

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA ELEKTROTECHNIKY A KOMUNIKAČNÍCH TECHNOLOGIÍ
ÚSTAV TELEKOMUNIKACÍ

FACULTY OF ELECTRICAL ENGINEERING AND COMMUNICATION
DEPARTMENT OF TELECOMMUNICATIONS

DETEKCE STÁTNÍ POZNÁVACÍ ZNAČKY PRO DOHLEDOVÉ
SYSTÉMY

BAKALÁŘSKÁ PRÁCE
BACHELOR'S THESIS

AUTOR PRÁCE
AUTHOR

DAVID SMÉKAL

BRNO 2013



VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY



**FAKULTA ELEKTROTECHNIKY A KOMUNIKAČNÍCH
TECHNOLOGIÍ**
ÚSTAV TELEKOMUNIKACÍ

FACULTY OF ELECTRICAL ENGINEERING AND COMMUNICATION
DEPARTMENT OF TELECOMMUNICATIONS

DETEKCE STÁTNÍ POZNÁVACÍ ZNAČKY PRO DOHLEDOVÉ SYSTÉMY

DETECTION OF REGISTRATION NUMBER FOR SURVEILLANCE SYSTEMS

BAKALÁŘSKÁ PRÁCE
BACHELOR'S THESIS

AUTOR PRÁCE
AUTHOR

DAVID SMÉKAL

VEDOUCÍ PRÁCE
SUPERVISOR

Ing. JIŘÍ PŘINOSIL, Ph.D.

BRNO 2013



**VYSOKÉ UČENÍ
TECHNICKÉ V BRNĚ**

**Fakulta elektrotechniky
a komunikačních technologií**

Ústav telekomunikací

Bakalářská práce

bakalářský studijní obor
Teleinformatika

Student: David Smékal

ID: 134405

Ročník: 3

Akademický rok: 2012/2013

NÁZEV TÉMATU:

Detekce státní poznávací značky pro dohledové systémy

POKYNY PRO VYPRACOVÁNÍ:

Prostudujte moderní metody číslicového zpracování obrazových signálů orientované na detekci státní poznávací značky. Na základě získaných teoretických znalostí navrhnete metodu státní poznávací značky České republiky ve statických obrazech s uvažováním reálných podmínek při snímání obrazu. Navrženou metodu následně implementujte ve vhodném programovacím jazyku, ověřte na reálných statických obrazech a stanovte podmínky její použitelnosti.

DOPORUČENÁ LITERATURA:

[1] Nixon, M., Aguado, A.: Feature Extraction & Image Processing, Academic Press, ISBN: 978-0-1237-2538-7, 2008.

[2] Parker, J.R.: Algorithms For Image Processing And Computer Vision, John Wiley & Sons, Inc., ISBN: 0-471-14056-2, 1997.

Termín zadání: 11.2.2013

Termín odevzdání: 5.6.2013

Vedoucí práce: Ing. Jiří Přinosil, Ph.D.

Konzultanti bakalářské práce:

prof. Ing. Kamil Vrba, CSc.

Předseda oborové rady

UPOZORNĚNÍ:

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

ABSTRAKT

Práce se zabývá teoretickým popisem zpracování obrazu a počítačového vidění, detekcí a rozpoznáním státních poznávacích značek, resp. registračních značek. Jsou zde zmíněny jednotlivé metody segmentací obrazu, jako jsou filtrace šumu, detekce hran, prahování. Zkoumá přítomnost registrační značky v obraze.

KLÍČOVÁ SLOVA

SPZ, RZ, státní poznávací značka, registrační značka, detekce, prahování, segmentace, počítačové vidění, zpracování obrazu, OpenCV, C++.

ABSTRACT

The bachelor thesis deal with teoretic image processing and computer vision, detection license plate. There are include some methods segmentation image for example filtration noise, detection edge, thresholding. Researched presence of licence plate in the picture.

KEYWORDS

LP, license plate, detection, thresholding, segmentation, computer vision, image processing, OpenCV, C++.

SMÉKAL, David *Detekce státní poznávací značky pro dohledové systémy*: bakalářská práce. Brno: Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií, Ústav telekomunikací, 2013. 42 s. Vedoucí práce byl Ing. Jiří Přinosil, Ph.D.

PROHLÁŠENÍ

Prohlašuji, že svou bakalářskou práci na téma „Detekce státní poznávací značky pro dohledové systémy“ jsem vypracoval samostatně pod vedením vedoucího bakalářské práce a s použitím odborné literatury a dalších informačních zdrojů, které jsou všechny citovány v práci a uvedeny v seznamu literatury na konci práce.

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

Brno

.....

(podpis autora)

PODĚKOVÁNÍ

Rád bych poděkoval vedoucímu práce panu Ing. Jiřímu Přinosilovi, Ph.D. za odborné vedení, konzultace, trpělivost a podnětné návrhy k práci. Za cenné poznámky a připomínky k obsahu i zpracování.

Brno

.....

(podpis autora)

OBSAH

1	Úvod	10
1.1	Představení práce	10
1.2	Důvody detekce	10
1.3	Struktura práce	11
2	Zpracování obrazu	13
2.1	Počítačové vidění	13
2.2	Technologie OCR	13
3	Segmentace obrazu	15
3.1	Filtrace šumu	15
3.1.1	Lokální vyhlazování obrazu	15
3.1.2	Lineární vyhlazování	16
3.1.3	Nelineární vyhlazování	16
3.2	Detekce hran	16
3.3	Prahování	17
4	Vývojové prostředí	20
4.1	Microsoft Visual Studio	20
4.2	OpenCV	21
5	Registrační značky	22
5.1	Specifikace značek	22
6	Návrh postupu detekce	24
6.1	Popis způsobu nalezení značky	25
7	Testování	27
7.1	Pořízené testovací fotografie	27
7.2	Použití knihovny Puma.NET	27
7.3	Zkoumání parametrů prahování a umístění RZ	29
7.3.1	Automatizované prahování a detekce hran	30
7.3.2	Zjištění plochy značky a klasifikace znaků	31
8	Podmínky nasazení	33
8.1	Předem stanovená pravidla	33
8.2	Skutečné vlastnosti použití	34

9	Výsledky studentské práce	35
9.1	Výsledky	35
9.2	Dosažené výsledky	36
10	Závěr	38
	Literatura	39
	Seznam příloh	41
A	Přílohy	42
A.1	Příložené CD	42

SEZNAM OBRÁZKŮ

1.1	Ukázka kontrolní mýtné brány a pevné otočné kamery [12]	11
1.2	Schéma funkce kontrolní mýtné brány [12]	11
3.1	Normovaná značka a obraz s aplikací Laplaceova operátora	17
3.2	Prahování s nižší hodnotou prahu	19
3.3	Prahování s vyšší hodnotou prahu	19
4.1	Logo Microsoft Visual Studio	20
4.2	Logo OpenCV	21
5.1	Registrační značka	22
6.1	Vývojový diagram	24
6.2	Dílčí kroky postupu	25
6.3	Rozměry značky	26
7.1	Ukázky zdárné detekce RZ s nadbytkem chybných znaků	28
7.2	Ukázky správné detekce znaků RZ	29
7.3	Ukázky chybné detekce RZ	29
7.4	Prahování Otsu metodou a Canny detektor	31
7.5	Prahování nalezené oblasti a rozpoznání znaků	32
8.1	Úhel detekce	33
9.1	Výsledky úspěšných detekcí registrační značky	36
9.2	Výsledky úspěšných detekcí registrační značky s použitím blesku	37
9.3	Chybná detekce registrační značky	37

SEZNAM TABULEK

5.1	Tabulka krajů a jejich písmenné označení	23
-----	--	----

1 ÚVOD

Tato práce řeší problematiku počítačového vidění, především detekci státní poznávací značky. Seznámíme se s digitalizací objektu, zpracováním obrazu a jeho následnou filtrací či segmentací. Dále s jednotlivými možnostmi a metodami detekcí objektů v obraze.

V současnosti se setkáváme se zpracováním obrazových dat na každém kroku. Najde se jen málo lidí, kteří by se nesetkali s digitální fotografií. Ať už pořízenou na své dovolené či výletu za zážitkem, nebo pro pracovní účely. Přitom jen malá hrstka lidí zná a chápe princip zpracování obrazu a počítačového vidění. Metody rozpoznávání obrazu a počítačového vidění nacházejí stále širší uplatnění v různých oborech. Především v dopravě, medicíně, robotice, atd. Jedním ze zajímavých a užitečných směrů toho odvětví je detekce a následné zpracování státní poznávací značky, resp. registrační značky automobilu.

1.1 Představení práce

Cílem práce je vytvořit vhodnou metodu, která bude rychlá, přesná a transparentní. Pokusíme se zahrnout všechny nároky na přesnou detekci. Seznámíme se možnostmi detekce objektů v obraze, vhodnou úpravou pořízených snímků a zpracováním výsledných dat. Za zdroj dat budeme považovat reálné statické snímky. V navazující bakalářské práci se pokusíme realizovat problematiku na kameru a zpracovávat data přímo z kamery. Naším cílem je vytvořit program, který nepotřebuje přítomnost lidského faktoru.

1.2 Důvody detekce

Proč vlastně detekovat objekty v obraze? Konkrétně registrační značku vozidla? V dnešní době to není neobvyklá věc, aby nás snímaly desítky kamer. Každá slouží na různou věc. Jedna střeží objekt proti zlodějům, další zjišťuje provoz v ulici, jiná sleduje aktuální počasí. Důvodů je hned několik. V různých odvětvích se používá kamerových systémů. Moderní medicína by se bez kamer neobešla. Operace pomocí robotů, různé kamerové sondy pro diagnostiku, jsou dnes běžně používanou věcí.

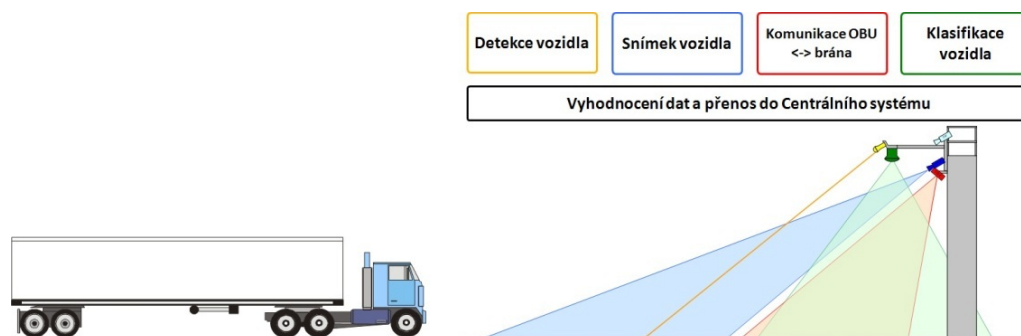
Naším tématem je detekce značek pro dohledové systémy. Na dálnicích již řadu let existuje systém elektronických mýt (obr. 1.2), které mají za úkol monitorovat sjízdnost silnic a především účtovat mýtné pro nákladní vozidla. Vše je řízeno počítačovým systémem a není zde potřeba lidské obsluhy. Tento systém se v České

republiky začal používat roce 2007 a stále je co zdokonalovat. Je třeba hlídat bezpečnost, maximální dovolenou rychlost, zda se vozidla pohybují ve správném směru jízdy apod.



Obr. 1.1: Ukázka kontrolní mýtné brány a pevné otočné kamery [12]

Jak uvádí Ředitelství silnic a dálnic ČR na [12], telematické aplikace (inteligentní dopravní systémy) jsou moderní informační a komunikační technologie, které přímo na pozemní komunikaci sledují a vyhodnocují konkrétní charakteristiky provozu, informují o aktuální dopravní situaci nebo provoz na komunikaci podle stanovených pravidel bezprostředně řídí.



Obr. 1.2: Schéma funkce kontrolní mýtné brány [12]

1.3 Struktura práce

Samostatná práce je rozdělena do několika částí:

- **Zpracování obrazu** – popisuje princip počítačového vidění jako vědní obor. Seznámí nás s uplatněním v dnešní době a představí technologii OCR.
- **Segmentace obrazu** – vytváří ucelené informace o pojmu segmentace. Zahrnuje různé metody zpracování obrazu, jejich úpravu pro další práci. Zaměřuje se především na filtraci šumu (kap. 3.1), detekce hran (kap. 3.2) a prahování (kap. 3.3).
- **Vývojové prostředí** – objasňuje princip vytváření softwaru a jsou zde zmíněny programy a knihovny, se kterými budeme pracovat.
- **Registrační značky** – tato část popisuje přesné zákonem definované typy a rozměry registračních značek.
- **Návrh postupu detekce** – seznamuje čtenáře o postupu, jakým způsobem bude docíleno nalezení registrační značky.
- **Testování** – shrnutí, jakým způsobem byly použité metody testovány a čím byla práce obohacena.
- **Podmínky nasazení** – specifikují předem stanovené podmínky použitelnosti.
- **Výsledky studentské práce** – ukazují dosažené výsledky v semestrální práci.

2 ZPRACOVÁNÍ OBRAZU

2.1 Počítačové vidění

Počítačové vidění je vědní disciplína, která se snaží technickými prostředky alespoň částečně napodobit lidské vidění [4]. V širším smyslu je chápáno za součást kybernetiky nebo umělé inteligence. Od tohoto poznatku však nejsme daleko od pravdy. Zobrazení pozorovaných obrazových dat, správná interpretace, je jádrem porozumění tohoto obrazu v počítačovém vidění. Cílem je snaha porozumět obecné trojrozměrné scéně za pomoci předchozích zkušeností a znalostí člověka. Takovéto porozumění je založeno na znalosti problému, cílech, tvorbě plánů k jejich dosažení a využití zpětných vazeb mezi různými úrovněmi zpracování.

Získání číslíkové reprezentace obrazových dat je ovlivněno řadou procesů. Mezi hlavní patří snímání scény, digitalizace (kvantizace), rušení (šum). Podle [1] je 5 hlavních příčin, které činí těžší celkové porozumění :

1. Ztráta informace při dvojrozměrném zobrazení z původně trojrozměrné scény.
2. Komplikovaný a nejednoznačný vztah mezi jasnem snímáče a tvarem povrchu 3D objektů ve scéně.
3. Veliké množství obrazových dat.
4. Šum, který je vždy přítomný v každé reálné scéně.
5. Vztah mezi pozorovaným detailem a zjišťovaným celkem.

Obor počítačového vidění se v poslední době stále rozvíjí, k čemuž dopomáhají levnější a stále dokonalejší kamerové systémy. Rozvoj oboru je přímo spojen s rozvojem počítačových zařízení, pomocí kterých se v maximální možné míře využívá výpočetní výkon. Zároveň je nutné zmínit také stále snadnější dostupnost různých knihoven a materiálů, zaměřujících se právě na zpracování obrazu.

2.2 Technologie OCR

Technologie OCR neboli optické rozpoznávání znaků (Optical Character Recognition) je systém digitálního zpracování textu do editovatelné podoby. Program obrázek převede na běžně upravitelný text, se kterým lze klasicky pracovat. Na kvalitě předlohy závisí kvalita výstupu. Čím kvalitnější předlohu algoritmus dostane, tím méně chyb může program udělat a závěrečná ruční korekce bude méně náročná. Technologii OCR je samozřejmě možné využít pro převod širokého spektra předloh. Nejvíce se této metody využívá pomocí scannerů, kdy se tištěná předloha převede

do digitální podoby a následně se jedná o textový soubor, který lze dále editovat (nejedná se o obraz).

Základem rozpoznávání je naučit systém případy, které mohou nastat a také jak tyto případy vypadají. Základními typy jsou myšleny písmena a speciální znaky jako čárka, tečka, otazník apod. Musíme tedy systému dodat předlohu a on si podle ní vyrobí popis daného znaku. Při rozpoznávání se pak hledané znaky porovnávají s naučenou databází. Celý OCR systém se skládá z několika kroků:

1. Scanování – převod do digitální podoby a následné prahování snímku (bude popsáno níže).
2. Vyčištění obrazu – Tato část má za úkol vyhladit převedený obraz a zbavit jej nepříznivých vlivů. Využívá se grafických filtrů jako například vyhlazení.
3. Lokalizace a segmentace – v další fázi se zkoumá rozložení textu jako celku na stránce. Například rozlišuje grafické prvky od textu. Poté se nejčastěji pomocí histogramu určí, kde jsou jednotlivé řádky a mezery. Následně se znaky izolují podle spojitých černých oblastí.
4. Rozpoznání znaků – úkolem této nejproblematictější části je klasifikovat upravený obraz podle známých znaků a převést do textu. Na to existuje několik metod. Některé popisují znak dle upraveného obrazu, jiné zjišťují specifické rysy, které daný znak charakterizují.
 - (a) Rozdělení do pásem – plocha symbolu se rozdělí podle mřížky na jednotlivá pásma. Poté se pomocí histogramu rozpoznají černé plochy v každém políčku. Výsledné hodnoty se porovnávají s naučenými znaky. Jakmile systém najde shodu, přiřadí k danému upravenému symbolu patřičné písmeno a tím složí celý text.
 - (b) Průsečíková metoda – je to metoda vektorová. V učicí fázi se jednotlivá písmena popisují pomocí průsečíků s přímkami. Počet, poloha a sklon jednotlivých přímek je předem dán. Stejně přímky jsou použity na upravený text. Takto se zjistí poloha a množství průsečíku. Tyto hodnoty se poté porovnávají s databází a při shodě je přiřazen příslušný symbol.

3 SEGMENTACE OBRAZU

Segmentace má za úkol nalezení objektů, které v obraze máme detekovat a rozlišit je od pozadí. Pro naše použití se musíme zaměřit na nalezení registrační značky umístěné na automobilu a následně získat její znakovou posloupnost. Pokud je výsledkem nalezení oblastí, které se nepřekrývají a odpovídají objektům předem stanovených podmínek, lze segmentaci považovat za kompletní. Jestli tomu tak není, nalezené objekty nesouhlasí, jde o částečnou segmentaci. Částečná segmentace je většinou výsledkem metod pracujících na principu homogenity oblastí. Veliký přínos po segmentaci je podstatné omezení objemu dat pro zpracování.

3.1 Filtrace šumu

Každý obrazový bod zatížený aditivním šumem není závislý na obrazové funkci, má nulovou střední hodnotu a směrodatnou odchylku. Při řešení principu lokální filtrace se setkáváme například s metodou, která využívá malého okolí reprezentativního pixelu k tomu, aby se docílilo výpočtu nové hodnoty pixelu. Tato metoda je založena na představě, že se celý obraz postupně (systematicky) prochází. Metody se dělí do dvou skupin, první skupina se zabývá tzv. vyhlazováním, což znamená, že zde usilujeme o potlačení šumu v obraze. Druhá skupina využívá detekci hran, pro odhadnutí derivace obrazové funkce [2].

Vyhlazování a detekce hran jsou v protikladu. Z tohoto důvodu jsou navrhovány nelineární metody, které jsou schopné obraz vyhladit a zároveň neporuší hrany a detaily v obraze. Předzpracování ve většině případů nepoužívá specifické znalosti o obraze. Pokud máme znalost o obraze k dispozici (může se jednat například o znalost statických parametrů šumu), tak tohoto prostředku může algoritmus zpracování s výhodou využít. Průběhem předzpracování nezískáváme žádnou novou informaci. Je možnost některou informaci potlačit či zvýraznit. Pokud potřebujeme zvýšit informační obsah obrazu, musíme užít lepší postup při získání (pořízení) obrazu.

3.1.1 Lokální vyhlazování obrazu

Pokud máme k dispozici pouze jediný obraz stejné předlohy, musíme se spolehnout na nadbytečnost údajů v obraze. Pixely, které jsou v těsné blízkosti, mají stejnou nebo podobnou hodnotu jasu. Hodnotu obrazových elementů, která je zkreslená šumem, můžeme opravit pomocí analýzy hodnot jasu ve vybraném okolí obrazu.

3.1.2 Lineární vyhlazování

Tyto metody vypočítávají novou hodnotu pixelu ve zkoumaném okolí. Ve zpracování obrazu se užívá třída lineárních filtrů. Nazýváme je invariantní filtry, a to z toho důvodu, že se chování filtru při změně polohy v obrázku nemění. Základní metodou pro vyhlazování obrazu je průměrování. Můžeme si pod tím představit to, že se ke každému bodu přiřadí nový jas, který je aritmetickým průměrem původních jasů v daném okolí.

3.1.3 Nelineární vyhlazování

Tyto metody částečně odstraňují problémy s rozmazáváním hran. Jedná se o snahu najít ve zkoumaném okolí jednu část, kam patří reprezentativní bod. Pixely této oblasti se využijí pro hledání určité hodnoty jasu, která bude reprezentovat celé okolí v obrázku.

3.2 Detekce hran

Detekce hran neboli rozpoznání obsahu obrazu. Abychom mohli vnímat obraz, potřebujeme mít důležitá místa v obraze, čím jsou významné hrany. Tato místa nám dávají více informací o obraze než místa jiná. U hran hraje důležitou roli osvětlení obrazu či místo pohledu. Jak je uvedeno v [4], nejdůležitějším faktorem pro segmentaci jsou informace o hranách v obraze. Tyto informace můžeme dělit na několik fází, nejprve musíme najít hrany a poté se vytváří cílová segmentace, kde se sledují hranice segmentů. Detekce hran slouží k označení body v obraze, u kterých mohou nejčastěji nastat změny hodnot okolních míst. Detekci nejčastěji užíváme v počítačovém vidění pro rozpoznání obrazu, sledování, apod.

Hrana je oblast obrazu, uvnitř které se sehrávají dané změny intenzity jasu [1]. Vznikají díky různým specifikům. Existuje určitá nespojitost v normále k povrchu, hloubce, barvě, či nespojitosti v osvětlení. Díky těmto faktorům detekujeme hrany. Existují metody pro rozpoznání nespojitostí jasu v obraze. Tyto metody se zaměřují na nalezení lokálních maxim dle první derivace signálu anebo v nalezení průchodu nulou u druhé derivace signálu.

Hranové operátory, jak již bylo naznačeno, lze rozdělit do dvou skupin. První se zaměřuje na aproximaci derivace obrazové funkce pomocí diferencí diskretních konvolucí. K odhalení první derivace se používá několik masek. Správný směr gradientu se odhaduje hledáním masky, která odpovídá největší velikosti gradientu. Pokud vyhledáváme hrany dle metody první derivace, tak využíváme Sobelův, Robertsův a Prewittův operátor. Druhá skupina operátorů hledá hrany v místech, kde druhá

derivace obrazové funkce prochází nulou. Vychází z poznatku přizpůsobení se vidění v přírodě, ze kterého vyplývá, že hrany a obrysy v obraze jsou pro porozumění nejdůležitější. Mezi hranové detektory patří Marra operátor (hledání průchodů nulou) nebo Cannyho hranový detektor [4].

Výsledkem první derivace funkce dvou proměnných je gradient, který je definován jako vektor [2]. Výpočet gradientu zjednodušen na tvar

$$\nabla f \approx |G_x| + |G_y|, \quad (3.1)$$

avšak v diskrétním obraze se derivace počítá z rozdílu dvou sousedních hodnot jasu pixelu

$$G_x \approx f(x + 1, y) - f(x, y). \quad (3.2)$$

Nalezení hranových bodů v obraze se využívá pro hledání hranic objektů. Detektor hran má snižovat nebo zcela vytlačit šum obrazu, ztenčit hrany a podobně upravovat daný obraz. Na obr. 3.1 je aplikované zvýrazňování hran pomocí Laplaceova operátoru.



Obr. 3.1: Normovaná značka a obraz s aplikací Laplaceova operátoru

3.3 Prahování

Prahování spočívá v převedení plnohodnotného obrazu na obraz se dvěma jasovými úrovněmi a to černou a bílou [2]. Používá se v případech, kdy se hledané objekty odlišují od pozadí. Dochází k transformaci vstupního obrazu na výstupní obraz, který dále budeme zpracovávat. Princip je zcela jednoduchý, je-li hodnota jasu v určitém bodě větší než stanovená hranice, přiřadí se pixelu černá a naopak je-li hodnota jasu v určitém bodě menší, pixelu se přiřadí barva bílá (viz rovnice 3.3 použita z [3]). Prahování může být globální nebo lokální.

$$f(i, j) = 1 \text{ je-li } g(i, j) \geq T, \text{ jinak } f(i, j) = 0, \quad (3.3)$$

kde T je hodnota prahu.

Jednodušším a často chybným způsobem je globální prahování, kdy je zvolená rozhodovací hodnota prahu aplikována na celý obraz. Tento způsob není vhodný

kvůli nerovnoměrnosti osvětlení. Lepšího výsledku lze docílit předzpracováním vstupního obrazu. Mezi globální prahování patří i procentní prahování. Zde se nezadáva konkrétní práh, ale procentní zastoupení bodů v obraze. Vzhledem k výpočetní nenáročnosti je nejrychlejší segmentační metodou. Lze ji provádět v reálném čase.

Naopak lokální prahování používá takové hodnoty prahu, které jsou měněny v různých částech obrazu. Respektuje rozdílné hodnoty jasu a tím ve výsledku docílí zvýrazněnost snímku. Je zkoumáno okolí každého pixelu a tím je stanovena hodnota prahu. Hodnotu prahu můžeme určit pokusem nebo pomocí metody automatického určování prahu. Tento princip je založen na analýze histogramu, kde rozlišujeme jeden nebo více dominantních vrcholů [3].

Níže jsou zobrazeny 2 snímky, které byli naprahovány. Každý však jinou hodnotou prahu. Jedná se o globální prahování s pevnou hodnotou prahu. Jestliže je obraz převeden do 256 stupňů šedi, máme zde vstupní hodnoty v rozsahu 0–255. V prvním případě na obrázku č. 3.2 je hodnota prahu 50 a snímek je docela jasný. Jelikož je zachyceno stříbrné auto, tak je téměř celé změněno v bílou a opět díky rozdílů barev na SPZ jsou znaky pěkně čitelné. V opačném případě je hodnota prahu vyšší a to 150. Tento jev je viditelný na obrázku č. 3.3. Snímek v levé části je tmavší, deformace znaků na SPZ není patrná a dokonce jsou znaky o něco širší a ve větším kontrastu. Tento práh je pro lidský vjem lepší.



Obr. 3.2: Prahování s nižší hodnotou prahu



Obr. 3.3: Prahování s vyšší hodnotou prahu

4 VÝVOJOVÉ PROSTŘEDÍ

Pro vytváření uceleného projektu je zapotřebí zvolit správné vývojové prostředí, nebo-li software pro zvýšení produktivity programátora pomocí provázaných komponent s uživatelským rozhraním. Jedná se především o zaměření na jeden konkrétní programovací jazyk. Námi vybraný programovací jazyk bude C++ případně C. U C++ se jedná o vyšší programovací jazyk procedurální, který se může zahrnout jak do strukturovaných, tak do objektově orientovaných přístupů. Jedním z úspěšným nasazením tohoto jazyku je téměř úplná kompatibilita C++ s C.

Naše vývojové prostředí bude Microsoft Visual Studio od společnosti Microsoft Corporation, které patří mezi nejpoužívanější programátorský software vůbec.

4.1 Microsoft Visual Studio

Řada produktů Microsoft Visual Studio podporuje proces vývoje softwarových aplikací od návrhu jejich architektury, přes tvorbu, testování až po nasazení. Vytvořené aplikace lze provozovat na různých platformách OS Windows a dalších zařízeních. Nezáleží na tom, zda děláte HTML5 aplikaci s moderním dotykovým rozhraním nebo serverové paralelní nativní výpočty v C++, Visual Studio poskytne vždy optimální produktivitu a výkon [8].

Visual Studio obsahuje editor kódu, integrovaný debugger a další vestavěné nástroje zahrnují designer formulářů pro tvorbu aplikací s GUI, designer webu, tříd a databázových schémat. Podporuje jazyky prostřednictvím jazykových služeb, což umožňuje, aby editor kódu a debugger podporoval jakýkoliv programovací jazyk. Mezi vestavěné jazyky patří C/C++, a C#. Dále jsou podporovány jazyky Oxygene, Python, XML/XSLT, HTML/XHTML, JavaScript a CSS.



Obr. 4.1: Logo Microsoft Visual Studio

4.2 OpenCV

OpenCV (Open Source Computer Vision) je otevřená knihovna počítačového vidění a strojového učení [9].

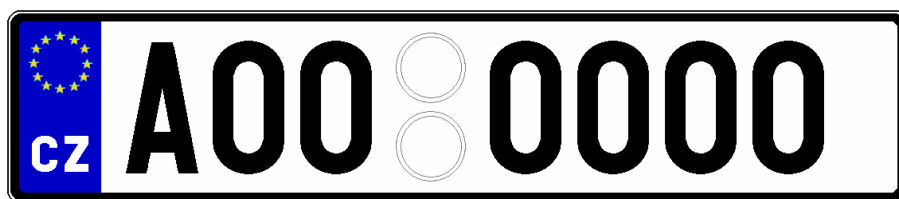
OpenCV spadá pod BSD licenci a proto je zdarma jak pro akademické použití, tak i komerční. Patří sem C++, C, Python, Java rozhraní a podpora Windows, Linux, Mac OS, iOS, a Androidu. OpenCV byl navržen pro výpočetní efektivitu a zaměřuje se na real-time procesory. Byl napsán a optimalizovaný především pro C / C++. Tato knihovna může využít multi-jádrové procesory. OpenCV byl přijat na celém světě, má více než 47 tisíc uživatelů a počet jeho stažení se odhaduje na více než 5 500 000. Rozsah jeho použití je velmi různorodý. Od interaktivního umění, procházení webů na internetu až po pokročilé úkoly v robotice.



Obr. 4.2: Logo OpenCV

5 REGISTRAČNÍ ZNAČKY

V České republice je mnoho typů registračních značek. Podle vyhlášky Ministerstva dopravy 243/2001 Sb. existuje několik typů [10]. My se budeme zaměřovat pouze na standardní značky pro všechna silniční motorová vozidla typu 101. Tento typ má rozměry 520 mm × 110 mm a je nejvíce používaný. Celkový počet černých znaků je 7 a jsou vyobrazena na bílém poli. Ukázková značka typu 101 je na následujícím obrázku (obr. 5.1).



Obr. 5.1: Registrační značka

5.1 Specifikace značek

Současné registrační značky mají dané parametry a vzhled od roku 2001. Značky nemají otvory pro přichycení k vozidlu, tzv. se „nacvakávají“ do plastových rámečků, které slouží k umístění značky. Tyto rámečky se nachází v přední a zadní části vozidla. Značky jsou vydávány krajskými městy. Dle krajů ČR jsou ustanoveny krajské kódy, které zobrazuje tabulka č. 5.1. Existují ještě další skupiny vozidel pod určitými písmeny. Např. písmeno F zahrnuje poznávací značky pro vozidla, která slouží ke zkušebním účelům, dále dle Vyhlášky z roku 2006 [11] se zde řadí i písmeno R, jež je uvedeno u vozidel, která slouží jako sportovní vozidla a v neposlední řadě můžeme nalézt na poznávacích značkách písmeno V, které je přiděleno veteránským vozidlům.

Registrační značky obsahují 5 až 7 znaků. Mezi tyto znaky patří arabské číslice a písmena latinské abecedy bez diakritiky. Pro znaky vyjádřené velkým písmem se používají písmena A, B, C, D, E, I, J, K, L, M, N, P, R, S, T, U, V, X, Y, Z. Pro znaky vyjadřující číslici se používají čísla 1, 2, 3, 4, 5, 6, 7, 8, 9, 0. Platná značka musí obsahovat nejméně jedno písmeno a alespoň jednu číslici. U značek platí zásada, že první písmeno, bráno z levé strany značky, označuje daný kód kraje, pod který náleží registrační místo, které registrační značku přiděluje jednotlivým vozidlům. Tato zásada ovšem neplatí pro registrační značky diplomatické, které začínají písmeny „DD“ a registrační značky cizinecké, kde se jako první písmena uvádějí „XX“. Dle této zásady je možné uvést fakt, že tímto vzniká téměř nevycerpitelná zásoba

Kraj	Písmeno	Kraj	Písmeno
Praha	A	Plzeňský	P
Středočeský	S	Jihočeský	C
Ústecký	U	Vysočina	J
Liberecký	L	Jihomoravský	B
Karlovarský	K	Olomoucký	M
Královéhradecký	H	Moravskoslezský	T
Pardubický	E	Zlínský	Z

Tab. 5.1: Tabulka krajů a jejich písmenné označení

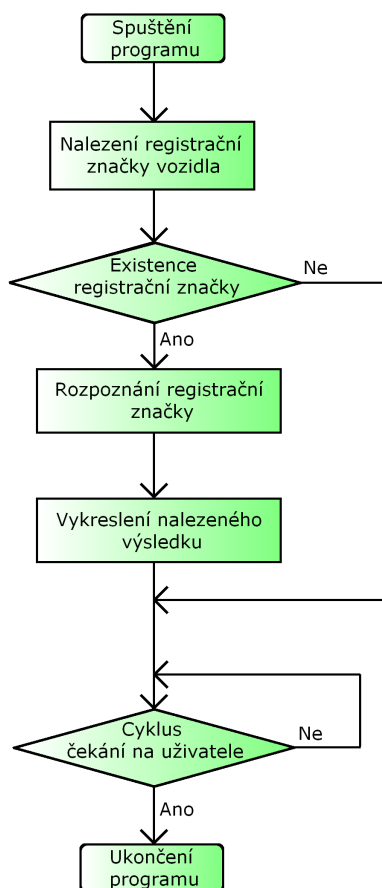
možných kombinací číslic a písmen. Od roku 2004, má každá registrační značka ve své levé části modrý pruh Evropské unie s rozlišovací značkou. U nás je tato značka CZ.

Existuje několik druhů tabulek registračních značek, pro všechny ale platí to, že jsou vyrobeny ze slitin lehkých kovů. Tabulka je po obvodu orámovaná barevným okrajem, který má šířku 4mm. Barva okraje se shoduje s barvou jednotlivých znaků uvedených na značce. Na tabulce jsou kruhové prostory, kde se umísťují nálepky o provedené technické kontrole a provedeném měření emisí škodlivin. Tyto prostory jsou prolisovány již při výrobě značky.

6 NÁVRH POSTUPU DETEKCE

V prvním kroku je důležité si objasnit, jak přesně má algoritmus pracovat. Cíle již máme určené, ale musíme si stanovit přesný postup průběhu programu. Postup celé detekce se bude dělit na 2 samostatné části. První úkol bude nalezení místa s registrační značkou v obrázku a následující druhý krok rozpoznání značky. Detekce značky nám zjistí přesné umístění ve snímku a následně nebudeme pracovat s celým obrazem, pouze s jeho částí, kde se nachází hledaný prvek. Po nalezení RZ se přechází do druhé části a to rozpoznání, u kterého použijeme vybraný OCR algoritmus pro klasifikaci alfanumerických znaků.

Následující vývojový diagram 6.1 nám zobrazuje postup výše popsaného algoritmu v základním tvaru. Každá část diagramu lze rozvinout do složitějších operací.

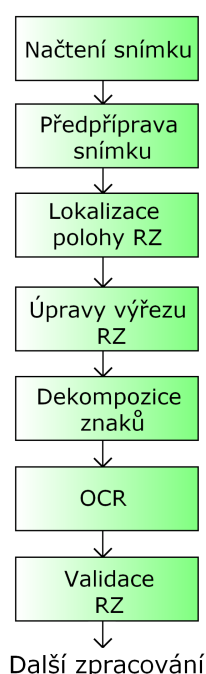


Obr. 6.1: Vývojový diagram

6.1 Popis způsobu nalezení značky

Základním rysem značky je velký počet svislých hran v RZ a kontrast černé a bílé barvy za předpokladu, že snímek byl pořízen zepředu bez jakéhokoliv zkreslení. Této skutečnosti lze využít v kroku odstranění oblasti, které nemají žádné svislé hrany [5]. Převědeme snímek do černobílé barvy, aplikujeme vybraný operátor na detekci hran a následně snímek naprahujeme. Použitím morfologické operace můžeme odstranit mezery mezi každou svislou hranou a spojit oblasti, které mají velký počet hran.

Pouze správná dekompozice problému a vyřešení jednotlivých částí systému rozpoznávání umožní dosažení dostatečné spolehlivosti celého systému a akceptovatelné pravděpodobnosti určení výsledků. Struktura takového systému je zobrazena na vývojovém diagramu 6.2.



Obr. 6.2: Dílčí kroky postupu

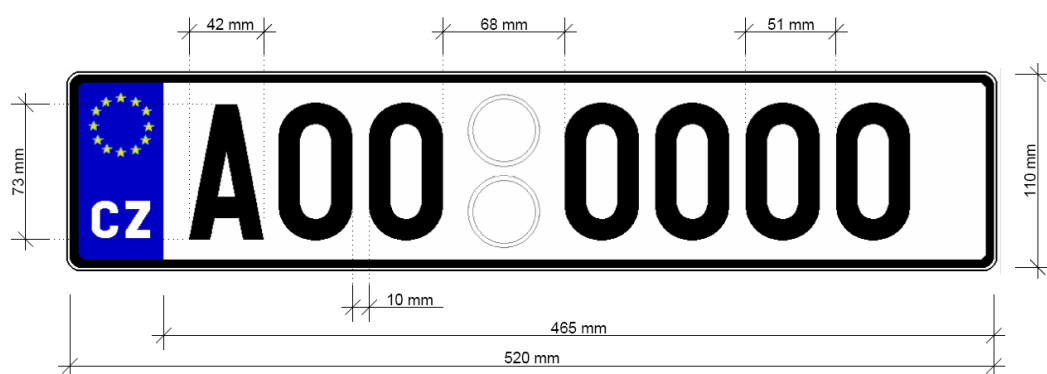
V řetězci rozpoznávání obrazové informace registrační značky je prvním a nejdůležitějším krokem její správná lokalizace polohy. Přesná lokalizace polohy nám následně ulehčuje navazující kroky, kterými jsou zejména přesné určení polohy jednotlivých znaků, a do značné míry také determinuje množinu nutných postupů k extrakci jednotlivých znaků.

Dalším navazujícím krokem je předúprava výřezu reprezentujícího RZ. Jde především o ořez oblasti, ekvalizace průběhu jasové funkce obrazu reprezentujícího právě tuto oblast, kompenzace natočení RZ, případně kompenzace dalších zkreslení [5].

Tyto úpravy, při znalosti rozměrů známých typů RZ a předpokládaných pozic jednotlivých znaků, dále umožňují predikovat pozice znaků. To velice usnadňuje následnou analýzu a segmentaci znaků.

Zpracování jednotlivých znaků je realizováno modulem OCR. Výhodou systému OCR určeného pro rozpoznávání znaků je omezený počet vstupních znaků, navíc s poměrně jednoznačně definovaným vzhledem použitého stylu písma [5]. Důležitá je především validita a případná korektura záměnných znaků (O–0, B–8...).

Obvyklým postupem lokalizace RZ je hledání předem definovaného bloku charakterizovaného přítomností velkého počtu vertikálních hran. K částečnému selhání dochází pouze ve speciálních případech, kdy je na vozidle přítomna oblast s podobnými vlastnostmi (např. mřížka vozidla se svislými hranami), či v případech, kdy jsou vertikální hrany znaků nevýrazné (např. z důvodu znečištění RZ). Na obr. 6.3 jsou vyneseny konkrétní rozměry znaků, mezery mezi nimi a umístění na značce.



Obr. 6.3: Rozměry značky

7 TESTOVÁNÍ

V práci jsem zohlednil různé metody čtení znaků. Zaměřil jsem se na porovnání více algoritmů, abych vyhodnotil, v jakých případech je lepší použít tu či onu metodu.

7.1 Pořízené testovací fotografie

Pro testování bylo pořízeno v Brně na různých místech celkem 120 fotografií. Registrační značky byly foceny převážně na předních částech automobilů. Focení probíhalo při různých podmínkách: slunečno, zataženo, šero, tma. Byly použity dva přístroje (digitální fotoaparát – 5 Mpx a mobilní telefon – 5 Mpx), oba s funkcí blesku. Toho bylo využito hlavně za tmy. Problém byl ve značném odlesku registrační značky díky materiálu, kterým je značka potažena. Avšak v dostatečné vzdálenosti byla značka velice dobře čitelná kvůli kontrastu černé a bílé barvy. Některé fotografie byly zachyceny i z úhlu. Z pořízených snímků bylo vyhodnoceno 96 jako použitelných pro testování.

7.2 Použití knihovny Puma.NET

Velice zajímavý způsob je použití knihovny Puma.NET,¹ která umožňuje snadné začlenění OCR funkcí téměř v každé aplikaci. Je třeba využít .NET Framework 2.0 a vyšší. Práce s touto knihovnou je velice jednoduchá a ve zdrojovém kódu lze při použití pouze pár řádků získat jakékoliv znaky z obrázku. Následující řádky zavádí použití knihovny ve vytvořeném programu.

```
var pumaPage = new PumaPage(obrazek);  
using (pumaPage)  
{  
    pumaPage.FileFormat = PumaFileFormat.RtfAnsi;  
    pumaPage.EnableSpeller = false;  
    pumaPage.Language = PumaLanguage.English;  
    pumaPage.RecognizeToString();  
}
```

Výstup lze zapsat do proměnné a následně vypsat, nebo uložit získané prvky do souborů TXT, RTF, HTML. Knihovna podporuje řadu vstupních souborů jako je BMP, GIF, EXIF, JPG, PNG. Dále různé jazyky, což v našem případě je zbytečné, jelikož čteme pouze znaky běžné latinky a arabské číslice.

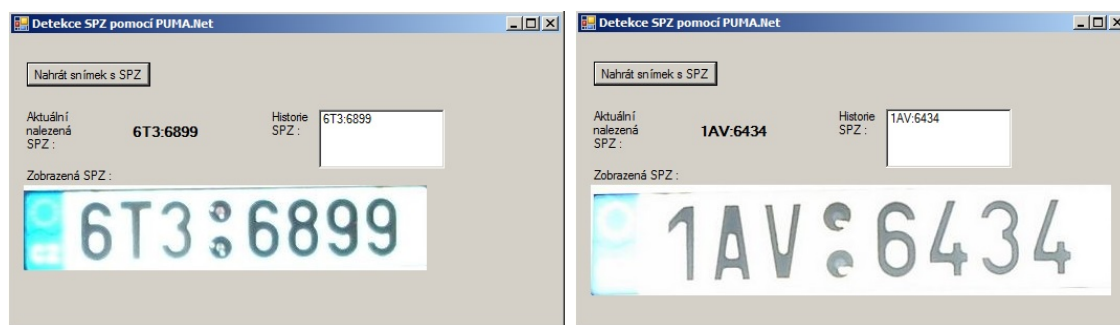
¹Copyright ©2009, Maxim Saplin (<http://pumanet.codeplex.com>)

Při testování snímků je podstatné, zda se jedná o přední nebo zadní poznávací značku. Zadní je totiž opatřena dvěma nálepkami s osvědčením o technické kontrole a stavu emisí. Tyto nálepky, které jsou umístěny mezi jednotlivými znaky, ve většině případů vyhodnocuje program jako dvojtečku nebo jiný nesmyslný znak. Tento znak se ve výsledku promítne i na výstupu znaků právě jako dvojtečka. Dále se často chybně detekuje začátek resp. konec značky a výstupním znakem je potom „!“ nebo „|“.

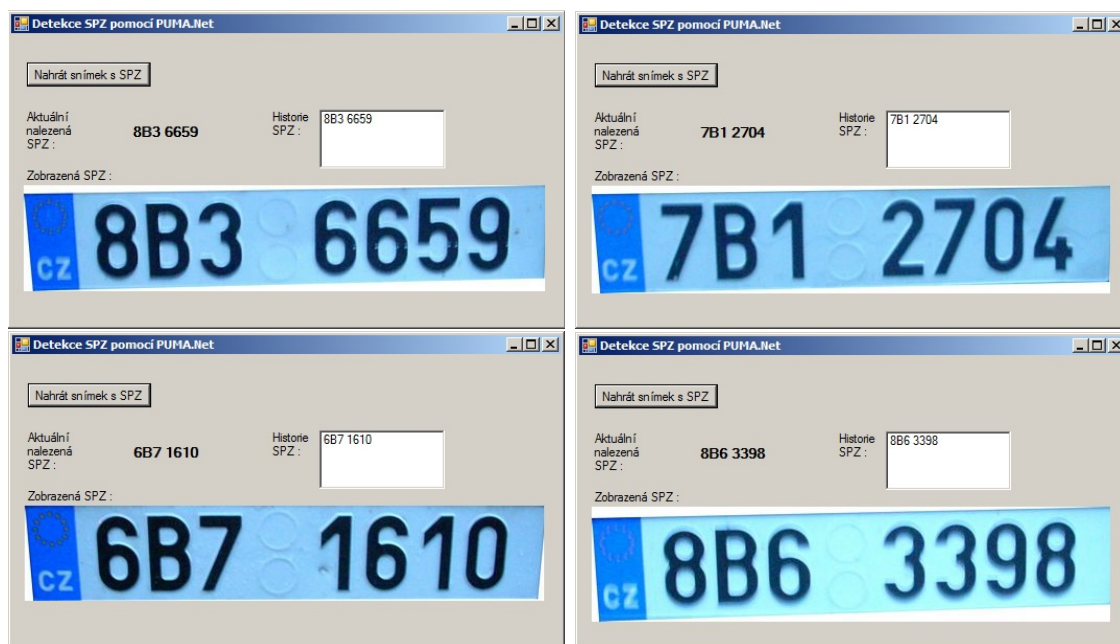
Test této aplikace zkoumal pouze správné vyhodnocení znaků ze značky, proto jako vstupní soubor byl použit snímek již oříznuté značky. Velice negativní vliv má znečištění značky. Následně se v posloupnosti znaků vyskytují, jak již bylo řečeno, dvojtečky, uvozovky, apostrofy, pomlčky atd. Program chybně detekoval znaky také v případech špatně pořízených fotografií. Zejména v případě odlesku světla, špatného pořizovacího úhlu, zastínění některých znaků. Záměna podobných znaků je také častým problémem. Nejvíce se zaměňovaly znaky „B“ a „8“, „A“ a „4“.

I přes tyto drobné nedostatky lze knihovnu označit za spolehlivou pro použití rozpoznávání registračních značek. Podmínkou ovšem zůstává kvalitní vstupní snímek. Z testovaných snímků registračních značek zcela bezchybně obstálo pouze necelých 19%. Dalších 63% obsahovalo zmíněné znaky navíc a 18% vykazovalo velké chyby (např. více jak deset nalezených znaků, symboly neodpovídající žádným písmenům ani číslicím).

Následující ukázka 7.1 znázorňuje zmíněné znaky navíc, které se objevují uprostřed značky. Na obr. 7.2 jsou zobrazeny výsledky správné detekce znaků. Poslední dva obrázky 7.3 zobrazují zcela chybnou detekci.



Obr. 7.1: Ukázky zdárné detekce RZ s nadbytkem chybných znaků



Obr. 7.2: Ukázky správné detekce znaků RZ



Obr. 7.3: Ukázky chybné detekce RZ

7.3 Zkoumání parametrů prahování a umístění RZ

V následujícím shrnutí se zaměřujeme na výsledky testování aplikace „Automatic Number Plate Recognition“. Tato aplikace byla převzata z výukové knihy OpenCV [7] a dále upravována a modifikována pro lepší a přesnější detekci českých registračních značek. Stěžejním úkolem bylo lépe naprahovat snímek. Jak již bylo zmíněno, fotografie byly pořízeny v různých situacích a tedy i světlost scény hrála velkou roli. Když byl snímek prahován nízkou resp. vysokou hodnotou prahu, ztrácely se ze snímku důležité informace o umístění RZ. Vyřešení tohoto problému je popsáno v kapitole č. 7.3.1.

Dalším důležitým bodem ve zlepšení aplikace bylo stanovit přesné parametry ohledně vlastností registrační značky. Detekované snímky zachycují zhruba 2 – 3 metry vzdálenou RZ.

7.3.1 Automatizované prahování a detekce hran

Pro přesné nalezení místa ve fotografii, kde je umístěna RZ je důležité snímek správně upravit. Nejdůležitějším krokem je již zmíněné prahování. Způsobů je hned několik, nejúčinnější se však osvědčila Otsu metoda. Jedná se o automatické nalezení prahu na základě výpočtu rozptylu. Přesněji by se měla však zabývat analýzou histogramu. Metoda vyniká především svojí jednoduchostí a výpočetní nenáročností. Příklad zápisu kódu je uveden níže.

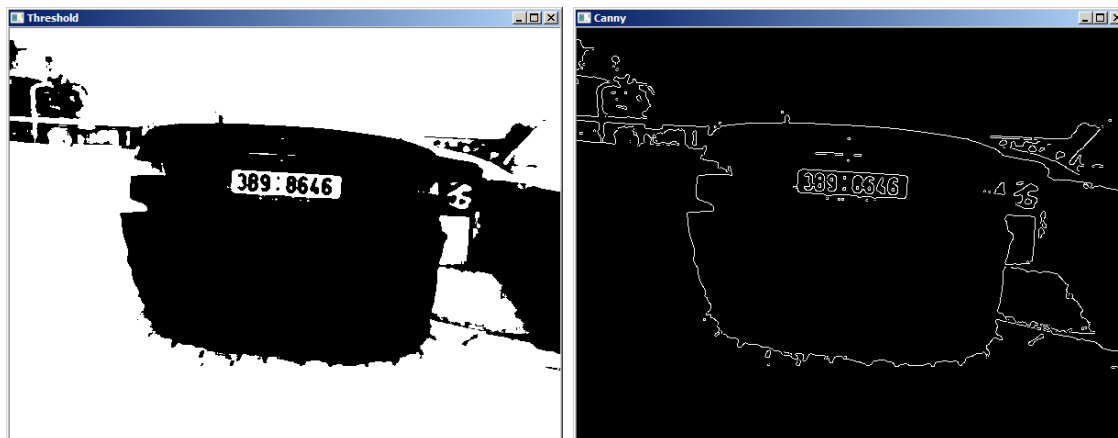
```
threshold(vstup_img_gray, vystup_img_threshold, 0, 255, CV_THRESH_OTSU);
```

Vstupem je černobílá fotografie a výstupem naprahovaný snímek.

Při testování všech fotografií byla úspěšnost nalezení správného prahu 71%. U této metody byl největší problém u aut s bílou resp. světlou barvou. Za denního světla fotografie obsahovala zvětší části bílou barvu a práh se stanovil takovým způsobem, že po prahování nebylo na snímku k rozeznání ani samotné auto, natož místo s RZ. Celkově se všemi testovanými světlými auty byl větší problém než s auty s tmavším odstínem.

Lepších výsledků se dosahovalo po implementaci Cannyho hranového detektoru do algoritmu. Důležité bylo stanovit, do jakého kroku tento detektor hran umístit. Jestliže se umístí za prahování metodou Otsu, výsledek to v případě světlých scén nezlepší, jelikož do Cannyho metody vstupují již zkreslená data. U ostatních fotografií se výsledky zlepší, což má přínos v přesnější detekci. Ve druhém případě kdy detekci hran provedeme jako prvního krok zpracování, nemá význam obraz následně prahovat a pouhé nalezení hran nestačí, takže tento způsob byl zavržen hned v počátku. Pro zlepšení výsledku testování světlých scén byl experimentálně zjištěn nejvhodnější způsob a to následující. Jako první se snímek naprahuje pevným prahem s nižší hodnotou mezi 100 až 125 a následně se aplikuje Cannyho detektor. Následné otestování navržené metody vykazovalo docela dobré výsledky. Po úpravě algoritmu byla z celkového počtu 19 světlých aut správně nalezena oblast obsahující RZ v 15 případech. Následný zdrojový kód zobrazuje umístění pevného prahování a Cannyho detektoru v testované aplikaci. Na obr. 7.4 je výsledek segmentace.

```
threshold(img_gray, img_threshold, 120, 255, THRESH_BINARY);  
Canny(img_threshold, img_canny, 0, 255, 3, FALSE);
```



Obr. 7.4: Prahování Otsu metodou a Canny detektor

Poslední způsob hledání prahu spočívá ve dvojitým či více násobném cyklickém opakování. V podstatě jde o to, že se nastaví pevný práh s hodnotou např. 200 a provede se detekce. Jestliže je výsledek nulový, provádí se prahování znovu s jinou, nyní nižší hodnotou např. 150. Tento způsob lze provádět vícekrát, avšak je důležité mít na paměti, aby se program nezacyklil v případě nenalezení žádné takové oblasti. Ve srovnání s propracovanější metodou Otsu nemá tento způsob význam aplikovat.

7.3.2 Zjištění plochy značky a klasifikace znaků

Uvažujeme pouze nalezené oblasti zjištěné na základě předem definovaných rozměrů. Hledaná oblast může být značka, která má poměr stran přibližně $520/110 = 4,727272$ (šířka/výška značky), s procentuální odchylkou 60 % a prostor minimálně 20 bodů a maximálně 125 bodů vysoký. Tyto hodnoty jsou vypočítány v závislosti na velikosti obrazu a polohy snímacího zařízení. Taty parametry byly zadány do zdrojového kódu a následně experimentálně vyhodnoceny.

```
float error=0.6;
float aspect=4.7272;
int min= 20*aspect*20;
int max= 125*aspect*125;
```

U každé nalezené kontury ověříme pravdivost, zda se může jednat o znak nebo ne. Podle známých rozměrů značky a jednotlivých symbolů na ní, jsme schopni zadat podmínky existence [7]. V našem případě jsou znaky 42 mm široké a 73 mm vysoké, jak je uvedeno na obr. 6.3. Poměr stran je tedy $42/73 = 0,57534$ a můžeme přijmout 35% chybu. Důležitým úkolem u rozpoznání jednotlivých znaků je nalézt správné kontury a následné použití umělých neuronových sítí pro klasifikaci jednotlivých vlastností ve třídě znaků.


```
float aspect=0.57534  
float charAspect= (float)r.cols/(float)r.rows;  
float error=0.35;  
float minHeight=15;  
float maxHeight=28;  
float minAspect=0.2;  
float maxAspect=aspect+aspect*error;
```



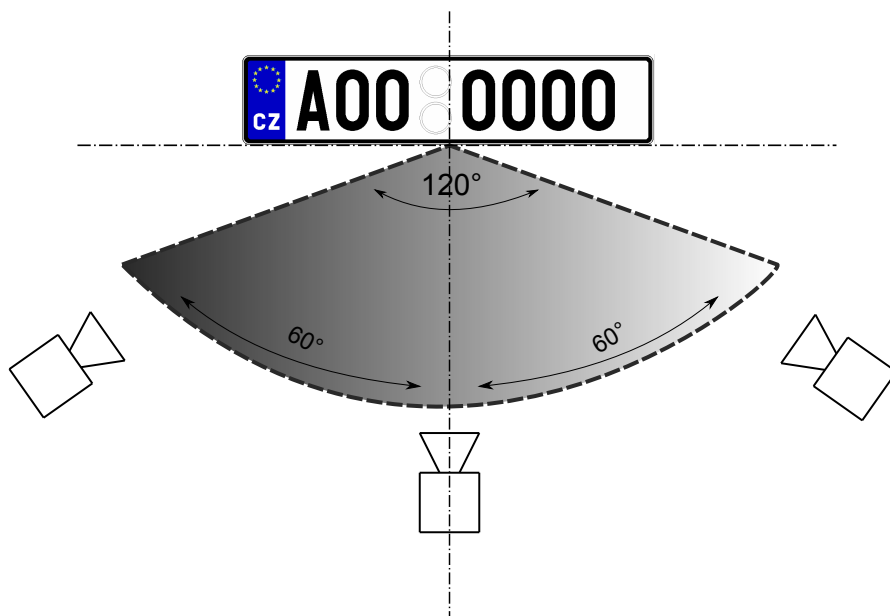
Obr. 7.5: Prahování nalezené oblasti a rozpoznání znaků

8 PODMÍNKY NASAZENÍ

Pro správnou funkčnost programu si musíme předem stanovit podmínky použitelnosti, abychom se v závěru práce vyhnuli případným chybám. Podmínek je celá řada, budeme se snažit poznamenat ty nejdůležitější.

8.1 Předem stanovená pravidla

- Při detekci registrační značky vozidla se budeme zaměřovat pouze na deklarovaný typ 101 (obr. 5.1) podle [10] jak bylo uvedeno v předchozí kapitole. Jedná se pouze o česká vozidla s „novou“ poznávací značkou.
- Úhel pohledu na značku při zachycování snímku bude stanoven v průběhu práce. Budeme se snažit vyhodnotit snímek s co nejvyšší přesností pod vysokým úhlem z každého směru. Grafické znázornění úhlů je vyznačeno na obrázku č. 8.1. Stanovený limit je 60° každým směrem od kolmého pohledu na značku.



Obr. 8.1: Úhel detekce

- Snímky, které budou pořízeny pro detekci musí být pořízeny za dne, pomocí denního světla. Foto pořízené za tmy, v šeru nebo při špatném osvětlení nebudou brány v potaz.

- Důležitou vlastností snímání je také ostrost. Většina současných fotoaparátů umožňuje ostřit i nastavovat expozici automaticky. Maximální vzdálenost detekovaného objektu od snímače bude do 5m (bez dalšího přiblížení pomocí snímače) .

8.2 Skutečné vlastnosti použití

V předchozí kapitole 8.1 jsme si na začátku práce stanovili podmínky, kterých bychom chtěli dosáhnout pro správnou funkčnost. Prvního kroku bylo dosaženo implementací vlastností a rozměrů naší registrační značky. Přesné rozměry jsou důležité pro nalezení oblasti RZ a hlavně pro určení jednotlivých znaků.

V dalším bodě jsme si stanovily maximální úhel detekce. Tohoto cíle nebylo dosaženo. Při testování jsme zjistili, že je metoda schopna rozpoznat RZ pod úhlem větší jak 30° pouze v okrajových případech. Při změně úhlu pohledu na značku se rozměry mění a algoritmus tuto plochu není schopen detekovat.

Také jsme si stanovili vlastnosti pořízených fotografií. Fotografování automobilů, jak již bylo řečeno, probíhalo za různých klimatických podmínek a výsledky byly následující. Snímky pořízené za tmy pomocí blesku měly výborné vlastnosti, jelikož RZ obsahuje reflexní materiál a ve výsledku na fotografii „zářila“ oblast umístění značky. Naopak fotografie bez blesku neměly žádný význam.

Maximální vzdálenost detekovaného objektu od snímače je 2–3 metry. Větší vzdálenost se v průběhu testování projevila jako nevyhovující.

9 VÝSLEDKY STUDENTSKÉ PRÁCE

9.1 Výsledky

Na následujících stranách jsou snímky, které nám aplikace vyhodnotila ať už úspěšně, nebo s chybou. Práce neobsahuje všechny registrační značky. Test byl prováděn ve stejný čas na stejném počítači. Procentuální zhodnocení je následující:

Puma.NET

- správně vyhodnocené RZ: 19%
- správně vyhodnocené RZ se zanedbatelnou chybou: 63%
- špatně vyhodnocené RZ: 18%

Automatic Number Plate Recognition

- správně naprahované snímky: 71%
- správně vyhodnocené RZ: 62%
- špatně vyhodnocené RZ: 38%

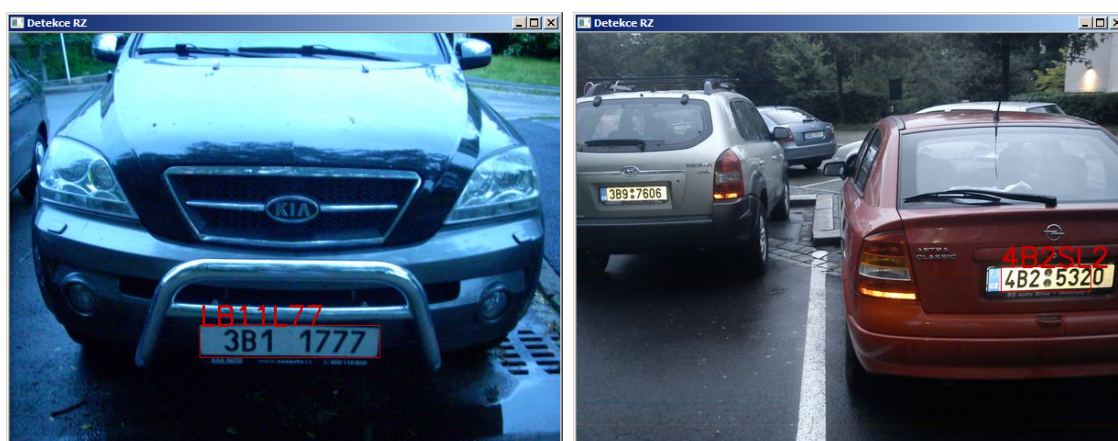
9.2 Dosažené výsledky



Obr. 9.1: Výsledky úspěšných detekcí registrační značky



Obr. 9.2: Výsledky úspěšných detekcí registrační značky s použitím blesku



Obr. 9.3: Chybná detekce registrační značky

10 ZÁVĚR

Úkolem této bakalářské práce bylo po prostudování možností detekce státní poznávací značky, následné implementování vhodné metody ve vybraném programovacím jazyku a stanovit podmínky funkčnosti. V teoretickém rozboru jsme se seznámili se zpracováním obrazu, jeho předzpracováním a segmentací. Aplikovali jednoduché metody na úpravu obrazu jako např. detekci hran a prahování.

Při použití knihovny Puma.NET jsme vše v kapitole 7.2 zhodnotili. Výsledky jsou dobré a knihovna je vhodná pro další použití. Podmínkou zůstává vhodné předzpracování vstupních dat. Podrobnější analýzu si vyžádala aplikace „Automatic Number Plate Recognition“. Zhodnocení je různorodé, záleží vše na vstupních snímcích. Výborně si obstály pečlivě pořízené snímky, jejichž ukázka je na obr. 9.1. Dále fotografie vytvořené za tmy s pomocí blesku vykazovaly dobré výsledky (obr. 9.2).

Pro zlepšení efektivity metod zde byl popsán princip prahování, který obohacuje systém a následné zpracování je přesnější. Velikou nevýhodou u fotografií pořízených pod úhlem je špatná nebo žádná detekce. Oblast značky mění své proporce a navržená metoda není schopná chybu eliminovat. Jedná se však o propracovaný systém, který vykazuje dobré výsledky. Vhodné by bylo spojení obou metod do jedné, a to tak, že by první část programu našla oblast RZ, vhodně ji upravila (což nyní provádí ANPR aplikace) a výstup z ní by vstupoval do OCR systému Puma.NET. Tohoto kroku již nebylo docíleno, jelikož se každá aplikace vytvářela v odlišném programovacím jazyku. Z výsledků testu je patrné, že metoda pracuje spolehlivě a má praktické využití.

LITERATURA

- [1] HLAVÁČ, Václav, SEDLÁČEK, Miloš. *Zpracování signálů a obrazů*. 1. vyd. Praha: Vydavatelství ČVUT, 2000, 221 s. ISBN 80-01-02114-9.
- [2] DOBEŠ, Michal. *Zpracování obrazu a algoritmy v C#*. 1. vyd. Praha: BEN – technická literatura, 2008, 143 s. ISBN 978-80-7300-233-6.
- [3] STRAKA, Stanislav. *Segmentace obrazu: diplomová práce*. Brno: Masarykova univerzita, Fakulta informatiky, 2009, 57 s.
- [4] HLAVÁČ, Václav a Milan ŠONKA. *Počítačové vidění*. Praha: Grada, 1992, 272 s. ISBN 80-854-2467-3.
- [5] DOBROVOLNÝ, M. *Rychlý algoritmus rozpoznání registračních značek vozidel*. Perner's Contacts, 2009, no. 15, s. 35-46. ISSN: 1801-674X.
- [6] LAGANIÈRE, Robert. *OpenCV 2 computer vision application programming cookbook*. 1. vyd. Brimingham: Packt Publishing, 2011, 287 s. ISBN 978-1-84951-324-1.
- [7] Baggio, D.; Emami, S.; Escrivá, D.; Ievgen, K.; Mahmood, N.; Saragih, J.; Shilkrot, R. *Mastering OpenCV with Practical Computer Vision Projects*. 1. vyd. Brimingham: Packt Publishing, 2012, 340 s. ISBN 978-1-84951-782-9.
- [8] MICROSOFT CORPORATION. *Microsoft Visual Studio – Vývoj od návrhu až po nasazení* [online]. 2012 [cit. 2012-12-01]. Dostupné z: <http://www.microsoft.com/cze/msdn/vstudio>
- [9] ITSEEZ: OpenCV Developers Team. *OpenCV* [online]. 2012 [cit. 2012-12-01]. Dostupné z: <http://opencv.org>
- [10] Česká republika. Registrační značky platné podle 243/2001 Sb. *243/2001* 2006. Dostupné z: <http://www.mdcz.cz/NR/rdonlyres/5C50406D-6CEE-4F03-BFD5-E29575BA674F/0/2regzna.pdf>
- [11] Česká republika. Sbírka zákonů 334/2006 Sb. *334/2006* 2006. Dostupné z: <http://aplikace.mvcr.cz/sbirka-zakonu/ViewFile.aspx?type=c&id=4949>
- [12] Telematické systémy - obecné informace. [online]. [cit. 2012-12-10]. Dostupné z: <http://www.dopravniinfo.cz/obecne-informace>
- [13] VIRIUS, M. *Jazyky C a C: kompletní kapesní průvodce programátora*. 1. vyd. Praha: Grada, 2006, 518 s. ISBN 80-247-1494-9.

- [14] PARKER, James. *Algorithms for image processing and computer vision*. New York: Wiley Computer Publishing, 1997, 417 s. ISBN 04-711-4056-2.
- [15] LIBERTY, Jesse. *Naučte se C++ za 21 dní: kompletní kapesní průvodce programátora*. 2. aktualiz. vyd. Brno: Computer Press, 2007, 796 s. ISBN 978-80-251-1583-1.

SEZNAM PŘÍLOH

A Přílohy	42
A.1 Přiložené CD	42

A PŘÍLOHY

A.1 Přiložené CD

Obsah CD:

- Bakalářská práce – David Smékal.pdf (elektronická verze BP)
- ANPR (složka s programem, zdrojovými kódy a testovacím foto)
- Puma (složka s programem, zdrojovými kódy a testovacím foto)
- Ukázky (složka s výstupními snímky)