

PŘÍRODOVĚDECKÁ FAKULTA UNIVERZITY PALACKÉHO
KATEDRA INFORMATIKY

BAKALÁŘSKÁ PRÁCE

Zpracování a rozpoznávání obrazu



2011

Václav Adamec

Anotace

Tato bakalářská práce se zabývá zpracováním a rozpoznáváním obrazu. Jsou zde představeny základní pojmy, metody a postupy z teorie zpracování obrazu a principy rozpoznávání objektů v obraze. Dále se práce věnuje rozboru a implementaci aplikace, která má za úkol nalézt a rozpoznat státní poznávací značku osobních vozidel. Součástí jsou i výsledky testování této aplikace na zkušební sadě fotografií.

Rád bych poděkoval Mgr. Eduardu Bartlovi, Ph.D. zejména za to, že se ujal vedení již rozpracované práce a pomohl mi ji tak dostat do konečné podoby. Dále děkuji svoji přítelkyni za podporu a trpělivost po celou dobu studia.

Obsah

1. Úvod	8
2. Zpracování obrazu	9
2.1. Obraz	9
2.2. Barevné modely	9
2.2.1. RGB	9
2.2.2. CMY a CMYK	10
2.2.3. CIELAB (CIE L*a*b*)	11
2.3. Snímání a digitalizace obrazu	12
2.3.1. Vzorkování	12
2.3.2. Kvantování	13
2.4. Předzpracování	13
2.4.1. Jasové transformace	13
2.4.2. Geometrické transformace	14
2.4.3. Filtrace a ostření	16
2.4.4. Matematická morfologie	18
2.5. Segmentace (nalezení objektů)	19
2.5.1. Prahování	20
2.5.2. Spojování oblastí	20
2.5.3. Dělení oblastí	20
2.5.4. Detekce hran	20
2.5.5. Srovnání se vzorem	24
2.6. Popis	24
2.6.1. Některé základní vlastnosti	24
2.7. Klasifikace	27
3. Rozbor úlohy	28
3.1. Státní poznávací značka silničního motorového vozidla a přípoj- ného vozidla	29
3.2. Cíle a vstupní předpoklady	30
4. Implementace	30
4.1. Fotografie a jejich předzpracování	31
4.2. Nalezení oblasti SPZ	31
4.3. Segmentace SPZ	32
4.4. Segmentace znaků	33
4.5. Klasifikace znaků	33
4.6. Výsledná aplikace	34
4.7. Výsledky testování	35
4.8. Shrnutí a hodnocení	35

Závěr	38
Conclusions	39
Reference	40
A. Tabulka nalezení a rozpoznání SPZ	41
B. Obsah přiloženého CD	43

Seznam obrázků

1.	Model RGB.	10
2.	Model CMYK.	10
3.	Model CIELAB.	11
4.	Chromatický diagram CIE 1931.	11
5.	Čtvercová a hexagonální mřížka.	12
6.	Značka a její histogram.	14
7.	Princip konvoluce.	17
8.	Gaussovo rozdělení.	17
9.	Dilatace.	18
10.	Eroze.	19
11.	2-D funkce Laplacian of Gaussian.	23
12.	Příklad kompaktního a nekompaktního objektu.	25
13.	Označení směrů pro 4-okolí a 8-okolí.	26
14.	Číslice „2“ a její řetězový kód.	26
15.	Schématické znázornění klasifikace.	27
16.	Proces rozpoznání SPZ.	28
17.	Standardní registrační značka.	30
18.	Šprávně lokalizovaná oblast SPZ.	32
19.	Univerzální maska SPZ.	33
20.	Prahovaná SPZ a extrahované znaky.	33
21.	Hledání průsečíků.	34
22.	Okno aplikace s rozpoznanou SPZ.	37

Seznam tabulek

1.	Úspěšnost nalezení SPZ.	35
2.	Úspěšnost rozpoznávání SPZ	35
3.	Tabulka nalezení a rozpoznání SPZ.	41
4.	Tabulka nalezení a rozpoznání SPZ (pokračování).	42

1. Úvod

S aplikacemi zpracovávající obraz se můžeme setkat v celé řadě odvětví lidské činnosti. Nalézají uplatnění v průmyslu při kontrole kvality (defektoskopie), v medicíně při automatickém vyhodnocování lékařských snímků, při předpovídání počasí a kartografii u družicových snímků, ve vojenství, v dopravě atd. Hlavní motivací pro zavádění těchto systémů je zautomatizování a zefektivnění práce s obrazem. Dříve bylo limitujícím faktorem velké množství dat a náročnost algoritmů při zpracování, ale s možnostmi dnešních technických zařízení a neustálým vývojem v oblasti zpracování obrazu se tato omezení ztrácejí. S podobnými aplikacemi se setkáváme na každém kroku stále častěji, často aniž bychom o tom měli vůbec tušení. Příkladem může být detekce obličeje nebo úsměvu, kterou již považujeme u běžný fotoaparátů za samozřejmost. Tato práce si klade za cíl nahlédnout do složité a obsáhlé problematiky zpracování obrazu a vytvořením jednoduché aplikace demonstrovat použití metod a postupů zpracování obrazu na konkrétním případě.

Práci lze rozdělit do dvou částí. V první jsou představeny základní pojmy a metody, které jsou využívány při práci s obrazem. Tato teoretická část je obsažena v kapitole 2. a popisuje pět základních fází, na které proces zpracování obrazu můžeme rozdělit:

- snímání a digitalizace,
- předzpracování,
- segmentace,
- popis,
- klasifikace.

Další část textu popisuje řešení konkrétní úlohy, kterou je lokalizace a rozpoznání (převedení do textové podoby) státní poznávací značky z fotografie vozidla. Tato úloha může být vhodným příkladem podobných aplikací, se kterými se setkáváme v běžném životě. Kapitola 3. obsahuje seznámení s předpisy a zákony řešící vzhled a používání SPZ v České Republice a seznamuje se základními problémy, které je nutné brát v potaz při snaze o jejich rozpoznávání. Kapitola 4. popisuje metody a konkrétní postupy použité při řešení dílčích kroků vytvářeného algoritmu. Kapitola 4.8. pak shrnuje poznatky získané při implementaci a testování aplikace na zkušební sadě fotografií a částečně poukazuje na možné způsoby řešení odhalených problémů. Součástí je i hodnocení dosažených výsledky s ohledem na možné uplatnění v praxi.

2. Zpracování obrazu

2.1. Obraz

Pojem obraz lze matematicky popsat jako spojitou funkci dvou nebo tří argumentů, ve tvaru [12]:

$$f = (x, y) \text{ případně } f = (x, y, z).$$

Tuto funkci nazýváme obrazovou funkcí, kde její proměnné x , y jsou reálná čísla, představující souřadnice v rovině, proměnná t může sloužit pro popis obrazu měnícího se v čase a hodnotou funkce je měřená fyzikální veličina. Nejčastější měřenou veličinou je jas, ale mohou to být i jiné veličiny, např. teplota apod. Hodnota obrazové funkce také nemusí být reprezentována pouze jedinou hodnotou. Například v případě použití barevného modelu RGB to budou hodnoty tři, pro každou barevnou složku jedna.

2.2. Barevné modely

Při vytváření této kapitoly jsem čerpal z [13]. Barevné modely vycházejí z chromatického diagramu CIE, který zobrazuje všechny barvy viditelného spektra. „*Barva je vlastnost objektů spojená s jejich schopností odrážet elektromagnetické vlnění různých vlnových délek.*“ [6] Model, který by popisoval všechny barvy viditelného spektra by byl velmi složitý, proto se omezujeme na modely, které jsou založeny na „míchání“ základních (většinou tří) barev. Tyto barvy tvoří v chromatickém diagramu trojúhelník, tzv. gamut, který zobrazuje škálu barev, které lze daným modelem popsat. Níže uvedené základní modely RGB a CMY jsou závislé na zařízení. Lidskému vnímání jsou přirozenější modely oddělující jasovou a barevnou složku. Oko je citlivější na změnu jasu než na změnu barvy. Mezi tyto modely patří například model HSV reprezentovaný šestibokým jehlanem. Dalším zde uvedeným modelem bude model CIELAB, ale barevných modelů je celá řada. Některé nacházejí své uplatnění v polygrafii, jiné např. při přenosu televizního signálu atd.

2.2.1. RGB

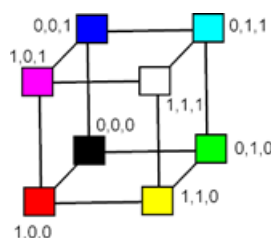
RGB model je aditivní model, jehož základní barvy jsou červená (red), zelená (green) a modrá (blue). Ostatní barvy vznikají kombinací těchto tří barev. Tento model vychází z citlivosti oka na vlnové délky světla zhruba odpovídající těmto barvám. Jeho využití najdeme především u monitorů.

Ve výpočetní technice se u modelu RGB standardně používá osmibitová reprezentace každé barvy, to znamená, že dokáže rozlišit $2^8(256)$ úrovní pro každou barvu. Celkově tedy máme k dispozici 2^{24} kombinací, což představuje 16777216

barev. Tomuto modelu říkáme True Colors. Pokud barevnou informaci nepotřebujeme udržovat, je vhodné převést obraz do odstínů šedé. Každý pixel pak v paměti zabere místo tří jen jeden byte a obraz přitom neztratí nic ze své hodnoty. Vzorrec pro převod barevného obrazu do odstínů šedé v sobě odráží odlišnou citlivost lidského oka na jednotlivé barvy, proto má každá složka při převodu jinou váhu:

$$f = 0,299R + 0,587G + 0,114B.$$

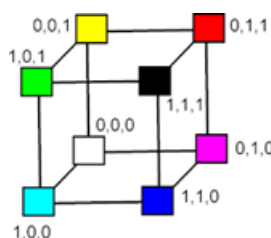
R představuje červenou složku barvy, G zelenou složku a B modrou složku.



Obrázek 1. Model RGB (zdroj: www.dmp.spsei.cz/digi/model.php).

2.2.2. CMY a CMYK

Základními barvami modelu CMY(K) jsou azurová (cyan), purpurová (magenta) a žlutá (yellow). Kombinací těchto tří barev však nevznikne dokonalá černá, proto byla k tomuto modelu přidána jako čtvrtá barva právě černá (CMYK). Jde o substraktivní model využívaný pro tisk.



Obrázek 2. Model CMYK (zdroj: www.dmp.spsei.cz/digi/model.php).

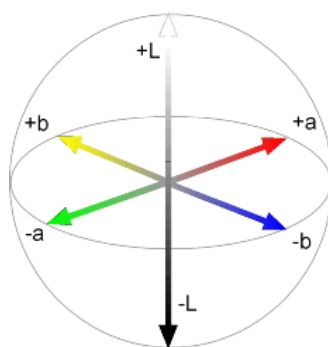
Převod mezi modely RGB a CMY lze provést podle následujícího vzorce. Nevýhodou je, že tyto modely nemají stejný gamut, jinými slovy nedokážou zobrazit stejné spektrum barev.

$$\begin{bmatrix} C \\ M \\ Y \end{bmatrix} = \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix} - \begin{bmatrix} R \\ G \\ B \end{bmatrix}.$$

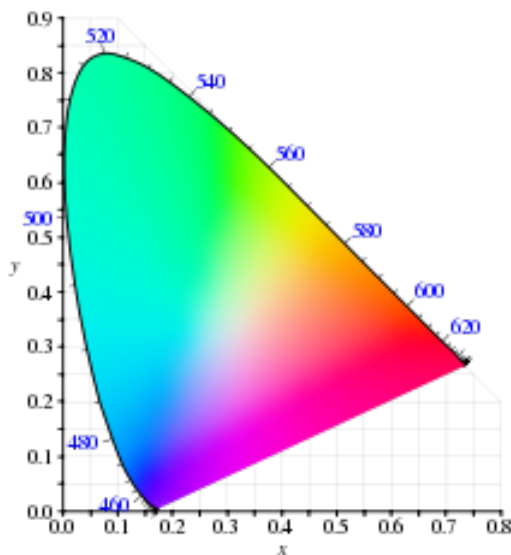
C představuje azurovou složku barvy, M purpurovou složku a Y žlutou složku v modelu CMY. R, G, B jsou složky v modelu RGB.

2.2.3. CIELAB (CIE $L^*a^*b^*$)

Chromatický diagram CIE definovaný v roce 1931 (obr.4.) je nerovnoměrný, proto byl v roce 1976 diagram modifikován a definován model CIELAB. Model je reprezentován koulí, jejíž svislá osa představuje jas a barvy jsou definovány na dvou horizontálních osách a a b . Osa a stanovuje hodnotu barvy mezi červenou a zelenou, osa b hodnotu mezi žlutou a modrou. Tento model je nezávislý na zařízení, může tak být použit jako referenční.



Obrázek 3. Model CIELAB (zdroj: www.codeproject.com).



Obrázek 4. Chromatický diagram CIE 1931 (zdroj: www.wikipedia.org).

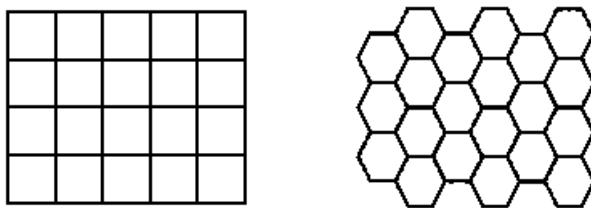
2.3. Snímání a digitalizace obrazu

Při vytváření této kapitoly jsem čerpal z [12]. Na pořizování snímku by jsme mohli z pohledu konečného zpracování nahlížet jako na okrajovou záležitost, opak je ale pravdou. Snímání je prvním krokem v posloupnosti činností zpracování obrazu, a jako takové má zásadní vliv na všechny kroky následující. Vhodně zvolenými podmínkami snímání, jako jsou vlastnosti snímacího zařízení, jeho poloha a osvětlení scény, lze pozitivně ovlivnit náročnost dalšího zpracování. Naopak nevhodně nasnímaný obraz a jeho špatná kvalita může být příčinou nezdaru celého procesu. Obraz také nemusíme pořizovat jen v části spektra viditelné člověku. Příkladem může být kamera snímající infračervené záření nebo termokamera pořizující teplotní záznam scény.

Po převodu optické veličiny na většinou spojitý elektrický signál, je pro další zpracování v počítači nutné tento signál digitalizovat. Digitalizaci rozumíme vzorkování spojitě obrazové funkce do matice $m \times n$ bodů (m je horizontální a n vertikální rozlišení) a kvantování do k jasových úrovní. Jinak řečeno převedeme analogovou obrazovou funkci na konečný počet obrazových bodů s konečným počtem jasových úrovní. Nejběžnější formou digitalizace snímků pro nás představují digitální fotoaparáty využívající CCD nebo CMOS snímače.

2.3.1. Vzorkování

Při vzorkování hraje důležitou roli vzorkovací interval. Při velké vzdálenosti vzorkovacích bodů může docházet ke ztrátě detailů. Aby k tomu nedocházelo, měl by být vzorkovací interval menší nebo roven polovině rozměru nejmenších detailů v původním obraze. Tuto skutečnost shrnuje Shannonova věta, která říká, že vzorkovací frekvence musí být alespoň dvakrát větší než nejvyšší frekvence vzorkovaného signálu. Každý vzorkovací bod představuje jeden obrazový bod (pixel) ve vzorkovací mřížce. Standardně se můžeme setkat s dvěma typy mřížek, s „klasickou“ čtvercovou a hexagonální, složenou z pravidelných šestiúhelníků.



Obrázek 5. Čtvercová a hexagonální mřížka.

Výhodou čtvercové mřížky je snadná implementace, u hexagonální mřížky zase snadněji určíme vzdálenosti, ale nemůžeme použít Furierovu transformaci.

Další variantou pro popis plochy obrazu je plocha složená z rovnostranných trojúhelníků.

2.3.2. Kvantování

Analogové hodnoty jasu obrazové funkce jsou kvantováním převedeny do diskrétního tvaru. Každé původní hodnotě jasu ve vzorkovacím bodě se přiřadí nejbližší kvantovací úroveň. Při rovnoměrně rozložených kvantovacích úrovních jde o uniformní kvantování, v opačném případě je kvantování neuniformní. Pokud bude kvantovacích úrovní málo, dojde k hrubému rozložení jasu v obraze a podstatnému snížení kvality rozlišení jasových úrovní. Lidské oko je schopno rozlišit přibližně padesát úrovní jasu, počet kvantovacích úrovní by tedy neměl být menší než tato hodnota.

$$k = 2^b,$$

kde k je počet jasových úrovní a b je počet bitů reprezentujících jeden obrazový bod. Při osmibitové reprezentaci jednoho pixelu dostaneme 256 jasových úrovní.

2.4. Předzpracování

Při vytváření této kapitoly jsem čerpal z [12],[1],[4]. Obraz je ve většině případů při snímání a následné digitalizaci nějakým způsobem „poškozen“. A právě tato poškození jako je šum nebo zkreslení se snaží odstranit metody předzpracování obrazu. Cílem je tedy opětovně zvýšit kvalitu obrazu zejména z hlediska dalšího zpracování. To znamená potlačit některé informace a naopak zvýraznit ty rysy obrazu, které jsou pro nás dále důležité a budeme s nimi v dalších krocích zpracování pracovat. Při tomto procesu je využito nadbytečné informace obsažené v obraze.

2.4.1. Jasové transformace

Jasové transformace patří mezi nejjednodušší transformace předzpracování obrazu. Tyto transformace provádějí operace s hodnotami jasu jednotlivých pixelů tak, že každému pixelu přiřadí novou hodnotu jasu podle transformační funkce. Jednoduchým příkladem může být negativ:

$$g(x, y) = 1 - f(x, y).$$

Dále v textu budou uvedeny některé další jasové transformace, nejdříve však objasníme pojem histogram.

Histogram je graf znázorňující četnost zastoupení jednotlivých jasových úrovní v obraze. Horizontální osa představuje jasové úrovně a vertikální počet pixelů v obraze, které mají odpovídající hodnotu jasu. Je využíván pro opravu

špatně exponovaných snímků, nebo například při metodách automatického prahování. Histogram určitým způsobem popisuje snímek, ale nemá výpovědní hodnotu z hlediska rozlišitelnosti snímku. Každý snímek má jen jeden histogram, ale přitom více snímků může mít histogram stejný. Nemůžeme tedy dva snímky vzájemně rozlišit pouze na základě znalosti jejich histogramu. Matematicky jej lze definovat následujícím vzorcem:

$$N = \sum_{k=0}^l h_k,$$

kde h_k je počet pixelů s úrovní jasu k a k až l je rozsah jasu.



Obrázek 6. Značka a její histogram.

Ekvalizace histogramu je transformace, jejíž snahou je rovnoměrnější zastoupení jednotlivých jasových úrovní v obraze. Můžeme tak docílit lokálního zvýšení kontrastu u jinak špatně rozlišitelných detailů v nevhodně exponovaném obraze.

$$g_k = \frac{1}{N} \sum_{j=0}^k n_j,$$

kde N je celkový počet pixelů a n_j je četnost j -té úrovně jasu.

2.4.2. Geometrické transformace

Geometrické transformace slouží pro cílené geometrické zkreslení obrazu nebo pro jeho opravu. Deformace objektů v obraze může být zapříčiněna nevhodnou polohou při snímání nebo nedokonalostí snímacího zařízení. Pro další práci s obrazem je někdy nutné provést jeho korekci. Základními geometrickými transformacemi jsou posunutí, změna měřítka a otočení. Výsledkem transformačních funkcí jsou souřadnice obrazového bodu, které nejsou celočíselné a je proto nutné je interpolací převést do diskrétní podoby. Nejjednodušší je interpolace nejbližším sousedem, sofistikovanějšími jsou pak bilineární či kubická interpolace.

Posunutí bodu A o vektor t je definováno vztahem:

$$\begin{aligned}x' &= x + t_x, \\y' &= y + t_y,\end{aligned}$$

kde x, y jsou souřadnice původního bodu a x', y' jsou souřadnice nového bodu posunutého o vektor t .

Přepsáním do homogenních souřadnic dostaneme maticový zápis:

$$\begin{bmatrix} x' & y' & 1 \end{bmatrix} = \begin{bmatrix} x & y & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ t_x & t_y & 1 \end{bmatrix}.$$

Maticový zápis umožňuje pohodlnější práci s transformacemi. Pro skládání transformací stačí vynásobit transformační matice jednotlivých transformací. Výslednou matici složené transformace pak jednoduše aplikujeme na jednotlivé body. Vyhneme se tak zbytečnému počítání jednotlivých transformací a místo toho provedeme výpočet transformace pouze jednou.

Změna měřítka s_x ve směru x a s_y ve směru y je definována vztahem:

$$\begin{aligned}x' &= x \cdot s_x, \\y' &= y \cdot s_y.\end{aligned}$$

Maticový zápis:

$$\begin{bmatrix} x' & y' & 1 \end{bmatrix} = \begin{bmatrix} x & y & 1 \end{bmatrix} \begin{bmatrix} s_x & 0 & 0 \\ 0 & s_y & 0 \\ 0 & 0 & 1 \end{bmatrix}.$$

Otočení kolem počátku o úhel φ je definováno vztahem:

$$\begin{aligned}x' &= x \cos \varphi - y \sin \varphi, \\y' &= x \sin \varphi + y \cos \varphi.\end{aligned}$$

Maticový zápis:

$$\begin{bmatrix} x' & y' & 1 \end{bmatrix} = \begin{bmatrix} x & y & 1 \end{bmatrix} \begin{bmatrix} \cos \varphi & \sin \varphi & 0 \\ -\sin \varphi & \cos \varphi & 0 \\ 0 & 0 & 1 \end{bmatrix}.$$

2.4.3. Filtrace a ostření

Před segmentací obrazu je vhodné, a ve většině případů i nutné, alespoň částečně zbavit obraz šumu. Šum se ve většině případů do obrazu dostává při jeho snímání a při přenosu. V obraze jsou pak zřetelné osamocené pixely, výrazně se lišící od svého okolí. Metody vyhlazování se snaží tyto extrémní obrazové body potlačit a nahradit je pravděpodobnější hodnotou, získanou na základě hodnot jasu bodů nacházejících se v okolí. Pro úplné odstranění šumu bychom museli znát původní obrazovou informaci, kterou v praxi většinou nemáme. Při vyhlazování šumu dochází ve většině případů i k nechtěnému jevu, rozmazání hran. Ty jsou totiž v obraze také charakterizovány výraznou změnou jasu, stejně jako šum. Naopak ostření, kdy se snažíme zvýraznit hrany, sebou nese zase i zvýraznění šumu.

Diskrétní konvoluce je matematická operace zpracovávající dvě funkce. V našem případě je první funkcí obrazová funkce (zpracováváný obraz) a tou druhou (označovanou jako jádro, nebo maska konvoluce) je filtr, který chceme na obraz aplikovat. Masku je zvykem zapisovat formou matice. Vhodnou volbou hodnot této matice docílíme u výsledného obrazu různých efektů. Diskrétní konvoluce je tedy vhodným prostředkem pro operace s obrazem jako je například vyhlazování nebo detekce hran.

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

Jak je vidět na obrázku 7. hodnoty masky udávají váhu jakou příslušné pixely v původním obraze ovlivňují výslednou hodnotu. Masku postupně posouváme po obraze a násobíme hodnoty pixelů v obraze a na příslušné pozici v masce. Následně vynásobené hodnoty sečteme. Aby nedocházelo k zesvětlování nebo ztmavování obrazu, je nutné, aby součet hodnot masky byl roven jedné. Proto musíme masku vynásobit tzv. koeficientem masky.

Vyhlazování průměrováním patří mezi lineární metody vyhlazování a někdy může vést i k přílišnému rozmazání hran. Pomocí konvoluce vypočítáme každému bodu v obraze novou hodnotu, která je průměrem hodnot jasů okolních bodů. Konvoluční jádro je rovno:

$$\frac{1}{9} \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}.$$

Pokud bychom použili větší masku, dojde k lepšímu vyhlazení, ale za cenu větší ztráty detailů. Lepší výsledky může mít maska, která přisuzuje bodům ve středu masky větší váhu. Omezíme tím rozmazání hran v obraze.



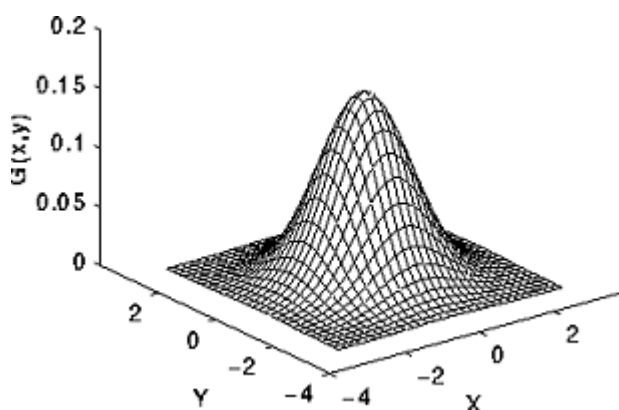
Obrázek 7. Princip konvoluce (zdroj: www.wikipedia.org).

Gaussovo vyhlazování je příkladem vyhlazování pomocí masky s různou vahou jednotlivých bodů. Konvoluční maska je vypočítána z funkce představující dvourozměrné Gaussovo rozdělení:

$$G(x, y) = \frac{1}{2\pi\sigma^2} e^{-\frac{x^2+y^2}{2\sigma^2}},$$

kde x, y jsou souřadnice obrazového bodu a σ je standardní odchylka (rozptyl). Konvoluční jádro je rovno:

$$\frac{1}{273} \begin{bmatrix} 1 & 4 & 7 & 4 & 1 \\ 4 & 16 & 26 & 16 & 4 \\ 7 & 26 & 41 & 26 & 7 \\ 4 & 16 & 26 & 16 & 4 \\ 1 & 4 & 7 & 4 & 1 \end{bmatrix}.$$



Obrázek 8. Gaussovo rozdělení se střední hodnotou (0,0) a $\sigma = 1$, [4].

Mediánová filtrace je metoda, kdy se nová hodnota obrazového bodu určí jako medián vybraných sousedních bodů. Medián je prostřední prvek v uspořádané posloupnosti hodnot. Je-li počet prvků sudý, určí se medián jako aritmetický průměr prostředních dvou prvků. Vyhlažování mediánovou filtrací patří mezi nelineární metody a přináší dobré výsledky u šumu typu „pepř a sůl“.

Ostření lze dosáhnout pomocí sečtení původního obrazu s jeho derivací. (Derivací obrazové funkce docílíme nalezení hran viz kapitola 2.5.4.) Není ale nutné počítat dvě transformace obrazu. Tuto operaci lze provést pomocí jediné masky. Zde je uvedeno konvoluční jádro pro superpozici obrazu a Laplaceova operátoru:

$$\begin{bmatrix} 0 & -1 & 0 \\ -1 & 5 & -1 \\ 0 & -1 & 0 \end{bmatrix}.$$

2.4.4. Matematická morfologie

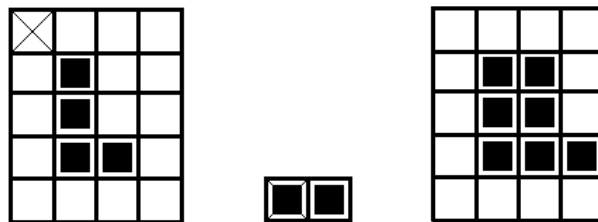
Autory této teorie jsou G. Matheron a J. Serra. Transformace matematické morfologie pracují s tvarem objektů a využíváme je nejen pro předzpracování obrazu, ale i pro jeho segmentaci. V binární matematické morfologii je obraz vyjádřen pomocí bodové množiny. Transformace jsou realizovány operacemi s bodovou množinou X (představující obraz) a menší bodovou množinou B (strukturním elementem). Základními transformacemi jsou dilatace a eroze, dalšími pak např. otevření a uzavření nebo transformace hit or mis.

Dilatace zvětšuje původní objekty a vyplňuje malé díry a zálivy. Je definována jako vektorový součet dvou bodových množin (Minkowského množinový součet).

$$X \oplus B = \{d \in E^2 : d = x + b, x \in X, b \in B\}.$$

Lze ji vyjádřit také jako sjednocení posunutých bodových množin.

$$X \oplus B = \bigcup_{b \in B} X_b.$$



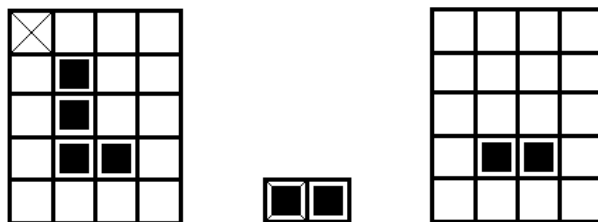
Obrázek 9. Příklad bodové množiny, strukturního elementu a výsledek dilatace.

Eroze zjednodušuje strukturu objektů. Objekty menší než strukturní element zmizí. Je definována jako vektorový rozdíl dvou bodových množin (Minkowského množinový rozdíl).

$$X \ominus B = \{d \in E^2 : d + b \in X \text{ pro všechna } b \in B\}.$$

Lze ji vyjádřit také jako průnik posunutých bodových množin.

$$X \ominus B = \bigcap_{b \in B} X_{-b}.$$



Obrázek 10. Příklad bodové množiny, strukturního elementu a výsledek eroze.

Otevření je eroze následovaná dilatací.

$$X \circ B = (X \ominus B) \oplus B.$$

Uzavření je dilatace následovaná erozí.

$$X \bullet B = (X \oplus B) \ominus B.$$

Otevření a uzavření se používá pro odstranění detailů při zachování tvaru objektů. Otevření rozděluje objekty spojené úzkými spoji. Uzavření tyto objekty naopak spojuje a vyplňuje malé díry, čímž může dojít k vyhlazení tvaru.

2.5. Segmentace (nalezení objektů)

Při vytváření této kapitoly jsem čerpal z [12],[1],[4]. Snahou segmentace je nalezení objektů, které nás v obraze zajímají a jejich rozlišení od pozadí. Pokud je výsledkem nalezení oblastí, které se nepřekrývají a odpovídají objektům reálného světa, hovoříme o kompletní segmentaci. V případě, kdy oblasti přesně nesouhlasí s objekty, jde o částečnou segmentaci. Kompletní segmentace je ve většině případů náročná a neobejdeme se při ní bez bližších znalostí scény. Částečná segmentace je většinou výsledkem metod pracujících na principu homogenity oblastí. Jiný přístup nabízí segmentace na základě hledání hranic objektů. Dalším neopomenutelným výsledkem segmentace, vzhledem k dalšímu zpracovávání, je mnohdy podstatné omezení objemu dat.

2.5.1. Prahování

Prahování je základní, implementačně velmi jednoduchou metodou segmentace obrazu, kdy na základě hodnoty jasu obrazového bodu a vhodně zvoleného prahu T rozhodujeme o příslušnosti obrazového bodu k popředí nebo pozadí. Výsledkem prahování je dvouúrovňový obraz, v němž jsou v ideálním případě odděleny objekty zájmu a pozadí.

$$g(x, y) = \begin{cases} 1 & \text{pro } f(x, y) \geq T \\ 0 & \text{pro } f(x, y) < T \end{cases}.$$

Stanovení úrovně prahu T je pro výsledek prahování rozhodující a při vhodném rozložení jasu v obraze lze hodnotu prahu stanovit pomocí některé z metod automatického určování prahu (např. Otsu [11]). Vhodným rozložením jasu máme na mysli, že objekty mají dostatečně odlišnou úroveň jasu od pozadí (histogram má dva výrazné vrcholy). Další variantou je lokální prahování, při kterém, na rozdíl od globálního, určujeme hodnotu prahu pro jednotlivé oblasti samostatně. Dostaneme tak lepší výsledky případě, že scéna není např. rovnoměrně osvětlena. V případech, kdy máme dostatečnou znalost scény a dokážeme určit procentuální zastoupení objektů a pozadí, je vhodné využít tzv. p-procentní prahování.

2.5.2. Spojování oblastí

Na začátku této metody je obraz rozdělen na malé části, které jsou pak postupně spojovány na základě stanovených podmínek. Počáteční dělení může být až na úrovni jednotlivých pixelů a jako podmínka pro spojení dvou oblastí může sloužit např. informace o jasu nebo barvě oblasti. Tento postup opakujeme dokud už nelze žádné oblasti v obraze spojit.

2.5.3. Dělení oblastí

Dělení oblastí představuje opačný postup než předešlá metoda. Celý obraz je na začátku v jediné oblasti, kterou postupně dělíme na menší, dokud daná oblast není homogenní. Sledovanou vlastností může být opět střední úroveň jasu v jednotlivých částech. V praxi se často používají metody kombinující oba postupy.

2.5.4. Detekce hran

Hranu lze definovat jako výraznou změnu jasu. Při detekci hran tedy v obraze hledáme oblasti, ve kterých k těmto prudkým změnám dochází. Metody detekce hran jsou založeny na využití první a druhé derivace obrazového signálu. V oblasti s výraznou změnou jasu se nachází lokální maximum první derivace obrazové funkce a druhá derivace prochází nulou. V praxi se pak využívá konvoluce obrazu s vhodně zvolenou maskou pro aproximaci první nebo druhé derivace.

Metody založené na první derivaci První derivaci obrazové funkce $f(x, y)$ popisuje její *gradient*. Gradient je vektorová funkce ∇ , definovaná jako vektor parciálních derivací:

$$\nabla f(x, y) = [G_x, G_y] = \left[\frac{\partial f}{\partial x}, \frac{\partial f}{\partial y} \right].$$

Velikost (modul) gradientu :

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

Směr gradientu:

$$\varphi(x, y) = \arctan \frac{G_y}{G_x} = \frac{\frac{\partial f}{\partial y}}{\frac{\partial f}{\partial x}}.$$

V diskrétním obraze derivace aproximujeme pomocí diferencí:

$$\Delta_x f(x, y) \approx f(x + 1, y) - f(x, y),$$

$$\Delta_y f(x, y) \approx f(x, y + 1) - f(x, y).$$

Robertsův operátor je velmi jednoduchý, ale také náchylný na šum.

$$G_x \approx f(x, y) - f(x + 1, y + 1).$$

Konvoluční jádro pro směr x a y :

$$\begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix}, \begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix}.$$

Masku pro směr y dostaneme pootočením masky pro směr x o 90° . Stejný postupem dostaneme masky i pro ostatní směry. (Masku můžeme rotovat i po 45° .)

Operátor Prewittové je definován jako:

$$G_x \approx f(x + 1, y) - f(x - 1, y).$$

Konvoluční jádro pro směr x a y :

$$\begin{bmatrix} -1 & 0 & 1 \\ -1 & 0 & 1 \\ -1 & 0 & 1 \end{bmatrix}, \begin{bmatrix} -1 & -1 & -1 \\ 0 & 0 & 0 \\ -1 & -1 & -1 \end{bmatrix}.$$

Sobelův operátor. Konvoluční jádro pro směr x :

$$\begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix}.$$

Ostatní masky opět dostaneme rotací.

Metody založené na druhé derivaci Pokud v obraze hledáme hranu a nezajímá nás její směr, můžeme využít detekci hran založenou na druhé derivaci obrazové funkce. V místě, kde je první derivace maximální, je druhá derivace rovna 0. Pokud můžeme v diskrétním obraze aproximovat první derivaci jako rozdíl hodnot jasu dvou sousedních bodů, bude druhá derivace určena rozdílem těchto dvou sousedních rozdílů. Dostáváme tak vztah:

pro směr x :

$$\frac{\partial^2 f(x, y)}{\partial x^2} \approx [f(x+1, y) - f(x, y)] - [f(x, y) - f(x-1, y)],$$

pro směr y :

$$\frac{\partial^2 f(x, y)}{\partial y^2} \approx [f(x, y+1) - f(x, y)] - [f(x, y) - f(x, y-1)].$$

Laplaceův operátor ∇^2 (Laplacián) pro funkci $f(x, y)$:

$$\nabla^2 f(x, y) = \frac{\partial^2 f(x, y)}{\partial x^2} + \frac{\partial^2 f(x, y)}{\partial y^2}.$$

Předchozí vztah můžeme u diskrétní funkce opět nahradit následovně:

$$\nabla^2 f(x, y) \approx f(x+1, y) + f(x-1, y) + f(x, y+1) + f(x, y-1) - 4f(x, y).$$

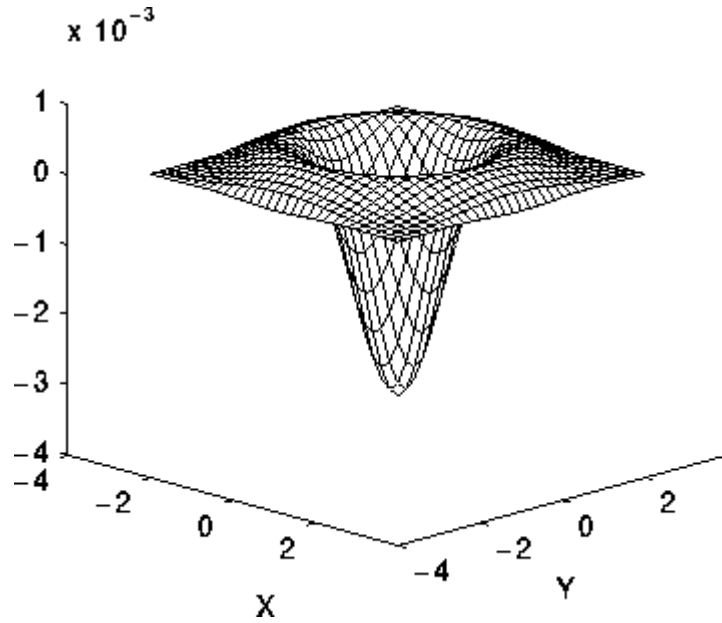
Konvoluční jádro Laplaceova operátoru:

$$\begin{bmatrix} 0 & 1 & 0 \\ 1 & -4 & 1 \\ 0 & 1 & 0 \end{bmatrix}.$$

Laplacian of Gaussian. Laplaceův operátor je citlivý na izolované body a tedy i šum. Pokud bychom nejdříve na obraz aplikovali Gaussův operátor dojde k vyhlazení šumu a tedy i jeho vlivu na výsledek následné konvoluce obrazu s Laplaceovým operátorem. Obě operace lze sloučit do jedné – Laplacian of Gaussian (LoG):

$$LoG(x, y) \approx -\frac{x^2 + y^2 - 2\sigma^2}{2\pi\sigma^6} e^{-\frac{x^2+y^2}{2\sigma^2}},$$

kde x, y jsou souřadnice obrazového bodu a σ je standardní odchylka (rozptyl).



Obrázek 11. 2-D funkce Laplacian of Gaussian, [4].

Zjednodušené konvoluční jádro operace LoG:

$$\begin{bmatrix} 0 & 0 & -1 & 0 & 0 \\ 0 & -1 & -2 & -1 & 0 \\ -1 & -2 & 16 & -2 & -1 \\ 0 & -1 & -2 & -1 & 0 \\ 0 & 0 & -1 & 0 & 0 \end{bmatrix}.$$

Další metody které je vhodné na tomto místě ještě zmínit jsou Houghova transformace a diskrétní Fourierova transformace. Tyto dvě metody zde nebudou představeny, ale v případě zájmu je možné se více dovědět například v [1], respektive [5].

2.5.5. Srovnání se vzorem

Hledání vzoru v obraze by se mohlo zdát jako jednoduchá a efektivní metoda, ve skutečnosti však naráží na mnoho překážek. V reálu se maska jen málokdy přesně shoduje s podobou hledaného objektu v prohledávaném obraze. Většinou obraz obsahuje šum a hledané objekty jsou nějakým způsobem geometricky deformovány. Již pouhé mírné pootočení může mít za následek neúspěch. Vytváření samostatného vzoru a jeho hledání pro každou možnou variantu ještě více zvyšuje výpočetní náročnost metody. Z těchto důvodů ve většině případů nehledáme přesnou kopii vzoru, ale omezuje se jen na dostatečnou podobnost. Tu lze vyjádřit pomocí korelačního koeficientu, který může nabývat hodnot $\langle -1, 1 \rangle$ a má tvar:

$$r = \frac{\sum_x \sum_y (f(x, y) - \bar{f})(g(x, y) - \bar{g})}{\sqrt{\sum_x \sum_y (f(x, y) - \bar{f})^2 \sum_x \sum_y (g(x, y) - \bar{g})^2}},$$

kde $f(x, y)$ je obrazová funkce obrazu A a $\bar{f}$ je střední hodnota jasu v A . $g(x, y)$ je obrazová funkce obrazu B a $\bar{g}$ je střední hodnota jasu v B .

2.6. Popis

Při vytváření této kapitoly jsem čerpal z [12]. Abychom mohli objekty nalezené při procesu segmentace přiřadit pomocí klasifikátoru k jednotlivým třídám, musíme je nejprve vhodným způsobem popsat. Možných popisů objektů je hodně a vždy se snažíme objekty charakterizovat s ohledem na následný klasifikační algoritmus, který na základě tohoto popisu objekty zájmu rozpozná. Rozeznáváme dva typy přístupu k tomuto problému.

První typ je kvantitativní popis charakteristik, tedy popis na základě elementárních číselných vlastností objektů, tzv. příznaků. Mohou to být vlastnosti jako je velikost, podlouhlost, kompaktnost, nebo třeba průměrná hodnota jasu. Výsledkem takového popisu je vektor příznaků, který co nejvýstižněji popisuje daný objekt.

Druhým typem je kvalitativní popis, který zkoumá i vztahy základních vlastností (primitiv) objektů. Tyto vztahy mezi primitivy (struktura objektu) mohou být reprezentovány grafy typu strom nebo řetězci. K popisu třídy v těchto strukturách je využívána gramatika definující třídu prostřednictvím svých pravidel. V ideálním případě je takový popis invariantní vůči změně velikosti nebo pootočení.

2.6.1. Některé základní vlastnosti

Velikost N jednoduše stanovíme součtem všech obrazových bodů objektu:

$$N = \sum_{i=0}^n P_i,$$

kde P_i jsou jednotlivé obrazové body objektu.

Podlouhlost L se stanovuje jako poměr stran nejmenšího opsaného obdélníku (pravoúhelníku) objekt:

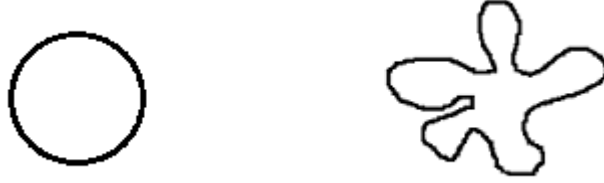
$$L = \frac{a}{b},$$

kde a je delší strana obdélníku a b je kratší strana obdélníku.

Kompaktnost C lze vyjádřit vztahem:

$$C = \frac{l^2}{N},$$

kde l je délka hranice objektu a N jeho velikost.



Obrázek 12. Příklad kompaktního a nekompaktního objektu.

Momenty představují vážený průměr intenzity jasu pixelů, vztažený k nějakému stanovenému bodu v obraze, většinou k těžišti. Mohou tak nést informaci charakteristickou pro popisovaný objekt. Momenty, které jsou invariantní vůči translaci, rotaci a velikosti objektu definoval Hu [7] následovně:

$$I_1 = \eta_{20} + \eta_{02},$$

$$I_2 = (\eta_{20} - \eta_{02})^2 + 4\eta_{11}^2,$$

$$I_3 = (\eta_{30} - 3\eta_{12})^2 + (3\eta_{21} - \eta_{03}),$$

$$I_4 = (\eta_{30} + \eta_{12})^2 + (\eta_{21} + \eta_{03})^2,$$

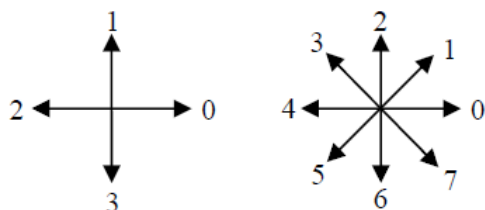
$$I_5 = (\eta_{30} - 3\eta_{12})(\eta_{30} + 3\eta_{12})[(\eta_{30} + \eta_{12})^2 - 3(\eta_{21} + \eta_{03})^2] + (3\eta_{21} - \eta_{03})(\eta_{21} + \eta_{03}) \\ [3(\eta_{30} + \eta_{12})^2 - (\eta_{21} + \eta_{03})^2],$$

$$I_6 = (\eta_{20} - \eta_{02})[(\eta_{30} + \eta_{12})^2 - (\eta_{21} + \eta_{03})^2] + 4\eta(\eta_{30} + \eta_{12})(\eta_{21} + \eta_{03}),$$

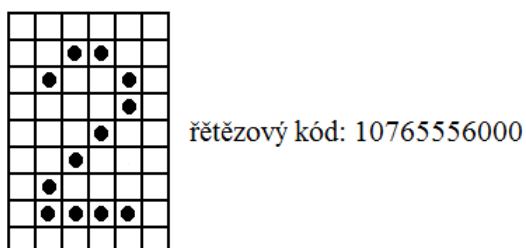
$$I_7 = (3\eta_{21} - \eta_{03})(\eta_{30} + \eta_{12})(\eta_{30} + \eta_{12})^2 - 3(\eta_{21} + \eta_{03})^2 + (3\eta_{21} - \eta_{03})(\eta_{21} + \eta_{03}) \\ [3(\eta_{30} + \eta_{12})^2 - (\eta_{21} + \eta_{03})^2],$$

kde $I_0, \dots, I_7$ jsou momentové invarianty a η_{pq} jsou normalizované centrální momenty.

Řetězové (Freemanovy) kódy se používají k popisu hranice. Jsou to posloupnosti čísel, reprezentující jednotlivé směry při pohybu po hranici objektu. Aby byl řetězový kód invariantní je nutné ho derivovat. Derivaci získáme jako první diferenci kódu mod 4 nebo mod 8. Takto normovaný řetězový kód pak ukazuje kolik pootočení o 90 nebo 45 stupňů je potřeba při postupu po hranici objektu.



Obrázek 13. Označení směrů pro 4-okolí a 8-okolí.



Obrázek 14. Číslice „2“ a její řetězový kód.

Další variantou popisu objektu pomocí jeho hranic jsou tzv. podpisy objektů (Singnitures)[\[1\]](#).

2.7. Klasifikace

Při vytváření této kapitoly jsem čerpal z [10]. Klasifikace je posledním krokem posloupnosti zpracování. Jejím účelem je pomocí klasifikátoru rozpoznat objekty, to znamená přiřadit je do jednotlivých tříd. Klasifikace úzce souvisí s popisem objektu v obraze (viz kapitola 2.6.). Klasifikátor je totiž algoritmus, který o příslušnosti objektu k dané třídě rozhoduje právě na základě popisu objektu.



Obrázek 15. Schématické znázornění klasifikace.

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. Klíčovou roli při klasifikaci hraje vedle vhodně zvolených charakteristik objektů nastavení klasifikátoru. K tomu může sloužit trénovací množina, což je soubor prvků s předem známou příslušností k třídě. Procesu nastavení klasifikátoru pomocí takovéto trénovací množiny říkáme učení s učitelem. Je zřejmé, že vzorové prvky trénovací množiny by měly zobecňovat i ostatní možné objekty, aby klasifikátor dobře fungoval nejen u objektů obsažených v trénovací množině. Pokud nemůžeme využít trénovací množinu, protože není k dispozici nebo neznáme počet tříd, můžeme využít metodu shlukové analýzy. Tato metoda rozděluje rozpoznávané objekty do shluků (množin), které obsahují blízké si prvky. O podobnosti je rozhodováno na základě popisu objektů.

Druhou možností klasifikace je rozhodování na základě struktury objektu (syntaktická analýza). K tomu slouží gramatika [9], která kontroluje syntaxi nalezených objektů. Základní vlastnosti objektů (primitiva) tvoří abecedu. Nalezené objekty jsou popsány pomocí řetězců primitiv (slov). Gramatika svými pravidly definuje jazyk nad danou abecedou. To znamená, že pomocí gramatiky můžeme rozhodnout o příslušnosti slova (popisu nalezeného objektu) k jazyku.

Jazyk generovaný gramatikou $G = \langle N, \sigma, P, S \rangle$:

$$L(G) = \{w \in \Sigma^*; S \Rightarrow^* w\},$$

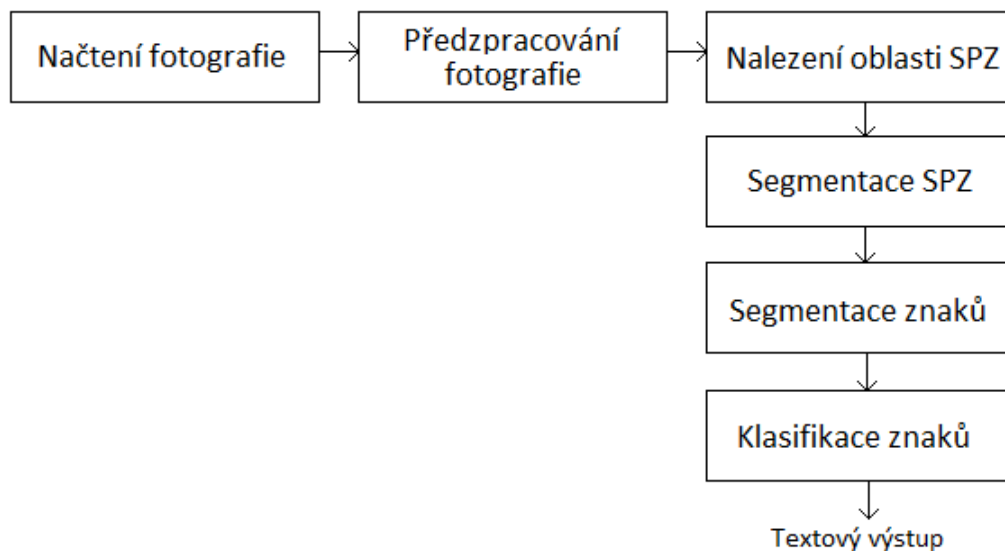
kde N je abeceda se symboly zvanými neterminály, σ je abeceda se symboly zvanými terminály, P je množina pravidel, S je počáteční symbol, w je slovo (řetěz terminálních symbolů) a $\Rightarrow^*$ je relace odvození.

3. Rozbor úlohy

Jak bylo řečeno již v úvodu, konkrétní řešenou úlohou je nalezení a následné rozpoznání státní poznávací značky (dále jen SPZ) na fotografii automobilu. Podobné aplikace nacházejí využití v dopravě při sledování a řízení provozu a společně s vhodnou databází jsou pak cennými pomocníky při zaznamenávání a identifikaci hříšníků, kteří spáchali dopravní přešestupek nebo třeba jako automatický vrátný kontrolující vjezd do areálu. Úspěšnost těchto systémů je z podstatné části ovlivnitelná vstupními parametry.

V praxi se setkáváme s celou řadou problémů, které je třeba pro dosažení správného výsledku brát v potaz. Jedná se například o volbu snímacího zařízení, která rozhoduje o kvalitě pořizovaného záznamu. Vhodná poloha kamery zase mimo jiné usnadňuje lokalizaci značky. Můžeme pak předpokládat výskyt značky v určité oblasti a vyhneme se prohledávání celého snímku, čímž snížíme možnost špatné lokalizace i výpočetní náročnost. Usnadnění určení polohy lze dosáhnout i pomocí pevných referenčních bodů, které mohou být tvořeny například viditelnými značkami na vozovce. Někdy se využívají i čidla zabudovaná přímo ve vozovce. Většina systémů pro rozpoznávání SPZ se používá ve venkovním prostředí, což sebou nese další problém, a tím jsou proměnlivé světelné podmínky při snímání scény vlivem denní doby a počasí. Problém s osvětlením lze v mnoha případech řešit přisvícením infračerveným světlem. Mnou vytvářená aplikace se některým těmto problémům vyhne volbou fotografií, pro které bude určena.

Rozdělení problému na jednotlivé části, které je nutné postupně vyřešit, je schématicky znázorněno na obr. 16.



Obrázek 16. Schématické znázornění procesu rozpoznání SPZ z fotografie.

3.1. Státní poznávací značka silničního motorového vozidla a přípojného vozidla

Při vytváření této kapitoly jsem čerpal z [3] a [8]. České republiky vydávají SPZ obce s rozšířenou působností podle zákona č. 56/2001 sb. v němž se například objevuje i nový termín „Registrační značka“. (Pojmy RZ a SPZ jsou ekvivalentní.) Vzhled, provedení a umístění RZ pak řeší vyhláška č. 243/2001 a její pozdější novely – vyhlášky č. 291/2004 sb. a č. 334/2006 sb.

RZ se skládá podle typu z pěti až sedmi alfanumerických znaků, přesněji z velkých písmen latinské abecedy (bez diakritiky) a arabských číslic. Některé znaky se nepoužívají, aby nedocházelo k záměně (G, O, Q, W). Každá standardní RZ musí obsahovat minimálně jedno písmeno a alespoň jednu číslici, přičemž první písmeno musí označovat kraj, ve kterém byla RZ vydána. Toto neplatí pro speciální typy značek, jakými jsou např. diplomatické RZ atd. Tyto speciální značky se od standardních mohou lišit nejen strukturou registračního čísla, ale i barevným provedením.

Tabulka s RZ dále může obsahovat kontrolní nálepky o provedení pravidelné technické kontroly a měření emisí výfukových plynů a může být provedena jako jednořádková a dvouřádková. RZ vydané po 1. 5. 2004 mají na levém okraji modrý „euro“ proužek, s označením státu. Současně s těmito RZ se ještě stále můžeme setkat se „starými“ značkami vydávanými do roku 2001. Ty jsou tvořeny trojicí písmen a dvěma dvojicemi číslic někdy oddělenými pomlčkou. První dvě písmena v těchto značkách označují okres, kde byla značka registrována, třetí písmeno pak sérii. Tabulka s RZ musí být až na výjimky podle vyhlášky umístěna na vozidle podle následujících pravidel:

- v předu i vzadu,
- uprostřed (pokud v technickém průkaze není povoleno jiné umístění),
- kolmo na směr jízdy,
- spodní hranou dolů,
- vodorovně k vozovce.

Jakékoliv zakrývání, ušpinění nebo zásahy do značky, které znemožňují či snižují její čitelnost jsou striktně zakázány a za jízdu s takovou značkou hrozí podle současného sazebníku pokuta od pěti do deseti tisíc korun a zákaz řízení na šest až dvanáct měsíců.



Obrázek 17. Standardní registrační značka o rozměrech 520 x 110 mm

3.2. Cíle a vstupní předpoklady

Cílem práce není vytvoření aplikace, která by svými parametry konkurovala sofistikovaným profesionálním systémům, ale snaha demonstrovat využití alespoň základních metod zpracování obrazu na jednoduchém příkladu. Komerční systémy používané v praxi mají úspěšnost rozpoznání přes 95% a rychlost zpracování 10 snímků za sekundu. Mým cílem je dosáhnout úspěšnosti rozpoznání kolem hodnoty 70%, přitom rychlost nebude klíčová, protože aplikace bude zpracovávat jen jednotlivé statické snímky zvolené uživatelem. Fotografie budou zobrazovat automobil pokud možno z přímého čelního, nebo zadního pohledu, bez jiných rušivých objektů. Požadovaným cílem bude nalezení a následné rozpoznání „nové“ české jednořádkové SPZ (viz. kap. 3.1.).

4. Implementace

Pro implementaci podobných aplikací zpracovávajících obraz lze využít celou řadu již vytvořených knihoven, ať už placených nebo volně dostupných na internetu, určených pro tuto problematiku. Příkladem je multiplatformní open source knihovna OpenCV společnosti Intel. Úlohu by také bylo možné vcelku pohodlně realizovat v Matlabu společnosti MathWorks, který pro práci s obrazem obsahuje Image processing toolbox. Pro práci jsem se však rozhodl nevyužít těchto možností a jít vlastní cestou vytvořením vlastní třídy, která bude mít v aplikaci zpracování obrazu na starost. Důvodem této volby bylo lepší nastudování a pochopení metod a algoritmů pro práci s obrazem. Před samotnou implementací jednotlivých metod je potřeba jim dobře porozumět, na rozdíl od „slepého“ používání již definovaných funkcí v případě využití jiného nástroje.

Práci jsem vytvářel v programovacím jazyce C# a pro volbu algoritmů byla vedle dosažitelných výsledků důležitá především jednoduchost implementace. Některé části kódu jsou převzaty nebo inspirovány knihou Michala Dobeše „Zpracování obrazu a algoritmy v C#“ [1]. Mnou vytvořená třída pro práci s obrazem se jmenuje „Obraz.cs“ a je využívána třídou „Algoritmus.cs“. Tato třída již jen využívá metody předchozí třídy a jak její název napovídá, shrnuje celý postup implementovaného algoritmu rozpoznávání SPZ do svojí hlavní metody

„Algoritmus.HLEDEJ(image)“. Proces rozpoznání bude postupně rozebrán dále v textu a pro podrobnější seznámení je možné nahlédnout do zdrojového kódu na přiloženém CD.

4.1. Fotografie a jejich předzpracování

Prvotní myšlenkou bylo využít pro testování aplikace fotografií z internetových autobazarů apod. Od tohoto záměru jsem však musel upustit z důvodu nesourodosti, různé kvality a mnohdy úmyslně znehodnocené oblasti fotografií obsahující SPZ. Pro potřeby ladění a testování jsem nakonec vytvořil vlastní soubor fotografií, který je dostupný na přiloženém CD. Fotografie byly pořizovány postupně v průběhu práce běžnými digitálními fotoaparáty a dále již nebyly žádným softwarem speciálně upravovány, pouze byly v programu Malování operačního systému Windows 7 upraveny na potřebnou velikost a případně oříznuty.

Fotografie byly pořizovány z různých vzdáleností a za různých světelných podmínek ovlivněných denní dobou a počasím, mohou proto mít rozdílnou ostrost a jas. Z tohoto důvodu jsou všechny fotografie před samotným hledáním oblasti SPZ, které probíhá na základě detekce kolmých hran, vyhlazeny. Při testování se ukázalo vhodné je ještě před vyhlazením paradoxně přiostrit. Na takto upravený obraz je aplikován hranový detektor, poté následuje prahování. V aplikaci je použito globálního prahování s pevně stanoveným prahem. Volba prahu je empiricky stanovena na úroveň 90 a získaný binární obraz je dále podstoupen dilataci. Cílem je odstranění drobných objektů (šumu) a zbytečných horizontálních hran z obrazu, před následným hledáním oblasti SPZ pomocí detekce kolmých hran. Zde je uveden použitý operátor:

$$\begin{bmatrix} -1 & -1 & -1 \\ -1 & 8 & -1 \\ -1 & -1 & -1 \end{bmatrix}.$$

4.2. Nalezení oblasti SPZ

Správná lokalizace oblasti SPZ ve vstupní fotografii je prvotním předpokladem pro pozdější úspěšné rozpoznání značky, respektive správné rozpoznání jejích jednotlivých znaků. Výhodou pro mnou vytvářenou aplikaci je předpokládaná poloha a velikost značky ve vstupním snímku. Tato znalost umožňuje urychlit výpočet omezením prohledávané části snímku. Algoritmus je tedy nastaven tak, aby prohledával pouze spodní polovinu fotografie.

Samotné určení polohy SPZ je provedeno na základě počtu kolmých hran, které se vyskytují v obraze po předchozích úpravách. Je zřejmé, že oblast výskytu SPZ v obraze bude takových hran obsahovat specifické množství a tohoto faktu využívá i realizovaný algoritmus. Obraz je postupně procházen a vždy je nalezena pozice s největším výskytem kolmých hran. Pokud tato pozice vyhovuje podmínce

na počet hran stanovené při ladění, je celá oblast předána k dalšímu zpracování. V opačném případě je oblast okolo nalezené pozice smazána a proces se opakuje.

Nevýhodou tohoto přístupu může být situace, kdy fotografie obsahuje i jiné oblasti generující stejný počet kolmých hran jako SPZ. Příkladem může být přední maska automobilu nebo textová reklama na kapotě. Z tohoto důvodu dokáže algoritmus pro nalezení oblasti značky označit více kandidátních oblastí, ve výsledné implementaci toho však není využito.



Obrázek 18. Správně lokalizovaná oblast SPZ (označena červeným křížkem).

4.3. Segmentace SPZ

Lokalizovaná oblast je „vyříznuta“ z původního vstupního snímku a opětovně upravena pro další zpracování. Úprava oblasti spočívá v přístřežení a následném globálním prahování. Pokusil jsem se hodnotu prahu stanovovat automaticky pomocí metody Otsu, ale tato metoda nepřinášela dobré výsledky a nakonec se ukázala jako nejlepší volba stanovit práh pevně. Úroveň prahu je tedy opět stanovena na 90. Při této hodnotě docházelo při testování k nejlepším výsledkům u nejvíce snímků. V takto připraveném obraze hledáme přesnou polohu SPZ.

Možným způsobem jak dosáhnout úplné segmentace značky je využití znalosti její velikosti a horizontálního histogramu. V aplikaci je však využito postupu inspirovaného článkem „Rychlý algoritmus rozpoznávání registračních značek vozidel“ [2]. Byla vytvořena univerzální maska SPZ a tento vzor je hledán v prahovaném kandidátním výřezu. Výsledkem je pozice s největší shodou prohledávaného obrazu a masky. Metoda je velmi jednoduchá na implementaci, komplikuje ji však skutečnost, že v reálu používané značky mohou mít více podob, pro vytvářenou aplikaci je ale dostačující.



Obrázek 19. Univerzální maska vytvořená pro segmentaci SPZ.

4.4. Segmentace znaků

Extrahování jednotlivých znaků podstatně ulehčuje pevná struktura SPZ a předpoklad známé velikosti značky. Pokud se podařila přesná segmentace značky, můžeme vycházet právě z její struktury. Využijeme tedy znalost předpokládané polohy každého znaku díky zákonem stanovené podobě značky. Stačí jen výřez se značkou rozdělit do odpovídajících částí. Alternativou by bylo získat jednotlivé znaky např. pomocí vertikálního histogramu, ale výše navržená metoda je podstatně jednodušší a u správně extrahovaných značek má velmi dobré výsledky. Ze získaného výřezu jsou odstraněny všechny malé objekty, respektive je ponechán pouze ten největší. Ostatní objekty budou zřejmě jen malé nečistoty na značce, šum nebo části jiného znaku v případě špatné lokalizace výřezu a jsou proto ignorovány. Největší nalezený objekt reprezentující hledaný znak je poté předán ke klasifikaci.



Obrázek 20. Prahovaná SPZ a extrahované znaky.

4.5. Klasifikace znaků

Problematika rozpoznávání textu a znaků obecně, neboli OCR (Optical Character Recognition), se v posledních letech dynamicky rozvíjela. Programy rozpoznávající text jsou dnes již standardním doplňkem různých skenerů a kopírek a mnoho jich lze nalézt na internetu. Tyto programy dosahují úspěšnosti až 99% a některé nabízejí i kontrolu a opravu špatně klasifikovaných znaků pomocí kontroly celého slova.

Komerční aplikace využívají různých sofistikovaných metod rozpoznávání. Příkladem může být software na třídění poštovních zásilek pracující na principu rozložení bodů v obraze. V obraze jsou hledány průsečíky rozpoznávaného znaku a vektorů na definovaných místech obr. 21.



Obrázek 21. Hledání průsečíků.

Mezi implementačně nejjednodušší metody rozpoznávání patří metoda srovnání se vzorem. Pro volbu této metody u vytvářené aplikace hovořilo snadné vytvoření vzorů díky omezené sadě používaných znaků, jejich jednotné velikosti a fontu. Byly tedy vytvořeny šablony pro všechny znaky, které se mohou vyskytovat v SPZ. (Vytvořená kolekce vzorů je součástí aplikace.) Každý segmentovaný znak je postupně porovnáván se všemi šablonami a určení znaku probíhá podle největší shody. V případě minimální shody s kterýmkoliv vzorem je zobrazen znak „?“ . Pokud se tak stane hned u prvního znaku, nejsou již ostatní znaky rozpoznávány a místo nalezených znaků se na všech pozicích zobrazí znak „?“ .

Tento stav s největší pravděpodobností nastane v situaci, kdy byla špatně určena poloha značky a nemá tak smysl snažit se rozpoznávat její jednotlivé znaky.

4.6. Výsledná aplikace

Aplikace umožňuje zpracovávat fotografie ve formátu jpg, gif, bmp, png a pro úspěšné rozpoznání SPZ je vlivem zvoleného algoritmu důležitý předpoklad správné velikosti značky na fotografii. Aplikace umožňuje uživateli tři režimy práce.

V základním zobrazení okna aplikace uživatel pouze načtením zvolí požadovanou fotografii a nemůže ovlivnit parametry automatického vyhledání a rozpoznání SPZ. Ovládání programu je v základním režimu intuitivní a neklade na uživatele žádné speciální nároky na znalost problematiky zpracování obrazu. Při automatickém režimu rozpoznávání lze přes volbu „Zobrazení – podrobnosti“ rozšířit okno o část, která poskytuje podrobnější informace o procesu rozpoznání.

„Pokročilé“ zobrazení je režim, kdy uživatel může ovlivnit parametry vyhledání a rozpoznání SPZ. Uživatel je postupně proveden sledem jednotlivých kroků předzpracování fotografie, které přesně kopírují postup automatického zpracování. Je mu umožněno jednotlivé kroky ovlivnit změnou parametrů, buďto přímo, nebo přes menu „Nastavení“. Vedle těchto dvou režimů může aplikace sloužit i jako základní editor pro úpravu fotografií. Tento režim je dostupný přes menu „Zobrazení – rozšířené“, případně je možné využít jednotlivé funkce v menu „Nástroje“. Podrobnější popis aplikace je k dispozici přímo v aplikaci v menu „Nápověda“, nebo na přiloženém CD.

4.7. Výsledky testování

Testování aplikace proběhlo na sadě celkem 66 fotografií, které byly pořízeny běžnými digitálními fotoaparáty, a kromě úpravy velikosti dále neprošly žádnými úpravami. Výsledky testování zobrazují tabulky 1. a 2. Kompletní výsledky rozpoznání jednotlivých fotografií zobrazují tabulky v příloze A.

	Počet	%
Celkem fotografii	66	100
Úspěšně nalezené oblasti pomocí hran	65	98,5
Úspěšné upřesnění pozice pomocí masky	62	95,4
Celkem přesně nalezených značek	62	94

Tabulka 1. Úspěšnost nalezení SPZ.

Celková úspěšnost přesné lokalizace značky je tedy 94%.

	Počet	%
Rozpoznávaných značek	62	100
0 chyb	50	80,7
1 chyba	9	14,5
2 chyby	1	1,6
3 chyby	1	1,6
4 chyby	1	1,6

Tabulka 2. Úspěšnost rozpoznávání SPZ

Do statistiky v tab. 2. jsou započítávány pouze přesně lokalizované SPZ. Správně rozpoznáno bylo 422 znaků z celkového počtu 434 znaků v přesně nalezených SPZ. To představuje úspěšnost rozpoznávání znaků 97,2%.

Celková pravděpodobnost úspěšného nalezení a bezchybného rozpoznání celé značky je tedy **75,8%**. Byly tak splněny cíle stanovené v kap. 3.2.

4.8. Shrnutí a hodnocení

Hledání oblasti výskytu značky je realizováno na základě jejího specifického počtu kolmých hran. Tento postup může selhávat v případech, kdy fotografie obsahuje oblasti s podobnými parametry jako hledaná značka. Rušivě může působit i maska automobilu, nebo případné textové reklamy na kapotě. Algoritmus proto dokáže označit více kandidátních oblastí. Navazující část algoritmu se ale nepodařilo dostatečně přesně nastavit, a v některých případech z více kandidátních

pozic vybral tu špatnou. Z tohoto důvodu nejsou ve výsledném řešení alternativní pozice hledány, nicméně klasifikátor se podařilo nastavit takovým způsobem, že v testované kolekci fotografií označil oblast chybně jen v jediném případě (fotografie č. 59).

V době, kdy již byla aplikace rozpracována, jsem narazil na článek [2], kde je navržen „rychlý algoritmus rozpoznání registračních značek vozidel“, který se téměř shoduje s realizovaným postupem popsaným v tomto textu. Podobně jako v tomto článku jsem vytvořil vzor značky, který je hledán v obraze, a pokusil jsem se do vytvářené aplikace implementovat jednoduchou korekci pootočení značky. V některých případech algoritmus „zbytečně“ otáčí i rovné značky, ale z celkového pohledu představuje korekce pootočení přínos. Obě úpravy tak měly pozitivní vliv na výsledek. Vzory značky byly nakonec vytvořeny dva. Jeden pro značky s modrým euro proužkem na straně a druhý bez něj, což přineslo další zlepšení při upřesňování polohy, která je kriticky důležitá pro bezchybnou segmentaci jednotlivých znaků. K té je totiž využita pouze znalost pevné struktury značky dané příslušným zákonem. Algoritmus tedy nevyhledává skutečnou pozici znaků, ale pouze předpokládá jejich výskyt na předem známém místě. To představuje značné zjednodušení problému a v případě správné velikosti značky a přesného určení její pozice je to i plně funkční způsob nalezení znaků.

Před samotnou klasifikací znaků jsou získané oblasti se znaky ještě upravovány. Úprava spočívá ve snaze odstranit všechny drobné objekty přítomné kvůli šumu, špatnému prahování atd., a ponechat ve výřezu pouze samotný znak. Takto získaný znak je poté porovnáván se sadou vzorů a o příslušnosti znaku k vzoru je rozhodnuto podle největší shody se vzorem, kdy některé vybrané pixely ve vzoru mohou mít větší rozhodovací váhu než ostatní. Sada vzorů je přímo součástí zdrojového kódu.

Úspěšnost rozpoznání dobře segmentovaných znaků ve značce dosahuje přes 97 %. Dobrou statistiku podporuje jednotný font znaků i možnost předpokládat výskyt číslic a písmen na určitých pozicích ve značce. Podle platného zákona musí značka obsahovat minimálně jedno písmeno a jednu číslici, přitom první písmeno musí odpovídat kraji, kde je značka registrována. Teoreticky jsou tak možné různé „prominentní“ kombinace písmen a čísel, jako PEPA007 a podobně. V současnosti se však stále striktně dodržuje způsob vydávání značek, kdy si zájemce nemá možnost podobu nové značky vybrat. Jsou tak vydávány značky pouze s jedním písmenem na druhé pozici ve značce a od roku 2009, kdy došlo k vyčerpání všech kombinací u značek vydávaných v Praze a později i ve Středočeském kraji, se může písmeno v těchto krajích vyskytovat i na třetí pozici. Na třetí pozici tak může být číslice i písmeno, což situaci lehce komplikuje. Při testování byla polovina špatně rozpoznávaných znaků právě na třetí pozici. Pro odstranění této skutečnosti by tedy bylo vhodné věnovat větší pozornost vytvořeným vzorům a jejich lepšímu vzájemnému odlišení. Možnou cestou vylepšení by bylo zvětšit extrahovaný znak na větší rozlišení, případně úplně pozměnit způsob klasifikace.

Realizace navrženého algoritmu tak potvrdila jeho slabiny, které byly zřejmé

již na počátku. Pro správnou lokalizaci a rozpoznání značky je důležitá její správná pozice a zejména velikost na fotografii. To může být při reálném nasazení v mnoha případech výrazně omezující podmínkou. V praxi by taková aplikace obstála pouze v situacích, kdy můžeme předpokládat předem známou velikost snímku a objektů v něm, tedy především u stacionárních kamer. S těmito omezeními vstupních parametrů však byla aplikace vytvářena již od začátku testování nakonec prokázalo, že zvolený postup je při dodržení zmíněných předpokladů životaschopný.



Obrázek 22. Okno aplikace s rozpoznanou SPZ.

Závěr

Účelem této práce bylo seznámení se s problematikou zpracování obrazových dat na nižší úrovni, která je běžně označována jako předzpracování obrazu i zpracování na vyšší úrovni, představující segmentaci a klasifikaci objektů v obraze.

Z logického pohledu lze práci rozdělit do dvou částí. Nejdříve jsou v textu představeny základní pojmy a postupy práce s obrazem používané při jeho zpracování (kap. 2.). Druhá část se pak věnuje výběru a implementaci algoritmů, které mají za úkol lokalizovat a rozpoznat státní poznávací značku ve fotografii automobilu (kap. 4.).

Součástí práce je i shrnutí a hodnocení dosažených výsledků (kap. 4.8.) získaných při testování vytvořené aplikace na zkušební sadě fotografií (kap. 4.7.).

Conclusions

The aim of this work is to make an introduction to the image processing at a lower level (so called image preprocessing). Moreover, the work deals with the processing at a higher level representing the segmentation and classification of the objects in an image.

From a logical point of view the work can be divided into two parts. First, basic concepts and approaches are introduced (Section 2.). The second part is devoted to selection and implementation of the algorithms for location and identification of car license plates from the photos (Section 4.).

The work also summarizes and evaluates the results (Section 4.8.) obtained from a set of the test images (Section 4.7.).

Reference

- [1] M. Dobeš: *Zpracování obrazu a algoritmy v C#*, BEN 2008, ISBN 978-80-7300-233-6.
- [2] M. Dobrovolný: *Rychlý algoritmus rozpoznávání registračních značek vozidel*.
- [3] D. Feuereisl: WWW stránky – www.feudal.cz.
- [4] R. Fisher a kol.: *Image processing learning resources*, 2004, The University of Edinburgh.
- [5] V. Hlaváč: *Lineární integrální transformace – Fourierova transformace*, Fakulta elektrotechnická ČVUT v Praze, Katedra kybernetiky, Centrum strojového vnímání.
- [6] V. Hlaváč, M. Sedláček: *Zpracování signálů a obrazů*, ČVUT 2005, ISBN 80-01-03110-1.
- [7] Hu: *Visual pattern recognition by moment invariants*. Information Theory, IEEE Transactions, 1962, ISSN 0018-9448.
- [8] Legislativa věnující se problematice SPZ: WWW stránky – www.mdcz.cz.
- [9] P. Martínek: *Základy teoretické informatiky*, 2006, Katedra informatiky, Přírodovědecká fakulta UP Olomouc.
- [10] V. Mařík, Olga Štěpánková, Jiří Lažanský a kol.: *Umělá inteligence (1)*, 1993, ISBN 80-200-0496-3.
- [11] N. Otsu: *A Threshold Selection Method from Gray-Level Histograms*, IEEE Transactionson Systems, Man and Cybernetics 1979.
- [12] M. Šonka, V. Hlaváč: *Počítačové vidění*, Grada 1992, ISBN 80-85424-67-3.
- [13] J. Žára: *Moderní počítačová grafika*, Computer Press 2004, ISBN 80-251-0454-0.

A. Tabulka nalezení a rozpoznání SPZ

snímek	nalezení oblasti	upřesnění pozice	SPZ	rozpoznaná SPZ
1.jpg	ANO	ANO	3E88031	0E88031
2.jpg	ANO	ANO	4H49903	4H49903
3.jpg	ANO	ANO	2C84152	2CB4152
4.jpg	ANO	ANO	7S85010	7S85010
5.jpg	ANO	ANO	4H75891	4H75891
6.jpg	ANO	ANO	6B54067	6B54067
7.jpg	ANO	ANO	6B54067	6B54067
8.jpg	ANO	ANO	6S57476	6S57476
9.jpg	ANO	ANO	4A16562	4A16562
10.jpg	ANO	ANO	3E89493	3E89493
11.jpg	ANO	ANO	6T81740	8PB0740
12.jpg	ANO	ANO	4C96824	4C96824
13.jpg	ANO	ANO	4E11598	4E11598
14.jpg	ANO	ANO	1T60582	1T60582
15.jpg	ANO	NE	1E44885	???????
16.jpg	ANO	ANO	2E63190	2E63190
17.jpg	ANO	ANO	4U76161	4U76161
18.jpg	ANO	ANO	4U76161	4U76161
19.jpg	ANO	ANO	2P47415	2P47415
20.jpg	ANO	ANO	2P47415	2P47415
21.jpg	ANO	ANO	1SJ6623	1SJ6623
22.jpg	ANO	ANO	1SU1399	1SU1399
23.jpg	ANO	ANO	4B40970	4B40970
24.jpg	ANO	ANO	2C41262	2C41262
25.jpg	ANO	ANO	3M70118	3M70116
26.jpg	ANO	ANO	8S36382	8S36382
27.jpg	ANO	ANO	9S02800	9S02800
28.jpg	ANO	ANO	4M21039	4M21039
29.jpg	ANO	ANO	5B25219	5B25219
30.jpg	ANO	ANO	5S48610	5S48610
31.jpg	ANO	ANO	4S13052	4S13052
32.jpg	ANO	ANO	4T38227	4T38227
33.jpg	ANO	ANO	3T45003	3T46003
34.jpg	ANO	ANO	1T95558	1T95558
35.jpg	ANO	ANO	4T37827	4T37827

Tabulka 3. Tabulka nalezení a rozpoznání SPZ.

snímek	nalezení oblasti	upřesnění pozice	SPZ	rozpoznaná SPZ
36.jpg	ANO	ANO	2T11264	2T11264
37.jpg	ANO	ANO	4T08128	4TD8325
38.jpg	ANO	ANO	3T73176	3T73176
39.jpg	ANO	NE	5S58812	???????
40.jpg	ANO	ANO	4S56973	4S96973
41.jpg	ANO	ANO	5U83001	5U83009
42.jpg	ANO	ANO	2S08993	2S08993
43.jpg	ANO	ANO	4B31881	4B31861
44.jpg	ANO	ANO	3E88031	3E88031
45.jpg	ANO	ANO	1SU5433	1SU5433
46.jpg	ANO	NE	7S47744	???????
47.jpg	ANO	ANO	5S48339	5S48339
48.jpg	ANO	ANO	1SN1937	1SN1937
49.jpg	ANO	ANO	6T85363	6FB5363
50.jpg	ANO	ANO	3Z19215	3Z19215
51.jpg	ANO	ANO	2AD4813	2AD4813
52.jpg	ANO	ANO	3T02277	3T02277
53.jpg	ANO	ANO	3U61220	3U61220
54.jpg	ANO	ANO	6T91292	3T02277
55.jpg	ANO	ANO	2A89470	2A89470
56.jpg	ANO	ANO	8A41225	8I41225
57.jpg	ANO	ANO	7B22810	7B22810
58.jpg	ANO	ANO	4T50713	4T50713
59.jpg	NE	NE	4T12955	???????
60.jpg	ANO	ANO	1SE6328	1SE6328
61.jpg	ANO	ANO	3Z43174	3Z43174
62.jpg	ANO	ANO	1SC2788	1SC2786
63.jpg	ANO	ANO	4H74928	4H74928
64.jpg	ANO	ANO	1SU0554	1SU0554
65.jpg	ANO	ANO	5C37411	5C37411
66.jpg	ANO	ANO	1AR6544	1AR6544

Tabulka 4. Tabulka nalezení a rozpoznání SPZ (pokračování).

B. Obsah přiloženého CD

`readme.txt`

Soubor obsahuje důležité informace pro spuštění a správnou funkčnost programu „Rozpoznávání SPZ“.

`bin/`

Složka obsahuje spustitelný program „Rozpoznávání SPZ.exe“ a soubor s nápovědou.

`doc/`

Složka obsahuje textovou část práce ve formátu PDF a zdrojový text práce, včetně všech použitých obrázků.

`src/`

Složka obsahuje kompletní zdrojové texty programu „Rozpoznávání SPZ“.

`data/`

Složka obsahuje testovací sadu fotografií.