



ČESKÉ VYSOKÉ UČENÍ TECHNICKÉ V PRAZE  
Fakulta jaderná a fyzikálně inženýrská



# Rozpoznávání objektů pro interakci člověk-robot

## Object recognition for human-robot interaction

Bakalářská práce

Autor: **Soňa Drocárová**  
Vedoucí práce: **Ing. Jitka Kostková, Ph.D.**  
Konzultant:  
Akademický rok: 2020/2021

## ZADÁNÍ BAKALÁŘSKÉ PRÁCE

Student:	Soňa Drocárová
Studijní program:	Aplikace přírodních věd
Studijní obor:	Matematické inženýrství
Studijní zaměření:	Aplikované matematicko-stochastické metody
Název práce (česky):	Rozpoznávání objektů pro interakci člověk-robot
Název práce (anglicky):	Object recognition for human-robot interaction

### Pokyny pro vypracování:

- 1) Proveďte základní rešerši metod pro popis jednoduchých 2D objektů.
- 2) Zvolte 1-2 metody vhodné pro plně viditelné objekty s neznámou orientací a velikostí.
- 3) Zvolené metody implementujte v jazyce MATLAB.
- 4) Vyzkoušejte jejich účinnost na kartách a objektech hry Dobble, studujte vliv parametrů použitých metod.
- 5) V této úloze vyřešte detekci a segmentaci objektů.

Doporučená literatura:

- 1) J. Flusser, T. Suk, B. Zitová, 2D and 3D image analysis by moments. John Wiley & Sons, 2016.
- 2) R. C. Gonzalez, R. E. Woods, Digital Image Processing (4th ed.). Pearson, 2018.
- 3) W. K. Pratt, Digital Image Processing (3rd ed.). John Wiley, New York, 2001.

Jméno a pracoviště vedoucího bakalářské práce:

Ing. Jitka Kostková

Ústav teorie informace a automatizace AV ČR, v.v.i., oddělení Zpracování obrazové informace,  
Pod Vodárenskou věží 4, 182 08 Praha 8

Jméno a pracoviště konzultanta:

Datum zadání bakalářské práce: 31.10.2020

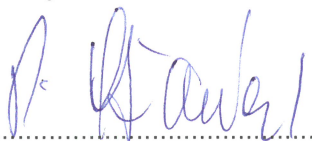
Datum odevzdání bakalářské práce: 7.7.2021

Doba platnosti zadání je dva roky od data zadání.

V Praze dne 30.10.2020



.....  
garant oboru



.....  
vedoucí katedry



  
.....  
děkan

*Pod'akovanie:*

Chcela by som sa poďakovať mojej školiteľke Ing. Jitke Kostkovej, Ph.D. za jej trpezlivosť, ochotu, ústretovosť, cenné rady a podnetné usmernenia pri písaní tejto bakalárskej práce.

*Čestné prehlásenie:*

Prehlasujem, že som túto prácu vypracovala samostatne a uviedla som všetku použitú literatúru.

V Prahe dňa 7. júla 2021

Soňa Drocárová

*Název práce:*

**Rozpoznávání objektů pro interakci člověk-robot**

*Autor:* Soňa Drocárová

*Obor:* Matematické inženýrství

*Zaměření:* Aplikované matematicko-stochastické metody

*Druh práce:* Bakalářská práce

*Vedoucí práce:* Ing. Jitka Kostková, Ph.D., Ústav teorie informace a automatizace AV ČR, v.v.i. oddělení Zpracování obrazové informace

*Konzultant:*

*Abstrakt:* Práce se zabývá rozpoznáváním objektů v obraze pomocí příznaků. Cílem bylo zvolit vhodnou metodu pro rozpoznávání symbolů karet společenské hry Dobble a implementovat ji v prostředí MATLAB. Na dosažení tohoto cíle bylo potřebné provést segmentaci a detekci objektů. Následně se testovali vlastnosti zvolené metody a její úspěšnost při hledání společného symbolu dvou různých karet. Testování potvrdilo účinnost použitého postupu pro danou úlohu.

*Klíčová slova:* analýza obrazu, Dobble, Fourierové deskriptory, invarianty, prahování, příznaky, segmentace

*Title:*

**Object recognition for human-robot interaction**

*Author:* Soňa Drocárová

*Abstract:* This thesis deals with object recognition in images using features. The objective was to find a suitable method for the recognition of symbols on cards from a game called Dobble and implementing this method in MATLAB. Achieving this goal required performing segmentation and detection of said objects. Properties of this method and its degree of success in object recognition were tested. Testing confirmed that the chosen approach was effective for the given task.

*Key words:* Dobble, features, Fourier descriptors, image analysis, invariants, segmentation, thresholding

# Obsah

Úvod	7
<b>1 Základné definície</b>	<b>9</b>
<b>2 Segmentácia</b>	<b>11</b>
2.1 Segmentácia zameraná na hranice . . . . .	11
2.2 Segmentácia orientovaná na oblasti . . . . .	12
<b>3 Príznamy</b>	<b>14</b>
3.1 Vizuálne príznaky . . . . .	15
3.2 Úplné vizuálne príznaky . . . . .	17
3.3 Transformačné koeficienty . . . . .	19
3.4 Momenty a momentové invarianty . . . . .	22
<b>4 Dobble</b>	<b>24</b>
<b>5 Praktická časť</b>	<b>26</b>
5.1 Segmentácia a detekcia objektov . . . . .	26
5.2 Výpočet príznakov . . . . .	28
5.3 Invariantnosť deskriptorov . . . . .	28
5.4 Invariantnosť a rozlišovacia schopnosť deskriptorov . . . . .	29
5.5 Citlivosť na šum . . . . .	30
5.6 Úspešnosť rozpoznávania v závislosti na počte deskriptorov . . . . .	31
5.7 Hľadanie spoločného objektu dvoch kariet . . . . .	32
5.8 Diskusia . . . . .	34
<b>Záver</b>	<b>35</b>

# Úvod

Vnímanie vizuálnych informácií je pre človeka prirodzenou súčasťou každodenného života. Zrak tvorí veľkú časť našich zmyslových vnemov. Aj z obyčajného obrázku vieme nadobudnúť obrovské množstvo informácií. Dôsledkom toho sa stali dôležitým prostriedkom medziludskej komunikácie a rovnako aj komunikácie ľudí so strojmi. Analýza obrazu v počítačoch je však výrazne komplikovanejší proces, ako u človeka. Tieto komplikácie sú spôsobené viacerými faktormi ako napríklad fakt, že človek si dokáže v určitých situáciách domyslieť chýbajúce informácie, v čom mu napomáhajú aj ostatné zmysly, dokáže vnímať kontext danej situácie a vie veľmi efektívne využívať znalosti, ktoré v priebehu života nadobudol. Ľudský zrak je rovnako schopný rozoznať daný objekt v rôznych polohách, orientáciách a pri rôznych veľkostiach. Na docelenie toho, aby počítač takouto schopnosťou disponoval sa vyžaduje zvolenie vhodnej metódy rozpoznávania nezávislej na daných transformáciách [3] čím sa bude práca zaoberať.

Spomínané skutočnosti ale aj mnoho ďalších robia analýzu obrazu pre počítač náročnou úlohou, ktorá je však dôležitá v praxi v mnohých oblastiach. Využitie nachádza pri detekcii a identifikácii tváre v obraze, ktorá sa dnes bežne objavuje napríklad v smartfónoch, alebo pri rozpoznávaní dopravných značiek v autách. Rovnako nachádza svoje uplatnenie aj v medicíne, kde sa používa napríklad na identifikáciu podozrivých tkanív v medicínskych snímkach.

Proces analýzy obrazu sa obvykle skladá z piatich krokov: získavanie obrazu, predspracovanie, extrakcia a výpočet príznakov, detekcia a klasifikácia.



Obr. 1: Postup analýzy obrazu

Vo fáze *získavania obrazu* zohráva významnú rolu voľba vzorkovacej frekvencie a počet úrovní kvantizácie. Pri voľbe je dôležité zachovanie čo najvyššej možnej kvality pri udržiavaní pomernej nízkej veľkosti obrazu. Pre ďalší postup procesu spracovania obrazu bude teda prospešné vylepšiť kvalitu získaného obrazu. Cieľom *predspracovania* je odstránenie, alebo potlačenie nedostatkov vzniknutých pri získavaní obrazu. Medzi tieto patria rôzne degradácie, ktorými sú napríklad zlé zaostrenie, pohyb v obraze alebo šum [3]. V tomto kroku musíme brať do úvahy spôsob, akým budeme obraz ďalej spracovávať, aby bolo jeho vylepšenie žiadúce pre naše špecifické použitie. Do tejto kategórie radíme procesy ako škálovanie, ekvalizácia histogramu, vyhladzovanie hrán, filtrácia obrazu a podobne [5]. Úlohou *detekcie* je lokalizovať objekt záujmu a oddeliť ho od pozadia a od ostatných objektov. Ďalším krokom je *výpočet*

*príznakov*, z ktorých sa v práci zaoberáme hlavne takými, ktoré sú nezávislé na určitých transformáciách a sú podrobne rozobrané v Kapitole 3. *Klasifikácia* je posledným krokom analýzy obrazu a priraduje každý neznámy objekt do určitej triedy na základe hodnôt príznakov [3].

Cieľom práce je spraviť prehľad o invariantných príznakových metódach rozpoznávania objektov v obraze, aplikovať vhodnú metódu v úlohe rozpoznávania symbolov kartovej hry Dobble. Na dosiahnutie tohto cieľa bolo ďalej potrebné previesť segmentáciu a detekciu objektov v obraze. Na zvolenej metóde sa testovali rôzne vlastnosti ako citlivosť na šum, invariantnosť, rozpoznávacía schopnosť a jej účinnosť pri rozpoznávaní.

Následujúca kapitola obsahuje základné pojmy týkajúce sa matematickej reprezentácie digitálneho obrazu. Kapitola 2 sa zaoberá metódami segmentácie, ktorá je pri rozpoznávaní symbolov na kartách nevyhnutná. V Kapitole 3 sa dostávame k hlavnej téme tejto práce - príznakom. Popisuje jednotlivé príznakové metódy, ktoré sú invarianté voči rotácii, posunu a zmene mierky, alebo je u nich možné invariantnosť dosiahnuť. Kapitola 4 predstavuje hru Dobble, z ktorej vznikli obrázky použité na experiment. Posledná Kapitola 5 popisuje experiment, použité metódy analýzy obrazu a samotné výsledky testovania.



# Kapitola 1

## Základné definície

Analýza obrazu v počítačoch vyžaduje svoje matematické nástroje, ale aj ľudskú intuíciu pri voľbe techník pre jeho spracovanie [5]. Táto kapitola sa preto zaoberá základnými pojmami a definíciami, ktoré charakterizujú digitálny obraz.

Matematickú reprezentáciu obrazu uvádza Definícia 1.

**Definícia 1.** Funkciou obrazu rozumieme po častiach spojitú reálnu funkciu  $f$  definovanú na kompaktnom nosiči  $\Omega \subset \mathbb{R}^d$ .

Digitálny obraz vzniká prevedením tejto spojitkej funkcie na diskretnu prostredníctvom vzorkovania a kvantizácie. Vzorkovaním digitalizujeme hodnoty funkcie obrazu v  $x$ -ových a  $y$ -ových súradniciach, kvantizáciou amplitúdy intenzity.

Typická reprezentácia obrazu je formou poľa, ktorého prvky pixely nesú hodnoty intenzity v danom bode. Prostredníctvom vzorkovania a kvantizácie teda dostávame zo spojitkej funkcie obrazu  $f$  maticu o  $M$  riadkoch a  $N$  stĺpcoch, ktorú reprezentuje nasledujúca rovnica [5]:

$$f(x, y) = \begin{pmatrix} f(0, 0) & f(0, 1) & \dots & f(0, N-1) \\ f(1, 0) & f(1, 1) & \dots & f(1, N-1) \\ \vdots & \vdots & \ddots & \vdots \\ f(M-1, 0) & f(M-1, 1) & \dots & f(M-1, N-1) \end{pmatrix}. \quad (1.1)$$

Každý digitálny obraz obsahuje jeden alebo viac farebných kanálov. Tieto kanály udávajú intenzitu farby v danom bode  $(x, y)$ . Medzi základné typy obrázkov patria napríklad: *binárne*, *šedotónové* alebo *farebné* [11], ktoré sú vyobrazené na Obrázku 1.1.

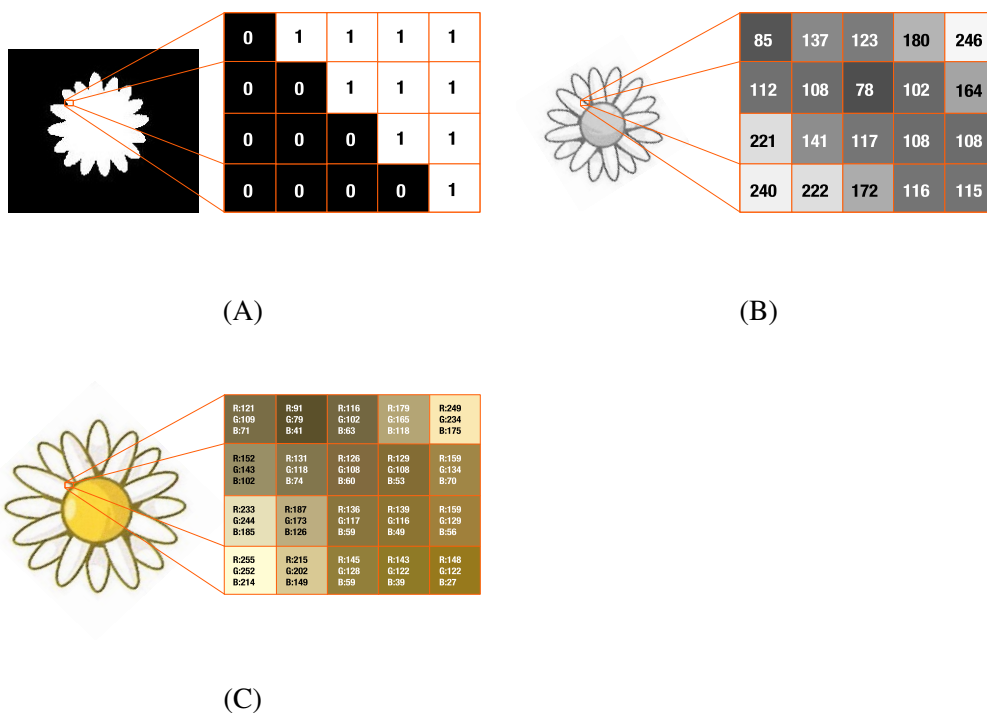
Binárne obrázky tvoria dvojrozmerné polia. Ich prvky nesú logické hodnoty 0 alebo 1. Pre pixel čiernej farby platí  $f(x_c, y_c) = 0$ , pre biely  $f(x_b, y_b) = 1$ .

Šedotónový obraz je rovnako ako binárny definovaný pomocou dvojrozmerného poľa. Hodnoty pixelov v tomto prípade reprezentujú intenzitu. Štandardne nadobúdajú reálne hodnoty v rozmedzí 0 – 1 alebo celočíselné hodnoty v rozmedzí 0 – 255. Najnižšia hodnota predstavuje čiernu farbu, najvyššia bielu.

Najpoužívanejšou reprezentáciou farebného obrazu je farebný priestor RGB. Skladá sa z troch kanálov: červený, zelený a modrý. Tvorí teda trojrozmerné pole, v ktorom každý pixel obsahuje tri hodnoty intenzity. Ďalšie trojrozmerné reprezentácie farebných obrázkov sú napríklad CMY, CMYK alebo HSI.

Medzi typmi obrazov možno prevádzať rôznymi spôsobmi. Napríklad prevod z obrázku vo farebnom priestore RGB na šedotónový dostávame pomocou transformácie:

$$g(x, y) = \alpha f_R(x, y) + \beta f_G(x, y) + \gamma f_B(x, y), \quad (1.2)$$



Obr. 1.1: Hodnoty pixelov pre binárny (A), šedotónnový (B) a farebný (C) obraz

kde  $g$  je vzniknutý šedotónnový obraz a  $f_R$ ,  $f_G$  a  $f_B$  sú jednotlivé farebné kanály pôvodného. Hodnoty váhových koeficientov  $\alpha$ ,  $\beta$ ,  $\gamma$  sú určené na základe toho, ako ľudské oko vníma farby. Štandardne majú hodnoty  $\alpha = 0,2989$ ,  $\beta = 0,5870$  a  $\gamma = 0,1140$  [11].

Pri rozpoznávaní objektov nás však nezaujíma obraz ako celok, ale jednotlivé objekty, ktoré sa na ňom nachádzajú. Na to aby sme mohli rozlišovať symboly na kartách, je potrebné ich najskôr oddeliť od pozadia. Dôležitým krokom pre účely tejto práce je preto proces segmentácie.

## Kapitola 2

# Segmentácia

Úlohou segmentácie je rozdeliť obraz na pozadie a jednotlivé objekty záujmu. Tento proces sa zameriava na vzájomný vzťah daného pixelu vzhľadom k okolitým ale aj rôznym ďalším pixelom v obraze. Výstupom segmentácie sú zväčša dáta, ktoré popisujú hranice daných objektov alebo objekty celé. Základné metódy sa delia na metódy zaoberajúce sa hranicami objektov a metódy orientované na oblasti.

### 2.1 Segmentácia zameraná na hranice

Segmentácia zameraná na hranice skúma ostré, lokálne rozdiely intenzity. Tieto metódy sú realizované prostredníctvom prvých a druhých derivácií. Gradient je vektor, ktorý v každom bode udáva smer a veľkosť maximálnej zmeny funkcie  $f$ . Preto ho považujeme za vhodný nástroj na detekciu hrán. Príkladom využitia gradientu je *Prewittovej* alebo *Sobelov filter*, ktoré získavajú jeho hodnotu pre daný pixel konvolúciou obrazu  $f$  a príslušného jadra  $\mathbb{M}$  ako:

$$g(x, y) = \sum_{i=I_{min}}^{I_{max}} \sum_{j=J_{min}}^{J_{max}} \mathbb{M}(i, j) f(x + i, y + j), \quad (2.1)$$

kde indexy  $i = 0, j = 0$  zodpovedajú stredu jadra, ktoré má rozmery  $(I_{max} - I_{min} + 1, J_{max} - J_{min} + 1)$  a  $x, y$  značia polohu pixelu [11]. Tieto jadrá sa nazývajú Prewittovej a Sobelov operátor. Prewittovej operátor má tvar:

$$\mathbb{M}_x = \begin{pmatrix} -1 & 0 & 1 \\ -1 & 0 & 1 \\ -1 & 0 & 1 \end{pmatrix}, \mathbb{M}_y = \begin{pmatrix} -1 & -1 & -1 \\ 0 & 0 & 0 \\ 1 & 1 & 1 \end{pmatrix}, \quad (2.2)$$

Sobelov:

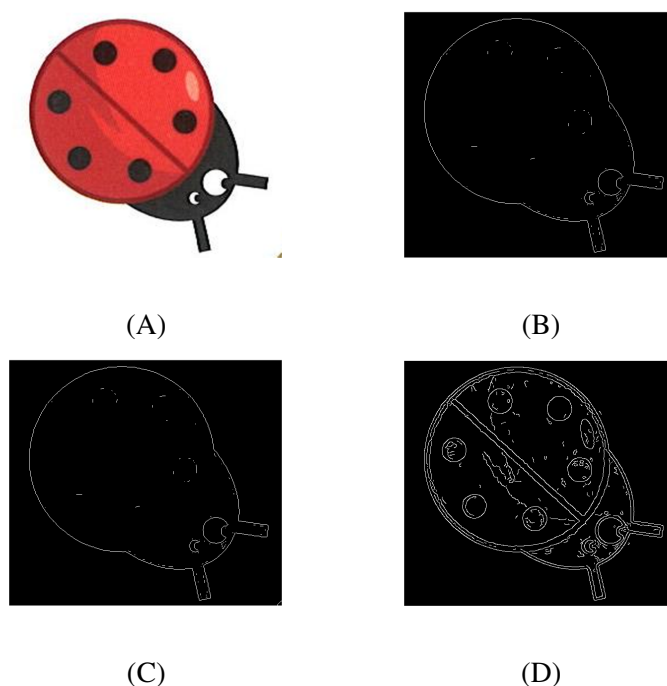
$$\mathbb{M}_x = \begin{pmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{pmatrix}, \mathbb{M}_y = \begin{pmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{pmatrix}. \quad (2.3)$$

Spomínané filtre však neurčujú priamo hranice, udávajú len náznak toho, kde sa môžu nachádzať. Pri samotnom hľadaní treba rovnako vziať do úvahy veľkosť gradientu, počínajúc ktorou sú dané body chápané ako hraničné, alebo šum, ktorý pri použití týchto metód môže spôsobiť, že za hraničný bude považovaný aj taký, čo do nej nepatrí [11]. Medzi pokročilejšie metódy zaoberajúce sa týmito skutočnosťami patria napríklad *Cannyho hranový detektor* a *Marr-Hildrethov hranový detektor*.

Ideálne by hranice mali byť spojité, k čomu nemusí vplyvom nedostatkov v obraze pri niektorých metódach dochádzať. Preto môže byť často detekcia hránic sprevádzaná algoritmi na ich zoskupenie.

Tieto metódy delíme na lokálne, to sú také, ktoré spočívajú v analýze pixelov v úzkom okolí hraničných a následnom spojení tých, ktoré sú podobné v závislosti na určitých kritériách a globálne metódy, pri ktorých detegujeme, či množina pixelov leží na krivke daného tvaru a túto krivku ďalej považujeme za hľadanú hranicu [5].

Výsledný obraz po prevedení segmentácie zameranej na hranice pomocou Sobelovho, Prewittovej filtru a Cannyho detektoru je zobrazený na Obrázku 2.1.



Obr. 2.1: Segmentácia obrázku (A) použitím Sobelovho (B), Prewittovej filtru (C) a Cannyho detektoru (D)

## 2.2 Segmentácia orientovaná na oblasti

Tieto metódy sa zaoberajú rozdelením obrazu na jednotlivé oblasti na základe nejakých preddefinovaných kritérií. Typickými príkladmi sú prahovanie, segmentácia štiepením, spájaním, narastaním oblastí.

### Prahovanie

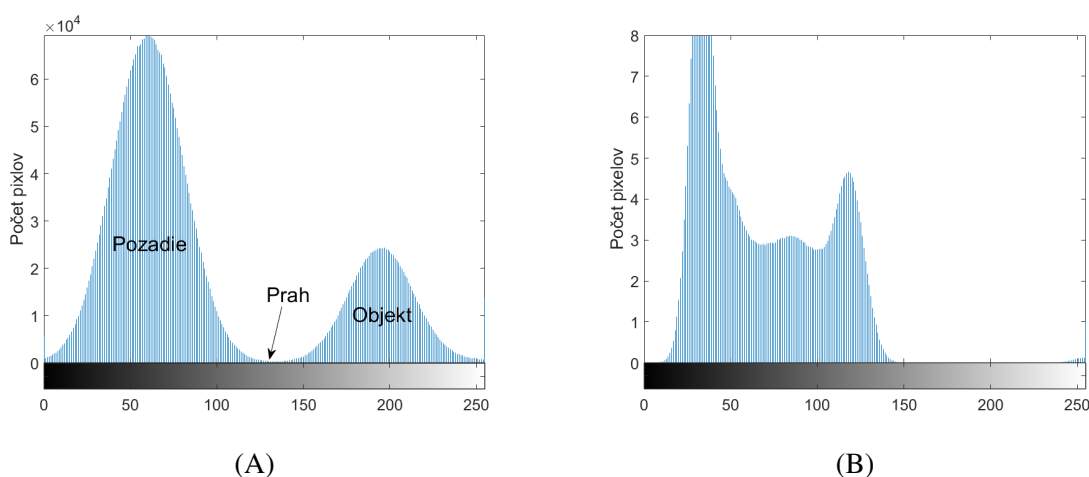
Prahovanie spočíva vo voľbe vhodného prahu intenzity obrazu  $T$ , ktorý oddeľuje pozadie od jednotlivých objektov. Výsledkom je binárny obraz vytvorený z pôvodného obrazu definovaný ako:

$$g(x, y) = \begin{cases} 1 & \text{ak } f(x, y) > T \\ 0 & \text{ak } f(x, y) \leq T, \end{cases} \quad (2.4)$$

kde  $f(x, y)$  je vstupný obraz. Každý bod, v ktorom platí  $f(x, y) > T$  je bodom objektu. Ak je možné použiť jednu konštantu  $T$  pre celý obraz, jedná sa o globálne prahovanie. Naopak pri premennej hodnote

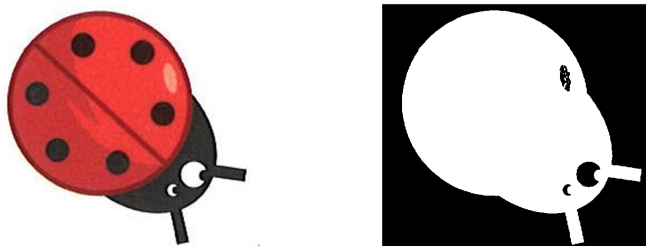
$T$  hovoríme o variabilnom prahovaní. Pokiaľ konštanta závisí priamo na súradniciach  $(x, y)$  variabilné prahovanie nazývame adaptívne alebo dynamické [5].

Globálne prahovanie je možné využiť pri dostatočnej odlišnosti intenzity objektov a pozadia. Na Obrázku 2.2 je uvedený ideálny prípad histogramu pre globálne prahovanie ako aj príklad toho, kedy táto metóda segmentácie dobre nefunguje.



Obr. 2.2: Ideálny prípad pre použitie globálneho prahovania (A) a príklad histogramu obrazu, pre ktorý táto metóda dobre nefunguje (B)

Hodnotu prahu vieme vypočítať z histogramu, v špecifických prípadoch môže byť zvolená manuálne, alebo pomocou Otsuovej metódy. Otsuova metóda vyberá zo všetkých možných taký prah, ktorý maximalizuje štatistický rozptyl medzi jednotlivými triedami. Podrobne je popísaná v [5] a jej výsledok zobrazuje Obrázok 2.2.

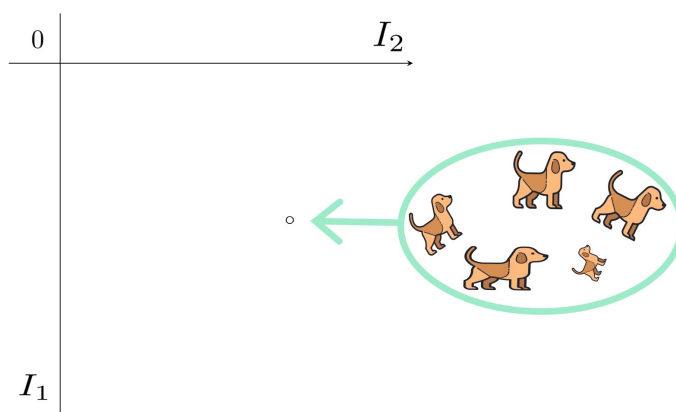


Obr. 2.3: Segmentácia pomocou globálneho prahovania Otsuovou metódou

## Kapitola 3

# Príznyky

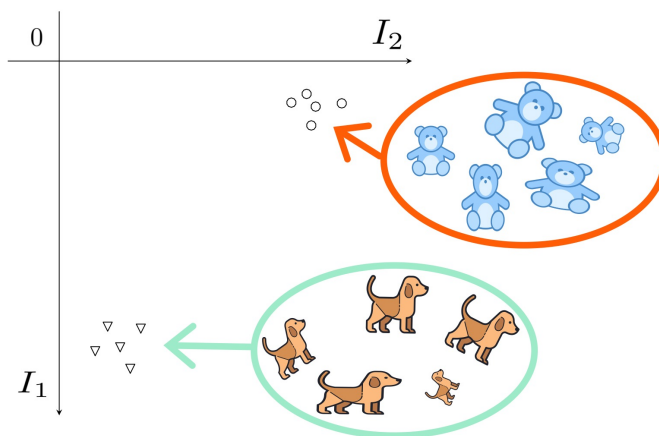
Na rozpoznávanie objektov sa v analýze obrazu často používajú príznakové metódy. Príznyky kvantitatívne popisujú vlastnosti objektov. Ich hodnoty sú považované za prvky funkčného priestoru. Aby bolo pomocou nich možné objekt rozpoznať v rôznych podmienkach, mali by byť jednoznačné a v závislosti na degenerácii, ktorá sa v danej aplikácii môže vyskytovať, invariantné voči rôznym transformáciám ale napríklad aj zmenám osvetlenia alebo uhlu pohľadu. Za invariantný príznak považujeme funkcionál  $I$ , ktorý obraz  $f$  prevádza do funkčného priestoru tak, že  $I(f)$  závisí na triede, do ktorej  $f$  patrí a nezávisí na jeho konkrétnom vzhľade. Ak  $\mathcal{D}$  je ľubovoľná trieda prípustných degradácií obrazu, potom  $I(f) = I(\mathcal{D}(f))$  pre invariant  $I$  [3]. Invariantnosť voči skupine transformácií teda znamená, že jej aplikáciou na obraz sa hodnoty príznakov nemenia [5]. Príklad tejto vlastnosti je zobrazený na Obrázku 3.1.



Obr. 3.1: Invariantnosť príznakov voči zobrazeným transformáciám

Invariantnosť však nie je pre príznaky postačujúca na to, aby bolo pomocou nich možné objekty rozpoznať. Napríklad ak  $I(f) = 0$  pre všetky  $f$ , potom je  $I$  invariantom, ale jeho rozpoznávaciu schopnosť je nulová. Rozšírenie invariantnosti nad určitú mieru môže naopak zhoršiť ich rozpoznávaciu schopnosť. Preto je pri rozpoznávaní objektov v obraze rovnako dôležité zvoliť vhodné príznaky tak, aby boli invariantné ale zároveň dokázali objekty odlíšiť [3]. Táto vlastnosť sa nazýva diskriminabilita a je protichodná k invariantnosti. Obrázok 3 znázorňuje skoro ideálny prípad pre rozpoznávanie, kedy príznaky

jedného transformovaného objektu danej triedy majú približne rovnaké hodnoty a zároveň sú tieto hodnoty dostatočne vzdialené od seba pre rôzne objekty. K ideálnym prípadom invariantov, kedy sa zobrazia do jedného bodu v praxi nedochádza kvôli tomu, že počítače pracujú s diskretnými objektami a majú konečnú presnosť, čoho následkom vznikajú zaokrúhľovacie chyby a podobné nedostatky, ktoré tieto hodnoty vychylujú.



Obr. 3.2: Ideálny prípad hodnôt invariantov pre dva rôzne transformované objekty

Na obrázkoch skúmaných v tejto práci, dochádza k otočeniu objektov, posunutiu a zmene mierky. Preto sa ďalej práca zaoberá príznakmi ktoré sú na daných transformáciách nezávislé, alebo u nich vieme invariantnosť dosiahnuť normalizáciou.

Translácia, rotácia a škálovanie sa dá vyjadriť v maticovom tvare ako:

$$\mathbf{x}' = s\mathbb{R}_\theta\mathbf{x} + \mathbf{t}, \quad (3.1)$$

kde  $\mathbf{x}$  značí vektor súradníc pôvodného obrazu,  $\mathbf{x}'$  je vektor súradníc po transformácii,  $\mathbf{t}$  je vektor posunu,  $s$  je faktor, ktorým obraz škálujeme a  $\mathbb{R}_\theta$  je matica rotácie o uhol  $\theta$  definovaná ako:

$$\mathbb{R}_\theta = \begin{pmatrix} \cos \theta & -\sin \theta \\ \sin \theta & \cos \theta \end{pmatrix}. \quad (3.2)$$

Vo väčšine prípadov je na rozpoznávanie potrebné použiť súčasne viacero príznakov. Spájame ich do takzvaného vektoru príznakov.

Literatúra uvádza mnoho kritérií, podľa ktorých sa dajú príznaky deliť do skupín. Medzi najpodrobnejšie patrí rozdelenie podľa matematického aparátu, ktorého základné kategórie sú: vizuálne príznaky, úplné vizuálne príznaky, transformačné koeficienty, príznaky textúry, diferenciálne invarianty, momentové invarianty a iné.

### 3.1 Vizuálne príznaky

Vizuálne príznaky sú založené na princípe zrovnania rozmerov objektu vzhľadom ku nejakému zvolenému referenčnému tvaru. Sú jednoduché na výpočet, invariantné voči rotácii, translácii aj uniformnému škálovaniu, ich schopnosť rozpoznávania je však pomerne nízka. Preto sa používajú viaceré súčasne. Príkladom vizuálnych príznakov je kompaktnosť, kruhovitosť, excentricita, konvexita a hrnatosť [3].

### Kompaktnosť

Kompaktnosť je daná rovnicou:

$$\text{kompaktnosť} = \frac{O(A)^2}{S(A)}, \quad (3.3)$$

kde  $S(A)$  je obsah a  $O(A)$  je obvod hranice oblasti  $A$ . Hranicu oblasti tvoria pixely, ktoré sa v nej nachádzajú a zároveň susedia aspoň s jedným pixelom pozadia. Pojmom hranice sa detailnejšie zaoberáme v sekcii o úplných vizuálnych príznakoch. Obsah je počet pixelov, ktoré sa v oblasti nachádzajú, obvod je dĺžka jej hranice. Pre kruhový tvar má hodnotu  $4\pi$  a pre štvorcový 16.

### Kruhovitosť

Veľmi úzko súvisiacim príznakom ku kompaktnosti je kruhovitosť. Definujeme ju ako:

$$\text{kruhovitosť} = \frac{4\pi S(A)}{O(A)^2} \quad (3.4)$$

s rovnakým značením ako v predošlom prípade. Nadobúda hodnoty v rozmedzí 0 až 1, jednej sa rovná len v prípade, že oblasť  $A$  je kruhová. Pre štvorec nám dáva hodnotu  $\pi/4$ .

### Excentricita

Excentricita oblasti je určená excentricitou elipsy, ktorá má rovnaký druhý centrálny moment ako daná oblasť. Podľa vzorca pre výpočet excentricity elipsy teda dostávame:

$$\text{excentricita} = \frac{\sqrt{a^2 - b^2}}{a} = \sqrt{1 - (b/a)^2}, \quad a \geq b. \quad (3.5)$$

Dĺžku hlavnej poloosi elipsy značíme  $a$ , dĺžku vedľajšej  $b$ . Excentricita má maximálnu hodnotu 1 pre oblasť tvaru čiary a minimálnu hodnotu 0 pre kruhovú oblasť [5].

### Konvexita

Konvexita určuje podobnosť objektu a jeho konvexného obalu ako:

$$\text{konvexita} = \frac{S(A)}{S(C_A)}, \quad (3.6)$$

kde konvexný obal značíme  $C_A$ . Oblasť nazývame konvexnou, ak pre každé dva body patriace do danej oblasti, do nej patrí aj úsečka, ktorá ich spája. Konvexný obal je teda najmenšia konvexná oblasť do ktorej objekt patrí. Hodnoty konvexity sa pohybujú medzi 0 a 1, hodnotu 1 nadobúda v prípade konvexného tvaru.

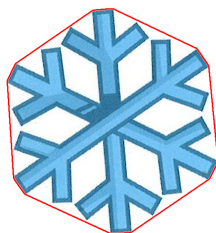
### Hranatosť

Hranatosť udáva mieru podobnosti oblasti a jej prislúchajúceho najmenšieho opísaného obdĺžnika. Znovu je veľkosť tejto veličiny kladná a nepresiahne hodnotu 1, ktorú nadobúda pre obdĺžnikový tvar. Vychádza z rovnice:

$$\text{hranatosť} = \frac{S(A)}{S(R_A)}, \quad (3.7)$$

kde  $S(R_A)$  je obsah spomínaného obdĺžnika. Pomocou najmenšieho opísaného obdĺžnika sa dá ďalej vypočítať *pozdlžnosť*, ktorá je daná pomerom jeho strán.





Obr. 3.3: Príklad konvexného obalu objektu

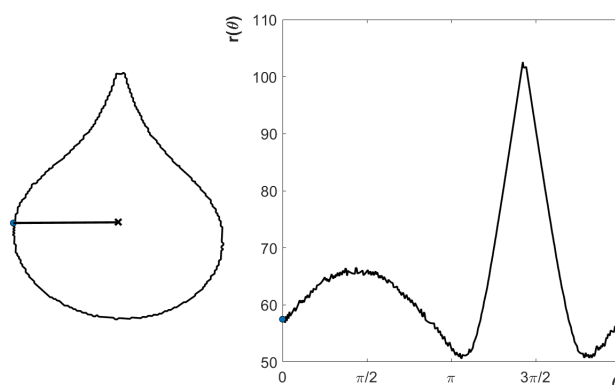
## 3.2 Úplné vizuálne príznaky

Pre prípad, že sa do databázy ukladajú len príznakové vektory a požadujeme, aby sa pomocou nich dal objekt znovu zrekonštruovať, musí byť ich vektor kompletný. To znamená, že ďalšie nezávislé príznaky pre daný objekt už neexistujú [3]. Úplné vizuálne príznaky popisujú objekty pomocou kompletných príznakových vektorov. Typickými príkladmi takýchto príznakov pre binárne objekty sú signatúra, tvarová matica alebo reťazový kód.

### Signatúra

Jednorozmerný funkcionál, ktorý popisuje dvojrozmernú hranicu objektu nazývame signatúrou. Existuje množstvo spôsobov, akými sa dá signatúra vytvoriť. Jedným z najjednoduchších je pomocou radiálnej funkcie [5]. Radiálna funkcia  $r(\theta)$  popisuje vzdialenosť hranice oblasti od stredu v závislosti na uhle  $\theta$ . Obrázok 3.4 zobrazuje príklad signatúry pre daný objekt.

Vzorkovaním z nej vieme získať *radiálny tvarový vektor*, ktorý možno priamo použiť ako vektor príznakov. Zmenou dĺžky tohto vektora vieme regulovať jeho rozpoznávaciu schopnosť. Keďže radiálna



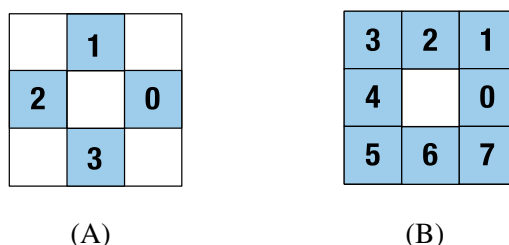
Obr. 3.4: Signatúra pre daný tvar hranice objektu

funkcia popisuje vzdialenosť od stredu objektu, je invariantná voči posunutiu. Zmena veľkosti objektu sa však prejaví premenou amplitúdy. Normalizáciou vektora jeho najväčším prvkom vieme dosiahnuť, aby sa hodnoty všetkých funkcií pohybovali v rovnakom rozmedzí. Pri zmene počiatočného bodu alebo rotácii objektu dochádza k posunu radiálnej funkcie. Preto je pri výpočte vzdialenosti medzi dvoma tvarovými vektormi potrebné hľadať minimum spomedzi všetkých posunov porovnávaného vektora [3].

Popis pomocou signatúry je jednoduchý nástroj na rozpoznávanie objektov. Táto metóda však funguje len pre hviezdicové objekty. Teda také, pre ktoré platí, že každá polpriamka začínajúca v strede objektu pretne jej hranicu len raz [3]. Nevýhodou tejto metódy je rovnako aj jej citlivosť na šum a fakt, že škálovanie radiálnej funkcie závisí len na jej minime a maxime [5].

### Reťazový kód

Reťazový kód popisuje hranicu objektu na základe vzájomných vzťahov pixelov. Pri rozdeľovaní pixelov na hraničné a vnútorné musíme brať do úvahy, ktoré z nich sa považujú za susedné vzhľadom ku skúmanému pixelu. Susedné môžu byť teda pixely, ktoré sa nachádzajú horizontálne a vertikálne vzhľadom ku skúmanému (4-susednosť), alebo všetky, ktoré ho obklopujú (8-susednosť). Majme pixel so súradnicami  $(x, y)$ , potom horizontálne budú mať polohu  $(x + 1, y)$ ,  $(x - 1, y)$  a vertikálne  $(x, y + 1)$ ,  $(x, y - 1)$ .



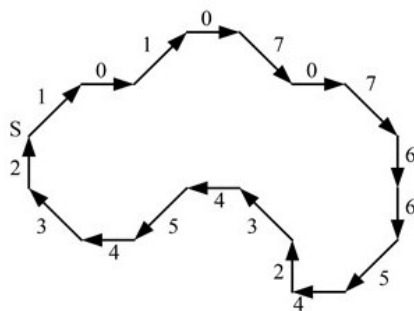
Obr. 3.5: Typy susednosti pixelov: 4-susednosť (A), 8-susednosť (B)

Reťazový kód vzniká opisom hranice pomocou vzájomnej pozície hraničných prvkov [9]. Počínajúc zvoleným hraničným bodom sa hranica prechádza a ukladajú sa čísla, ktoré popisujú smery postupu. V závislosti na type susedstva sú pridelené jednotlivým smerom čísla zobrazené v Obrázku 3.5. Kód tvorený týmito číslami sa nazýva *Freemanov reťazový kód*.

Invariantnosť voči zmene počiatočného bodu dosiahneme tak, že budeme kód cyklicky posúvať až dokým nebude odpovedať najmenšiemu možnému celému číslu.

Pre dosiahnutie nezávislosti na rotácii sa používa diferenčný reťazový kód. Vypočítame ho ako počet zmien smeru pri prechode z daného bodu na nasledujúci v protismere hodinových ručičiek. Pri poslednom bode hranice počítame zmenu vzhľadom k prvému bodu.

Normalizácia pre zmenu veľkosti sa dá vyriešiť prevzorkovaním hraníc pred výpočtom samotného kódu, čo je dôležité aj kvôli zamedzeniu chybám, ktoré vznikli pri segmentácii alebo prostredníctvom šumu [5].

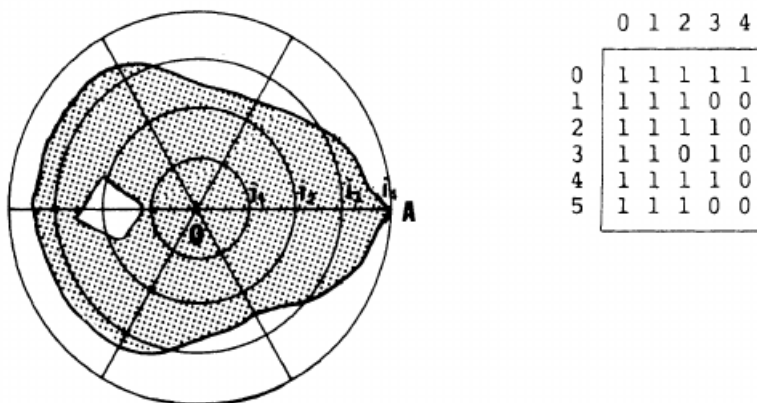


Obr. 3.6: Príklad 8-susednej hranice objektu, ktorého reťazový kód je: 101070766542345432 [14]

Reťazový kód je však v praxi málo používaný, pretože neexistuje žiadna jednoduchá metrika, ktorá by odpovedala skutočnému rozdielu tvaru objektov. Porovnávanie kódov preto prebieha pomocou hľadania najdlhšieho spoločného podreťazca, ktoré je časovo náročné [3].

### Tvarová matica

Táto metóda popisu objektov prevedie daný tvar na binárnu maticu o zvolenej veľkosti. Je invariantná voči rotácii, posunu aj škálovaniu a je vhodná aj pre objekty, ktoré majú v sebe diery alebo sú rozdelené na viaceré izolované časti. Tvarovú maticu rozmerov  $m \times n$  dostaneme, ak úsečku spájajúcu stred objektu  $O$  a jeho najvzdialenejší bod od stredu  $A$  rozdelíme na  $n-1$  rovnako dlhých úsekov. Cez objekt nakreslíme sústredné kružnice so stredom v bode  $O$  a polomeri  $L/(n-1), 2L/(n-1), \dots, (n-1)L/(n-1)$ , kde  $L$  značí dĺžku úsečky  $OA$ . Počínajúc úsečkou  $OA$  v protismere hodinových ručičiek rozdelíme kružnice na  $m$  oblúkov s veľkosťou uhlu  $d\theta = 360/m$  stupňov. Ak časť mriežky na pozícii  $(iL/(n-1), j(360/m))$  leží v objekte, potom prvok matice  $m_{ij}$  bude mať hodnotu 1 [6]. Príklad tvarovej matice objektu znázorňuje Obrázok 3.7.



Obr. 3.7: Matica pre daný tvar objektu [6]

## 3.3 Transformačné koeficienty

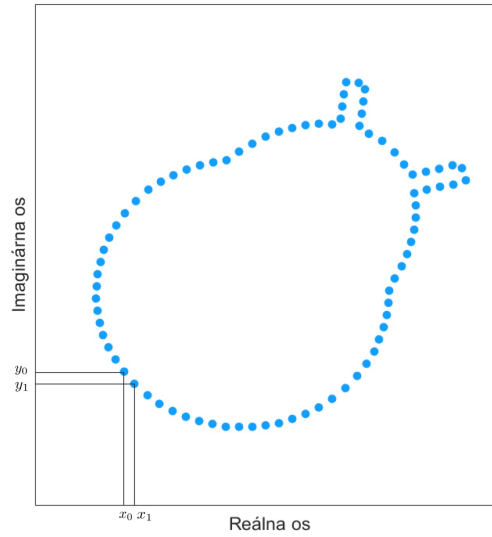
Transformačné koeficienty dostávame transformáciou hranice objektu. Medzi používané transformácie patria napríklad Fourierová, Walsh-Hadamardova, ale aj iné podobné transformácie.

### Fourierové deskriptory

Majme hranicu objektu, pozostávajúcu z  $N$  pixeov na pozíciách  $(x_0, y_0), (x_1, y_1), \dots, (x_{N-1}, y_{N-1})$ . Pomocou týchto súradníc vieme vyjadriť body hranice objektu vo forme komplexného čísla ako:

$$z(k) = x(k) + iy(k), \quad (3.8)$$

kde  $x(k) = x_k$  a  $y(k) = y_k$  pre  $k = 0, 1, \dots, N-1$ . Hranicu objektu v komplexnej rovine znázorňuje Obrázok 3.8. Fourierové deskriptory vznikajú aplikáciou diskretnej Fourierovej transformácie na postupnosť komplexných čísel (3.8) [5].



Obr. 3.8: Príklad hranice objektu v komplexnej rovine

Fourierovú transformáciu diskretnej funkcie  $\tilde{f}$  vieme z definície vyjadriť rovnicou:

$$\mathcal{F}[\tilde{f}(t)](\mu) = \tilde{F}(\mu) = \int_{-\infty}^{+\infty} \tilde{f}(t) e^{-i2\pi\mu t} dt. \quad (3.9)$$

Funkcia  $\tilde{f}$  má tvar:

$$\tilde{f}(t) = \sum_{k=-\infty}^{+\infty} f(t) \delta(t - k\Delta T), \quad (3.10)$$

kde  $\Delta T$  je interval medzi jednotlivými vzorkami funkcie  $f$  a  $\delta$  je diskretný impulz definovaný ako:

$$\delta(x) = \begin{cases} 0 & \text{ak } x \neq 0 \\ 1 & \text{ak } x = 0. \end{cases} \quad (3.11)$$

Dosadením tohto tvaru  $\tilde{f}$  do Fourierovej transformácie dostávame rovnosť:

$$\begin{aligned} \tilde{F}(\mu) &= \int_{-\infty}^{+\infty} \tilde{f}(t) e^{-i2\pi\mu t} dt = \int_{-\infty}^{+\infty} \sum_{k=-\infty}^{+\infty} f(t) \delta(t - k\Delta T) e^{-i2\pi\mu t} dt = \\ &= \sum_{k=-\infty}^{+\infty} \int_{-\infty}^{+\infty} f(t) \delta(t - k\Delta T) e^{-i2\pi\mu t} dt = \sum_{k=-\infty}^{+\infty} f_k e^{-i2\pi\mu k\Delta T}, \end{aligned} \quad (3.12)$$

v ktorej  $f_k = \int_{-\infty}^{+\infty} f(t) \delta(t - k\Delta T) dt = f(k\Delta T)$  značí  $k$ -tu vzorku funkcie  $f$ . Aby sme získali  $M$  rovnomerne rozložených vzoriek Fourierovej transformácie zavedieme frekvenciu  $\mu$  v tvare:

$$\mu = \frac{m}{M\Delta T} \quad \text{pre } m = 0, 1, \dots, M-1 \quad (3.13)$$

a substitúciou  $\mu$  v (3.12) definujeme diskretnú Fourierovú transformáciu ako:

$$\tilde{F}(m) = \sum_{k=0}^{M-1} f(k) e^{-i2\pi mk/M} \quad \text{pre } m = 0, 1, \dots, M-1, \quad (3.14)$$

Pre diskretnú Fourierovú transformáciu existuje rovnako aj inverzná transformácia tvaru:

$$f(m) = \frac{1}{M} \sum_{k=0}^{M-1} F(k) e^{i2\pi mk/M} \quad \text{pre } m = 0, 1, \dots, M-1. \quad (3.15)$$

Ak máme hranicu objektu v tvare (3.8), potom budú deskriptory určené rovnicou:

$$Z(m) = \sum_{k=0}^{M-1} z(k) e^{-i2\pi mk/M} \quad \text{pre } m = 0, 1, \dots, M-1. \quad (3.16)$$

Rotácia o uhol  $\theta$  sa v komplexných súradniciach prejaví násobením bodu konštantou  $e^{i\theta}$ . Pri rotácii objektu teda pre jednotlivé deskriptory platí:

$$Z_r(m) = \sum_{k=0}^{M-1} z_r(k) e^{-i2\pi mk/M} = \sum_{k=0}^{M-1} z(k) e^{-i2\pi mk/M} e^{i\theta} = Z(m) e^{i\theta} \quad (3.17)$$

Posun o  $\Delta_{xy} = \Delta x + i\Delta y$  sa prejaví nasledovne:

$$\begin{aligned} Z_t(m) &= \sum_{k=0}^{M-1} z_t(k) e^{-i2\pi mk/M} = \sum_{k=0}^{M-1} (z(k) + \Delta_{xy}) e^{-i2\pi mk/M} = \sum_{k=0}^{M-1} z(k) e^{-i2\pi mk/M} + \Delta_{xy} \sum_{k=0}^{M-1} e^{-i2\pi mk/M} \\ &= Z(m) + \Delta_{xy} \delta(m), \end{aligned} \quad (3.18)$$

kde platnosť poslednej rovnosti vychádza z faktu, že Fourierová transformácia konštanty nám dáva impulz nachádzajúci sa v počiatku [5].

Škálovanie konštantou  $\alpha$  nám dáva:

$$Z_s(m) = \sum_{k=0}^{M-1} z_s(k) e^{-i2\pi mk/M} = \sum_{k=0}^{M-1} \alpha z(k) e^{-i2\pi mk/M} = \alpha Z(m). \quad (3.19)$$

Ak sa počiatkový bod hranice zmení zo  $z(0)$  na  $z(k_0)$  dochádza k nasledujúcej zmene deskriptorov [5]:

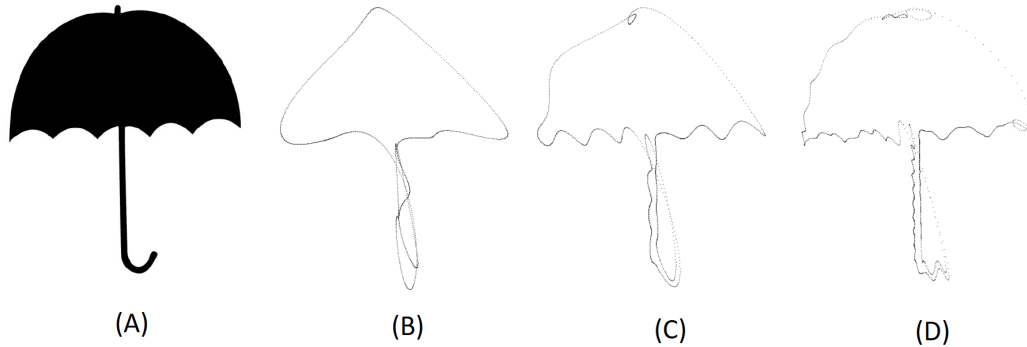
$$Z_p(m) = \sum_{k=0}^{M-1} z_p(k) e^{-i2\pi mk/M} = \sum_{k=0}^{M-1} z(k - k_0) e^{-i2\pi mk/M} = Z(m) e^{-i2\pi k_0 m/M}. \quad (3.20)$$

Z Rovnice (3.18) možno usúdiť, že na polohe objektu závisí len prvý koeficient transformácie. Jeho odstránením dosiahneme nezávislosť deskriptorov na posunutí. Invariantnosť voči rotácii možno jednoducho zabezpečiť tým, že sa miesto samotného koeficientu bude brať jeho absolútna hodnota. Tým rovnako dosiahneme aj nezávislosť na voľbe počiatkového bodu. Normalizáciu voči zmene mierky možno docieľiť delením každého deskriptoru absolútnou hodnotou prvého [3]. Výsledné Fourierové deskriptory invariantné voči uvedeným transformáciám majú tvar:

$$C(m) = \frac{|Z(m)|}{|Z(1)|} \quad \text{pre } m = 2, 3, \dots, M-1. \quad (3.21)$$

Prvky transformácie o vysokých frekvenciách popisujú detaily hranice, prvky s nízkymi frekvenciami celkový tvar objektu. Použitie množstvo koeficientov preto ovplyvňuje rozpoznávaciu schopnosť deskriptorov. V Obrázku 3.9 vidíme, že pri rekonštrukcii hranice z vyšších počtov deskriptorov pripomína jej tvar pôvodný objekt s väčšou presnosťou.

V praxi sa často Fourierová transformácia počíta pomocou algoritmu rýchlej Fourierovej transformácie [5].



Obr. 3.9: Rekonštrukcia hranice objektu (A) pomocou zvyšujúceho sa množstva Fourierových deskriptorov, ktorých je pre tvar (B) 10, (C) 20, (D) 50 [4]

### 3.4 Momenty a momentové invarianty

Posledným typom príznakov ktoré v tejto práci spomenieme sú momentové invarianty. Používajú sa pre binárny, šedotónový, farebný, ale aj vektorový obraz. Tieto príznaky sú špeciálnymi funkciami obrazových momentov.

**Definícia 2.** *Bud'  $\{\pi_p(\mathbf{x})\}$  polynomická báza o  $d$ -premenných na priestore obrazových funkcií definovaných na  $\Omega$ . Nech  $\mathbf{p} = (p_1, \dots, p_d)$  je multiindex kladných celých čísel, ktorý udáva najvyššiu mocninu premenných v  $\pi_p(\mathbf{x})$ . Potom moment  $M_p^{(f)}$  obrazu  $f$  je definovaný ako:*

$$M_p^{(f)} = \int_{\Omega} \pi_p(\mathbf{x}) f(\mathbf{x}) d\mathbf{x}. \quad (3.22)$$

Číslo  $|p| = \sum_{k=1}^d p_k$  je rád momentu.

Momenty kvantitatívne opisujú tvar grafu funkcie  $f$ . Funkciu premietajú do bázy polynómov. Vzhľadom k polynomickej báze rozlišujeme rôzne druhy momentov: geometrické, komplexné, ortogonálne atď. Najpoužívanejšími sú vďaka jednoduchosti polynomickeho základu geometrické momenty. Sú však závislé na posunutí, rotácii a zmene mierky.

Posunutie o vektor  $t = (a, b)^T$  sa prejaví:

$$\begin{aligned} m'_{pq} &= \int_{-\infty}^{+\infty} \int_{-\infty}^{+\infty} x^p y^q f(x-a, y-b) dx dy = \int_{-\infty}^{+\infty} \int_{-\infty}^{+\infty} (x+a)^p (y+b)^q f(x, y) dx dy = \\ &= \sum_{k=0}^p \sum_{j=0}^q a^k b^j m_{p-k, q-j}. \end{aligned} \quad (3.23)$$

Nezávislosť dosiahneme posunutím objektu tak, aby sa jeho ťažisko nachádzalo v počiatku súradnicovej sústavy. Tým dostávame takzvané centrálné geometrické momenty:

$$\mu_{pq} = \int_{-\infty}^{+\infty} \int_{-\infty}^{+\infty} (x-x_c)^p (y-y_c)^q f(x, y) dx dy, \quad (3.24)$$

kde  $x_c = m_{10}/m_{00}$  a  $y_c = m_{01}/m_{00}$  sú súradnice stredy objektu.

Pri škálovaní obrazu faktorom  $s$  dostávame:

$$\begin{aligned}\mu'_{pq} &= \int_{-\infty}^{+\infty} \int_{-\infty}^{+\infty} (x - x_c)^p (y - y_c)^q f(x/s, y/s) dx dy = \\ &= \int_{-\infty}^{+\infty} \int_{-\infty}^{+\infty} s^p (x - x_c)^p s^q (y - y_c)^q f(x, y) dx dy = s^{p+q+2} \mu_{pq}.\end{aligned}\quad (3.25)$$

Momenty normalizované voči zmene mierky majú tvar:

$$\nu_{pq} = \frac{\mu_{pq}}{\mu_{00}^w}, \quad (3.26)$$

kde  $w = \frac{p+q}{2} + 1$ .

Medzi najslávnejšie momenty nezávislé na rotácii okolo počiatku patria napríklad Huove momenty, ktoré sú dané nasledujúcimi rovnicami:

$$\phi_1 = m_{20} + m_{02} \quad (3.27)$$

$$\phi_2 = (m_{20} - m_{02})^2 + 4m_{11}^2 \quad (3.28)$$

$$\phi_3 = (m_{30} - 3m_{12})^2 + (3m_{21} - m_{03})^2 \quad (3.29)$$

$$\phi_4 = (m_{30} + m_{12})^2 + (m_{03} + m_{21})^2 \quad (3.30)$$

$$\begin{aligned}\phi_5 &= (m_{30} - 3m_{12})(m_{30} + m_{12}) \left( (m_{30} + m_{12})^2 - 3(m_{21} + m_{03})^2 \right) + \\ &\quad (3m_{21} - m_{03})(m_{21} + m_{03}) \left( 3(m_{30} + m_{12})^2 - (m_{03} + m_{21})^2 \right)\end{aligned}\quad (3.31)$$

$$\phi_6 = (m_{20} - m_{02}) \left( (m_{30} + m_{12})^2 - (m_{21} + m_{03})^2 \right) + 4m_{11}(m_{30} + m_{12})(m_{21} + m_{03}) \quad (3.32)$$

$$\begin{aligned}\phi_7 &= (3m_{21} - m_{03})(m_{30} + m_{12}) \left( (m_{30} + m_{12})^2 - 3(m_{21} + m_{03})^2 \right) - \\ &\quad (m_{30} - 3m_{12})(m_{21} + m_{03}) \left( 3(m_{30} + m_{12})^2 - (m_{03} + m_{21})^2 \right).\end{aligned}\quad (3.33)$$

Ak geometrické momenty  $m_{pq}$  v Rovniciach (3.27) - (3.33) nahradíme normalizovanými  $\nu_{pq}$  z (3.26), sú tieto momenty invariantné voči rotácii, škálovaníu a translácii. Ich rozpoznávacíu schopnosť je však limitovaná a sú na sebe závislé. Hlavným nedostatkom Huovej teórie je, že pomocou nej nedokážeme odvodiť rotačné invarianty z momentov vyšších rádov. Ucelenú teóriu rotačných invariantov (z komplexných momentov s polynomickým základom  $\pi_{pq}(x, y) = (x + iy)^p (x - iy)^q$ ) publikoval Flusser v deväťdesiatych rokoch.

Okrem momentov nezávislých na rotácii existujú napríklad aj momenty invariantné voči afínnym transformáciám alebo rozmazaniu. Tieto, ale aj ďalšie druhy momentových invariantov sú podrobne vysvetlené v [3].

## Kapitola 4

# Dobble

Účinnosť zvolenej metódy z predošlej kapitoly budeme testovať na kartách hry Dobble. Keďže je táto hra dôležitou súčasťou práce, v tejto kapitole vysvetlíme cieľ hry a jej jednotlivé komponenty.

Dobble (alebo Spot It!) je spoločenská kartová hra, ktorá sa skladá z päťdesiatich piatich kariet okrúhleho tvaru (zobrazené na Obrázku 4.1). Na každej z nich sa nachádza osem rôznych objektov. Dokopy nájdeme v celej hre päťdesiat sedem symbolov, ktoré môžu byť na rôznych kartách inak otočené a môžu mať iné rozmery. Dôležitým faktorom v tejto hre je, že na každých dvoch kartách sa nachádza práve jeden spoločný symbol.



Obr. 4.1: Balíček kariet Dobble (Spot It!) [15]

Princíp, akým boli vytvorené karty, ktoré spĺňajú tieto požiadavky je založený na matematickej teórii konečných projektívnych rovín. Tento kombinatorický objekt pre množinu symbolov určuje Definícia 3.

**Definícia 3.** *Majme konečnú množinu symbolov  $S$  a množinu niektorých ich podmnožín (kariet)  $\mathcal{K} = \{K_1, K_2, \dots, K_m\} \subset 2^S$ . Usporiadanú dvojicu  $(S, \mathcal{K})$  nazývame konečnou projektívnou rovinou, ak spĺňa axiómy:*

1. každé dve množiny (karty)  $K, L \in \mathcal{K}$  majú spoločný práve jeden symbol, t.j.  $|K \cap L| = 1$ ,
2. pre každé dva symboly  $s, t \in S$  existuje práve jedna množina (karta)  $K \in \mathcal{K}$  pre ktorú  $s, t \in K$ ,
3. existuje množina štyroch symbolov  $M \subset S$ , z ktorých sa žiadne tri nevyskytujú na rovnakej karte, t.j.  $|M \cap K| \leq 2$  pre všetky karty  $K \in \mathcal{K}$  [12].

Rád konečnej projektívnej roviny značíme  $n$  a je daný rovnicou  $n = |K| - 1$ , kde  $K \in \mathcal{K}$  značí ľubovoľnú kartu a  $| \cdot |$  počet prvkov množiny  $K$ . Pomocou nasledujúcej vety vieme odvodiť počet kariet v balíčku pri danom počte symbolov na jednu kartu.

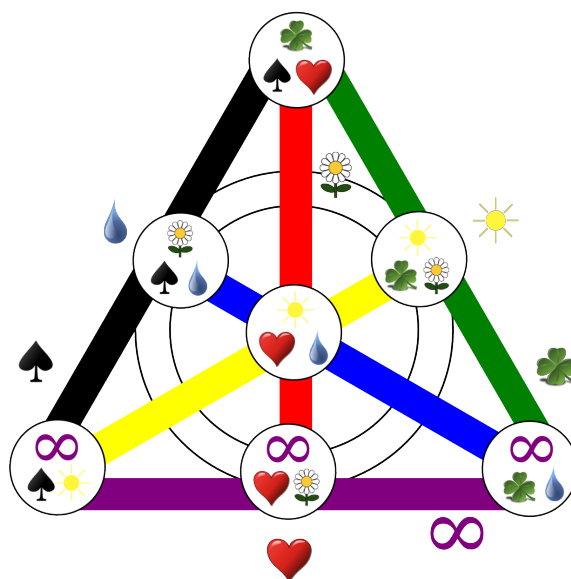


**Veta 1.** Ak je  $n \in \mathbb{N}, n \geq 2$ , potom konečná projektívna rovina  $(S, \mathcal{K})$  spĺňa:

1. každý symbol  $s \in S$  sa nachádza práve na  $q = n + 1$  kartách  $K \in \mathcal{K}$ ,
2.  $|S| = |\mathcal{K}| = n^2 + n + 1 = q^2 - q + 1$  [12].

Pre prípad ôsmich symbolov na kartu bude teda počet kariet rovný päťdesiatšesť. V samotnej hre však dve karty chýbajú pretože karty sú štandardne vyrábané po päťdesiatich piatich kusoch.

Všeobecne sa v teórii projektívnych rovín miesto pojmu kariet a symbolov používajú pojmy priamok, ktoré v Defínícii 3 značíme  $S$  a bodov, ktorých množina je označená  $\mathcal{K}$ . V Euklidovskej rovine predpokladáme, že každé dve priamky sa pretínajú v jednom bode alebo sú rovnobežné. Projektívna rovina je teda „rozšírením“ Euklidovskej roviny, kde sa každé dve priamky pretínajú práve v jednom bode (axióm 1. z Defínície 3). Konečná projektívna rovina má konečný počet priamok a bodov [2]. Príkladom najjednoduchšej takejto roviny so siedmimi bodmi a siedmimi priamkami je Fanova rovina. Vznik kariet na tejto rovine znázorňuje Obrázok 4.2. Každá priamka symbolizuje jeden objekt a body, v ktorých sa čiary pretnú, budú symbolizovať vzniknuté karty.



Obr. 4.2: Príklad vzniku kariet Dobble pre 7 objektov vo Fanovej rovine [1]

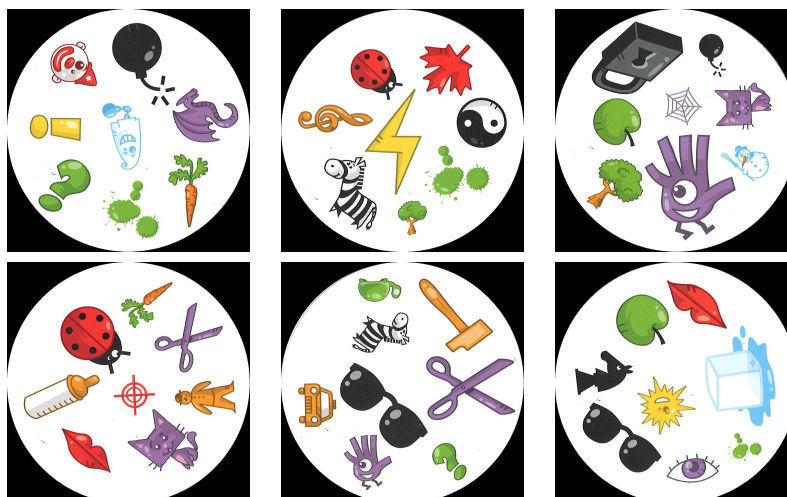
Existuje viacero verzií hry, ktoré sa dajú s kartami hrať. Cieľom všetkých z nich je nájsť spoločný symbol dvoch kariet, nahlas ho pomenovať a kartu s identifikovaným objektom vziať, umiestniť alebo zahodiť.

## Kapitola 5

# Praktická časť

Účelom tejto práce bolo zoznámenie sa s metódami popisu jednoduchých 2D objektov v obraze a následná implementácia zvolenej metódy v programovacom prostredí MATLAB. Metódy boli testované na sérii päťdesiatich piatich obrázkov kariet hry Dobble.

Obrázky kariet boli vytvorené skenovaním, následne boli orezané na rovnakú veľkosť a bola na ne aplikovaná kruhová maska.



Obr. 5.1: Príklad obrázkov použitých v experimente

### 5.1 Segmentácia a detekcia objektov

Pre segmentáciu bola zvolená metóda manuálneho prahovania. Metódu sme pre tento prípad zvolili pretože pozadie obrázkov bolo homogénne a objekty mali od neho výrazne odlišnú intenzitu. Testovaná bola aj Otsuova metóda, ktorá pre obrázky dobre nefungovala a pri použití získaného prahu splynuli objekty, ktoré mali vysoké hodnoty intenzity s pozadím.

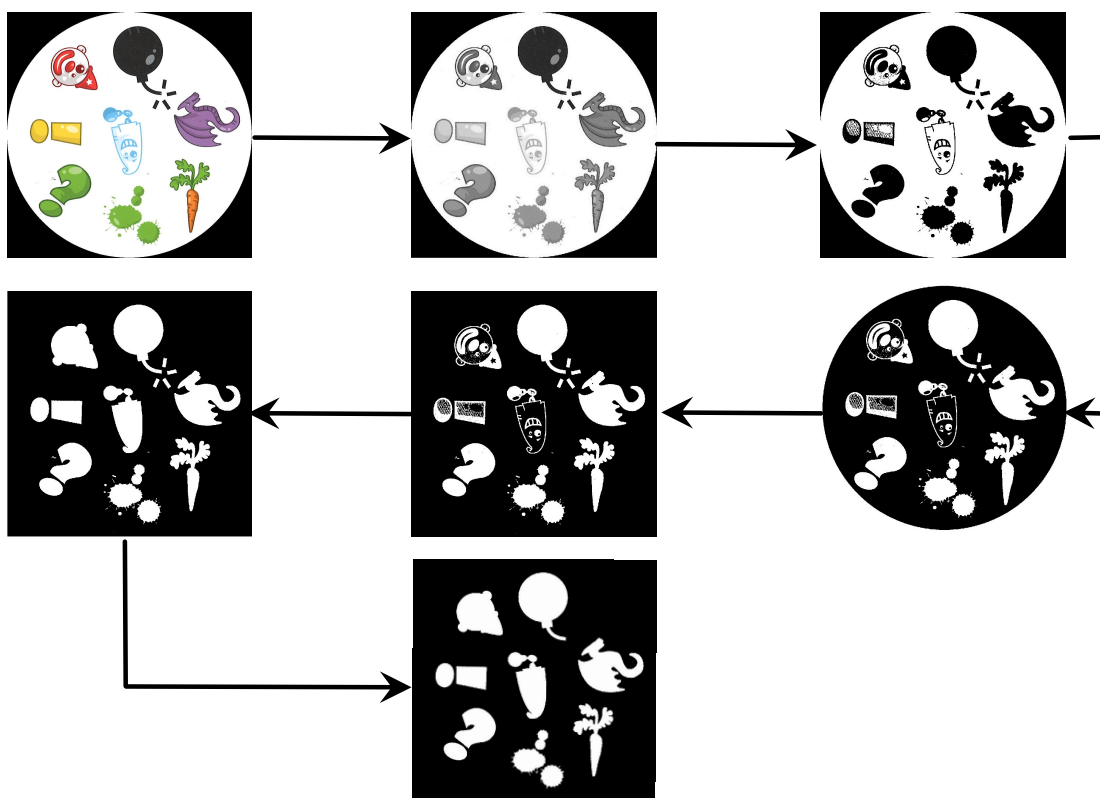
Obrázky boli pred segmentáciou prevedené na šedotónové. Výber prahu spočíval v testovaní rôznych hodnôt prahu na vybraných obrázkoch, ktoré obsahovali objekty s najvyššími hodnotami intenzity ako je duch, snehuliak alebo kocka ľadu. Tento výber bol prevádzaný pomocou aplikácie „Image Segmenter“ z „Image Processing Toolbox“.

Keďže má biele pozadie najvyššiu intenzitu a prahovanie obrazu je proces definovaný rovnicou (2.4), prah bol určený ako najmenšia hodnota intenzity  $T$ , ktorá je väčšia ako hodnota intenzity pixelov všetkých objektov zo všetkých kariet. Týmto spôsobom dostaneme binárny obraz, v ktorom budú mať pixely objektov nulovú a pixely bieleho pozadia jednotkovú hodnotu. Preto sa hodnoty nula a jedna musia v obraze zameniť, čo dosiahneme jeho invertovaním.

Vzhľadom k tomu, že na obraz bola použitá kruhová maska, po invertovaní budú mať pixely na ktorých mieste sa maska nachádzala jednotkové hodnoty. Tieto hodnoty vynulujeme pomocou funkcie na odstránenie hraníc (`imclearborder()`). Táto funkcia v binárnom obraze priradí nulovú hodnotu všetkým spojitým útvarom, ktoré majú jednotkové hodnoty pixelov a dotýkajú sa hraníc obrazu.

Vo vzniknutom binárnom obraze bola ďalej použitá funkcia na vyplnenie dier v objektoch a aplikáciou filtra boli vylúčené drobné bodky, ktoré vznikli pri skenovaní kariet a nepatrili medzi objekty.

Detekcia objektov bola prevedená pomocou funkcie, ktorá označí všetky pixely každého objektu celým číslom od jedna po počet objektov.



Obr. 5.2: Postup pri spracovaní od pôvodného obrázku po jeho výzor pred detekciou objektov

### Postup

1. Vstupný obraz prevedieme na šedotónový
2. Segmentácia šedotónového obrazu manuálnym prahovaním
3. Invertovanie vzniknutého binárneho obrazu

4. Odstránenie objektov dotýkajúcich sa hranice obrazu
5. Vyplnenie dier v objektoch
6. Použitie filtra na odstránenie malých objektov, ktoré medzi hľadané objekty nepatria

## 5.2 Výpočet príznakov

Keďže majú skúmané objekty dostatočne odlišné tvary a vyžadujeme invariantnosť príznakov voči rotácii, posunutiu a zmene mierky, zvolili sme pre ich jednoduchosť a rýchlosť výpočtu ako príznaky Fourierové deskriptory. Fourierové deskriptory sú ako už bolo vysvetlené v Kapitole 3, vypočítané Fourierovou transformáciou hranice objektu. Skôr než teda dôjde k výpočtu samotných deskriptorov, musia byť nájdené hranice objektov.

Hranice objektov sme získali pomocou funkcie, ktorá využíva algoritmus sledovania hranice. Pre každý označený objekt bola takto vytvorená matica, ktorá obsahuje  $x$ -ové a  $y$ -ové súradnice hraníc. Matica bola následne prevedená na vektor komplexných čísel, ktorého prvky mali tvar (3.8).

Na vektor komplexných čísel bola použitá funkcia pre výpočet diskretnej Fourierovej transformácie pomocou algoritmu rýchlej Fourierovej transformácie. Deskriptory sme následne normalizovali voči rotácii, posunutiu a zmene mierky podľa rovnice (3.21), kde sme namiesto normalizácie prvým deskriptorom použili normalizáciu na dĺžku hranice. Aby sa príznaky pohybovali v približne rovnakých hodnotách rozptylov pridali sme vyššiu váhu príznakom vyšších rádov tak, že sme každý vynásobili číslom o jedno väčším, ako je číslo jeho poradia.

Medzi objektami sa vyskytujú tri, ktoré majú rovnaký tvar hranice a to kruhový. Keďže deskriptory zohľadňujú len tvar hranice, boli k nim pridané príznaky priemernej farby. Priemerná farba bola získaná prevedením obrázku na typ double, aby sa jej hodnoty pohybovali v rozmedzí nula až jedna a následným výpočtom priemeru hodnôt intenzity pixelov každého farebného kanálu. Tieto tri hodnoty boli použité v príznakovom vektore spolu s deskriptormi.

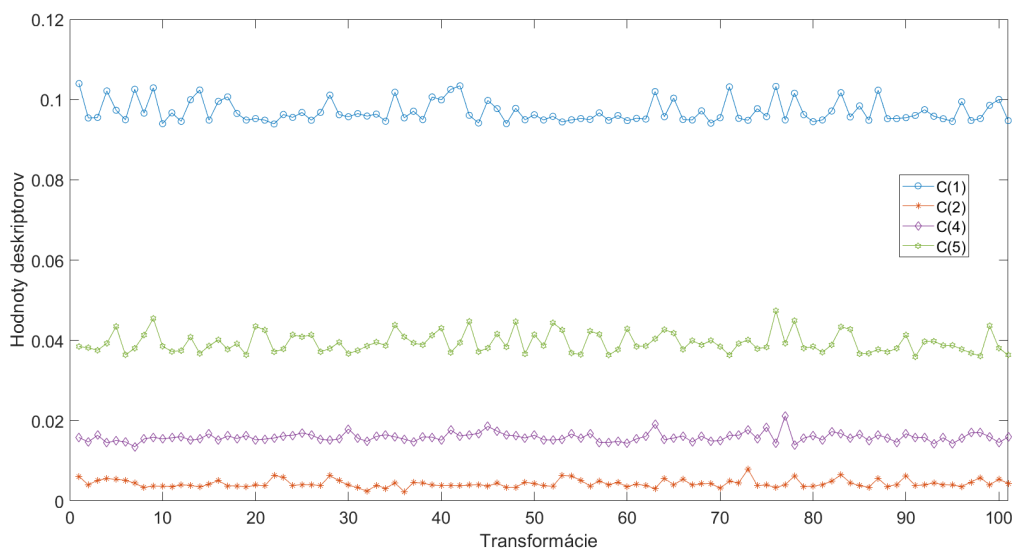
Pre klasifikáciu objektov bola použitá klasifikácia podľa minimálnej vzdialenosti (minimum distance classifier), ktorá spočíva v tom, že vektor príznakov priradí do triedy dosahujúcej minima vzdialenosti medzi týmto vektorom a všetkými triedami [3]. Pre výpočet vzdialenosti je možné zvoliť rôzne metriky (napríklad Euklidovská, Mahalanobisová, Manhattanská). V našom prípade sme pri klasifikácii pomocou deskriptorov využili Euklidovskú metriku, ktorá je pre dva vektory  $\mathbf{a}$ ,  $\mathbf{b}$  dĺžky  $n$  definovaná rovnicou:

$$\rho(\mathbf{a}, \mathbf{b}) = \sqrt{\sum_{i=1}^n (a_i - b_i)^2}. \quad (5.1)$$

## 5.3 Invariantnosť deskriptorov

Dôležitou vlastnosťou deskriptorov, ktorú v tejto práci skúmame je ich invariantnosť voči rotácii, zmene mierky a posunutiu. K testovaniu tejto vlastnosti sme si zvolili náhodne jeden objekt (pery), ktorý sme získali po segmentácii a detekcii. Obrázok objektu sme náhodne rotovali a škálovali stokrát pri použití bikubickej interpolácie. Uhol náhodnej rotácie sa pohyboval v rozmedzí 1 – 359 stupňov a škálovací faktor mal veľkosť v rozmedzí od 0,3 do 5. Po každej transformácii boli v transformovanom obrázku vypočítané Fourierové deskriptory a hodnoty pre prvý, druhý, štvrtý a piaty sú znázornené na Obrázku 5.3.

Ideálne by sa hodnoty deskriptorov mali pri použitých transformáciách meniť čo najmenej a znázornené hodnoty v grafe by mali pripomínať priamky. Vo vzniknutom Obrázku 5.3 však môžeme pozorovať



Obr. 5.3: Invariantnosť deskriptorov číslo 1, 2, 4, 5 voči náhodným transformáciám objektu, ktorými je rotácia a škálovanie

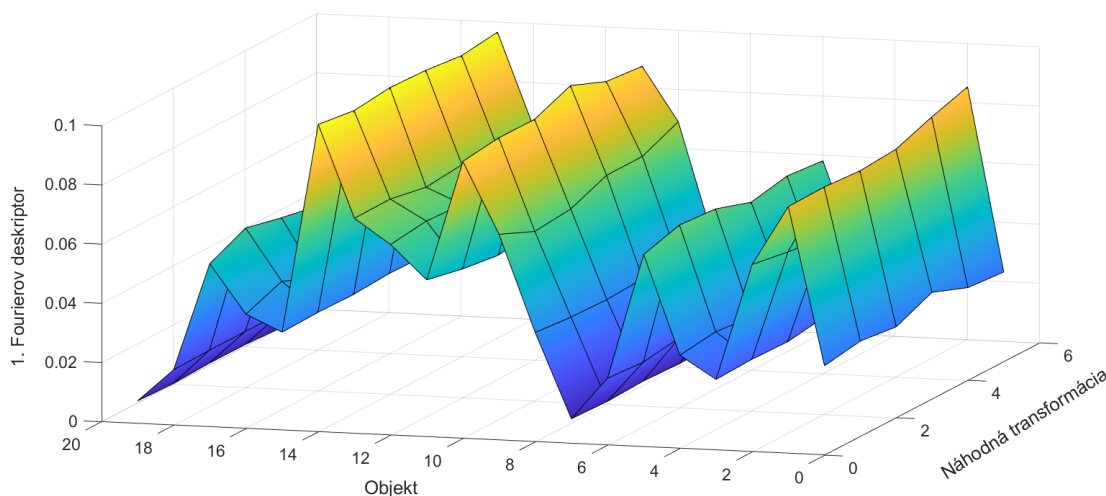
mierne fluktuácie hodnôt. Toto kolísanie hodnôt mohlo vzniknúť v dôsledku prevzorkovacích chýb pri transformácii, ktoré môžu zmeniť tvar hranice z ktorej deskriptory počítame. Ďalej vznikali chyby pri vysokých hodnotách škálovacieho faktoru, ktoré slúžili na otestovanie invariantnosti aj pri väčších zmenách mierky, ale v samotných obrázkoch k tejto degenerácii nedochádza. Môžeme teda predpokladať, že pre účely tejto práce sú deskriptory dostatočne nezávislé na použitých transformáciách.

## 5.4 Invariantnosť a rozlišovacia schopnosť deskriptorov

Ako už bolo spomínané v Kapitole 3, je dôležité, aby invarianty mali zároveň dobrú rozlišovaciu schopnosť. Na Fourierových deskriptoroch preto chceme ukázať, že majú dostatočne odlišné hodnoty pre rôzne objekty a zároveň sú invariantné voči rotácii, posunu a zmene mierky.

Aby sme túto vlastnosť otestovali, zvolili sme prvý Fourierov deskriptor. Tento deskriptor sme spočítali vždy pre jeden rovnaký objekt, ktorý sme dostali segmentáciou a detekciou z obrázku karty. Objekt sme šesťkrát náhodne otáčali v rozmedzí 1 – 359 stupňov a škálovali faktorom o náhodnej hodnote medzi 0,3 až 5. Takýto postup sa opakoval pre dvadsať rôznych objektov a výsledné hodnoty prvého deskriptora sú znázornené na Obrázku 5.4.

V ideálnom prípade požadujeme čo najmenšie fluktuácie hodnôt pri náhodných transformáciách a čo najväčšie rozdiely hodnôt pre rôzne objekty. Výsledný graf približne odpovedá týmto požiadavkám. Kolísanie hodnôt pri transformáciách je len mierne a hodnoty pre rôzne objekty sú dostatočne odlišné. Najmenší rozdiel hodnôt deskriptora pozorujeme pre objekt číslo dvadsať a objekt číslo osem (tvar slnka a tvar snehovej vločky). Táto podobnosť hodnôt môže byť spôsobená tým, že prvý Fourierov deskriptor popisuje celkový tvar hranice, ktorý majú tieto symboly dostatočne podobný. Keďže pri porovnávaní používame viacero príznakov, nielen prvý, malý rozdiel medzi hodnotami pre tieto dva objekty je indierentný.



Obr. 5.4: Invariantnosť a rozlišovacia schopnosť prvého Fourierového deskriptoru

## 5.5 Citlivosť na šum

Ďalšou skúmanou vlastnosťou je citlivosť na šum. Za šum považujeme náhodné degradácie obrazu, ktoré môžu vzniknúť napríklad pri jeho získavaní, prenose alebo spracovaní. Na to, aby rozpoznávanie pomocou deskriptorov fungovalo pre obrázky vytvorené za rôznych podmienok, mali by deskriptory byť voči nemu čo najmenej citlivé. Obraz, ktorý bol použitý na tento experiment vznikol skoro v ideálnom prostredí a obsahuje minimum šumu. Preto ho budeme pridávať a skúmať chovanie deskriptorov.

Aditívny Gaussovský šum je dobrou aproximáciou reálneho šumu vyskytujúceho sa v praxi [13]. Hodnoty jeho intenzity v danom pixely sú náhodné veličiny s hustotou pravdepodobnosti

$$p(x) = \frac{1}{\sigma \sqrt{2\pi}} e^{-\frac{x^2}{2\sigma^2}},$$

kde  $\sigma$  značí rozptyl. Miera šumu v obraze sa určuje v decibeloch pomocou SNR (signal-to-noise ratio) ako:

$$SNR = 10 \log_{10} \left( \frac{D(f)}{D(n)} \right).$$

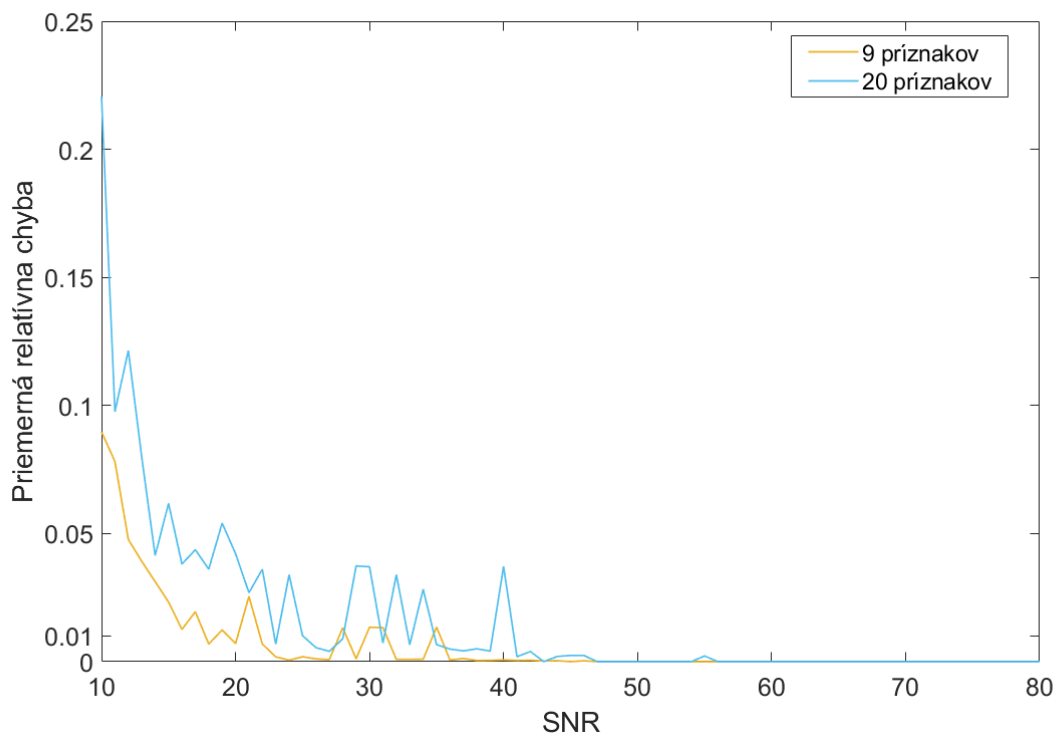
$D(f)$  je rozptyl signálu bez šumu,  $D(n)$  rozptyl pre šum [7].

Do každého kanálu farebného obrázku objektu sme postupne pridávali aditívny Gaussovský šum 70-krát v rozmedzí SNR od 10 dB (najviac šumu) do 80 dB (najmenej šumu). Následne sme obraz segmentovali, detegovali v ňom objekt a počítali pre neho príznakový vektor. Vektor sa skladal z deviatich (pri opakovaní pre porovnanie aj z dvadsiatich) príznakov vrátane priemernej farby pre každý farebný kanál. Príznačky sme porovnávali s príznakmi objektu v obraze bez šumu a počítali priemernú relatívnu chybu ako priemer z hodnôt relatívnej chyby vypočítaných pomocou rovnice:

$$\text{relatívna chyba} = \frac{|\text{príznačky po pridaní šumu} - \text{príznačky z databázy}|}{|\text{príznačky z databázy}|}.$$

Výsledný graf znázorňujúci túto chybu v závislosti na SNR je pre počet deskriptorov rovný dvadsať a deväť zobrazený v Obrázku 5.5. Priemerná relatívna chyba je v oboch prípadoch vysoká pri nízkych hodnotách SNR. Graf, na ktorého výpočet bolo použitých deväť deskriptorov dosahuje chyba okolo

1% pre hodnoty SNR vyššie ako 22 dB, ktoré odpovedajú relatívne nízkemu množstvu šumu. Použité príznaky sú teda citlivé na šum a metóda by za horších podmienok získavania obrazu nefungovala. Táto citlivosť je pravdepodobne spôsobená faktom, že deskriptory vznikajú transformáciou hranice, ktorá sa aplikáciou šumu bude meniť. Ďalej si na Obrázku 5.5 môžeme všimnúť, že deskriptory vyšších rádov sú na šum podstatne citlivejšie, čo je spôsobené tým, že popisujú detaily hranice.



Obr. 5.5: Citlivosť deskriptorov na šum

## 5.6 Úspešnosť rozpoznávania v závislosti na počte deskriptorov

Použitým množstvom deskriptorov zvyšujeme alebo znižujeme ich rozpoznávaciu schopnosť. Preto je dôležité zistiť, pre aký počet deskriptorov počítač identifikuje objekty čo najpresnejšie.

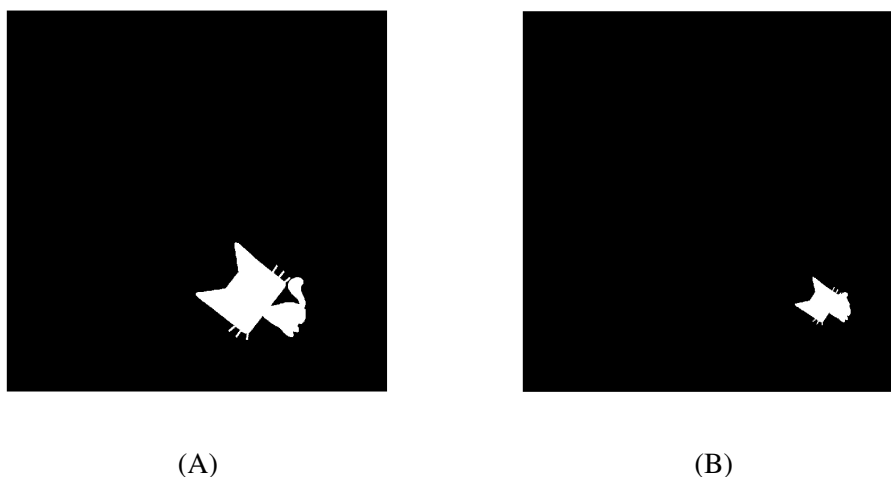
Na to, aby sme mohli úspešnosť vyhodnocovať, sme si vytvoril databázu príznakov pre objekty. V tejto štruktúre sa nachádzajú príznakové vektory pre každý skúmaný objekt. Pre jednotlivé obrázky sme teda vypočítali príznakové vektory objektov a porovnávali ich s tými, ktoré boli uložené v databáze. Podľa toho sme pomenovali jednotlivé objekty na obrázku. Úspešnosť sme vyhodnocovali porovnávaním získaných názvov a ozajstných názvov objektov v obraze. Následne sme zmenili počet používaných deskriptorov a opakovali rovnaký proces. Počet chybné identifikovaných objektov v závislosti na meniacom sa počte príznakov (kde prvé tri príznaky sú príznaky priemernej farby) sme zaznamenali do Tabuľky 5.1.

Pre menší počet príznakov vznikol pri identifikácii väčší počet chýb, čo sa dalo očakávať. Pre deväť až päťdesiatosem príznakov však počet chybné identifikovaných objektov ostal rovný jednej. Táto chyba pri väčšom počte príznakov bola spôsobená vždy jedným objektom z rovnakej karty. Jej príčinou bolo, že objekt bol na tejto karte príliš malý a dôsledkom toho sa počas segmentácie spojili časti objektu, ktoré

Počet príznakov	Počet chýb
4	36
5	20
6	9
7	6
8	2
9 a viac	1

Tabuľka 5.1: Počet chýb rozpoznávania objektov v závislosti na počte príznakov

boli na všetkých zvyšných kartách a teda aj na objekte uloženom v databáze oddelené (vid' Obrázok 5.6). Hranica tohto objektu mala potom na danej karte iný tvar než jej zástupca v databáze.



Obr. 5.6: Tvar objektu uloženého v databáze (A) a tvar objektu, ktorý spôsoboval chybu pri identifikácii (B)

Z výsledkov je zrejmé, že nižší počet ako šesť Fourierových deskriptorov vykazuje viditeľne horšie výsledky. Zvyšovaním počtu týchto deskriptorov však rastie aj čas výpočtu, na čo musíme pri voľbe dĺžky príznakového vektora dbať. Výsledky tohto experimentu naznačujú, že pre počet deskriptorov vyšší ako šesť sa pridávaním ďalších rozpoznávacích schopnosť nijako výrazne nezlepšuje.

## 5.7 Hľadanie spoločného objektu dvoch kariet

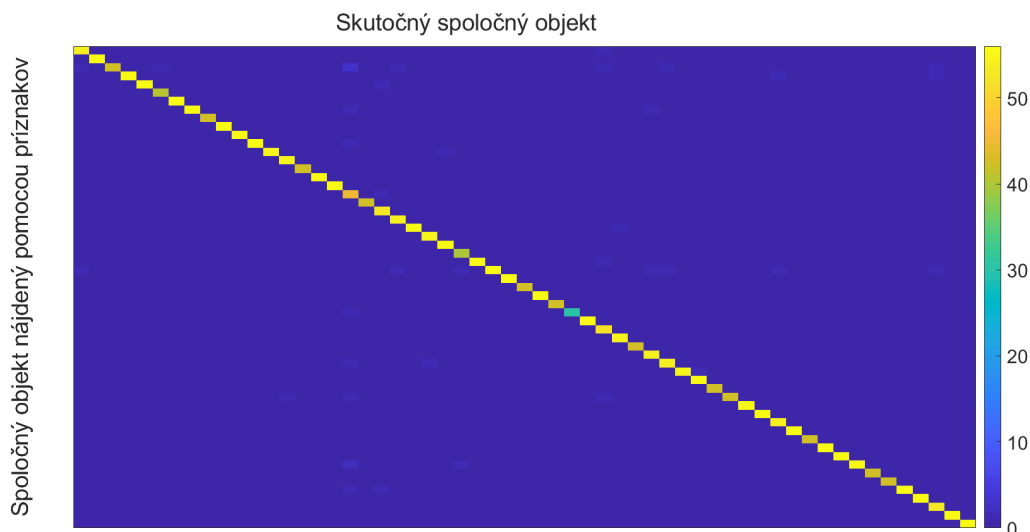
Ako bolo spomínané v Kapitole 4 cieľom hry Dobble je nájdenie spoločného objektu dvoch kariet. Preto zisťujeme úspešnosť deskriptorov pri hľadaní tohto objektu medzi dvoma obrázkami kariet.

Úspešnosť sme vyhodnocovali porovnávaním obrázku každej naskenovanej karty so všetkými ostatnými kartami z balíka, ktoré boli náhodne otáčané v rozmedzí 1 – 359 stupňov. Ako vektor príznakov sme použili priemernú farbu všetkých troch farebných kanálov a prvých šesť Fourierových deskriptorov. Takýmto spôsobom bolo porovnávanie prevedené spolu 2790-krát, z čoho bolo zaznamenaných 39 nesprávne identifikovaných objektov.

Obrázok 5.7 zobrazuje maticu zámen pri hľadaní správneho spoločného objektu dvoch kariet. Na diagonále sa nachádza počet správne klasifikovaných spoločných objektov a mimo diagonály počet tých, ktoré boli určené nesprávne. Na obrázku vidíme, že nastáva takmer ideálny prípad, kedy je matica



skoro diagonálna. Objekt, ktorý bol nájdený správne najmenej krát bol symbol snehuliaka. Dôvodom je skutočnosť, že (ako bolo spomínané v Kapitole 4) v hre chýbajú dve karty a snehuliak sa mal nachádzať na oboch z nich. Preto je tento symbol spoločným objektom pre menšie množstvo kariet.



Obr. 5.7: Matica zámen (confusion matrix) pre hľadanie spoločného objektu kariet

Účinnosť metódy pri klasifikácii určujú metriky ako je presnosť (precision), úplnosť (recall) a špecificita (specificity). Vypočítame ich z matice zámen  $C$  pomocou množstva skutočne pozitívnych (TP), skutočne negatívnych (TN), falošne pozitívnych (FP), falošne negatívnych (FN) klasifikácii pre danú triedu  $m$  [8], ktoré sú určené rovnicami:

$$\begin{aligned} TP_m &= C_{m,m} \\ TN_m &= \sum_{i=1, i \neq m}^k \sum_{j=1, j \neq m}^k C_{i,j} \\ FP_m &= \sum_{i=1, i \neq m}^k C_{m,i} \\ FN_m &= \sum_{i=1, i \neq m}^k C_{i,m} \end{aligned}$$

Presnosť definujeme ako:

$$\text{Presnosť} = \frac{TP_m}{FP_m + TP_m}$$

a určuje aká časť z objektov priradených do danej triedy skutočne do tejto triedy patrila. Úplnosť (alebo senzitivita) vyjadruje efektívnosť pri určovaní správnej triedy a vypočíta sa ako:

$$\text{Úplnosť} = \frac{TP_m}{FN_m + TP_m}.$$

Špecificita je daná rovnicou:

$$\text{Špecificita} = \frac{TN_m}{FP_m + TN_m}$$

a určuje, ako efektívne klasifikátor vyhodnotí, že daný objekt nepatrí do určitej triedy [10].

Z matice zámen, ktorá je znázornená na Obrázku 5.7 sme ďalej vypočítali priemernú presnosť, úplnosť a špecificitu, ich minimá, maximá a rozptyly. Tieto hodnoty sú zaznamenané v Tabuľke 5.7.

Metrika	Priemer [%]	Maximum [%]	Minimum [%]	Rozptyl [%]
Presnosť	98,73	100	82,35	0,089
Úplnosť	98,74	100	80,36	0,0916
Špecificita	99,98	100	99,69	0,00003

Tabuľka 5.2: Tabuľka vyhodnotenia úspešnosti hľadania spoločného objektu dvoch kariet

Správne klasifikovaných bolo 98,6 % objektov, priemerná presnosť klasifikácie 98,73 %, úplnosť nadobúdala v priemere hodnotu 98,74 % a špecificita 99,98 %. Všetky metriky majú maximálnu hodnotu 100 % a ich minimálne hodnoty neklesajú pre žiadnu triedu pod 80 %. Tieto výsledky značia relatívne vysokú úspešnosť rozpoznávania spoločných objektov dvoch kariet, z čoho môžeme usúdiť, že metóda bola pre danú úlohu vhodne zvolená.

## 5.8 Diskusia

Testovaním sme overili funkčnosť zvolenej metódy popisu objektov pri rozpoznávaní symbolov kariet. Výsledky dokázali, že príznaky sú do určitej miery nezávislé na rotácii, posune, zmene mierky a ich hodnoty sú odlišné pre rôzne objekty. Testovaný bol optimálny počet deskriptorov pre popis objektov. Výsledky naznačujú, že pre počet deskriptorov vyšší ako šesť sa rozpoznávacía schopnosť výrazne nezvyšuje. Za daných podmienok dosahovala metóda výborné výsledky pri rozpoznávaní spoločných objektov dvoch kariet. Test citlivosti na šum však zaznamenal vysoké chyby pri výpočte príznakov aj v obraze s nízkymi hodnotami pridaného šumu.

Metóda bola vhodná pre danú úlohu, kedy snímky boli zaznamenané v ideálnych podmienkach. Ak by sa však snímanie kariet uskutočnilo iným spôsobom predpokladáme, že metóda nebude fungovať. Ako ukázal test citlivosti na šum, aj pri malých nedostatkoch v obraze je chyba pri výpočte príznakov relatívne vysoká. Rovnako by príznaky nefungovali, ak by karty boli snímané z rôznych uhlov pohľadu, čo by spôsobilo ďalšie možné transformácie objektov voči ktorým nie sú deskriptory invariantné.

# Záver

V tejto práci som sa zaoberala rozpoznávaním objektov v obraze využitím invariantných príznakov. Cieľom bolo nájsť vhodnú metódu pre rozpoznávanie objektov s neznámou orientáciou a veľkosťou a jej účinnosť otestovať na snímkach kariet hry Dobble.

Skôr ako došlo k samotnému výpočtu príznakov, bola prevedená segmentácia pomocou manuálneho globálneho prahovania a následná detekcia objektov. Objekty som popísala pomocou Fourierových deskriptorov, ktoré vznikli Fourierovou transformáciou hranice a príznakov priemernej farby objektu, ktoré boli pridané kvôli výskytu troch objektov s rovnakým tvarom hranice. Ďalej som použila klasifikáciu podľa vzdialenosti.

Testovala sa invariantnosť Fourierových deskriptorov voči rotácii, posunu a zmene mierky. Výsledky ukázali, že sú do určitej miery invariantné voči týmto transformáciám. Hľadaný bol aj optimálny počet deskriptorov pre rozpoznávanie. Z experimentu vyplynulo, že pre počet príznakov väčší ako deväť sa rozpoznávacía schopnosť už výrazne nezvyšuje. Pri hľadaní spoločného objektu dvoch kariet dosahovala metóda úspešnosť 98,6 %. Analýza citlivosti na šum ukázala pomerne vysokú senzitivitu deskriptorov voči týmto degeneráciám.

Metóda bola testovaná na snímkach, ktoré vznikli v ideálnych podmienkach. Pre úlohu rozpoznávania objektov v obraze získanom týmto spôsobom boli zvolené príznaky efektívne. Predpokladáme však, že za menej ideálnych podmienok by použitá metóda nefungovala. Ak by boli snímky zhotovené z iného uhlu pohľadu, transformácie objektov by zahŕňali napríklad aj nerovnomerné škálovanie, voči ktorému nie sú deskriptory invariantné. Ďalší problém by tvorila ich citlivosť na šum, ktorá spôsobuje, že pri výskyte týchto degenerácií v obraze by vznikali chyby v ich výpočte.

Ak by sme chceli simulovať hru Dobble v reálnom živote, karty by mali byť snímané z uhla pohľadu hráča. Rozpoznávanie spoločných objektov by za takýchto podmienok vyžadovalo voľbu príznakov, ktoré sú invariantné voči afinným transformáciám a odolnejšie voči šumu. Na tieto účely by mohli slúžiť napríklad momentové invarianty, semi-diferenciálne invarianty, metóda SIFT alebo SURF. Úloha by bola rovnako riešiteľná pomocou konvolučných neurónových sietí. Ďalej by bolo možné skúmať reakčné časy ľudí pri hraní hry a podľa ich štatistického rozloženia nastaviť počítač tak, aby bola hra vyrovnaná. Pre zjednodušenie experimentu by sa dalo generovať databázu objektov automaticky pomocou algoritmu k-means.

# Literatúra

- [1] *combinatorics* - What is the math behind the game Spot It? Mathematics Stack Exchange. URL: <https://math.stackexchange.com/questions/36798/what-is-the-math-behind-the-game-spot-it> (cit. 15.05.2021).
- [2] Micky Dore. *Dobble — Theory and Implementation*. Medium. 19.feb. 2021. URL: <https://mickydore.medium.com/dobble-theory-and-implementation-ff21ddb5318> (cit. 15.05.2021).
- [3] Jan Flusser, Tomáš Suk a Barbara Zitová. *2D and 3D Image Analysis by Moments*. English. 1st edition. Chichester, West Sussex, United Kingdom ; Hoboken, NJ: Wiley, dec. 2016. ISBN: 978-1-119-03935-8.
- [4] *Fourier Descriptors*. Wolfram Demonstrations Project. URL: <http://demonstrations.wolfram.com/FourierDescriptors/> (cit. 05.07.2021).
- [5] Rafael Gonzalez a Richard Woods. *Digital Image Processing*. English. 4th edition. New York, NY: Pearson, mar. 2017. ISBN: 978-0-13-335672-4.
- [6] Ardeshir Goshtasby. “Description and Discrimination of Planar Shapes Using Shape Matrices”. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* PAMI-7.6 (1985), s. 738–743.
- [7] Don H Johnson. “Signal-to-noise ratio”. In: *Scholarpedia* 1.12 (2006), s. 2088.
- [8] Thomas Kautz, Bjoern M. Eskofier a Cristian F. Pasluosta. “Generic performance measure for multiclass-classifiers”. In: *Pattern Recognition* 68 (1. aug. 2017), s. 111–125. ISSN: 0031-3203. URL: <https://www.sciencedirect.com/science/article/pii/S0031320317301073>.
- [9] Mark Nixon. *Feature Extraction and Image Processing for Computer Vision*. English. 3rd edition. Oxford: Academic Press, okt. 2012. ISBN: 978-0-12-396549-3.
- [10] Marina Sokolova a Guy Lapalme. “A systematic analysis of performance measures for classification tasks”. In: *Information Processing & Management* 45.4 (), s. 427–437. ISSN: 03064573. URL: <https://linkinghub.elsevier.com/retrieve/pii/S0306457309000259>.
- [11] Chris Solomon a Toby Breckon. *Fundamentals of Digital Image Processing: A practical approach with examples in Matlab*. John Wiley & Sons, 2011.
- [12] Petr Stehlík. “Matematika za karetní hrou dobbble”. In: *Pokroky matematiky, fyziky a astronomie* 64.2 (2019), s. 69–90.
- [13] Milan Šonka, Václav Hlaváč a Roger Boyle. *Image Processing, Analysis, and Machine Vision*. English. 4th edition. Stamford, CT, USA: Cengage Learning, jan. 2014. ISBN: 978-1-133-59360-7.

- [14] Jianzhong Zhou et al. “Shaft orbit identification for rotating machinery based on statistical fuzzy vector chain code and support vector machine”. In: *Journal of Vibroengineering* 16.2 (2014). Number: 2 Publisher: JVE International Ltd., s. 713–724. ISSN: 1392-8716. URL: <https://www.jvejournals.com/article/14874>.
- [15] Zygomatic. *Spot it!* URL: <https://www.spotitgame.com/> (cit. 11.05.2021).