Image processing using MATLAB*. 2nd ed. S.I.: Gatesmark Pub., c2009. ISBN 978-0-9820854-0-0.

[11] MAŠEK, Jan. *Detekce objektů v obraze s pomocí Haarových příznaků*. Brno, 2012. Diplomová práce. VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ.

[12] VIOLA, P. a M. JONES. Rapid object detection using a boosted cascade of simple features. In: *Proceedings of the 2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition. CVPR 2001*. IEEE Comput. Soc, 2001, I-511-I-518. DOI: 10.1109/CVPR.2001.990517. ISBN 0-7695-1272-0. ISSN 1063-6919. Dostupné také z: http://ieeexplore.ieee.org/document/990517/

[13] DALAL, N. a B. TRIGGS. Histograms of Oriented Gradients for Human Detection. In: *2005 IEEE Computer Society Conference on Computer Vision and Pattern*

Recognition (CVPR'05) [online]. IEEE, 2005, s. 886-893 [cit. 2018-04-16]. DOI: 10.1109/CVPR.2005.177. ISBN 0-7695-2372-2. ISSN 1063-6919. Dostupné z: http://ieeexplore.ieee.org/document/1467360/

[14] HORÁK, Karel, Ilona KALOVÁ, Petr PETYOVSKÝ a Miloslav RICHTER. *Počítačové vidění*. Brno, 2008.

[15] Color Models. *Intel® Developer Zone* [online]. [cit. 2018-04-16]. Dostupné z: https://software.intel.com/en-us/node/503873

[16] HRÚZ, Marek. *LBP, HoG* [online]. 2015, , 9 [cit. 2018-04-27]. Dostupné z: http://www.kky.zcu.cz/uploads/courses/mpv/05/materialy05.pdf

[17] *Barvy.jpg*. [online]. [cit. 2018-04-27]. Dostupné z: http://fyzweb.cz/clanky/img/00107/barvy.jpg

[18] KOTEK, Zdeněk. *Metody rozpoznávání a jejich aplikace*. Praha: Academia, 1993. ISBN 80-200-0297-9

SEZNAM PŘÍLOH

Tištěné přílohy

I. Tabulka výsledků testování správnosti detekce a rozpoznání herních kamenů
II. Tabulka naměřených hodnot časů – Měření 1
III. Tabulka naměřených hodnot časů – Měření 2
IV. Tabulka naměřených hodnot časů – Měření 3
V. Vývojový diagram výpočtu bodů ve vertikální řadě

Elektronické přílohy – obsah CD

Software – Zdrojové kódy aplikace vizuálního rozhodčí

I. Tabulka výsledků testování správnosti detekce a rozpoznání herních kamenů

Č. testu	Tvar						Celkem
	Circle	Square	Diamond	Criss-Cross	Starburst	Clover	
1	0	0	0	0	0	3	3
2	0	0	0	0	2	2	4
3	0	0	0	1	3	2	6
4	0	1	0	0	3	1	5
5	0	0	0	0	3	0	3
6	0	2	0	0	2	0	4
7	0	2	0	0	2	1	5
8	0	2	0	0	1	1	4
9	0	0	0	0	1	0	1
10	0	0	0	0	1	0	1
11	0	3	0	0	0	1	4
12	0	0	0	0	0	2	2
13	0	0	0	0	0	3	3
14	0	1	0	0	2	0	3
15	0	2	0	0	2	1	5
16	0	0	0	0	2	0	2
17	0	0	0	0	0	3	3
18	0	0	0	0	3	0	3
19	0	1	0	0	2	0	3
20	0	0	0	0	1	3	4
1	0	2	0	0	0	1	3
2	0	0	0	0	0	0	0
3	0	0	0	0	1	0	1
4	0	0	0	0	0	0	0
5	0	1	0	0	1	1	3
6	0	2	0	0	0	1	3
7	0	1	0	0	1	1	3
8	0	1	0	0	0	1	2
9	0	2	0	0	0	1	3
10	0	0	0	0	0	0	0
11	0	2	0	0	0	1	3
12	0	1	0	0	0	1	2
13	0	2	0	0	0	1	3
14	0	1	0	0	0	0	1
15	0	0	0	0	0	1	1
16	0	2	0	0	0	2	4
17	0	0	0	0	0	0	0
18	0	2	0	0	0	0	2
19	0	0	0	0	0	0	0
20	0	2	0	0	0	2	4

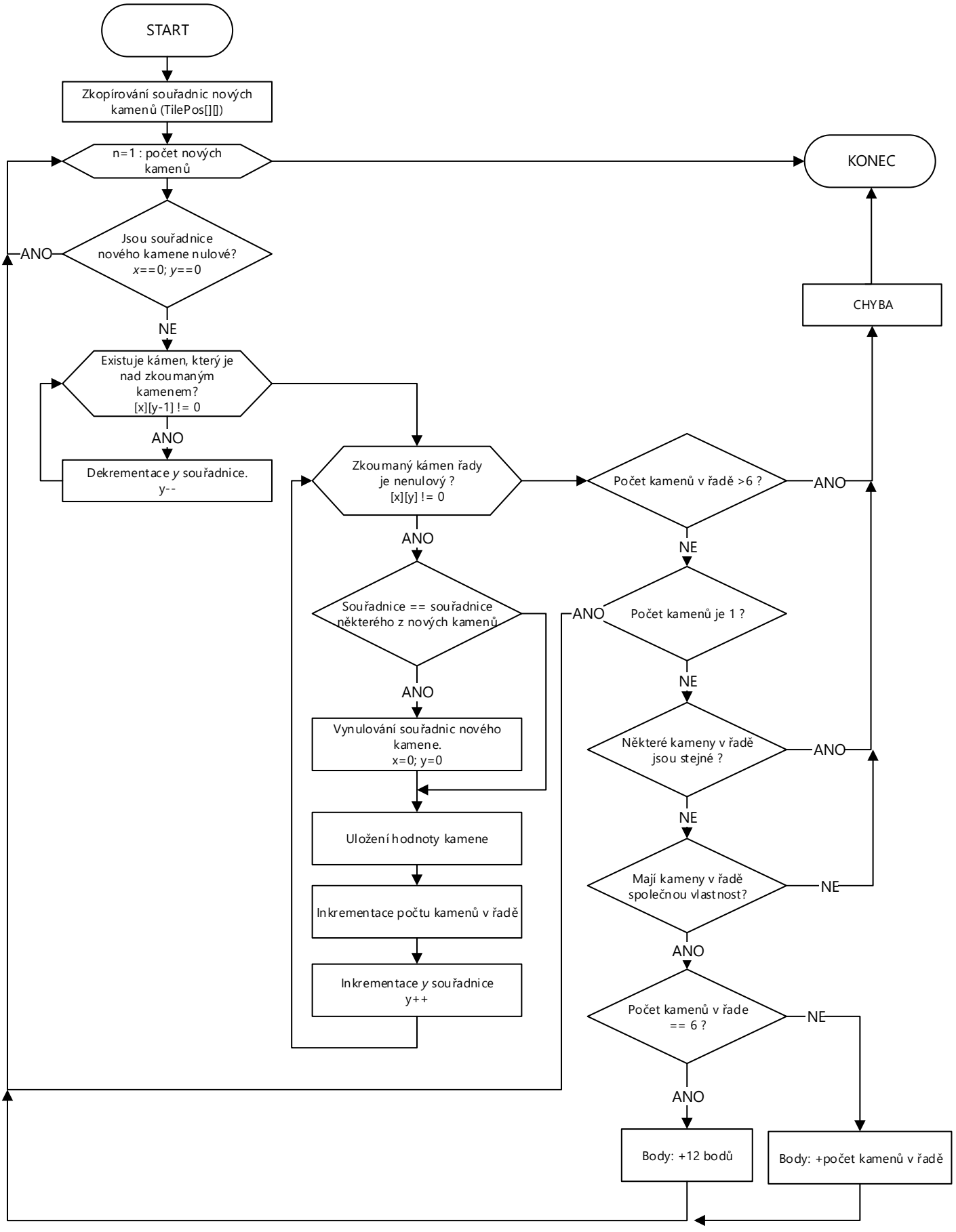
II. Tabulka naměřených hodnot časů – Měření 1

Počet herních kamenů [ks]	Čas zpracování obrazu [s]	Čas analýzy hry [s]	Čas vykreslení herního pole [s]	Celkový čas [s]
3	0,453467692	0,004836718	0,180460103	0,638764513
4	0,56551118	0,003325538	0,24498359	0,813820308
5	0,712764513	0,004380102	0,278177641	0,995322257
6	0,802569641	0,005774564	0,327550359	1,135894564
7	0,937316513	0,005389949	0,403619898	1,346326359
8	1,004667488	0,006548512	0,444406359	1,455622359
9	1,176283282	0,005996513	0,488268718	1,670548513
10	1,539242667	0,007028103	0,586580103	2,132850872
11	1,64142318	0,010348718	0,684811282	2,33658318
12	1,887217641	0,009507077	0,751955282	2,64868
13	2,000877539	0,010795077	0,885056821	2,896729436
14	2,086568821	0,011698462	0,844277949	2,942545231
15	2,306116308	0,015434052	0,918280205	3,239830564
16	2,306125949	0,01292718	0,972248616	3,291301744
17	2,585389949	0,013912206	1,045555077	3,644857231
18	2,727849231	0,014073641	1,107361231	3,849284103
19	2,687956103	0,013860103	1,148152616	3,849968821
20	2,81430359	0,014284923	1,182732513	4,011321026
21	3,059353436	0,017898257	1,291519385	4,368771077
22	3,064725129	0,016082872	1,322294564	4,403102564
23	3,212307692	0,017706872	1,516938667	4,746953231
24	3,384608	0,019073847	1,601324308	5,005006154
25	3,470434257	0,02197518	1,65199118	5,144400616
26	3,945985641	0,019811897	1,590634462	5,556432
27	4,094098667	0,019310154	1,640699488	5,754108308
28	4,211054769	0,017836718	1,732913436	5,961804923
29	4,337944616	0,016908513	1,734730675	6,089583804
30	4,50257518	0,017041641	1,886794257	6,406411077
31	3,923568205	0,017183179	1,815692513	5,756443898
32	4,208242872	0,016829949	1,818366564	6,043439385
33	5,177477539	0,020557539	2,066524513	7,26455959
34	4,980539487	0,017551795	1,914182359	6,912273641
35	4,790275488	0,016212923	2,409114872	7,215603282
36	5,020416411	0,017932923	2,180916923	7,219266257

III. Tabulka naměřených hodnot časů – Měření 2

Počet herních kamenů [ks]	Čas zpracování obrazu [s]	Čas analýzy hry [s]	Čas vykreslení herního pole [s]	Celkový čas [s]
3	0,425828103	0,003130051	0,181931692	0,610889846
4	0,450775385	0,003059692	0,215460924	0,669296
5	0,648477744	0,004070154	0,290097436	0,942645334
6	0,752050257	0,00354441	0,347932103	1,10352677
7	0,866936616	0,006140718	0,469586257	1,34266359
8	0,970820923	0,006779487	0,466298462	1,443898872
9	1,019798974	0,004955693	0,538749539	1,563504205
10	1,300451283	0,006852307	0,619443898	1,926747488
11	1,258415385	0,005263179	0,755165334	2,018843898
12	1,41119241	0,005490872	0,718089436	2,134772718
13	1,58860041	0,007013539	0,852254359	2,447868308
14	1,746105847	0,008574564	0,83797518	2,59265559
15	1,813112821	0,00911118	0,890852718	2,713076718
16	1,825350975	0,008255179	0,955073846	2,78868
17	2,025249846	0,008195282	0,992043898	3,025489026
18	2,578216	0,009268103	1,139774564	3,727258667
19	2,380766564	0,010276308	1,116964923	3,508007795
20	2,527850462	0,009168	1,168225641	3,705244103
21	2,167209847	0,009004308	1,145560616	3,32177477
22	2,342054359	0,008230974	1,221368821	3,571654154
23	2,425163077	0,008063794	1,255153231	3,688380103
24	3,150535385	0,009277334	1,490921436	4,650734154
25	3,094524513	0,009968	1,456117949	4,560610462
26	2,762633231	0,009225846	1,466244308	4,238103385
27	3,725833436	0,012250667	1,612770257	5,350854359
28	4,162178257	0,010880616	1,581625641	5,754684513
29	5,18415118	0,013065641	1,762606975	6,959823795
30	5,003041847	0,012378871	1,787715488	6,803136206
31	5,789896205	0,012645333	1,878294565	7,680836103
32	5,616758154	0,011021744	2,008257641	7,636037539
33	7,120960206	0,015913435	2,121026667	9,257900308
34	5,736246154	0,017021949	2,027316718	7,780584821
35	6,054632205	0,014811488	2,102791385	8,172235077
36	5,74510359	0,010896	2,126738462	7,882738051

IV. Tabulka naměřených hodnot časů – Měření 3

Počet herních kamenů [ks]	Čas zpracování obrazu [s]	Čas analýzy hry [s]	Čas vykreslení herního pole [s]	Celkový čas [s]
3	0,351539487	0,008057436	0,194713026	0,554309949
4	0,467797334	0,003057641	0,238048	0,708902975
5	0,497965539	0,003719589	0,297523283	0,799208411
6	0,647681846	0,005414154	0,359122052	1,012218052
7	0,754080821	0,005511795	0,425593231	1,185185846
8	0,866207795	0,007511589	0,492188103	1,365907487
9	0,976269744	0,007345847	0,544194256	1,527809846
10	1,119490667	0,009591794	0,617159385	1,746241847
11	1,144046154	0,010688206	0,656553641	1,811288
12	1,231041436	0,008683488	0,677741744	1,917466667
13	1,14269641	0,008726769	0,728338051	1,879761231
14	1,298675282	0,009361436	0,76134359	2,069380308
15	1,377770872	0,008677744	0,822176821	2,208625436
16	1,432734974	0,008877949	0,877546051	2,319158974
17	1,525877539	0,008732308	0,962754872	2,497364718
18	1,571006359	0,014674666	0,994314872	2,579995897
19	1,965854154	0,011211898	1,115125744	3,092191795
20	1,979123487	0,011484718	1,165858051	3,156466256
21	1,767739077	0,011518769	1,175490667	2,954748513
22	1,811790564	0,009297846	1,21460677	3,03569518
23	1,920140718	0,009077333	1,359692923	3,288910975
24	2,002934359	0,00932759	1,327905436	3,340167385
25	2,311684103	0,009230359	1,442563898	3,763478359
26	2,19372677	0,009387897	1,479673231	3,682787898
27	2,172569231	0,010738051	1,533782359	3,717089641
28	2,460448206	0,011006974	1,624900718	4,096355898
29	2,523170872	0,017820718	1,703980308	4,244971898
30	2,274450667	0,010960205	1,695852103	3,981262974
31	2,669060103	0,015103385	1,980049847	4,664213334
32	2,73684	0,012533744	1,807741128	4,557114872
33	2,780657846	0,010828513	1,81809477	4,609581129
34	2,851834406	0,01132788	1,951001063	4,814163349
35	2,929317862	0,014497265	2,005169194	4,948984321
36	3,006801318	0,01766665	2,059337325	5,083805293

V. Vývojový diagram výpočtu bodů ve vertikální řadě