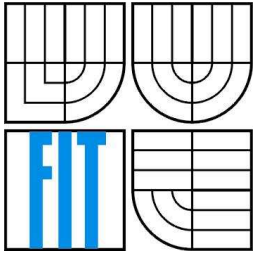


VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY



FAKULTA INFORMAČNÍCH TECHNOLOGIÍ
ÚSTAV POČÍTAČOVÉ GRAFIKY A MULTIMÉDIÍ

FACULTY OF INFORMATION TECHNOLOGY
DEPARTMENT OF COMPUTER GRAPHICS AND MULTIMEDIA

DETEKCE POZNÁVACÍ ZNAČKY V OBRAZE

IMAGE-BASED LICENCE PLATE RECOGNITION

BAKALÁŘSKÁ PRÁCE
BACHELOR'S THESIS

AUTOR PRÁCE
AUTHOR

MILOŠ HOMOLA

VEDOUCÍ PRÁCE
SUPERVISOR

Ing. VÍTĚZSLAV BERAN

BRNO 2007

ZADÁNÍ

1. Prostudujte dostupnou literaturu týkající se tématu. Proved'te rozbor dosavadního stavu problematiky.
2. Na základě nastudovaných informací navrh'nete postup řešení zadání. Diskutujte zvolené metody a vysvětlete navržené řešení.
3. Vytvořte anotovanou testovací sadu dat pro závěrečné testování Vašeho řešení.
4. Navrh'nete a implementujte knihovnu realizující Vámi navržený postup. Dbejte na programovou čistotu, modulárnost a přenositelnost knihovny. Knihovnu dokumentujte.
5. Vyhodno'te Vaše řešení pomocí testovacích dat, srovnajte Vaše výsledky s výsledky jiných prací a proved'te jejich diskuzi.

LICENČNÍ SMLOUVA POSKYTOVANÁ K VÝKONU PRÁVA UŽÍT ŠKOLNÍ DÍLO

uzavřená mezi smluvními stranami:

1. Pan/paní

Jméno a příjmení:

Bytem:

Narozen/a (datum a místo):

(dále jen „autor“)

a

2. Vysoké učení technické v Brně

Fakulta informačních technologií

se sídlem Božetěchova 2, 612 66, Brno

jejímž jménem jedná na základě písemného pověření děkanem fakulty:

.....

(dále jen „nabyvatel“)

Čl. 1 Specifikace školního díla

1. Předmětem této smlouvy je vysokoškolská kvalifikační práce (VŠKP):

- disertační práce
 - diplomová práce
 - bakalářská práce
 - jiná práce, jejíž druh je specifikován jako
- (dále jen VŠKP nebo dílo)

Název VŠKP:
Vedoucí/ školitel VŠKP:
Ústav:
Datum obhajoby VŠKP:

VŠKP odevzdal autor nabyvateli v*:

- tištěné formě – počet exemplářů
- elektronické formě – počet exemplářů

* hodící se zaškrtněte

2. Autor prohlašuje, že vytvořil samostatnou vlastní tvůrčí činností dílo shora popsané a specifikované. Autor dále prohlašuje, že při zpracovávání díla se sám nedostal do rozporu s autorským zákonem a předpisy souvisejícími a že je dílo dílem původním.
3. Dílo je chráněno jako dílo dle autorského zákona v platném znění.
4. Autor potvrzuje, že listinná a elektronická verze díla je identická.

Článek 2

Udělení licenčního oprávnění

1. Autor touto smlouvou poskytuje nabyvateli oprávnění (licenci) k výkonu práva uvedené dílo nevýdělečně užít, archivovat a zpřístupnit ke studijním, výukovým a výzkumným účelům včetně pořizování výpisů, opisů a rozmnoženin.
2. Licence je poskytována celosvětově, pro celou dobu trvání autorských a majetkových práv k dílu.
3. Autor souhlasí se zveřejněním díla v databázi přístupné v mezinárodní síti
 - ihned po uzavření této smlouvy
 - 1 rok po uzavření této smlouvy
 - 3 roky po uzavření této smlouvy
 - 5 let po uzavření této smlouvy
 - 10 let po uzavření této smlouvy(z důvodu utajení v něm obsažených informací)
4. Nevýdělečné zveřejňování díla nabyvatelem v souladu s ustanovením § 47b zákona č. 111/ 1998 Sb., v platném znění, nevyžaduje licenci a nabyvatel je k němu povinen a oprávněn ze zákona.

Článek 3

Závěrečná ustanovení

5. Smlouva je sepsána ve třech vyhotoveních s platností originálu, přičemž po jednom vyhotovení obdrží autor a nabyvatel, další vyhotovení je vloženo do VŠKP.
6. Vztahy mezi smluvními stranami vzniklé a neupravené touto smlouvou se řídí autorským zákonem, občanským zákoníkem, vysokoškolským zákonem, zákonem o archivnictví, v platném znění a popř. dalšími právními předpisy.
7. Licenční smlouva byla uzavřena na základě svobodné a pravé vůle smluvních stran, s plným porozuměním jejímu textu i důsledkům, nikoliv v tísní a za nápadně nevýhodných podmínek.
8. Licenční smlouva nabývá platnosti a účinnosti dnem jejího podpisu oběma smluvními stranami.

V Brně dne:

.....
Nabyvatel

.....
Autor

ANOTACE

Tento projekt se zabývá automatickou detekcí značky vozidla založené na hledání úseček ve vstupním obraze. V úvodu je popsán hlavní problém současných metod a stanovav cíl práce. Následuje popis několika základních metod, které se danou problematikou zabývají. Jsou diskutovány jejich výsledky a každá metoda je hodnocena. Závěrem této kapitoly je na základě získaných informací a vzhledem k cílu práce navržen algoritmus pro detekci značky založené na hledání úseček. Další kapitoly se věnují podrobnému popisu mé práce. Je kladen důraz na přesnou a spolehlivou detekci úseček a následně jejich seskupování do konzistentních celků. Na konci práce je provedeno několik experimentů pro demonstraci účinnosti a spolehlivosti zvolené metody. V závěru jsou diskutovány dosažené výsledky a je hodnocena vhodnost tohoto přístupu pro detekci značky vozidla.

KLÍČOVÁ SLOVA

Detekce značky, Houghova transformace.

ABSTRACT

This project is concerned with automatic detection of licence plate based on finding abscissa in entering image. In the preface the main problem of presents' methods is described. Then the definition of some basic methods, which are concerned with mentioned problematic, follows. Their results are discussed and each way of solution is assessed. In conclusion of the chapter, the algorithm for detection of licence plate is designed. It is done on the base of gained information and because of the aim of the work. Next chapter is devoted to the detailed description of my work. It is underlined emphasis on exact and reliable detection of abscissae and following clustering into consistent units. In the end of the work a few experiments for demonstration of efectiveness and reliability of selected method are made. In conclusion, gained results are discussed and the convience of this approach of detection of numberplate is assessed.

KEY WORDS

Local Plate Recognition, Hough transform.

PROHLÁŠENÍ

Prohlašuji, že svou bakalářskou práci na téma "Téma práce" 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 jsem si plně vědom následků porušení ustanovení § 11 a následujících autorského zákona č. 121/2000 Sb., včetně možných trestněprávních důsledků vyplývajících z ustanovení § 152 trestního zákona č. 140/1961 Sb.

V Brně dne

.....

(podpis autora)

PODĚKOVÁNÍ

Děkuji vedoucímu bakalářské práce Ing. Vítězslavu Beranovi, za užitečnou metodickou a odbornou pomoc a další cenné rady při zpracování mé bakalářské práce.

V Brně dne

.....

(podpis autora)

Obsah

1. ÚVOD	1
1.1. DEFINICE PROBLÉMU	1
1.2. CÍL PRÁCE.....	1
2. CHARAKTERISTIKA VYBRANÝCH METOD PRO DETEKCI ZNAČKY.....	2
2.1. DETEKCE ZNAČKY ZALOŽENÁ NA HISTOGRAMOVÁNÍ A MATEMATICKÉ MORFOLOGII	2
2.2. METODA PRAHOVÁNÍ VSTUPNÍHO OBRÁZKU.....	5
2.3. METODA VERTIKÁLNÍCH HRAN A HORIZONTÁLNÍ ZÁVORY	6
2.4. SČÍTÁNÍ GRADIENTU JASOVÉ FUNKCE.....	8
2.5. NÁVRH ALGORITMU.....	9
3. DETEKCE ÚSEČEK V OBRAZE.....	11
3.1. DETEKCE HRAN	11
3.2. HOUGHOVA TRANSFORMACE.....	12
4. SHLUKOVÁNÍ ÚSEČEK.....	14
4.1. CLUSTERIZACE HOUGHOVA PROSTORU.....	15
4.2. METODA POROVNÁVÁNÍ PŘÍMEK	18
4.3. SHLUKOVÁNÍ PŘÍMEK S VYUŽITÍM HISTOGRAMU SOUSEDNOSTI.....	19
4.4. NAHRAZENÍ CLUSTERU MINIMÁLNÍM POČTEM ÚSEČEK	22
5. EXPERIMENTY	26
6. ZÁVĚR	28
7. SEZNAM LITERATURY.....	29
8. SEZNAM PŘÍLOH	30
9. SEZNAM OBRÁZKŮ	30

1. Úvod

V současné době je velice populární automatické rozpoznávání vozidel pomocí detekce jejich registrační značky. Algoritmů, které celou problematiku řeší, je v dnešní době celá řada a každý většinou pracuje na mírně odlišném principu a využívá jiných rysů charakteristických pro hledanou oblast.

1.1. Definice problému

Většina metod pro detekci registrační značky je založena na několika základních typech informace. Jsou to například jasová informace, velikost, hustota či odchylka gradientu obrazu a často je kamera pro snímání obrazu umístěna v předem známých světelných podmínkách a snímaná vozidlo projíždí předem definovanou oblastí (např. jízdní pruh) nebo stojí na určitém místě (parkovací pole), což detekci umožňuje implementovat mnohem efektivněji a algoritmus pracuje rychle.

1.2. Cíl práce

U algoritmů, které jsem měl možnost prostudovat, převládá jeden společný limitující rys. Značky jsou spolehlivě detekovatelné, pakliže jsou umístěny rovnoběžně se spodním okrajem obrazu. Pokud bychom snímanou značku natočili, celá detekce bude stále méně přesná a spolehlivá. Právě tento nedostatek se ve své práci snažím eliminovat.

Jelikož se mi nepodařilo nalézt žádnou studii či jiný materiál zabývající se nebo alespoň hodnotící metodu detekce registrační značky založenou na hledání přímek v obraze, rozhodl jsem se implementovat metody, které umožní detekci významných úseček, které se stanou základem pro zjištění pozice celé značky. Cílem je, aby výsledné úsečky byly konzistentní a pokud možno co nejdelší.

2. Charakteristika vybraných metod pro detekci značky

Tato kapitola je určena k popisu metod, které se problematikou detekce značky zabývají. Kromě popisu principu činnosti každé metody se pokusím zhodnotit její vlastnosti jako robustnost, rychlost a implementační náročnost. Na konci této kapitoly se pokusím navrhnout algoritmus, který odstraňuje určité nevýhody popsanych metod.

2.1. Detekce značky založená na histogramování a matematické morfologii

Jednou z nejjednodušších metod detekce značky je metoda založená na histogramování a matematické morfologii. Využívá jednu z typických vlastností pro každou značku, a to sice skutečnost, že v oblasti, kde se značka vyskytuje dochází k vysokému rozdílu kontrastu (většinou černé písmo na bílém podkladu).

Popis

Vlastní proces detekce má dvě části – hrubá a přesná detekce. V první fázi jsou hledány kandidátní oblasti, které jsou u této metody tvořeny horizontálním výřezem obrazu. Ve druhé fázi jsou tyto kandidátní oblasti lépe analyzovány a je určena přesná pozice značky.

Hrubá detekce

Vlastní proces hrubé detekce spočívá využití již zmíněného vysokého kontrastu v oblasti značky. Lze vypočítat průměrnou hodnotu jasu a následně provést rozdíl tohoto průměru od jasové hodnoty každého bodu obrazu. V oblasti značky budou tyto rozdíly nejvíce patrné. Můžeme vypočítat vertikální gradient pomocí vzorce 1.

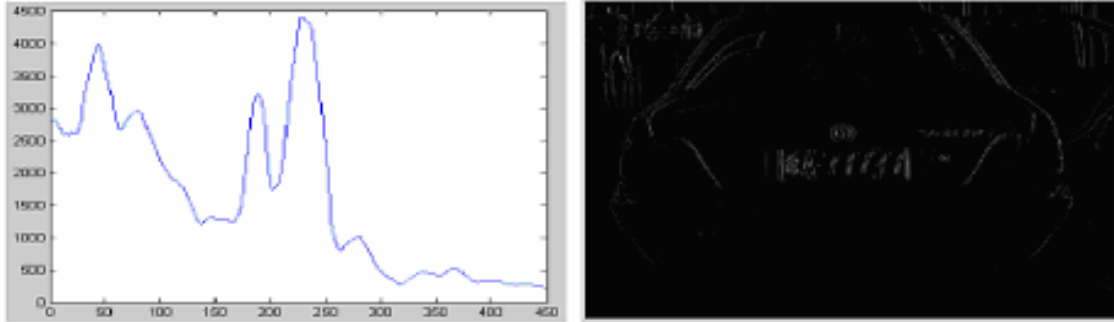
$$g_v(i, j) = |f(i, j-1) - f(i, j)| \quad (1)$$

kde g_v je funkce gradientu a $f(i, j)$ představuje hodnotu pixelu obrazu o souřadnicích $[i, j]$.

Abychom byli schopni detekovat část obrazu, kde se značka nachází, využijeme právě vertikálního gradientu, přičemž počítáme jeho horizontální projekci pomocí vzorce 2. Výstup této operace (vyhlazený Gaussovým filtrem) ukazuje obr. 3.

$$T_H(i) = \sum_{j=1}^n g_v(i, j) \quad (2)$$

kde TH je hodnota projekce.



Obr. 1: Horizontální projekce gradientu a zdrojový obraz gradientu

Protože v oblasti značky lze očekávat vždy vysoký rozdíl gradientu, budeme hledat v získané horizontální projekci lokální maxima (oblast značky nemusí vždy nutně korespondovat s maximálním výběžkem v grafu projekce). Tato lokální maxima a jejich (vhodně zvolené) blízké okolí označíme jako kandidátní oblasti.



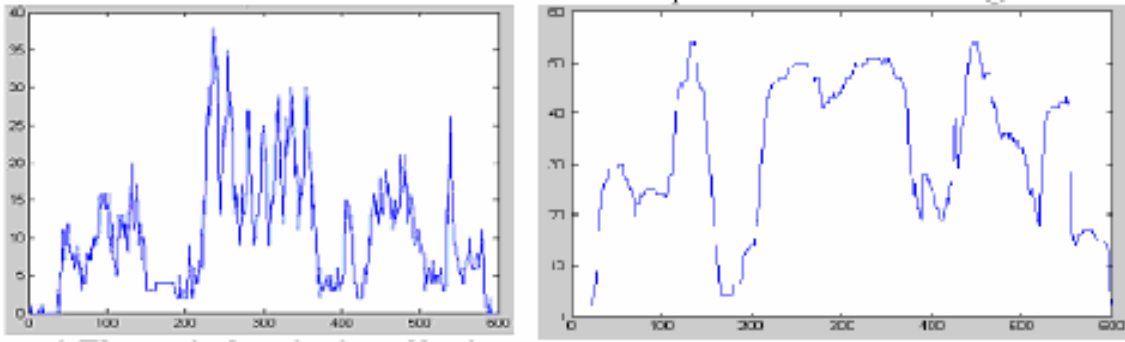
Obr. 2: Kandidátní oblast vyňatá z originálního obrazu

Přesná detekce

Abychom z kandidátní oblasti získali značku, využijeme základních morfologických operací (dilatace, eroze). Vypočítáme horizontální gradient pomocí vzorce 3 a jeho vertikální projekci pomocí vzorce 4. Výsledek vertikální projekce je zobrazen v obr. 3.

$$g_H(i, j) = |f(i+1, j) - f(i, j)| \quad (3)$$

$$T_V(j) = \sum_{i=1}^m g_H(i, j) \quad (4)$$



Obr. 3: Vertikální projekce gradientu před a po aplikaci morfologických operátorů

V tomto kroku je použito tzv. uzavřené aritmetiky, abychom dosáhli odlišení jednotlivých částí grafu (obr. 3). Detailně je princip aplikace morfologických popsán v [6].

Poslední fází postupu detekce značky popisovaného v této podkapitole je prahování a binarizace. Výstup uvedený v obr. 3 je podroben prahování podle vzorců 5 a 6, přičemž je tvořen binární obraz (obr. 4).

$$f_T(1,i) = 1 \quad \text{pro} \quad f_T(1,i) \geq T \quad (5)$$

$$f_T(1,i) = 0 \quad \text{pro} \quad f_T(1,i) \leq T \quad (6)$$

Kde $f_T(1,i)$ je vertikální projekce po provedení matematické morfologie a T je práh. Klíčovým problémem bývá problém jak zvolit hodnotu prahu – ta je určena pomocí vzorce 7.

$$T = t * aver \quad (7)$$

Kde $aver$ je průměrná hodnota $f_T(1,i)$ a „ t “ váhový parametr. Pomocí experimentů autoři algoritmu určili hodnotu $t=1.23$.



Obr. 4: Binární obraz

Bílé místa v obr. 4 vyznačují potenciální výskyty značky. Tyto oblasti jsou dále zkoumány a analyzovány, aby bylo možné určit pravděpodobnost, s jakou mohou skutečně reprezentovat značku. Kritérii pro tuto analýzu je např. poměr stran (výška, šířka) atd. Tento postup aplikujeme na všechny

kandidátní oblasti jednu po druhé a vybereme kandidáta s nejvyšší pravděpodobností výskytu značky – nebo žádného, pokud výsledky nesplňují podmínky výskytu značky (špatný poměr výška/šířka apod.).

Hodnocení metody

Výhody – metoda (podle experimentů uváděných v původní dokumentaci [7]) se zdá být poměrně rychlá a spolehlivá. Snadno lze pochopit její základní princip.

Nevýhody – v původní dokumentaci jsou nejasně nebo neúplně popsány rozhodující kroky jako např. výběr vhodné kandidátní oblasti z více nalezených. Metoda je také koncipována, aby spolehlivě detekovala značku umístěnou vodorovně se spodním okrajem obrazu. Při případném natočení se značně snižuje robustnost metody. Stejně tak pokud bude obraz obsahovat rušivé pozadí nebo jiné vysoko-contrastní vzory, může nastat situace, že značka se v grafu horizontální projekce gradientu neprojeví jako lokální maximum a nebude nalezena.

2.2. Metoda prahování vstupního obrázku

Další metodou využívající pro detekci značky jasovou informaci obrazu je metoda založena na prahování vstupního obrázku za účelem získání binárního obrázku, kde znaky značky jsou černé a pozadí bílé.

Princip

Prahaný obrázek se prochází od spoda vzhůru. Nalezne-li se černá oblast, obchází se kolem dokola za účelem zjištění šířky a výšky oblasti. Vyhovuje-li celkový rozměr a poměr stran (aspekt) oblasti předpokladům, je tato označena jako možný znak registrační značky. Jestliže je nalezeno dostatečně množství takovýchto oblastí protínajících jeden řádek obrázku, je toto místo označeno jako místo výskytu značky. Jestliže žádné takovéto místo nebylo nalezeno, zvýšíme práh a proceduru opakujeme. Jestliže ani pro všechny prahy nenalezneme odpovídající místo, vstupní obrázek neobsahuje žádnou značku. Tento postup je podrobněji popsán v [7].

Pro správnou funkčnost tohoto algoritmu je nutné splnit několik podmínek:

- Znaky značky musí být možné segmentovat prahováním. To vyžaduje stálé podmínky osvětlení přes celou značku a dobrý kontrast obrázku značky.
- Znaky značky se nesmí dotýkat. Jestliže se dotýkají, potom jeden nebo více znaků může být prahováním spojeno. Spojením vznikne oblast, jejíž rozměry nebudou odpovídat požadovaným a značka nebude nalezena.
- Značka musí být bez trhlin, špíny, šroubů nebo jiných poškození, které by způsobily spojení znaků při prahování.

Hodnocení

Zaručení výše uvedených podmínek je v praxi nemožné. Jediná věc, kterou lze kontrolovat, je velikost značky v obrázku. Lze mít požadavek na velikost značky s ohledem na zvětšení kamery, což značně snižuje již tak nízkou robustnost celého algoritmu. Ani detekce pootočené značky nepřipadá v úvahu a proto je tato metoda pro použití v praxi i přes svoji zjevnou jednoduchost nevhodná.

2.3. Metoda vertikálních hran a horizontální závory

Uvedli jsme, že segmentace prahováním potřebuje pro správnou funkci v praxi nereálné podmínky snímání obrazu. Jestliže se vyhneme potřebě prahování, můžeme sestavit robustnější systém. Algoritmus popisovaný v této kapitole je založen na skutečnosti, že obrázek značky obsahuje význačný počet vertikálních hran.

Popis

Vertikální hrany jsou charakteristické velkým rozdílem jasu sousedících pixelů a jsou patrné jako skoky v horizontálním řezu značkou. Tento krok algoritmu identifikuje všechny části obrázku, které mohou reprezentovat značku.

Práci této části algoritmu lze popsat takto:

1. Na každém n -tém řádku proved' horizontální řez vstupním obrázkem. Číslo n nesmí být větší než nejmenší výška znaku. To zaručuje protnutí značky alespoň jedním horizontálním řezem.

2. Pro každý pixel z tohoto horizontálního řezu, jestliže je rozdíl mezi hodnotami jasu aktuálního pixelu a následujícího větší než nějaký práh T , označ tento aktuální pixel jako vertikální hranu.
3. Seskup všechny body vertikálních hran na daném horizontální řezu do skupin tak, že horizontální vzdálenost mezi body vertikálních hran ve stejné skupině je menší než šířka nejširšího očekávaného znaku. Toto funguje za předpokladu, že každý znak ve značce produkuje minimálně jednu vertikální hranu a vzdálenost mezi dvěma libovolnými znaky není větší než šířka znaku. Je-li vzdálenost mezi body dvou vertikálních hran větší než šířka znaku, předpokládá se, že tyto body musí být z jiných značek.
4. Spoj sousední skupiny do jedné. Každá skupina potom reprezentuje možnou značku. Pozice hranových bodů v každé skupině určují horizontální pozice mezer a levý a pravý extrém značky.

Nyní je nutné filtrovat jednotlivé skupiny bodů za účelem nalezení korektních značky:

1. Vezmi horizontální úsečku mezi levým a pravým extrémem značky a rozděl ji do několika stejných intervalů. Pro každý interval najdi horní a dolní hranici značky. Umíst'uj horizontální závoru postupně po celém intervalu a pro každou pozici posunuj postupně o jeden pixel tuto závoru vertikálně nahoru nebo dolů. Počítej přitom kolik vertikálních hran se nachází na této závoře. V místě nad znakem bude na závoře málo vertikálních hran (ideálně žádná). Maximální výška do které lze závoru posunovat odpovídá největšímu přípustnému znaku.
2. Metodou nejmenších čtverců urči přímky rovnoběžníka značky z odhadů horní hranice značky pomocí horizontální závory.

Hodnocení

Nespornou výhodou metody je její přímocharost, snadná pochopitelnost a patrně i nízká náročnost implementace. Nevýhodou této metody je potřeba znát předem určité informace, které se mohou při použití v praxi lišit. Jsou to například nejmenší výšku znaku nebo nejširší znak. Také podmínka, že každý znak produkuje minimálně jednu vertikální hranu nemusí být ve všech případech splněna, což může mít za následek nesprávnou detekci (může být detekována pouze část značky nebo úbec nic).

2.4. Sčítání gradientu jasové funkce

Charakteristika metody

Charakteristickým znakem poznávací značky je, že obsahují velký počet významných nespojností v relativně malé oblasti. To může být jednoduše změřeno součtem gradientů jasové funkce v malých obdélníkových oblastech. Využijeme k tomu rovnost:

$$S(x, y) = \sum_{i=x-\frac{x_s}{2}}^{i=x+\frac{x_s}{2}} \sum_{j=y-\frac{y_s}{2}}^{j=y+\frac{y_s}{2}} \sqrt{\left(\frac{\partial f}{\partial y}(x, y)\right)^2 + \left(\frac{\partial f}{\partial x}(x, y)\right)^2} \quad (8)$$

kde x_s a y_s je velikost malé obdélníkové oblasti.

Oblasti, které budou mít velký součet gradientů, mohou být potom vyšetřeny mnohem detailněji. Můžeme potvrdit nebo vyvrátit přítomnost značky a případně aproximovat její hranici.

Hodnocení

Bohužel tato technika pracuje správně pouze pro dostatečné rozlišení obrázku. Jestliže je rozlišení na značce malé, gradient má tendenci se zmenšovat vyhlazovacím filtrem, který je součástí většiny hranových detektorů. Naopak kdybychom filtr nepoužívali, objeví se přídavné gradienty způsobené šumem.

Tímto je uzavřena kapitola, která se věnovala především popisu již existujících metod, se kterými jsem měl možnost se seznámit. Pakliže se zaměříme obecně na princip výše popsaných způsobů detekce značky, lze najít jeden společný rys a to sice, že většina algoritmů se dostává do problémů ve chvíli, kdy dojde k natočení značky vzhledem ke spodnímu okraji obrazu. Tento problém se v mé práci snažím odstranit za použití postupů a principů popsaných dále v tomto dokumentu.

2.5. Návrh algoritmu

V této kapitole popíší návrh algoritmu, který si klade za cíl odstranit výše popsanou restrikcí většiny dnešních metod – problém detekovat značku pootočenou vůči okrajům obrazu. Aby tento cíl bylo možné splnit, nelze použít přístupů popsaných v předešlé kapitole nebo jim podobných. Proto jsem se rozhodl použít ve své práci jako základ Houghovu transformaci, která je vzhledem k natočení obrazu odolná a stejná vlastnost, jak jsem již předeslal, se od celého algoritmu očekává.

Preprocessing

V první fázi celého procesu detekce značky je vhodné uplatnit preprocessing v podobě ekvalizace histogramu. Jasové rozdíly v obraze jsou potom více patrné a je tak zlepšena následná detekce hran.

Detekce hran

Hrany jsou v mé práci detekovány pomocí Sobelova hranového detektoru. Celá detekce je rozdělena do dvou kroků, kde jsou odděleně detekovány hrany ve směru osy x a y . Výsledné obrazy jsou následně spojeny do jednoho celku, který poskytuje silnou odezvu pro všechny typy hran.

Prahování

Abychom mohli na hranový obraz aplikovat Houghovu transformaci, je nutné provést nejprve prahování, což je proces odstraňující vliv šumu a falešnou detekci čar.

Detekce čar

Základem pro detekci značky je správná, přesná a rychlá detekce čar, respektive úseček. Pro jejich detekci je základem Houghova transformace. Tyto úsečky jsou následně vhodně spojovány tak, aby vždy s jednou fyzickou hranou korespondovala v optimálním případě jediná úsečka.

Výběr kandidátů

Získané úsečky jsou dále analyzovány, jsou hledány jejich průsečíky a následně tvořeny uzavřené oblasti. Každá uzavřená oblast je třeba transformovat tak, že z původního obecně n-úhelníku získáme čtyřúhelník. Takto vznikne kandidátní oblast.

Analýza kandidátních oblastí

Posledním krokem detekce značky je analýza všech získaných kandidátních oblastí. Zde můžeme uplatnit několik rysů typických pro oblast značky, abychom byli schopni určit pravděpodobnost výskytu značky. Těmito rysy jsou poměr stran, poměr barev nalezených uvnitř oblasti (značka většinou obsahuje bílé pozadí a černé znaky), kontrastnost a další.

V následujícím textu a celé mé práci se budu soustředit především na detekci čar, protože se jedná o stěžejní a po všech stránkách (výpočetně, implementačně i principiálně) nejnáročnější a nejsložitější část celého výše popsaného algoritmu. Dalším důvodem je cíl celé mé práce – odstranit neodolnost proti pootočení. Tento problém lze vyřešit právě pomocí přesné a robustní detekce úseček, na které klademe několik specifických požadavků:

- Každá úsečka pokrývá právě jednu fyzickou hranu.
- Každá fyzická hrana je přesně určena optimálně jednou úsečkou.
- Úsečky se nepřekrývají.

Pokusil jsem se vyvinout několik metod, které se snaží výše uvedené vlastnosti splnit. V následující kapitole bude celá problematika podrobně popsána.

3. Detekce úseček v obraze

V této kapitole se pokusím popsat a zhodnotit celou svoji práci. Protože se mi nepodařilo najít žádný zdroj, který by se věnoval problematice detekce značky založené na detekci čar, rozhodl jsem se pojmout tuto práci jako část výzkumu, ve kterém se pokouším navrhnout vhodný způsob jak tuto problematiku řešit.

3.1. Detekce hran

Prvním krokem je detekce hran, která se skládá z několika dílčích bodů. Ty popíši v této podkapitole.

Preprocessing

Jelikož je detekce hran typicky založena na rozdílu jasů dvou sousedících bodů, je jako preprocessing aplikována na vstupní obraz ekvalizace histogramu, kdy využíváme beze zbytku celou šířku jasové stupnice a pakliže původní obraz nevyužívá toto pásmo celé, po ekvalizaci jsou jeho jasové rozdíly výraznější, což umožní lepší detekci hran.

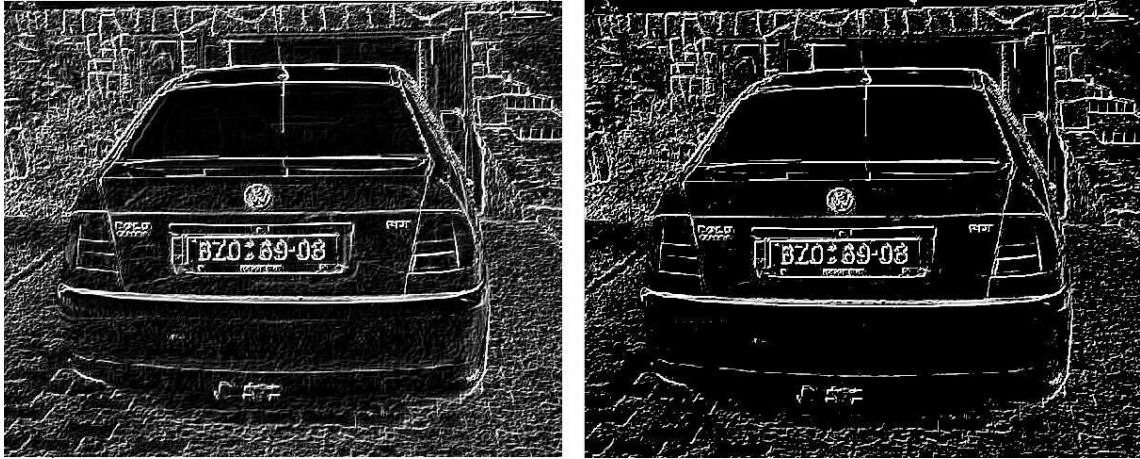
Hranový detektor

Detekce hran je základní krok, který umožňuje dále detekovat přímky nebo úsečky pomocí Houghovy transformace. Lze využít jeden ze dvou základních hranových detektorů – Sobelův nebo Cannyho. Pro účely této práce byl vybrán Sobelův hranový detektor pro jeho silnější odezvu.

Abychom dosáhli maximálně silné odezvy, je provedena odděleně detekce hran ve směru obou os. Ve druhé fázi jsou obrazy sečteny. Použití postupu, kdy bychom detekovali hrany ve směru obou os současně má za následek mnohem slabší odezvu, proto je přístup, který jsem v této práci zvolil vhodnější.

Prahování

Abychom v obraze před vlastní aplikací Houghovy transformace ponechali pouze skutečné hrany a obraz očistili od šumu, lze použít metodu prahování. Během prahování je spočítána průměrná hodnota jasů obrazu, na základě které je zvolen práh. Veškeré body, jejichž hodnota je menší než hodnota prahu jsou nulovány.



Obr. 5: Originální a prahovaný obraz

3.2. Houghova transformace

Houghova transformace aplikovaná na výstup získaný z některého hranového detektoru umožňuje vyjádřit hrany analyticky ve formě přímk v normálovém tvaru (standardní Houghova transformace) nebo úseček definovaných jejich koncovými body (tzv. pravděpodobnostní Houghova transformace).

Obecná aplikace

V první řadě se při používání Houghovy transformace musíme rozhodnout, na který z hranových detektorů ji budeme aplikovat.



Obr. 6: Nedetekované úsečky

V této práci je Houghova transformace aplikována na Sobelův detektor.. Na obrázku 6 je znázorněna odezva Houghovy transformace při obecném použití (červenou barvou jsou vykresleny detekované úsečky). Pro detekci úseček byla použita pravděpodobnostní varianta Houghovy transformace.

Lze si všimnout, že některé hrany zůstávají houghovou transformací nedetekovány (některé jsou označeny zelenou elipsou). Je to způsobeno velikostí oblasti, z níž jsou akumulovány jednotlivé body Houghovou transformací do houghova prostoru. Intenzita šumu je potom tak vysoká, že její odezva je vyšší, než odezva – byť jen krátké – hrany. Tyto nedostatky odstraňuje další krok – aplikace Houghovy transformace pouze na část obrazu pomocí nastavení tzv. okna zájmu (Region Of Interest – ROI).

Nastavení okna zájmu (ROI)

Tímto procesem se značně eliminuje vliv šumu, poněvadž oblast, která přispívá do Houghova prostoru je menší a hrana, která nebyla dříve detekována vlivem odezvy šumu se nyní projeví jako lokální maximum v Houghově prostoru.

Nevýhodou tohoto přístupu může být skutečnost, že velikost okna ROI v podstatě definuje maximální délku nadetekované úsečky. Pokud se v obraze vyskytuje úsečka, jejíž délka je větší než velikost okna ROI, bude fragmentována na více částí. Na následujících obrázcích je patrný rozdíl oproti výstupům bez použití okna ROI.



Obr. 7: Aplikace okna zájmu

Pravděpodobnostní variantu Houghovy transformace aplikujeme v několika cyklech na část obrazu získaného jako výstup Cannyho hranového detektoru. Po skončení každého průchodu získáme množinu úseček pro dané podokno. Než bude program pokračovat ve zpracování dalšího podokna, je nutné nejprve (jelikož souřadnice koncových bodů – kterými je úsečka určena – mají svůj počátek právě v levém horním rohu podokna) přepočítat souřadnice koncových bodů, aby jejich počátkem byl levý horní roh celého obrazu – tedy přičteme k souřadnicím koncových bodů posunutí okna ROI.

Úsečky, které jsme v předchozím kroku úspěšně detekovali trpí několika nedostatky:

- Jsou fragmentované – jednu skutečnou dlouhou hranu může detekce znázornit jako několik menších úseček.
- Překrývají se – tato vlastnost je způsobena posuvem okna ROI, kde platí, že krok posunutí ve směru osy x je vždy menší než šířka okna ROI.
- Mezi úsečkami, které by měly definovat jednu společnou hranu mohou být mezery. Výsledná hrana tedy není pokryta úplně.

Cílem je, aby každou hranu pokrývala nejvýše jedna úsečka, přičemž množina hran, které jsou od sebe velice málo vzdáleny mohou být reprezentovány jedinou úsečkou, která svojí polohou a velikostí reprezentuje všechny hrany z této množiny s určitou tolerancí. Za tímto účelem musíme podniknout následující kroky:

- Převod úseček na přímky v normálovém tvaru
- Sloučení přímek ležících v určitém blízkém okolí
- Analýza získaných přímek a zpětný převod na konzistentní úsečky

V následujícím textu budou tyto důležité kroky podrobněji popsány.

4. Shlukování úseček

Další fází mého algoritmu následující po základní detekci úseček je proces jejich shlukování a následné nahrazení více úseček jedinou. Do této chvíle je každá hrana pokryta velkým množstvím krátkých úseček. V této kapitole budou metody, které tento problém odstraňují, popsány a u každé bude provedena stručná diskuse výsledků.

4.1. Clusterizace houghova prostoru

Jelikož jsou všechny přímky v normálovém tvaru, lze je tedy vyobrazit ve dvourozměrném prostoru, kde první rozměr představuje vzdálenost přímky od počátku souřadnic (ρ) a druhý úhel natočení (θ). Za využití faktu, že přímky, které vznikly jako odezva na společnou hranu, leží s určitou tolerancí v jedné oblasti houghova prostoru (který lze nazvat clusterem) můžeme na tento prostor aplikovat vhodný algoritmus pro detekci zmíněných shluků a každý shluk přímek nahradit jedinou – tvořenou průměrem všech členů shluku.

Princip

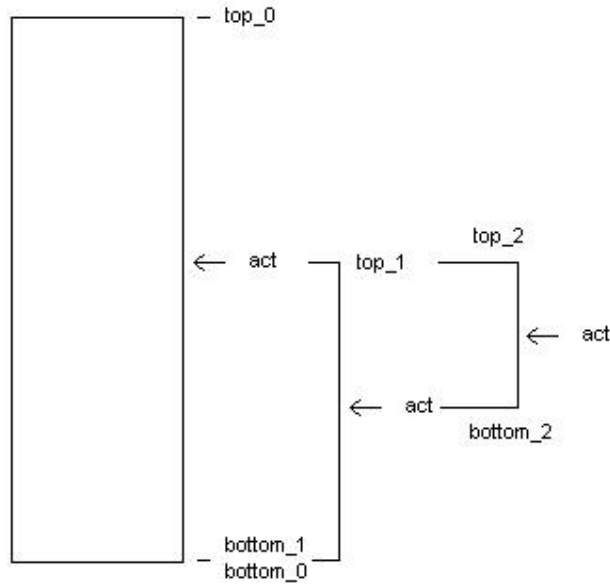
Při clusterování samotném narazíme ovšem na problém. Neznáme předem počet clusterů, které chceme vytvořit. Řešením je provést proces clusterizace vždy několikrát po sobě a měřit chybu, kterou představuje průměrná vzdálenost všech přímek clusteru od středu. Tento přístup s sebou ovšem nese několik nevýhod:

- Opakovaně prováděný proces clusterizace, který je výpočetně náročný.
- Je velice obtížné zvolit hodnotu chyby, při které už se nebude opakovat další iterace, tak, aby byla univerzální a vhodná pro obecnou aplikaci.

Hledání vhodného počtu clusterů

Při hledání počtu clusterů vycházíme z požadované chyby. Algoritmus iteruje tak dlouho, dokud se výsledná chyba nebude maximálně blížit požadované hodnotě. Pro dosažení tohoto cíle lze využít nejjednoduššího principu, který se nabízí: postupně navyšovat počet clusterů, provádět clusterizaci a kontrolovat chybu. Tento proces je nutné opakovat tak dlouho, dokud nebude hodnota chyby splňovat požadované kritérium.

Pakliže očekáváme, že počet vyhovujících clusterů se bude blížit horní nebo spodní hranici celkového počtu úseček, můžeme ji nastavit jako výchozí a poté postupně snižovat nebo zvyšovat. I přes tuto částečnou optimalizaci je však hledání optimálního počtu clusterů příliš výpočetně a časově náročné. Abychom celý proces co nejvíce urychlili, je vhodné využít následující postup znázorněný na obr. 8.



Obr. 8: Hledání optimálního počtu clusterů

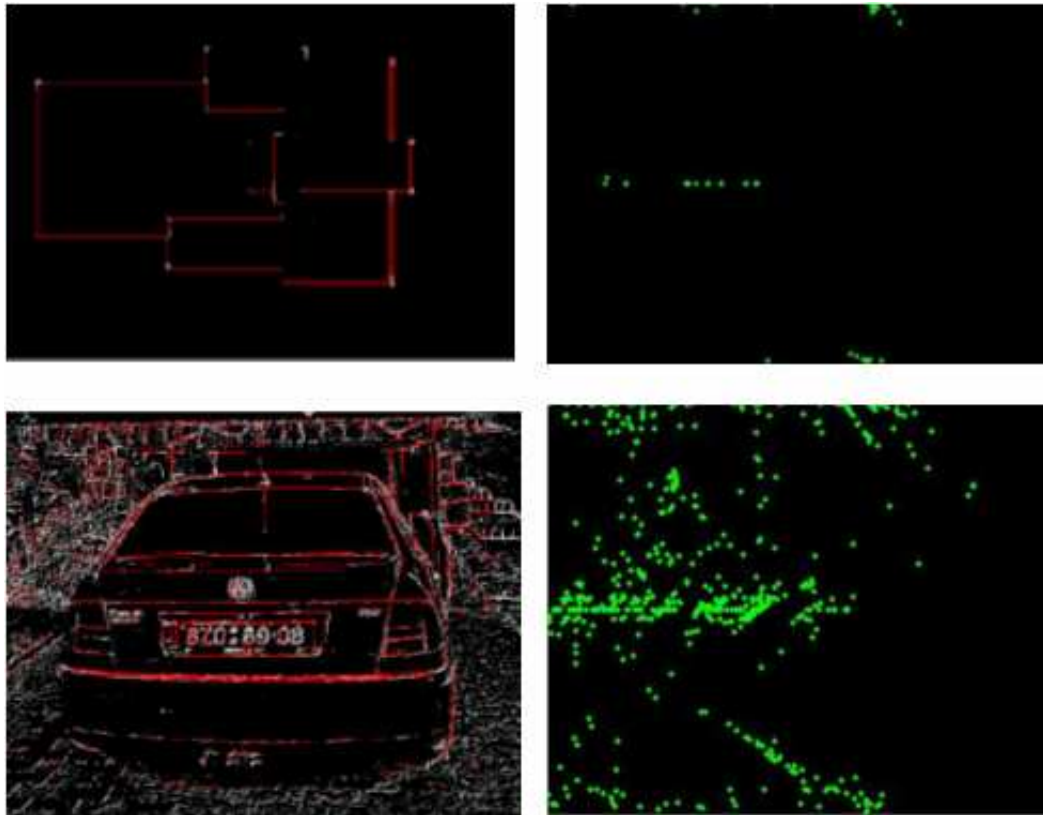
Výše nastíněný postup by se dal popsat v několika následujících krocích:

- 1) Spodní a horní hranici počtu úseček označme na počátku $bottom_0$ a top_0 .
- 2) Provedeme první clusterování s počtem clusterů nastaveným doprostřed intervalu $\langle bottom, top \rangle$.
- 3) Upravíme vyhledávací interval a to následujícím způsobem. Pakliže je výsledná chyba menší než požadovaná (viz. obr.), můžeme vyhledávání omezit pouze na část intervalu mezi aktuálním počtem clusterů, který označíme jako top_1 , a spodní hranicí tohoto intervalu, kterou zachováme. V opačném případě bychom zachovali horní hranici a $bottom_?$ Nastavili na aktuální hodnotu počtu clusterů.
- 4) Opět nastavíme hodnotu aktuálně zkoumaného počtu clusterů do prostřed intervalu $\langle bottom, top \rangle$ a provedeme clusterování.
- 5) Celý postup opakujeme tak dlouho, dokud se chyba clusterování maximálně neblíží požadované hodnotě.

Ačkoli se první z problémů – rychlost algoritmu podařilo z části odstranit použitím vhodné metody upravování počtu clusterů, druhá komplikace stále přetrvává.

Základním předpokladem použití výše zmíněného přístupu je clusterizovatelnost houghova prostoru. Tzn., že generované přímky skutečně shluky vytváří. Čím rovnoměrněji jsou přímky v

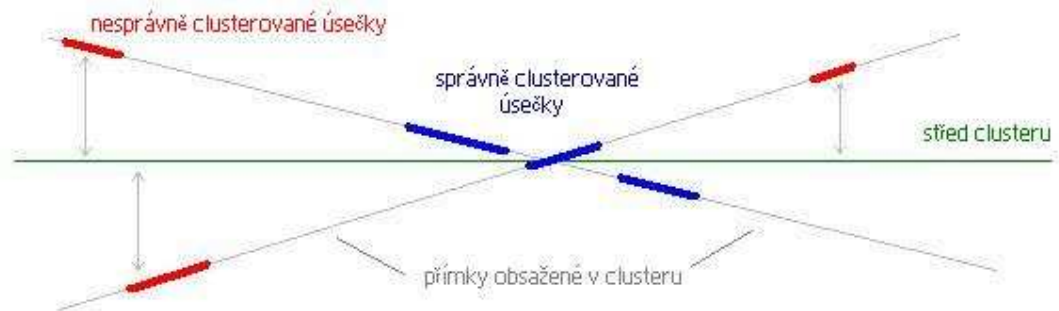
houghově prostoru rozmístěny, tím méně je potom tato úloha řešitelná za pomoci principu clusterování. Následující obrázky ukazují různé clusterizovatelné prostory.



Obr. 9: Ukázka obrazů a příslušných houghových prostorů

Výše jsou demonstrovány dva různě clusterizovatelné prostory. Obr. 9 se skládá celkem ze 4 podobrazů, přičemž první je složen z několika obdélníků, proto jeho houghův prostor vytváří viditelně oddělené shluky bodů, které lze bez problémů a s vysokou přesností clusterizovat. Oproti tomu obrázek automobilu je mnohem složitější a komplexnější. Jsou v něm obsaženy úsečky různého natočení a houghův prostor je obtížněji clusterizovatelný.

Abychom byli schopni aproximovat každou hranu minimálním počtem úseček (v optimálním případě jedinou), pouze informace obsažená v houghově prostoru nestačí, protože se zde jedná o vyjádření přímky, ale ve skutečnosti potřebujeme shlukovat úsečky. Následující obrázek demonstruje problém, při kterém při použití clusterizace na základě houghova prostoru může docházet.



Obr. 10: Problém obecné clusterizace

Hodnocení

Na obr. 10 je zobrazeno 6 úseček, které ovšem leží pouze na dvou odlišných přímkách. Při clusterizaci potom může dojít k vyznačenému problému: do clusteru budou sice zahrnuty všechny úsečky (nelze je od sebe nikterak odlišit, protože jsou reprezentovány pouze úhlem natočení a posuvem), ale pouze některé do tohoto clusteru skutečně patří. Přímky zvýrazněné červenou barvou jsou v clusteru obsaženy chybně – jejich vzdálenost od středu clusteru je příliš velká.

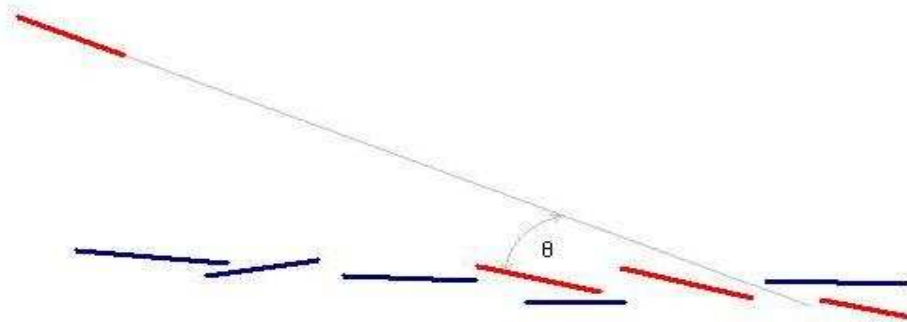
4.2. Metoda porovnávání přímek

Princip činnosti

První z této skupiny metod se snaží na základě porovnání dvou přímek rozhodnout, zda patří do stejné skupiny. Každé dvě přímky jsou porovnány pomocí porovnávací funkce, kde je analyzován pouze rozdíl jejich úhlů a posunutí vzhledem k počátku souřadného systému. Algoritmus pracuje tak, že načte jednu přímku a porovná ji se všemi ostatními. Pakliže se dané přímky se zadanou tolerancí shodují, jsou zahrnuty do stejné skupiny. Po analýze všech přímek získáme konečnou množinu tímto způsobem se sebou souvisejících entit a nabízejí se dvě možnosti pokračování:

- Ukončit analýzu a prohlásit první přímku za střed clusteru.
- Přepočítat střed clusteru jako průměr všech členských přímek.

Při využití způsobu a) může nastat situace, kdy přímka, která byla zvolena jako střed clusteru leží mimo některé důležité hrany, ale vlivem porovnávací funkce, která vždy musí mít určitou toleranci, se k ní mohou přiřadit přímky korespondující s jinou výraznou hranou (viz. obr. 11).



Obr. 11: Příklad nesprávného určení středu clusteru

Pakliže se přikloníme ke druhému ze způsobů, musíme se také vypořádat se skutečností, že vlivem změny pozice středu clusteru nemusí již všechny do této doby členské přímky po zpětné analýze patřit do tohoto clusteru a naopak jiné přímky do něj nově patřit mohou. Což vede k nutnosti znovu přepočítat polohu clusteru, dalšímu posunutí a opakování výše popsání postupu. Abychom optimalizovali rychlost algoritmu, je vhodné zavést maximální počet iterací nebo hodnotu posunutí clusteru, při níž proces přepočítávání polohy clusteru ukončíme.

Ačkoliv zde zcela jistě dochází k chybě, lze ji akceptovat, poněvadž algoritmus s každou další iterací aproximuje svoji konečnou polohu a s každým dalším krokem je chyba nižší. Výhodou je, že se výsledná poloha clusteru více blíží významným hranám.

Hodnocení

Tato metoda je základním přístupem pro shlukování přímek a pro svoji jednoduchost je také snadno implementovatelná. Bohužel stále pracuje pouze s informací houghova prostoru a jak již bylo řečeno, při práci s obecným typem obrazu se na jeho clusterizovatelnost nelze spoléhat. Proto se tento přístup ukázal pro další rozvoj jako ne příliš vhodný. Lze jej efektivně uplatnit pouze u obrazů, které mají tu vlastnost, že houghův prostor přímek detekovaných houghovou transformací je clusterizovatelný.

Na základě informace získané z houghova prostoru nelze v obecných případech detekci shluků přímek realizovat. Následující metody se pokouší využít houghův prostor v kombinaci s dalším typem informace, jako je např. poloha přímky v obraze.

4.3. Shlukování přímek s využitím histogramu sousednosti

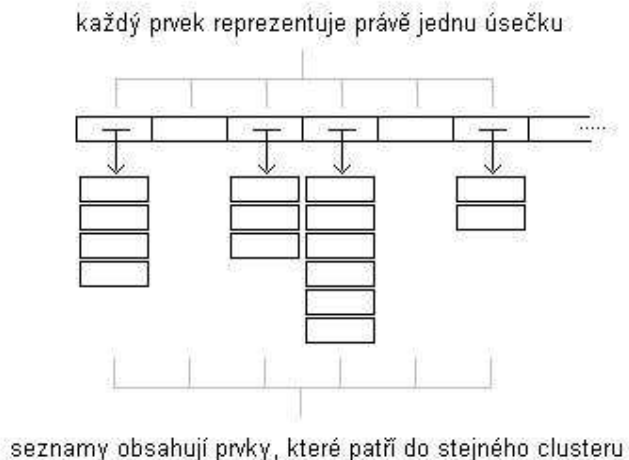
Jedná se o rozšíření a upravení výše diskutované metody. Přepočet hodnoty clusteru vede k tzv. „cestování“ clusteru houghovým prostorem a zvolení náhodné přímky jako střed clusteru a

následnou analýzou zjišťovat, které další přímký mohou do stejné skupiny patřit s sebou nesou určité komplikace. Tato metoda se pokouší odstranit podobné nevýhody hledáním těch nejvýznamnějších clusterů. Za tímto účelem si vytváří určitý typ histogramu – nazvěme jej histogram sousednosti, na základě kterého pro každou přímku zjistí, s kolika dalšími je ochotná cluster vytvořit. Tato přímka tedy zakládá skupinu sousedících přímek. Následně může být analyzována každá dílčí množina přímek a pro každý prvek jsou do této množiny přidány všechny jeho sousedící přímký.

Histogram sousednosti

V zásadě existují dva hlavní způsoby, kterých tato metoda využívá, jak histogram vytvořit.

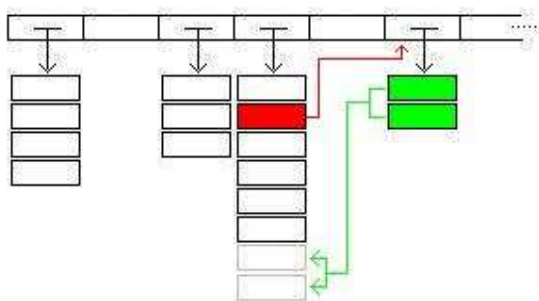
- **Četnostní histogram:** Má tolik polí, kolik je úseček – korespondují spolu svými indexy. Histogram obsahuje informaci a počtu sousedících úseček, ale neobsahuje už informaci o tom, které to jsou.
- **Seznamový histogram:** Každý prvek takového histogramu je potenciální seznam úseček. Pro každou nově analyzovanou úsečku je nejprve ověřeno, zda-li nepatří do některého z již vytvořených clusterů. Pakliže nikoliv, sama se stává prvním prvkem nového seznamu.



Obr. 12: Histogram sousednosti

Tento histogram lze také vytvořit více způsoby. Lze se držet základního výše popsaného principu, nebo pro každou přímku (podobně jako u četnostního histogramu) vytvářet seznam jejích sousedů. Po vytvoření takového histogramu je ovšem po zpracování určitého clusteru nutno vhodnou kontrolou zajistit, aby tímto clusterem obsažené úsečky nebyly podruhé zpracovány jako členové jiného clusteru. Tento problém lze řešit například metodou „kopírování sousedů“, kdy se vytvořený

histogram prochází a do každého clusteru jsou přidáni také sousedé jeho členů. Clustery, které tyto členové založili jsou smazány.



Obr. 13: Kopírování sousedů

Výsledné zbylé skupiny přímek umožní vypočítat střed clusteru. Ten však musí být dále analyzován a upraven dvěma kroky:

- Vyjmutí z clusteru příliš vzdálených úseček (přímka, na které tato úsečka leží sice může na základě úhlu a posunutí zdánlivě patřit do určitého clusteru, ale samotná pozice úsečky rozhodne, zda ji lze do takového clusteru zařadit.
- Přidání úseček, které na základě ρ a θ sice nevyhovují, ale jsou blízko celému clusteru. Aby se zamezilo akceptování úseček s příliš rozdílným úhlem, lze vhodným způsobem kontrolovat pouze směr úsečky bez důrazu na posunutí.

Hodnocení

Metoda založená na histogramu sousednosti dosahuje mnohem lepších a přesnějších výsledků než metody pracující bez histogramu. Hlavní výhodou spočívá v přesnosti detekce výsledných čar. Pro další případnou analýzu je příjemná skutečnost, že na základě využití histogramu můžeme úsečky procházet od těch s nejvyšší odezvou až k úsečkám s odezvou nižší.

Výše popsané metody umožňují detekovat shluky úseček, které korespondují s jedinou hranou. Abychom mohli skutečně nahradit více úseček jejich minimálním počtem, je nutné implementovat metody, které umožní zjistit míru pokrytí hlavní přímky (kterou tvoří střed celého clusteru) s množinou korespondujících úseček a následně vhodným postupem eliminovat nespojitosti. Touto problematikou se zabývá následující kapitola.

4.4. Nahrazení clusteru minimálním počtem úseček

Převod obecného clusteru – nebo-li shluku – na jeho reprezentaci v podobě minimalního počtu úseček se skládá ze čtyř základních kroků:

- Projekce úseček na přímku reprezentující střed clusteru
- Analýza těchto projektovaných přímek a odstranění některých nespojitostí
- Nalezení úseček modifikovaných odstraněním nespojitostí
- Zpětná projekce úseček

V následujícím textu budou podrobně popsány všechny dílčí kroky.

Projekce úseček

Mějme obecnou úsečku ležící mimo určitou přímku (viz. obr. 14). Můžeme aplikovat kolmou projekci koncových bodů úsečky na tuto přímku, abychom získali míru překrytí těchto dvou geometrických entit.

Jeden ze způsobů, jakým realizovat kolmou projekci, je analytické řešení celé situace. Pouze připomeňme, že přímku je v normálovém tvaru a dále provádím projekci pouze koncových bodů úsečky – nikoliv úsečky jako celku. Tedy jedná se v podstatě o řešení projekce bodu na přímku. Postup je následující:

- Převědeme normálový tvar přímky na obecnou rovnici.
- Bodem proložíme kolmici k přímce.
- Vypočítáme průsečík obou kolmých přímek.

Po odvození lze celý postup shrnout následujícími vzorci:

$$x = -\frac{-\cos(\theta) \cdot \rho + \sin(\theta) \cdot (y_0 \cdot \cos(\theta) - x_0 \cdot \sin(\theta))}{\cos^2(\theta) + \sin^2(\theta)} \quad \text{pro } \theta \neq 0 \quad (9)$$

$$y = -\frac{\cos(\theta)}{\sin(\theta)} \cdot x + \frac{\rho}{\sin(\theta)} \quad \text{pro } \theta \neq 0 \quad (10)$$

nebo

$$x = -\frac{\sin(\theta)}{\cos(\theta)} \cdot y + \frac{\rho}{\cos(\theta)} \quad \text{pro } \theta = 0 \quad (11)$$

$$y = y_0 \quad \text{pro } \theta = 0 \quad (12)$$

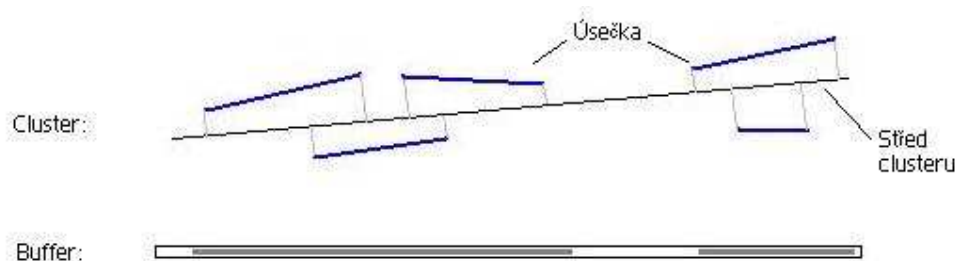
Kde $[x, y]$ jsou souřadnice projektovaného bodu, $[x_0, y_0]$ jsou souřadnice původního bodu, θ a ρ jsou parametry normálově vyjádřené přímky.

Analýza přímek, odstranění nespojitostí

Pro analýzu projektovaných přímek se využívá **buffer** v podobě celočíselného pole, které má délku rovnu vzdálenosti bodů, ve kterých úsečka reprezentující střed clusteru protíná okraje obrazu. Celé pole je zpočátku vynulováno. Do tohoto bufferu se každý bod ležící na této přímce zobrazí pomocí své vzdálenosti od levého průsečíku (index tohoto průsečíku by byl roven 0). Nejprve jsou získány pro každou přímku indexy koncových bodů a interval mezi nimi je vyplněn hodnotami 1. Tento postup se aplikuje pro všechny úsečky náležící k danému clusteru.

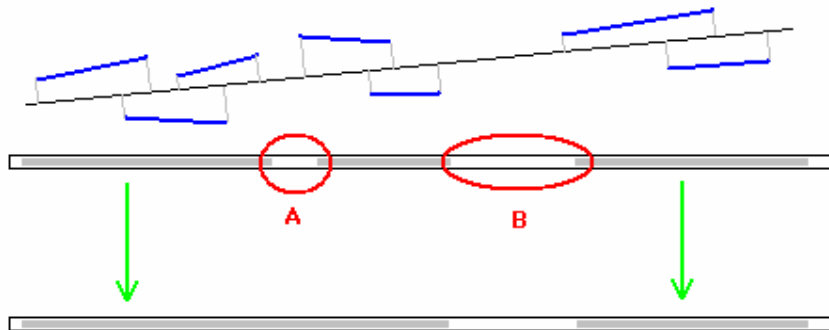
Odstranění nespojitostí

Výsledkem celého procesu je skutečnost, že překrývající se úsečky automaticky splynou v jednu (obr. 14). Celý buffer je následně analyzován, přičemž jsou hledány nespojitosti mezi jednotlivými úseky.



Obr. 14: Ukázka množiny úseček, středu clusteru a bufferu

Algoritmus pro odstranění nespojitostí v první fázi hledá mezeru mezi dvěma spojitými bloky, následně na základě délky těchto bloků určí maximální délku mezery, při které se mohou tyto dva bloky spojit. Velikost této mezery je dána relativně v procentech vzhledem k součtu délek sousedících bloků a je jedním z argumentů spojovací funkce. Pakliže velikost mezery nepřesahuje tuto hranici, dojde ke spojení sousedících bloků v jeden celek (viz. obr. 15).



Obr. 15: Odstranění nespojitosti

Na výše uvedeném příkladu lze vidět dvě různé mezery a dva různé typy chování spojovací funkce. V případě A dojde ke spojení fragmentů, protože velikost mezery není příliš velká. Oproti tomu v případě B je nespojitost zachována, jelikož je poměr mezery a délky sousedících fragmentů velká.

Jakmile jsou nespojitosti v bufferu odstraněny, je třeba je zpětně analyzovat, provést hledání úseček – a to modifikovaných či nemodifikovaných – a tyto úsečky opět promítnout zpět do 2D prostoru.

Hledání úseček v bufferu

Algoritmus pracuje velice jednoduše – prochází vstupní buffer prvek po prvku (připomeňme pouze, že buffer je binární – skládá se pouze z hodnot 0 a 1) a jakmile dojde ke změně hodnoty, určí se podle jejího charakteru, zda se jedná o počáteční bod úsečky, nebo byl nalezen koncový bod. Ve druhém případě je provedena zpětná projekce bodu na přímkou. Tento algoritmus je vykonáván, dokud nejsou zpracovány všechny body bufferu.

Zpětná projekce

Nyní zbývá už pouze vrátit body zpět na úsečku do 2D prostoru. Informace, které pro tento účel máme k dispozici jsou:

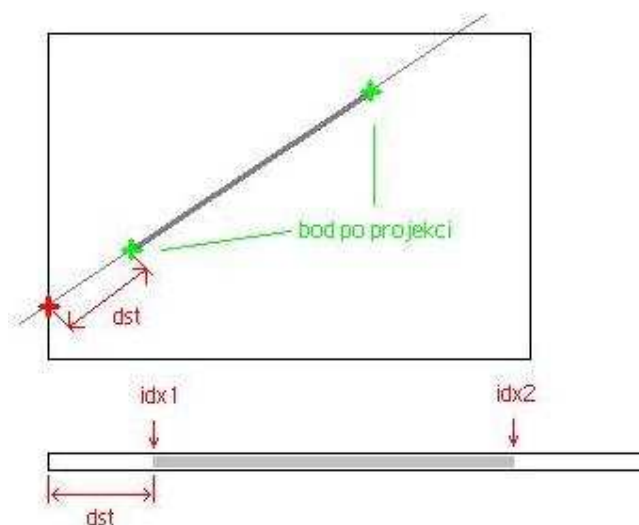
- Normálové vyjádření přímky, na kterou chceme projekci provést.
- Souřadnice průsečíků této přímky s okrajem obrazu.
- Vzdálenost hraničních bodů úsečky od levého průsečíku přímky s hranicí obrazu.

Zpětnou projekci realizujeme opět pro každý bod izolovaně. Algoritmus je následující:

- Postupně volíme vhodné hodnoty x -osy a následně vypočítáme odpovídající souřadnici y bodu ležícího na přímce.
- Ověříme vzdálenost získaného bodu od levého průsečíku.
- Pakliže je tato vzdálenost větší, než požadovaná (index daného bodu v bufferu), snížíme souřadnici x a postup opakujeme.

Na počátku výpočtu je třeba nastavit proměnnou x na hodnotu maximální vzdálenosti od levého průsečíku přímky s okrajem obrazu, které by projektovaný bod dosáhl v případě rovnoběžnosti této přímky s osou x .

Výše uvedený postup samozřejmě pracuje správně pouze pro přímky, které nejsou rovnoběžné s osou y . Pakliže je tedy $\theta = 0$, není pro projekci bodu použit zmíněný algoritmus, ale pouze je k y souřadnici spodního průsečíku přímky s okrajem obrazu přičtena hodnota indexu daného bodu v bufferu. Celý postup dokresluje obrázek 23.



Obr. 16: Princip zpětné projekce

5. Experimenty

V této kapitole uvedu několik experimentů, ve kterých ověřím odolnost celého procesu, který jsem ve své práci popisoval, proti pootočení obrazu. Experimenty byly aplikovány na tentýž obraz s různým natočením, takže můžeme zkoumat odolnost této metody proti úhlu natočení bez vlivů kvality obrazu na detekci čar.

Experiment č. 1 – Nenatočený obraz

Zde můžeme vidět obraz, který není natočen oproti osám obrazu. Nyní pojďme porovnat výsledky, kterých je dosahováno u obrazů s určitým natočením.



Obr. 17: Nenatočený obraz

Experiment č. 2 – Natočení o 10°

Jak můžeme vidět na obr. 18, natočení o úhel 10° působí určité komplikace, oblast značky je v podstatě nedetekovaná.



Obr. 18: Natočení o 10°

Na druhé straně stále zůstává zachován jeden důležitý rys – jsou poměrně přesně detekovány nejdůležitější a nejvýraznější hrany obrazu.

Experiment č. 3 – Natočení o 20°

Následně si můžete všimnout, že u jiného natočení je oblast značky detekována, hrany jsou indikovány s poměrně vysokým pokrytím.



Obr. 19: Natočení o 20°

Závěr

Z výše uvedených experimentů lze vypožorovat, že metoda je náchylná k natočení obrazu a nejsou detekovány některé výrazné hrany. Jako pozitivní vlastnost této metody hodnotím skutečnost, že řada hran je správně a přesně pokryta a úsečky, které tyto hrany pokrývají jsou konzistentní a zpravidla mají délku hrany.

6. Závěr

Cílem této práce bylo poskytnout kvalitní základ pro robustní a univerzální metodu detekce značky vozidla na základě detekce přímků pomocí Houghovy transformace. Celá práce se zabývala analýzou a řešením řady problémů a omezení, které s sebou nese tento způsob zpracování obrazu. Přínosem této práce je vyzkoušení nového – zatím neznámého – způsobu detekce značky, jeho charakteristika a doporučení (které uvádím v závěru této kapitoly) o použitelnosti vyzkoušené metody.

Ve své práci jsem popsal princip několika metod, které značku detekují a nastínil princip detekce značky na základě hledání úseček. Tato metoda s sebou ovšem přinesla řadu komplikací, které ve své práci v příslušných kapitolách popisuji. Vzniklé problémy jsem se pokusil řešit vždy několika způsoby, z nichž jsem si pro další pokračování vybral ten nejlepší.

Výsledky, které můžeme vidět na základě provedených experimentů sice v některých případech splňují cíl, který jsem si na začátku této práce stanovil, nicméně spolehlivost metody je diskutabilní a s různým natočením obrazu se mění, přičemž nezávisí přímo na úhlu natočení.

Vzhledem k výše uvedenému jsem dospěl k závěru, že metoda detekce značky založená na hledání přímků či úseček je pro toto použití nevhodná nebo přinejmenším velice komplikovaná a proto ji nelze doporučit pro další vývoj.

7. Seznam literatury

- [1] Hlaváč, V.: Zpracování signálů a obrazů. Praha, ČVUT, 2005
- [2] Rafael C. Gonzalez, Richard E. Woods: Digital Image Processing. Prentice-Hall, 2002
- [3] Galbiati, L. J.: Machine Vision and Digital Image Processing Fundamentals. Prentice-Hall, 1990
- [4] Sunghoon Kim, Daechul Kim, Younbok Ryu and Gyeonghwan Kim: A Robust Licence-Plate Extraction Method under Complex Image Conditions. Sogang University
- [5] Varsha Kamat and Subramaniam Ganesa: An Efficient Implementation of the Hough Transform for Detecting Vehicle Licence Plates Using DSP'S. Oakland University, Rochester
- [6] Feng Yang, Zheng Ma: Vehicle Licence Plate location Based on Histogramming and Mathematical Morphology. University of Electronic Science and Technology of China (UESTC)
- [7] Setchell J. C.: Applications of computer vision to road-traffic monitoring. Faculty of Engeneering, Bristol, 1997.
- [8] Draghici S.: A Neural Network Based Artificial Vision System for Licence Plate Recognition. Department of Computer Science, Wayne State University.
- [9] Barroso J., Rafael A., Dagle E. L., Bulas-Cruz J.: Number Plate Reading Using Computer Vision. Department of Electrical and Electronic Engineering, University Walk, UK.
- [10] Belza L.: Metody detekce a reprezentace hran v obraze [diplomová práce]. Fakulta informačních technologií, Brno, 2006
- [11] Blaha P.: Detekce významných bodů v obraze [diplomová práce]. Fakulta informačních technologií, Brno, 2006

8. Seznam příloh

Příloha 1: CD s aplikací demonstrující výsledky práce, manuálem aplikace, anotovanými testovacími daty a elektronickou formou této práce

9. Seznam obrázků

Obr. 1: Vertikální gradient obrazu a horizontální projekce gradientu.....	3
Obr. 2: Kandidátní oblast vyňatá z originálního obrazu	3
Obr. 3: Vertikální projekce gradientu před a po aplikaci morfologických operátorů.....	4
Obr. 4: Binární obraz.....	4
Obr. 5: Originální a prahovaný obraz	12
Obr. 6: Nedetekované úsečky.....	12
Obr. 7: Aplikace okna zájmu	13
Obr. 8: Hledání optimálního počtu clusterů	16
Obr. 9: Ukázka obrazu a příslušného houghova prostoru	17
Obr. 10: Problém obecné clusterizace	18
Obr. 11: Příklad nesprávného určení středu clusteru.....	19
Obr. 12: Histogram sousednosti.....	20
Obr. 13: Kopírování sousedů	21
Obr. 14: Ukázka množiny úseček, středu clusteru a bufferu.....	23
Obr. 15: Odstranění nespojitosti	24
Obr. 16: Princip zpětné projekce.....	25
Obr. 17: Nenatočený obraz.....	26
Obr. 18: Natočení o 10°.....	26
Obr. 19: Natočení o 20°.....	27