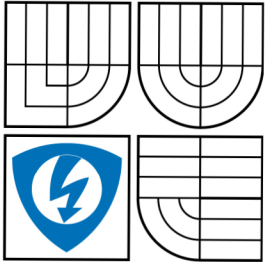


VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ  
BRNO UNIVERSITY OF TECHNOLOGY



FAKULTA ELEKTROTECHNIKY A  
KOMUNIKAČNÍCH TECHNOLOGIÍ  
ÚSTAV AUTOMATIZACE A MĚŘÍCÍ TECHNIKY

FACULTY OF ELECTRICAL ENGINEERING AND COMMUNICATION  
DEPARTMENT OF CONTROL AND INSTRUMENTATION

## VYHLEDÁVÁNÍ TEXTU

TEXT SEPARATION

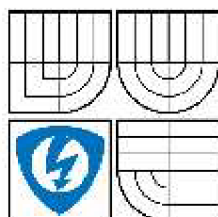
BAKALÁŘSKÁ PRÁCE  
BACHELOR'S THESIS

AUTOR PRÁCE  
AUTHOR

VLADIMÍR BURLAK

VEDOUCÍ PRÁCE  
SUPERVISOR

Ing. MILOSLAV RICHTER, Ph.D.



VYSOKÉ UČENÍ  
TECHNICKÉ V BRNĚ

Fakulta elektrotechniky  
a komunikačních technologií

Ústav automatizace a měřicí  
techniky

## Bakalářská práce

bakalářský studijní obor  
Automatizační a měřicí technika

**Student:** Burlak Vladimír  
**Ročník:** 3

**ID:** 78579  
**Akademický rok:** 2007/2008

**NÁZEV TEMATU:**

Vyhledávání textu

**POKYNY PRO VYPRACOVÁNÍ:**

Navrhněte metody pro vyhledávání textu ve snímcích.

Nastudujte problematiku zpracování obrazu v oblasti zpracování textů. Navrhněte a realizujte testovací databázi pro různé kategorie textu a obrazu, a jejich kombinace. Navrhněte program pro vyhledání a separaci oblastí podle kategorií předloh.

**DOPORUČENÁ LITERATURA:**

Žára J., Benaš B., Sochor J., Felkel P.: Moderní počítačová grafika, Computer Press, 1998, ISBN 80-251-0454-0

Hlaváč V., Šonka M.: Počítačové vidění, Grada, Praha 1992, ISBN 80-85424-67-3

Faugeras O.: Three-Dimensional Computer Vision, The MIT Press 1993

Kraus K.: Photogrammetrie 1 und 2, Ummeler / Bonn, 1996

**Termín zadání:** 1.2.2008

**Termín odevzdání:** 2.6.2008

**Vedoucí práce:** Ing. Milošlav Richter, Ph.D.

prof. Ing. Pavel Jura, CSc.  
*předseda oborové rady*

**UPOZORNĚNÍ:**

Autor bakalářské práce nesmí při vytváření bakalářské práce porušit autorská práva třetích osob, zejména nesmí zasahovat nedovoleným způsobem do dílech 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 § 152 trestního zákona č. 140/1961 Sb.

## **Anotace**

Tato bakalářská práce popisuje základní způsoby segmentace obrazu za účelem oddělení textu od pozadí. V první části jsou popsány obecné metody, určené pro zpracování obrazu. Je zde popsáno prahování, detekce hran a redukce šumu. V druhé části jsou definovány kategorie textu v obraze. Třetí část je zaměřena na program, pomocí kterého bude obraz segmentován.

## **Klíčová slova**

Prahování, detekce hran, redukce šumu, histogram

## **Annotation**

This bachelor's thesis describes basic methods of picture segmentation for the purpose of separation of text from a background. In first part are described general methods meant for picture processing. There is described verging, edge detection and interference reduction. In second part are defined categories of text in picture. Third part is focused on program which will help with picture segmentation.

## **Keywords**

Thresholding, edge detection, interference reduction, histogram

## **Bibliografická citace**

BURLAK, V. *Vyhledávání textu*. Brno: Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií, 2008. 50 s. Vedoucí bakalářské práce Ing. Miloslav Richter, Ph.D.

## **P r o h l á š e n í**

„Prohlašuji, že svou bakalářskou práci na téma „Vyhledávání textu“ 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:

## **P o d ě k o v á n í**

Děkuji vedoucímu bakalářské práce Ing. Miloslavu Richterovi, Ph.D. za účinnou metodickou, pedagogickou a odbornou pomoc a další cenné rady při zpracování mé bakalářské práce.

V Brně dne :

Podpis:

## OBSAH

|                                                            |           |
|------------------------------------------------------------|-----------|
| <b>1. ÚVOD .....</b>                                       | <b>12</b> |
| <b>2. REPREZENTACE OBRAZOVÝCH DAT.....</b>                 | <b>13</b> |
| 2.1 Obrazová funkce .....                                  | 13        |
| 2.2 řetězcové kódy .....                                   | 14        |
| <b>3. POJMY A METODY SPOJENÉ S DISKRÉTNÍM OBRAZEM.....</b> | <b>15</b> |
| 3.1 Histogram.....                                         | 15        |
| 3.2 Konvoluce nad diskretním obrazem .....                 | 15        |
| <b>4. OPERACE NAD DISKRÉTNÍM OBRAZEM.....</b>              | <b>17</b> |
| 4.1 Prahování .....                                        | 17        |
| 4.1.1 Globální prahování .....                             | 18        |
| 4.1.2 Globální prahování s hysterezí .....                 | 21        |
| 4.1.3 Adaptivní prahování .....                            | 22        |
| 4.2 reDukce Šumu.....                                      | 22        |
| 4.2.1 Gaussův filtr.....                                   | 22        |
| 4.2.2 Mediánový filtr .....                                | 23        |
| 4.3 detekce hran .....                                     | 23        |
| 4.3.1 Robertsův operátor.....                              | 24        |
| 4.3.2 Sobelův operátor .....                               | 25        |
| 4.3.3 Detekce hran pomocí Laplaceova operátoru .....       | 26        |
| 4.3.4 Cannyho hranový detektor.....                        | 27        |
| <b>5. TEXT V OBRAZE OBECNĚ .....</b>                       | <b>29</b> |
| 5.1 Pojmy text a znak .....                                | 29        |
| 5.2 Vlastnosti textu .....                                 | 30        |
| 5.3 charakteristický vektor řádku.....                     | 30        |
| <b>6. KATEGORIE TEXTU.....</b>                             | <b>32</b> |
| 6.1 Prostý text .....                                      | 32        |
| 6.2 Předlohy obsahující „nepopsané“ obrázky .....          | 32        |
| 6.3 Tabulky .....                                          | 33        |
| 6.4 Rozmanité obrazy obsahující text.....                  | 34        |
| 6.5 Hodnoty indikované přístroji .....                     | 34        |

|                                                           |           |
|-----------------------------------------------------------|-----------|
| 6.6 Text obsahující vzorce .....                          | 35        |
| 6.7 Některé zvláštní případy .....                        | 36        |
| <b>7. POSTUP VYHLEDÁVÁNÍ TEXTU .....</b>                  | <b>37</b> |
| 7.1 Načtení a přeDzpracování obrazu .....                 | 37        |
| 7.2 segmentace obrazu .....                               | 39        |
| 7.2.1 Obalové čáry .....                                  | 39        |
| 7.2.2 Segmentace uzavřených oblastí .....                 | 43        |
| 7.3 Prvotní fitrace .....                                 | 44        |
| 7.4 Hledání vztahů mezi oblastmi a jejich slučování ..... | 44        |
| 7.5 Druhotná filtrace .....                               | 46        |
| 7.6 Reprezentace výsledku .....                           | 46        |
| <b>8. APLIKACE VT .....</b>                               | <b>47</b> |
| <b>9. ZÁVĚR .....</b>                                     | <b>49</b> |
| <b>10. LITERATURA .....</b>                               | <b>50</b> |



## SEZNAM OBRÁZKŮ

|                                                                                                             |    |
|-------------------------------------------------------------------------------------------------------------|----|
| Obrázek 2.1 Popis směrů Freemanova kódu .....                                                               | 14 |
| Obrázek 4.1 Převodní charakteristika prahování pro pozadí tmavší nežli text<br>(hodnota prahu 0,5) .....    | 17 |
| Obrázek 4.2 Převodní charakteristika prahování pro pozadí světlejší nežli text<br>(hodnota prahu 0,5) ..... | 17 |
| Obrázek 4.3 Prahovaná předloha .....                                                                        | 19 |
| Obrázek 4.4 Histogram obrázku 4.3 .....                                                                     | 19 |
| Obrázek 4.5 Invertovaný výsledek prahování obrázku 4.3.....                                                 | 19 |
| Obrázek 4.6 Prahovaná předloha .....                                                                        | 20 |
| Obrázek 4.7 Invertovaný výsledek.....                                                                       | 20 |
| Obrázek 4.8 Histogram předlohy 4.6 .....                                                                    | 21 |
| Obrázek 5.1 Příklady textů a jejich char. vektoru psaných českou abecedou.....                              | 31 |
| Obrázek 5.2 Příklady textů a jejich char. vektorů psaných anglickou abecedou.....                           | 31 |
| Obrázek 6.1 Příklad tabulky a jejího řádkového a sloupcového vektoru.....                                   | 33 |
| Obrázek 7.1 Nevhodné použití adaptivního prahování.....                                                     | 38 |
| Obrázek 7.2 Vhodné použití adaptivního prahování.....                                                       | 38 |
| Obrázek 7.3 Ukázka vyhodnocení testovaného okolí .....                                                      | 39 |
| Obrázek 7.4 Modifikovaný způsob vyhodnocení testovaného okolí (pro pohyb<br>doprava a dolů).....            | 41 |
| Obrázek 7.5 Modifikovaný způsob vyhodnocení testovaného okolí (pro pohyb<br>doleva a nahoru).....           | 42 |
| Obrázek 7.6 Ukázka výsledku segmentace obalovými čarami .....                                               | 43 |
| Obrázek 7.7 Ukázka výsledku segmentace uzavřenými oblastmi .....                                            | 43 |
| Obrázek 8.1 Hlavní okno aplikace .....                                                                      | 47 |
| Obrázek 8.2 Formulář pro nastavení aplikace.....                                                            | 48 |

## SEZNAM TABULEK

|                                      |    |
|--------------------------------------|----|
| Tabulka 7-1 Změna směrů posunu ..... | 40 |
| Tabulka 7-2 Změna směrů posunu ..... | 41 |

## SEZNAM ZKRATEK

| Zkratka/Symbol | Popis                                            |
|----------------|--------------------------------------------------|
| $f(x,y)$       | Obrazová funkce                                  |
| $b(x,y)$       | Šedotónová obrazová funkce                       |
| $d(x,y)$       | Černobílá obrazová funkce                        |
| $t$            | Prahová hodnota                                  |
| $H(i)$         | Četnost výskytu $i$ -té jasové hodnoty           |
| $h(i,j)$       | Konvoluční jádro                                 |
| $\theta$       | Směr gradientu                                   |
| $L_4$          | Konvoluční jádro pro Laplaceův filtr pro 4-okolí |
| $L_8$          | Konvoluční jádro pro Laplaceův filtr pro 8-okolí |
| $v(i)$         | Četnost výskytu jasové hodnoty v $i$ -tém řádku  |
| $w(i,j)$       | Funkce, popisující tvar řádku                    |

## 1. ÚVOD

Optické rozpoznávání znaků je metoda, která pomocí kamery, fotoaparátu, skeneru, apod. umožňuje digitalizaci textů v obrazu, s nimiž lze pak pracovat jako s běžným počítačovým textem. Převedený text je vždy v závislosti na kvalitě předlohy třeba vhodně korigovat, aby obsah mohl být správně interpretován. Je to ovšem technicky náročný proces, který vyžaduje speciální SW podporu řešení.

V současné době existuje celá řada knihoven, které jsou určeny pro rozpoznávání textu. Jako velký problém se však ukázalo už samotné „strojové“ vyhledávání oblastí, které by mohly textovou informaci obsahovat a to především s ohledem na graficky velice náročné předlohy, které se všude kolem nás běžně vyskytují. Proto většinou nezbyvá, než se zpracovávanou předlohu pokusit před samotným procesem rozpoznávání dostatečně upravit, aby úspěšnost v získávání výsledků byla co nejvyšší, odfiltrovat nadbytečné informace a usnadnit tak práci „rozpoznávači“. Po převodu obrazu na text je dále třeba z pozice v předloze určit tok textu.

Cílem této práce bylo navrhnout knihovnu, pomocí níž by bylo možné odseparovat textové informace z obrazové předlohy, a označit oblasti, ve kterých by se mohl nalézat text určený k rozpoznávání znaků pomocí programu.

Po popisu nejrůznějších obecných metod, určených k práci s diskrétním obrazem, které jsou jako základní stavební kámen návrhu popsány v první části této práce je další její část zaměřena na rozdělení kategorií textu a obrazu, popis jejich specifických vlastností, způsobu práce s nimi a zvažuje případná úskalí, která se mohou vyskytnout při samotném vlastním procesu vyhledávání. Závěrečná část popisuje řešení použitá v průběhu návrhu uživatelského rozhraní testovací aplikace, zkušenosti z jejich testování a zdůvodnění vybraných metod.

Další potřebné informace jsou uvedeny na přiloženém CD.

## 2. REPREZENTACE OBRAZOVÝCH DAT

### 2.1 OBRAZOVÁ FUNKCE

Základním pojmem této práce jsou obrazová data, neboť tvoří data vstupní i výstupní.

Digitální obrazová data tvoří dvourozměrnou diskrétní funkci, kde proměnnými jsou souřadnice daného bodu (pixelu) a jejíž hodnota je vlastnost bodu, která je závislá na barevném prostoru, ve kterém obraz zpracováváme. Tuto funkci nazýváme obrazovou funkcí. Nejčastěji bývá použit prostor RGB, ve kterém je hodnota obrazové funkce tvořena hodnotami intenzity červené, zelené a modré složky. Používají se i jiné barevné prostory (např. RGBA, CMY, CMYK, aj.), které jsou popsány v [1]. My se omezíme na barevný prostor RGB. Pro veškeré operace pro zpracování obrazu budeme zpravidla používat jednodušší způsob reprezentace obrazových dat, a to obraz reprezentovaný širokou škálou šedých odstínů (dále jen šedotónový obraz). Obrazová funkce v šedotónovém obraze vrací jasovou hodnotu daného pixelu. Při převodu na šedotónový obraz musíme zohlednit fakt, že lidské oko není stejně citlivé na jednotlivé barevné složky. Proto není vhodné počítat s aritmetickým průměrem všech tří složek. Při převodu obrazu z barevného prostoru RGB na šedotónový obraz musíme přidělit jednotlivým barevným složkám váhové koeficienty, které kompenzují výše zmíněnou rozdílnou citlivost lidského oka na různé odstíny. Máme-li obrazová data reprezentována v barevném prostoru RGB s obrazovou funkcí  $f(i,j)$ , kde koeficienty  $i, j$  jsou souřadnice aktuálně převáděného pixelu, potom hledáme obrazovou funkci  $b(i,j)$  v šedotónovém prostoru pomocí vztahu (2.1) pro každý bod obrazu.

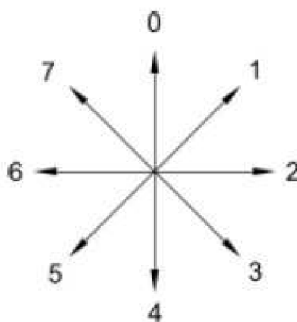
$$b(i, j) = 0,299R(f(i, j)) + 0,587G(f(i, j)) + 0,114B(f(i, j)) \quad (2.1)$$

V této kapitole bylo čerpáno z [1].

## 2.2 ŘETĚZCOVÉ KÓDY

Některé metody potřebují zaznamenat hranici oblasti. Řetězcové kódy budou v této úloze klíčovým pomocníkem.

Řetězcový kód je tvořen posloupností značek (nejčastěji číslic), které nám popisují trajektorii, po které jsme se dostali z počáteční pozice na pozici aktuální. Jedním z řetězcových kódů je Freedmanův kód. Pomocí číslic 0 až 7 popisuje všechny možné směry trajektorie do jedno-pixelového okolí (osmi-okolí). Jestliže odebereme směry, popsané číslicemi 1, 3, 5 a 7, získáme soubor hodnot pro čtyř-okolí aktuálního pixelu.



Obrázek 2.1 Popis směrů Freemanova kódu

### 3. POJMY A METODY SPOJENÉ S DISKRÉTNÍM OBRAZEM

#### 3.1 HISTOGRAM

Histogram je jedním z fundamentálních nástrojů pro kvalifikaci vlastností obrazů. Histogram je vektor, který udává absolutní četnost výskytu jasových hodnot obrazu.

Obsahuje-li obraz  $x \cdot y$  bodů, musí platit vztah (3.1)

$$\sum_{i=0}^{\max} H(i) = x \cdot y \quad (3.1)$$

kde  $H(i)$  je četnost výskytu  $i$ -té složky jasu. V šedotónovém obraze dosazujeme za intenzity hodnoty jeho obrazové funkce, v obraze v barevném prostoru RGB bývá často histogram rozdělen na tři histogramy (pro každou základní barvu zvlášť). Výhodou histogramu je absence informace o poloze jednotlivých intenzit v obraze. Pomocí histogramu jsme schopni zjistit například vhodnou hodnotu prahu pro prahování, zda je obraz dostatečně kontrastní a mnohé další.

V této kapitole bylo čerpáno z [1].

#### 3.2 KONVOLUCE NAD DISKRÉTNÍM OBRAZEM

Konvoluce dvou funkcí je matematická metoda, jejíž definice pro jednorozměrnou spojitou funkci je popsána vztahem (3.2).

$$f(x) * h(x) = \int_{-\infty}^{\infty} f(x - \alpha)h(\alpha)d\alpha \quad (3.2)$$

kde  $h(x)$  je funkce, kterou označujeme jako konvoluční jádro. Konvoluci si lze představit jako posouvání konvolučního jádra nad funkcí  $f(x)$ . Po úpravě do diskrétního tvaru a převodu na funkci dvou proměnných získáme vztah (3.3).

$$b(i, j) * h(i, j) = \sum_{x=-k}^k \sum_{y=-k}^k f(i-x, j-y)h(x, y) \quad (3.3)$$

Konvoluční jádro v tomto případě tvoří čtvercovou matici o hodnotě  $2k+1$ . Konvoluce tvoří základní stavební kámen pro mnoho operací nad obrazem (například detekce hran, šumové filtry).

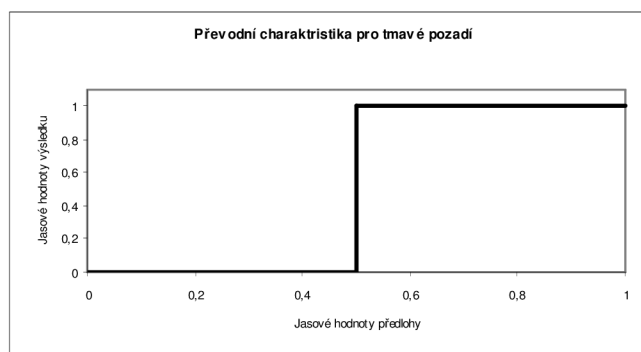
V této kapitole bylo čerpáno z [2].



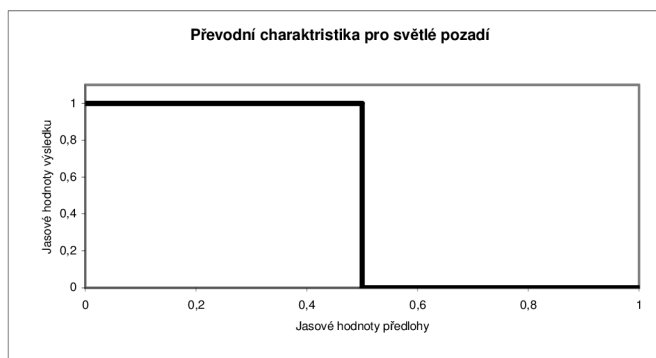
## 4. OPERACE NAD DISKRÉTNÍM OBRAZEM

### 4.1 PRAHOVÁNÍ

Mezi nejjednodušší metody oddělení objektů od pozadí patří prahování. Vstupem do této funkce bývá nejčastěji šedotónový obraz a výstupem je obraz černobílý. Tato metoda má několik variant prahování : globální, globální s hysterezí, adaptivní, ...



**Obrázek 4.1 Převodní charakteristika prahování pro pozadí tmavší nežli text  
(hodnota prahu 0,5)**



**Obrázek 4.2 Převodní charakteristika prahování pro pozadí světlejší nežli text  
(hodnota prahu 0,5)**

#### 4.1.1 Globální prahování

U metody globálního prahování zpravidla určíme jedinou hodnotu prahu přes celou oblast. Jestliže máme obrazovou funkci  $b(i, j)$ , kde  $b(i, j) \in \langle 0; 1 \rangle$ , koeficienty  $i, j$  jsou souřadnice aktuálního bodu (pixelu) a máme hodnotu prahu  $t \in (0; 1)$ , potom určíme obrazovou funkci  $d(i, j)$ , kde  $d(i, j) \in \{0; 1\}$  dle vztahu (4.1):

$$d(i, j) = \begin{cases} 0 & \text{pro } b(i, j) < t \\ 1 & \text{pro } b(i, j) \geq t \end{cases} \quad (4.1)$$

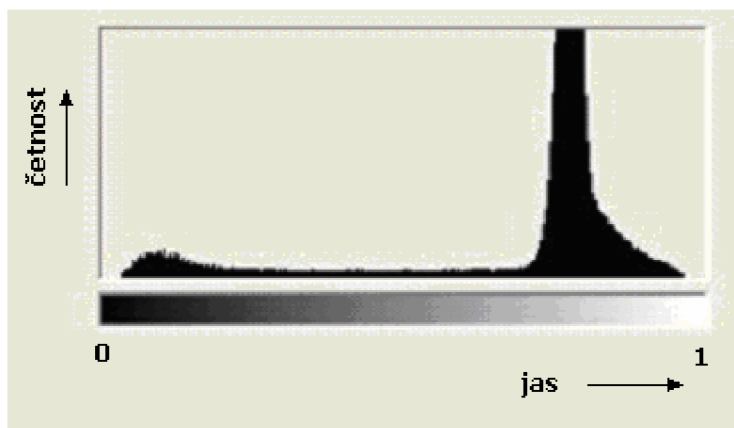
Tato metoda je vhodná především pro obrazy, jejichž histogram je tvořen pouze dvěma výraznými maximy. Jako hodnotu prahu určíme hodnotu mezi těmito maximy, přičemž předpokládáme, že jedním maximem je odstín pozadí a druhým je odstín vyhledávaných objektů (v našem případě tedy textu). Nevýhodou je také citlivost na rušení. Tato metoda se tedy hodí pouze na obrazy, které jsou rovnoměrně osvětlené a jejichž histogram nevykazuje přítomnost výrazných rušivých elementů.

Vzorec (4.1) používáme pouze v případě, že vyhledáváme světlý text na tmavém pozadí. Jeho převodní charakteristika je naznačena na obrázku Obrázek 4.1. Pokud však máme tmavý text na světlém pozadí, musíme ve vzorci (4.1) prohodit jasové hodnoty, abychom dosáhli jednotného výstupu. Zda má předloha tmavé nebo světlé pozadí poznáme z histogramu podle toho ve které jeho části se nachází globální maximum, které ve většině případů náleží právě pozadí. Tato skutečnost se vyskytuje i ve všech dalších variantách prahování.

Obrázek 4.3 tvoří předlohu, jejíž histogram (**Chyba! Nenalezen zdroj odkazů.**) obsahuje dvě maxima. Protože se její globální maximum nachází v oblasti vysokých jasových hodnot, musíme použít převodní charakteristiku dle obrázku Obrázek 4.2. Z histogramu je patrné, že lokální maximum jasových hodnot textu je výrazně nižší, než maximum odpovídající pozadí. Na obrázku Obrázek 4.5 je zobrazen invertovaný výsledek globálního prahování.

To communicate critical information, infrared images only are no longer enough. With revolutionary IR-Fusion technology, one can better identify details, manage and analyze images by combining both the infrared and visible light images. IR-Fusion technology simultaneously captures pixel-for-pixel infrared and visible light images and allows full image optimization with 5 different on-camera as well as software viewing modes. With the integrated laser pointer visible on the images, precise and accurate (faulty) component identification is very easy.

**Obrázek 4.3** Prahovaná předloha



**Obrázek 4.4** Histogram obrázku 4.3

To communicate critical information, infrared images only are no longer enough. With revolutionary IR-Fusion technology, one can better identify details, manage and analyze images by combining both the infrared and visible light images. IR-Fusion technology simultaneously captures pixel-for-pixel infrared and visible light images and allows full image optimization with 5 different on-camera as well as software viewing modes. With the integrated laser pointer visible on the images, precise and accurate (faulty) component identification is very easy.

**Obrázek 4.5** Invertovaný výsledek prahování obrázku 4.3

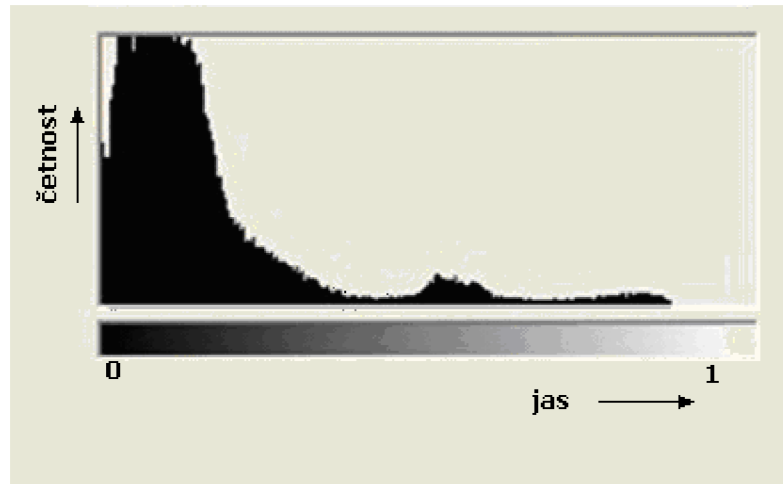
Na obrázku Obrázek 4.6 je zobrazena další předloha, na kterou aplikujeme globální prahování. Při pohledu na histogram předlohy (**Chyba! Nenalezen zdroj odkazů.**) však není patrné, že je tato předloha vhodným adeptem na globální prahování. Pokud ovšem vyjdeme ze známé skutečnosti, že se jedná o informaci, zobrazenou na svítivém displeji, můžeme nastavit hodnotu prahu v blízkosti lokálního maxima odpovídajícího nejvyšší jasové hodnotě. Výsledek tohoto zpracování je zobrazen invertovaně na obrázku Obrázek 4.7.



Obrázek 4.6 Prahovaná předloha



Obrázek 4.7 Invertovaný výsledek



Obrázek 4.8 Histogram předlohy 4.6

#### 4.1.2 Globální prahování s hysterezí

Na rozdíl od globálního prahování potřebuje tato metoda dva prahy  $t_H \in (0;1)$  a  $t_L \in (0;1)$  a zároveň  $t_H > t_L$ . Potom získáme černobílou obrazovou funkci  $d(i,j)$  podle vztahu (4.2):

$$d(i, j) = \begin{cases} 0 & \text{pro } b(i, j) < t_L \\ 1 & \text{pro } b(i, j) > t_H \\ \text{dle okolí} & \text{pro } b(i, j) \in \langle t_L, t_H \rangle \end{cases} \quad (4.2)$$

Pojmem „dle okolí“ je myšlen test izolovanosti daného bodu. Je-li ve čtyř-okolí nebo v osmi-okolí bod, který již byl detekován jako bod objektu (byl nastaven na hodnotu 1), bude i tento bod nastaven na hodnotu 1. Jestliže se však v okolí aktuálního bodu nenachází bod, který by měl hodnotu 1, je tomuto bodu přiřazena hodnota 0. Tato metoda dokáže poměrně účinně potlačit šum, jehož jasová hodnota se nachází mezi hodnotami  $t_L$  a  $t_H$ . Problémem však zůstává nutnost zvolit prahové hodnoty konstantní pro celý obraz. Tento problém lze odstranit rozdělením obrazu na menší celky, pro které budou voleny hodnoty prahů vždy podle aktuálních potřeb

dané podoblasti nebo přímo pro každý pixel zvlášť podle předem určeného okolí pixelu.

### 4.1.3 Adaptivní prahování

Jestliže dochází na pozadí obrazu ke značným změnám, popř. obraz obsahuje velké množství šumu nebo jiných rušivých přechodů, není možné použít pro celý obraz pouze jednu hodnotu prahu. Obraz tedy musíme rozdělit na menší celky, pro které určíme hodnotu prahu zvlášť (obecně každému pixelu). Prahová hodnota je tedy určena nejen odstínem šedé, ale i lokální vlastností prahovaného pixelu. Obecně tedy můžeme napsat:

$$t = f(x, y, b(x, y), h(x, y)) \quad (4.3)$$

kde  $x$  a  $y$  jsou souřadnice prahovaného bodu,  $b(x, y)$  je jasová hodnota bodu a  $h(x, y)$  je již zmiňovaná lokální vlastnost daného pixelu. Otázkou zůstává, jak nalézt vhodnou hodnotu prahu. Odpověď na tuto otázku získáme pomocí metod Chow a Kanenko a Otsu (metoda optimálního prahu), nebo některou ze statistických hodnot (například aritmetický průměr nebo medián).

V této kapitole bylo čerpáno z [2], [4].

## 4.2 REDUKCE ŠUMU

### 4.2.1 Gaussův filtr

Jak již bylo uvedeno výše, Gaussův filtr pro svoji činnost používá diskrétní konvoluci. Rozměr samotného filtru zpravidla stanovujeme na takový, aby ke své činnosti používal jedno- nebo dvou-pixelové okolí. Hustota normálního rozložení pro funkci jedné proměnné je dána vztahem (4.4):

$$g(x) = \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{(x-\mu)^2}{2\sigma^2}} \quad (4.4)$$

kde  $\mu$  je střední hodnota a  $\sigma^2$  je rozptyl. Normální rozložení bývá zapisováno ve tvaru  $N(\mu, \sigma^2)$ . Po dosazení normovaného tvaru  $N(0,1)$  a převodu funkce hustoty pravděpodobnosti na tvar dvou proměnných získáváme vztah (4.5):

$$G(x, y) = \frac{1}{2\pi} e^{-\frac{x^2+y^2}{2}} \quad (4.5).$$

Na závěr této kapitoly je nutno podotknout, že všechny filtry šumu, pracující na principu diskrétní konvoluce, musí mít jednu společnou vlastnost, a to, že součet prvků matice konvolučního jádra musí být roven jedné, aby nám neměnil jas v obraze.

#### 4.2.2 Mediánový filtr

Na rozdíl od Gaussova šumového filtru nevyužívá mediánový filtr diskrétní konvoluce. Mediánový filtr nahrazuje hodnotu aktuálně upravovaného pixelu mediánem získaným z okolí tohoto bodu. Mezi jeho nevýhody patří především přerušování tenkých čar. Pro vylepšení vlastností bývá klasické čtvercové okolí nahrazeno například křížem.

V této kapitole bylo čerpáno z [4].

### 4.3 DETEKCE HRAN

Jednou ze základních a zároveň důležitých metod pro segmentaci obrazu a následné vyhledání požadované informace je nalezení hran. Hrana je místo v obraze, kde obrazová funkce skokově mění svoji hodnotu. Zajímají nás tedy pozice, kde obrazová funkce obsahuje vysoké frekvence. Tato metoda je opačná k metodám odstraňujícím šum, ve kterých naopak vysoké frekvence potlačujeme.

Pro určení hrany potřebujeme znát odhad gradientu obrazové funkce. Gradient je dán vztahem (4.6), velikost vztahem (4.7) a směr vztahem (4.8):

$$\nabla f(x, y) = \left( \frac{\partial f(x, y)}{\partial x}, \frac{\partial f(x, y)}{\partial y} \right) \quad (4.6)$$

$$|\nabla f(x, y)| = \sqrt{\left(\frac{\partial f(x, y)}{\partial x}\right)^2 + \left(\frac{\partial f(x, y)}{\partial y}\right)^2} \quad (4.7)$$

$$\theta = \arctg \frac{\frac{\partial f(x, y)}{\partial y}}{\frac{\partial f(x, y)}{\partial x}} \quad (4.8)$$

V některých případech nebývá velikost počítána dle vztahu (4.7), nýbrž jeho aproximací pomocí vztahu (4.9):

$$|\nabla f(x, y)| = \left| \frac{\partial f(x, y)}{\partial x} \right| + \left| \frac{\partial f(x, y)}{\partial y} \right| \quad (4.9)$$

Protože pracujeme s diskretním obrazem, používáme odhad gradientu. Pro vypočtení odhadu gradientu se nejčastěji používají metody analyzující okolí s použitím konvolučního operátoru.

#### 4.3.1 Robertsův operátor

Robertsův operátor je zřejmě nejjednodušší metoda pro určení gradientu v diskretním obraze. Pro svoji činnost používá aktuálně vybraný pixel a jeho tři sousedy. Velikost gradientu je pak dána vztahem (4.10):

$$|\nabla b(i, j)| = |b(i, j) - b(i + 1, j + 1)| + |b(i + 1, j) - b(i, j + 1)| \quad (4.10)$$

Ze vztahu (3.14) je patrné, že tento postup určuje velikost gradientu z velikosti změn v hlavní a vedlejší diagonále. Robertsův operátor tedy velice dobře detekuje hrany se sklonem  $45^\circ$ . Pro odhad gradientu obrazové funkce je však možné použít i dvojici konvolučních jader, jako například (4.11) a (4.12) v kombinaci s níže uvedeným vztahem (4.14).



$$h = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & -1 \end{bmatrix} \quad (4.11)$$

$$\bar{h} = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & -1 & 0 \end{bmatrix} \quad (4.12)$$

Na principu rozdělení výpočtu do dvou složek pracuje i dále popsáný Sobelův operátor.

#### 4.3.2 Sobelův operátor

Sobelův operátor detekuje hrany pomocí dvou konvolučních jader, která jsou navzájem pootočená o  $90^\circ$  kolem svého středu. Tento operátor aproximuje první derivaci obrazové funkce. Používáme-li pro výpočet velikosti gradientu druhou mocninu či absolutní hodnotu, nezáleží nám na směru rotace konvoluční masky ( $90^\circ$  nebo  $-90^\circ$ ). Podobně jako ve vztazích pro spojitou funkci (4.7) a (4.9) počítáme velikost gradientu pomocí vztahu (4.13) nebo ve zjednodušené variantě dle vztahu (4.14):

$$|G(i, j)| = \sqrt{(f(i, j) * h(i, j))^2 + (f(i, j) * \bar{h}(i, j))^2} \quad (4.13)$$

$$|G(i, j)| = |f(i, j) * h(i, j)| + |f(i, j) * \bar{h}(i, j)| \quad (4.14)$$

kde  $h(i, j)$  je konvoluční jádro Sobelova operátoru a  $\bar{h}(i, j)$  je komplementární matice k  $h(i, j)$  otočená o  $90^\circ$  kolem svého středu. Příkladem matic tvořících komplementární pár pro získání odhadu gradientu pomocí Sobelovy metody jsou matice (4.15) a (4.16). Velkou výhodou Sobelova operátoru je jeho značná odolnost vůči šumu.

$$h = \begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix} \quad (4.15)$$

$$\bar{h} = \begin{bmatrix} 1 & 2 & 1 \\ 0 & 0 & 0 \\ -1 & -2 & 1 \end{bmatrix} \quad (4.16)$$

U Cannyho hranového filtru, používaného pro ztenčování, je navíc třeba znát i směr gradientu. Výpočet směru gradientu je dán vztahem (4.17)

$$\theta = \operatorname{arctg} \frac{G_y}{G_x} \quad (4.17)$$

kde  $G_y = b(i, j) * \bar{h}(i, j)$  a  $G_x = b(i, j) * h(i, j)$ . V případě potřeby znalosti směru gradientu však již není možné otáčet konvolučním jádrem libovolným směrem pro získání komplementární matice.

Sobelova a Robertsova metoda odhadují gradient pomocí první derivace. Některé metody neaproximují první derivaci, ale používají derivaci druhou. Jestliže hledáme lokální extrémy gradientu, je početně jednodušší použít jeho derivaci, neboť nalezení místa, kde funkce prochází nulou, je jednodušší nežli hledání jejího extrému. Těmto metodám říkáme metody zero-crossing. Jejich velkou nevýhodou je fakt, že jsou ještě citlivější na všudypřítomný šum. Příkladem metody, která aproximuje druhou derivaci obrazové funkce, je metoda založená na Laplaceově operátoru.

### 4.3.3 Detekce hran pomocí Laplaceova operátoru

Pro spojité funkce více proměnných je Laplaceův operátor definován pomocí vztahu (4.18):

$$\Delta f(x, y) = \frac{\partial^2 f(x, y)}{\partial^2 x} + \frac{\partial^2 f(x, y)}{\partial^2 y} \quad (4.18)$$

Pro diskrétní obrazovou funkci použijeme konvoluční jádro, ve kterém pracujeme se čtyř-okolím (4.19) nebo osmi-okolím (4.20). Použitím tohoto postupu uvidíme problém ve formě dvojité odezvy na hranu (na její vzestupné a sestupné straně). Možné záporné výsledky, které můžeme konvolucí získat, je možné odstranit posunutím do kladných hodnot, oříznutím nebo převedením na absolutní hodnotou.

$$L_4 = \begin{bmatrix} 0 & 1 & 0 \\ 1 & -4 & 1 \\ 0 & 1 & 0 \end{bmatrix} \quad (4.19)$$

$$L_8 = \begin{bmatrix} 1 & 1 & 1 \\ 1 & -8 & 1 \\ 1 & 1 & 1 \end{bmatrix} \quad (4.20)$$

Další varianty konvolučních jader jsou uvedeny v [4]. Některé metody používají matice o větších rozměrech, což jim zvýší odolnost vůči šumu, ovšem za cenu vyšší výpočetní náročnosti. Všechna konvoluční jádra, určená pro detekci hran, musí splňovat vlastnost, že součet všech prvků konvoluční matice musí být roven nule. Pouze při splnění této podmínky získáme nulovou odezvu konvoluce pro pixel, jehož hodnota obrazové funkce je rovna hodnotám obrazové funkce uvažovaného okolí a neobsahuje tedy hranu.

#### 4.3.4 Cannyho hranový detektor

Abychom dosáhli optimálních výsledků při nalezení hran, nestačí použít pouze některý z výše popsaných algoritmů. Pro optimální výsledek je výhodné použít Cannyho hranový detektor, který někdy bývá označován jako optimální hranový detektor. Popisovaný algoritmus si klade za cíl splnit následující podmínky:

- Co nejmenší počet chyb (rozlišení hran a šumu)

- Maximální přesnost (maximální přesnost polohy nalezené hrany)
- Jednoznačnost (každá hrana musí být v obraze pouze jednou)

Celý algoritmus můžeme rozdělit na několik částí:

V prvním kroku aplikujeme na obraz Gaussův filtr, pomocí kterého snížíme hladinu šumu, aniž bychom výrazně poškodili hrany předlohy.

Dále musíme zjistit velikost a směr gradientu. Pro tuto operaci je velice výhodný výše popsán Sobelův hranový detektor, který není příliš citlivý na šum a tudíž poskytuje dobré výsledky.

Poté co jsme snížili hladinu šumu v předloze a získali velikost a směr gradientu, přichází ke slovu ztenčování (thinning). Ztenčování má za úkol vybrat z obrazu pouze lokální extrém, čímž zvýšíme přesnost pozice hrany. Algoritmus pracuje tak, že za bod hrany uzná jen takový bod, jehož oba sousedé, ve směru kolmém na směr gradientu, mají nižší hodnotu gradientu, nežli je hodnota gradientu právě testovaného pixelu.

Na závěr je ještě nutné výsledek vyprahovat prahováním s hysterezí, čímž zajistíme co nejnižší počet chyb detekce.

V této kapitole bylo čerpáno z [1], [4].

## 5. TEXT V OBRAZE OBECNĚ

### 5.1 POJMY TEXT A ZNAK

Pojem text lze definovat jako posloupnost znaků, pomocí kterých je objektivním způsobem vyjádřena informace.

Znak je symbolem, který má pro širší skupinu čtenářů stejný význam. Zároveň má s jistou mírou podobnosti definovaný tvar.

Pro člověka, který se naučil tyto tvary, je tedy naprostou samozřejmostí nalezení tohoto tvaru a jeho dekodování. Pro vyhledávání textu pomocí stroje je tato úloha však značně složitější. Existuje totiž nepřeborné množství variant tvarů, které náleží jednomu znaku. K dalším problémům patří fakt, že ne každý tvar symbolu, který připomíná znak, je žádoucí za znak považovat. Samotné vyhledávání textu v obrázku tvoří velmi složitou úlohu, jejíž složitost spočívá především v její obrovské neurčitosti.

Základní rozdělení textu lze provést na text psaný strojově a text psaný ručně.

Při vyhledávání textu, který je napsán strojem, se musí algoritmus vypořádat se dvěma základními problémy. Prvním problémem je nepřeborné množství fontů, které náleží jedné abecedě. Druhým problémem je existence některých znakových sad, pomocí kterých lze textově vyjádřit i symboly, které by čtenáře ani nenapadlo považovat za text.

Při zpracování textu psaného ručně bývá situace ještě komplikovanější. Každý z nás má totiž svůj originální rukopis a dokonce stejné znaky psané jedním člověkem nejsou zcela shodné. Při psaní textu ručně navíc většinou dochází k vynechávání některých znaků, změnám velikosti znaků a sklonu písma. V různých kombinacích znaků dochází pak dokonce k významným změnám jejich tvaru.

## 5.2 VLASTNOSTI TEXTU

Algoritmus, který slouží k samotnému vyhledání textu a jeho přípravě pro proces rozpoznávání má zásadní problém v tom, že „neumí číst“. Nezbyvá tedy, než se zaměřit na charakteristické vlastnosti, které text většinou má a definovat omezující podmínky.

První vlastností, která je v textu přítomna téměř vždy, je organizace znaků do slov a řádků.

Abychom byli úlohu vůbec schopni řešit, musíme zavést řadu omezení, která ovšem příliš nezúží samotné použití algoritmu.

Mezi tato omezení je možno zařadit jednotnost barvy znaků alespoň na úrovni slov. Existují sice i předlohy, pro které je toto omezení zcela nepřijatelné, nicméně pro většinu předloh není toto omezení zásadní.

Mezi vhodná omezení patří i omezení se na znaky stejné velikosti opět alespoň na úrovni slov.

## 5.3 CHARAKTERISTICKÝ VEKTOR ŘÁDKU

Pokud se nám podaří vhodně sloučit znaky do dostatečně dlouhého řádku, můžeme použít pro samotné rozhodnutí, zda vybraný shluk je či není textem, vektor obsahující četnosti výskytu jasných složek textu nebo vektor, obsahující četnosti výskytu přechodů mezi jasnými hodnotami textu a pozadí. Pro tuto metodu je vždy nutno co nejpřesněji znát orientaci řádku. Pokud se algoritmus, byť jen nepatrně, odchýlí od této orientace, dochází ke značné deformaci charakteristiky.

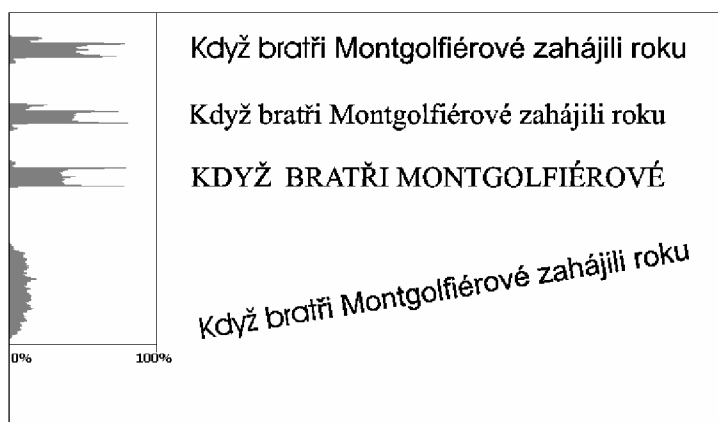
Oba tyto vektory je možno popsat vztahem (5.1)

$$v(j) = \sum_{i=L}^P s(i, w(i, j)) \quad (5.1)$$

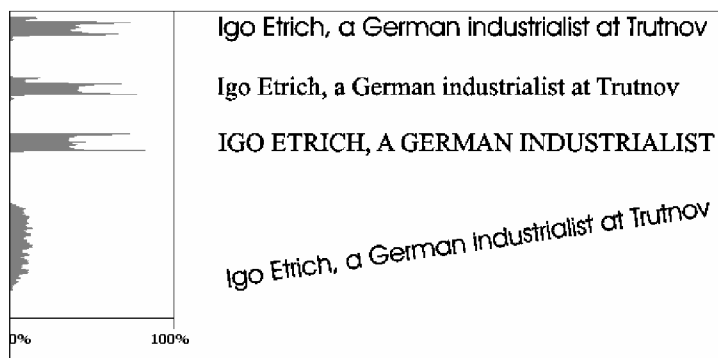
kde  $j$  je index vektoru  $v$ ,  $s$  představuje funkci, která má hodnotu 1 pro hodnotu obrazové funkce odpovídající textu (nebo na rozmezí text-pozadí), konstanta  $L$  představuje souřadnici na ose  $x$  bodu, který je nejvíce vlevo, konstanta  $P$  představuje souřadnici na ose  $x$  bodu, který je nejvíce vpravo a funkce  $w(i, j)$  popisuje přepočít

souřadnice  $y$  dle orientace řádku. Tento přepočítání nemusí mít pouze lineární charakter (pouhý náklon) ale může popisovat i zakřivení.

Popis charakteristiky pro ručně psaný text lze nalézt v [7]. My se zde budeme dále zabývat především textem psaným strojově. Na obrázcích Obrázek 5.1 a Obrázek 5.2 jsou znázorněny různé varianty textu a jejich řádkové vektory. Z těchto obrázků je patrný problém této metody. Pokud text obsahuje pouze znaky anglické abecedy a neobsahuje dostatek velkých znaků, ztrácí se lokální maximum v horní části charakteristiky. Pokud řádek neobsahuje znaky, jejichž body zasahují pod pomyslnou linku, ztrácí se lokální maximum i ve spodní části charakteristiky. Pokud text obsahuje pouze velká písmena, dochází ke značné deformaci hodnot vektoru. Problematické je i algoritmické hodnocení hodnot vektoru. Samozřejmě se v textu může vyskytnout i více sloučených obrazců, které mají stejný charakteristický vektor jako text a přitom textem nejsou.



Obrázek 5.1 Příklady textů a jejich char. vektoru psaných českou abecedou



Obrázek 5.2 Příklady textů a jejich char. vektorů psaných anglickou abecedou

## 6. KATEGORIE TEXTU

Tato kapitola si klade za cíl základní rozdělení obrazových předloh, které obsahují textové informace. Dostatečně přesné definování kategorie textu v obraze dokáže na únosnou míru zúžit obecnost úlohy a zvyšuje tedy úspěšnost procesu vyhledávání. V některých případech ani není žádoucí nalézt všechnen text, který předloha obsahuje, ale pouze některé jeho oblasti. Tyto kategorie se samozřejmě mohou téměř libovolně prolínat. V takovém případě je při procesu vyhledávání textu nutno předlohu vhodně rozdělit na dílčí celky.

### 6.1 PROSTÝ TEXT

Nejjednodušší kategorií je obraz, který obsahuje text napsaný na pozadí, jehož jasová hodnota je vždy hodně kontrastní s jasnem samotného textu. Jasové hodnoty textu si jsou velice blízké a pozadí je poměrně monotónní. V této kategorii, stejně jako ve všech ostatních, musí být správně zjištěno, zda se jedná o tmavý text na světlém pozadí či opačně. Pro tuto kategorii je vhodné použít prahování.

Hlavní rozdělení lze provést na předlohy, které na jedné úrovni obsahují pouze jeden řádek a na předlohy, které obsahují více navzájem spolu nesouvisících řádků různých odstavců. V předloze se navíc mohou vyskytovat rámečky a dlouhé čáry, které je třeba vždy odstranit.

Příkladem takovéto předlohy jsou obrazy získané skenováním některých stránek skript, některých knih, novinových článků, atd.

### 6.2 PŘEDLOHY OBSAHUJÍCÍ „NEPOPSANÉ“ OBRÁZKY

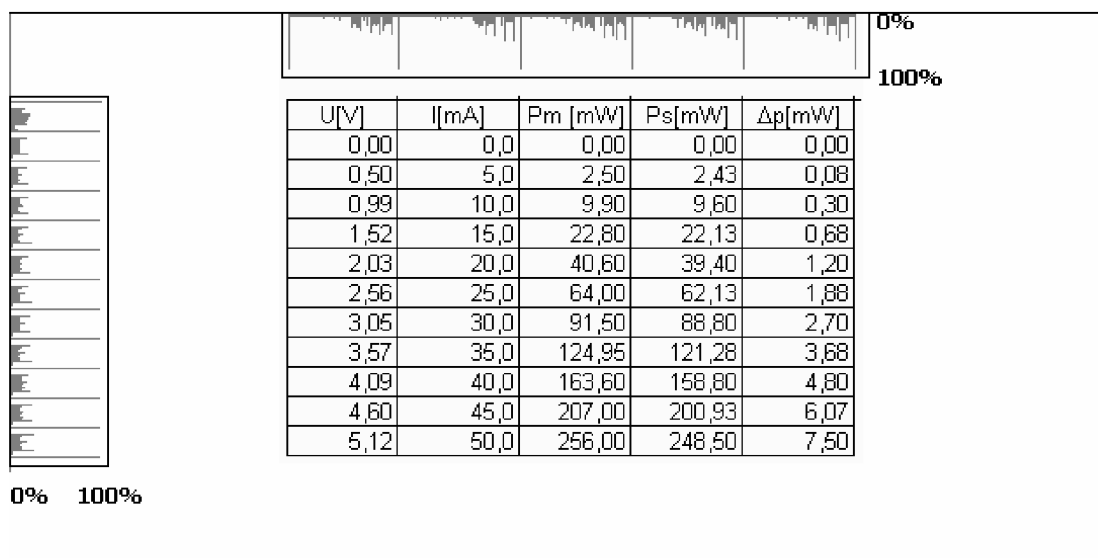
Většina předloh bývá ovšem mnohem rozmanitějších. Mezi časté objekty, vyskytující se v předloze, patří nepopsané obrázky. Pojmem nepopsaný obrázek je myšlen obrázek, který je třeba nalézt a z předlohy vyfiltrovat. V oblasti, kde se nachází obrázek, tedy neproběhne žádný pokus o nalezení textové informace. Při vyhledávání obrázku lze (v některých případech) využít jeho výrazného okraje. Zda se jedná o obrázek či nikoli můžeme poté zjistit například pomocí histogramu. Lze předpokládat, že histogram v této oblasti bude mnohem rozmanitější nežli je tomu



u oblastí obsahující pouze text a pozadí. Samotné vyhodnocení rozmanitosti histogramu je však úlohou velice složitou. U obrázků také nebývá vždy přítomen výrazný okraj. Zbylý postup je již totožný s kategorií předchozí.

### 6.3 TABULKY

Zcela zvláštní kategorií jsou tabulky. Tabulku lze definovat jako dvě množiny rovnoběžných čar, jejichž linie jsou na sebe kolmé. Ty potom tvoří hranice buněk, ve kterých se může nacházet text. K rozpoznání takovéto tabulky může být použit řádkový a sloupcový vektor, obsahující četnosti výskytu bodů náležící tabulce. V místech, kde se nalézají výrazná maxima, je umístěna linie. Tabulka a její řádkový a sloupcový vektor jsou znázorněny na obrázku Obrázek 6.1



**Obrázek 6.1 Příklad tabulky a jejího řádkového a sloupcového vektoru**

Tabulky však mohou být mnohem složitější. Mohou obsahovat sloučené buňky nebo dokonce nemusí být mřížka vyjádřena vůbec. V neposlední řadě je třeba ještě definovat, co tabulkou je a co není. Pokud bychom za tabulku považovali vše, co se nachází uvnitř mřížky, mohli bychom se dostat do problémů s textem, který je na orámované stránce. Je tedy nutno nastavit maximální výšku buňky nebo například minimální počet buněk, které musí tabulka obsahovat.

Jednou z možností vypořádání se s tabulkou je ji ignorovat. Druhou možností je pokusit se rozdělit tabulku na buňky a uložit jejich vzájemnou pozici.

#### **6.4 ROZMANITÉ OBRAZY OBSAHUJÍCÍ TEXT**

Tuto kategorii - co se týče vyhledání textu- můžeme bezesporu označit za nejsložitější. Tato kategorie může totiž obsahovat úplně libovolné kombinace textu a pozadí. Řádek textu zdaleka nemusí tvořit úsečka. Musíme tedy vyjít z předpokladu, že barva textu- aby text byl pro čtenáře snadno čitelný- musí být ve značném kontrastu s okolními objekty. Přitom se můžeme omezit na znaky, které jsou tvořeny pouze pixely s hodně blízkou jasovou hodnotou. Toto omezení v této kategorii může být však omezující až příliš. Nabízí se tedy možnost nalezení kontur znaků. V prvním kroku musíme na obraz aplikovat některý z hranových detektorů. Po dokončení detekce hran je třeba nalézt kontury objektů (a to i uvnitř jiné kontury). Na kontury lze aplikovat řetězcové kódy, které byly popsány výše. Následně je třeba zjistit, jestli se v blízkosti této oblasti nenachází další oblast, která rovněž splňuje požadavky na výšku nebo alespoň odpovídá svou jasovou hodnotou (další znak nebo částí téhož znaku). Pokud byla předchozím procesem nalezena alespoň jedna oblast splňující předchozí podmínku, je možno poskytnout tuto oblast procesu rozpoznávání textu. Oblasti, jejichž okraj tvoří vnitřní kontura znaku, mohou obsahovat další kontury.

Pro získání dobrých výsledků musíme vyjít z výše uvedeného předpokladu, že samotné znaky hledaného textu jsou dostatečně velké, neboť pro malé písmo nezískáme dostatečně kvalitní kontury a tudíž bude metoda velice často selhávat.

Mezi příklady této kategorie patří například reklamní letáky, billboardy, obrazové encyklopedie a obrázky z bulvárních časopisů.

#### **6.5 HODNOTY INDIKOVANÉ PŘÍSTROJI**

Řada přístrojů poskytuje operátorům informaci v podobě textu. Tento text v některých případech však nemá vlastnosti běžného textu. Přístroje také mohou obsluhu informovat pomocí znakových displejů. Tento obraz však lze zařadit do některé z předchozích kategorií a proto se jím nebudeme v této kapitole dále zabývat.

Často používaným textovým výstupem přístrojů jsou segmentové displeje. Obrazec jimi sestavený však netvoří uzavřenou oblast, ale několik dílčích uzavřených oblastí, mezi kterými jsou určité mezery. Tyto segmenty však mají předem daný tvar i svou možnou vzájemnou pozici. Pokud jsou tyto displeje tvořeny pomocí LED, nejlepším způsobem získání segmentů je globální prahování s prahem, který se nachází těsně pod lokálním maximem s nejvyšší jasovou hodnotou (viz **Chyba! Nenalezen zdroj odkazů.**).

Další skupinu tvoří přístroje, jež poskytují operátorům informaci pomocí stupnice, která je po určitém počtu dílků popsána. Pokud předem známe tvar stupnice (lineární, kruhová, oblouková), jsme schopni tuto stupnici odfiltrovat, protože její jednotlivé dílky by byly většinou také vyhodnoceny jako text, což by nám značně zpomalilo a často také zkreslilo výsledek. Při následném vrácení stupnice, spočítání dílků od okraje a nalezení vhodných rozpoznávaných hodnot jejího popisu můžeme určit zobrazovanou hodnotu.

Dalším způsobem, jak může přístroj reprezentovat hodnotu svého stavu, jsou mechanická počítadla, kterých je využito například v elektroměrech, plynoměrech nebo počítadlech provozních hodin. Specialitou tohoto způsobu zobrazení je stav, kdy číselný údaj nemusí být čitelný celý (může nastat stav, kdy je na některém segmentu viditelná jen část číslice předchozí a část číslice následující současně). Jednotlivé řády jsou však vždy ohraničeny rámečkem. Tato důležitá a specifická vlastnost je přítomna vždy.

Jako poslední příklad této kategorie je zde uveden způsob, jímž se vybírá pomocí bodového indikátoru (např. LED) mezi napsanými texty v okolí. Zpravidla je třeba vyfiltrovat vše kromě textu, který je nejbližší k tomuto indikátoru, jenž mívá velice vysokou jasovou hodnotu.

## 6.6 TEXT OBSAHUJÍCÍ VZORCE

Ke vzorcům je třeba přistupovat podobně jako k tabulkám. Nemá smysl provádět vyhledávání a následné rozpoznávání všech textových informací bez přesného určení jejich významu. Vzorce lze tedy, stejně jako tabulky, buď zcela ignorovat, nebo se pokusit o jejich zpracování a vytvořit sadu informací o tom, jak je

vybraná oblast umístěna v předloze. Při porovnání s tabulkou je však ještě komplikovanější určit, co vzorcem je a co není.

## 6.7 NĚKTERÉ ZVLÁŠTNÍ PŘÍPADY

Až doposud jsme se zabývali předlohami, ze kterých jsme se snažili získat většinu textových informací, které předloha obsahuje (kromě hodnot přístrojů). Existují ovšem i případy, kdy se nesnažíme získat všechnen text, ale jen jistou část, která má v dané úloze smysl. Příkladem této kategorie může být např. potřeba získat státní poznávací značku (SPZ) vozidla. Dané vozidlo může být přitom „popsáno“ nejrozmanitější reklamou, nebo jeho fotografie může obsahovat část reklamní tabule. Při samotné detekci tedy musíme nejprve nalézt oblast, kde se SPZ vozidla nachází a teprve poté zahájit přesné vyhledávání jejího obsahu. Musíme nejdříve nalézt uzavřenou obdélníkovou oblast, jejíž poměr stran je přesně definován. Teprve poté můžeme v této oblasti zahájit vyhledávání znaků s filtrací obrázků. Dalším takovým případem může být třeba rozpoznání PSC či celé adresy z pohlednice a dopisu, nebo nalezení tabule nesoucí název ulice či popisné číslo domu.

Zástupci těchto kategorií, včetně seznamu jejich původu, jsou umístěny na doprovodném CD.

## 7. POSTUP VYHLEDÁVÁNÍ TEXTU

Cílem popisovaných metod je získání hranic oblastí, u nichž je vysoká míra pravděpodobnosti, že by mohly obsahovat text. Jak již bylo uvedeno výše, zvolené řešení se snaží popsat vztah mezi jednotlivými elementárními oblastmi a následně provést jejich porovnání se vztahy mezi znaky či slovy v textu.

Samotné jádro je tvořeno čtyřmi třídami. Tyto třídy byly napsány v prostředí Borland® C++ Builder™ 6.0. Pro uložení hodnot, charakterizujících dílčí (elementární) oblast slouží třída `subshluk`. Pro popis sloučených dílčích oblastí slouží třída `shluk`. Většina metod pro vyhledávání je obsažena ve třídě `obr`. Poslední třídou je třída `hist`, která je nezbytná pro detekci tabulek nebo pro získání charakteristického vektoru řádku. Tyto operace však nebyly implementovány. Popis všech tříd je umístěn na doprovodném CD.

Úlohu lze rozdělit na několik dílčích částí:

- Načtení předlohy a základní předzpracování obrazové předlohy (převod na šedotónový a následně na černobílý obraz, šumový filtr)
- Segmentace obrazu na dílčí části a jejich popis
- Prvotní filtrace (odstranění některých dílčích částí)
- Hledání základních vztahů mezi oblastmi, které odpovídají vlastnostem textu a jejich sloučení
- Druhotná filtrace (filtrace na úrovni sloučených dílčích částí)
- Reprezentace výsledku

### 7.1 NAČTENÍ A PŘEDZPRACOVÁNÍ OBRAZU

Načtení a předzpracování obrazu jsou prvními kroky celého procesu. Pro tyto operace byla použita knihovna OpenCV (Open Source Computer Vision Library) od firmy Intel®. Tato knihovna obsahuje všechny funkce a struktury, potřebné pro tuto dílčí část zpracování. Především umožňuje přístup k jednotlivým pixelům obrazu prostřednictvím pole. Dále podporuje převod barevné předlohy na šedotónový obraz

a v neposlední řadě obsahuje funkce pro redukci šumu v obraze, prahování (globální i adaptivní) a detekci hran.

Pro účely této aplikace se nejvíce osvědčilo adaptivní prahování. Při jeho použití se však musíme vypořádat se dvěma problémy. Prvním problémem je volba okolí, ve kterém se zjišťuje hodnota prahu pro právě prahovaný bod. Aby byla předloha dobře převedena, musí být zvoleno dostatečně velké okolí, aby obsahovalo dostatek bodů pozadí i textu. Druhý problém se projeví v okamžiku, kdy předloha obsahuje světlý text na tmavém podkladu a tmavý text na světlém pozadí. Jestliže předloha obsahuje obě tyto varianty, bude po úpravě jedna varianta prahována dobře (Obrázek 7.2), druhá kombinace jasových hodnot však bude zpracována tak, že text i pozadí budou mít barvu černou a na rozmezí text-pozadí vznikne bílý obal, jehož šířka bude určena velikostí okolí pro prahování (Obrázek 7.1). Pokud má předloha charakter tmavého textu na světlém pozadí, lze použít adaptivní prahování s výstupní funkcí dle obrázku Obrázek 4.2, pokud má předloha opačný charakter, je nutné před provedením prahování získat negativ šedotónové předlohy. Tím získáme charakter tmavého textu na světlém pozadí a může být provedeno prahování. Pokud předloha obsahuje obě varianty, je nutné provést celý proces vyhledávání dvakrát pro obě varianty.



**Technology**

Obrázek 7.1 Nevhodné použití adaptivního prahování



**visible**

Obrázek 7.2 Vhodné použití adaptivního prahování

Hranové detektory jsou o něco robustnější, při jejich použití není problém s volbou okolí ani s jasovým vztahem textu a pozadí. Jako jejich největší nevýhoda se však ukázal vznik vnitřních oblastí v rámci jednoho znaku a také ztráta informace o tom, kolika body je oblast tvořena.

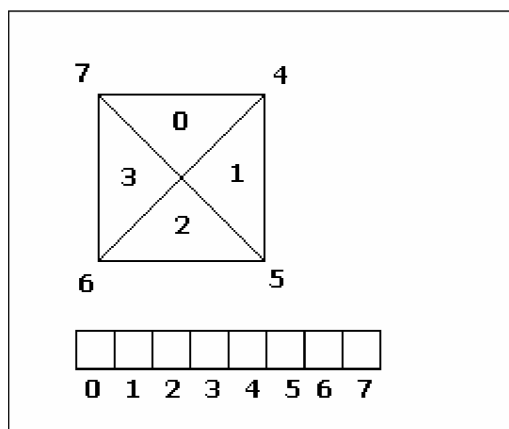
## 7.2 SEGMENTACE OBRAZU

Tato kapitola popisuje testované způsoby segmentace obrazu za účelem separace textu a následné získání oblastí, které by mohly tvořit slova nebo řádky; zde ukážeme jejich výhody a nevýhody a zdůvodníme volbu metody.

Při tomto procesu již máme převedenou předlohu na obraz černobílý, který tvoří vstupní data procesu.

### 7.2.1 Obalové čáry

Obalové čáry pracují na principu, kterým by bílé oblasti od černého pozadí odděloval člověk. Problémem, se kterým je třeba se vypořádat, je nespojitost hranice textu. V textu totiž lze většinou obalit samostatné znaky nebo více znaků, které se vlivem digitalizace předlohy sloučily v jeden celek. Algoritmus tedy musí pracovat nejen s bodem, jehož pohybem dochází k vykreslení hranice (obalové čáry), ale také s jeho okolím. Musí neustále sledovat přítomnost všech bílých bodů v tomto okolí a podle jejich pozic upravovat svůj směr postupu. Masku, kterou je sledováno okolí, ukazuje Obrázek 7.3.



Obrázek 7.3 Ukázka vyhodnocení testovaného okolí

Každá obalová křivka je popsána počátečním bodem a řetězcovým kódem, který určuje trajektorii, jež končí opět v počátečním bodě. Pro následné zpracování obrazu se ukázalo výhodnější použít řetězcový kód pro čtyř-okolí. Jelikož se jedná o hledání vnější hranice, byl zvolen postup z výchozího bodu po směru hodinových ručiček.

Celý proces začíná nalezením prvního bílého bodu, který není ohraničen žádnou obalovou křivkou. Následuje posun o velikost poloviny uvažovaného okolí, apriorní volba směru doprava a posun vztažného bodu. V dalším kroku nejprve proběhne naplnění pole, nesoucí četnosti výskytu bílých bodů v daných oblastech. Pokud jsou četnosti v oblastech ve směru postupu nenulové, následuje změna směru posunu dle Tabulka 7-2 Změna směrů posunu“. Pokud nastane situace, že výsledkem sledování okolí je jediný bílý bod masky, který leží na kraji diagonály, nastává opět změna směru, tentokrát dle Tabulka 7-1 Změna směrů posunu“. Tento postup se opakuje tak dlouho, dokud není dosaženo počátečního bodu.

Tento postup se však ukázal jako značně komplikovaný. Mezi jeho největší problémy patří možnost uváznutí například mezi řádky, vznik stavu, při kterém získáme nulový výskyt bílých bodů ve sledovaných oblastech, vznik hranic v již ohraničených oblastech a především nutnost vhodné volby sledovaného okolí, které zpravidla bývá neúnosně velké.

Problému uváznutí se lze zbavit tak, že nastavíme maximální délku řetězcového kódu a pokud je tato délka překročena před dosažením počátečního bodu, následuje smazání celé dosud vyznačené hranice, úprava velikosti okolí a opětovné spuštění algoritmu.

| Stávající směr | Sledovaná oblast (viz obr. 7.3) | Hodnota pro změnu | Nový směr |
|----------------|---------------------------------|-------------------|-----------|
| doleva         | 4                               | >0                | nahoru    |
| doprava        | 6                               | >0                | dolu      |
| nahoru         | 5                               | >0                | doprava   |
| dolu           | 7                               | >0                | doleva    |

**Tabulka 7-1 Změna směrů posunu**

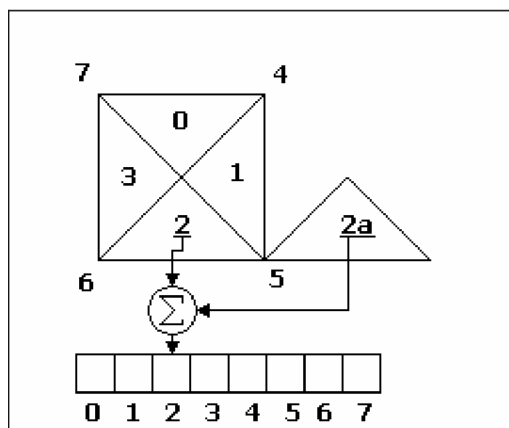


| Stávající směr | Sledovaná oblast (viz obr. 7.3) | Hodnota pro změnu | Nový směr |
|----------------|---------------------------------|-------------------|-----------|
| doleva         | 3                               | >0                | dolu      |
| doprava        | 1                               | >0                | nahoru    |
| nahoru         | 0                               | >0                | doleva    |
| dolu           | 2                               | >0                | doprava   |

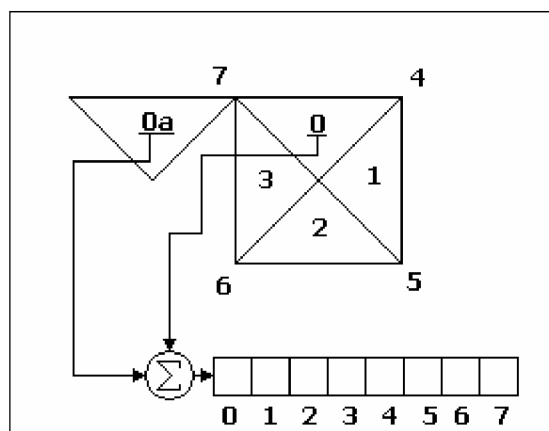
**Tabulka 7-2 Změna směrů posunu**

Problém se ztrátou výskytu bílých bodů lze odstranit uložení předchozího směru pohybu a jeho použitím v této situaci.

Velikost okolí lze snížit úpravou tvaru sledovaného okolí tak, jak ukazuje Obrázek 7.4 a Obrázek 7.5.



**Obrázek 7.4 Modifikovaný způsob vyhodnocení testovaného okolí (pro pohyb doprava a dolů)**



**Obrázek 7.5 Modifikovaný způsob vyhodnocení testovaného okolí (pro pohyb doleva a nahoru)**

Metoda obalových čar se však ukázala zbytečně komplikovaná a ne vždy poskytovala dostatečné výsledky, neboť při velkém obsahu šumu v předloze se občas stávalo, že výsledkem byla jediná obalová křivka, která uzavírala předlohu přes její okraj.

Obalové křivky lze také realizovat tak, že program vyznačí oblasti, kde přechází jasová hodnota obrazové funkce z hodnoty odpovídající barvě pozadí (černé) na jasovou hodnotu znaků (bílé). Následuje pak pospojování dostatečně blízkých hraničních bodů ve směru horizontálním i vertikálním. Takto získané hranice je pak možno obalit křivkou stejně jako v předchozím případě s tím rozdílem, že při této metodě stačí sledovat výskyt hraniční jasové hodnoty pouze v rámci jeho osmi-okolí.

Nevýhodou tohoto postupu je nutnost přesného odhadu maximální vzdálenosti pro pospojování hraničních bodů. Při špatně nastavené vzdálenosti může dojít k nutnosti současného shlukování a segmentace oblastí. Tuto metodu je možno použít jak v kombinaci s hranovými detektory, tak i s prahováním. Při použití hranového detektoru se však algoritmus musí vypořádat s problémem spojeným s obalením vnitřních hran znaků (například pro znaky O, A, ...).

**Volitelně může být přidána možnost nájezdu na polohu zadanou číselně, volba rychlosti pojezdu a podobně.**

**Výhodou je možnost pomocí manuálního režimu definovat polohy, které se využijí jako odkládací místa v automatickém režimu.**

### Obrázek 7.6 Ukázka výsledku segmentace obalovými čarami

#### 7.2.2 Segmentace uzavřených oblastí

Tento postup používá zcela jiný přístup. Vychází z výše definovaného omezení, že znaky jsou tvořeny body stejné barvy.

Celý proces začíná převodem předlohy na černobílý obraz pomocí prahování. Poté probíhá procházení matice, ve které jsou uloženy jasové hodnoty vyprahované předlohy, dokud není nalezen první bod o jasové hodnotě odpovídající bílé barvě. Následuje vyplnění této oblasti pomocnou barvou (šedou). Poté, co je dokončen proces vyplňování uzavřené oblasti, následuje získání elementárních hodnot popisujících tuto oblast. Mezi tyto hodnoty patří hodnota souřadnice  $y$  nejnižšího a nejvyššího bodu, hodnota souřadnice  $x$  bodu nejvíce vlevo a vpravo a počet bodů, které odpovídají vyznačené oblasti. Tyto hodnoty jsou zásadní pro další zpracování, tedy shlukování a filtraci oblastí. Následuje opět vyplnění této oblasti a to barvou pozadí (černou). Postup je opakován tak dlouho, dokud není dosaženo pravého dolního rohu předlohy bez nalezení další oblasti.

Velkou výhodou tohoto postupu, na rozdíl od metody obalových čar, je získání výšky znaků. Tato metoda však vyžaduje, aby byla obrazová předloha skutečně kvalitní, znaky byly tvořeny uzavřenými oblastmi (alespoň po částech) a aby při prahování nedošlo ke spojení znaku s nějakým dalším objektem (například rámečkem tabulky). Tuto metodu je možno použít v kombinaci s hranovými detektory, nicméně samotné hrany nám neposkytnou dostatek informací, které potřebujeme k dalšímu zpracování.

**Volitelně může být přidána možnost nájezdu na polohu zadanou číselně, volba rychlosti pojezdu a podobně.**

**Výhodou je možnost pomocí manuálního režimu definovat polohy, které se využijí jako odkládací místa v automatickém režimu.**

### Obrázek 7.7 Ukázka výsledku segmentace uzavřenými oblastmi

### 7.3 PRVOTNÍ FITRACE

Vstup tohoto procesu je tvořen souborem instancí tříd, které obsahují popis všech dílčích oblastí, které jsme získali prahováním. Tato filtrace má za cíl odstranění všech dílčích oblastí, které nemohou být znakem, případně souborem znaků.

Filtraci lze provést (dle přednastavených hodnot) podle následujících kritérií:

- Výšky větší než je přednastavená hodnota
- Výšky menší než je přednastavená hodnota (například znak, aby byl čitelný, musí mít výšku alespoň sedm bodů)
- Šířky větší než je přednastavená hodnota
- Šířky menší než je přednastavená hodnota (zde je však třeba dát pozor na nechtěné odstranění některých úzkých znaků, například i)
- Poměru počtu bodů nalezené oblasti k ploše obdélníka, opsaného okolo oblasti, který má strany rovnoběžné s okrajem obrázku. Pokud má tento poměr hodnotu dostatečně malou, jedná se pravděpodobně o rámeček, graf nebo dlouhou šikmou linií

Jak je patrné ze seznamu možností filtrace, umožňuje tento krok výraznou redukci dílčích oblastí. Při špatném nastavení hodnot filtrů však může dojít k odstranění dílčích celků, které by mohly být znaky. Dále tento filtr může odstranit diakritiku a interpunkci. Jejich odebrání však není zásadní pro samotné vyhledávání, neboť další kroky zpracování s ní nepracují. K jejímu navrácení dochází až v okamžiku závěrečné reprezentace výsledku.

### 7.4 HLEDANÍ VZTAHŮ MEZI OBLASTMI A JEJICH SLUČOVÁNÍ

Vstup tohoto procesu tvoří, stejně jako tomu bylo u prvotní filtrace, soubor dílčích oblastí. Cílem tohoto kroku je vytvořit větší celky, které by skutečně mohly být textem. Princip spočívá v nalezení dílčích oblastí, které jsou umístěny za sebou

a jejichž výška odpovídá textu stejné velikosti. Při řešení byl použit předpoklad, že znaky jsou umístěny za sebou a to zleva doprava.

Celý postup probíhá následujícím způsobem:

- Vzestupné seřazení všech dílčích oblastí podle  $x$ -ové souřadnice bodu dílčí oblasti, který je nejvíce vlevo.
- Vyzvednutí první oblasti jako aktuální.
- Hledání další oblasti, která se z hlediska  $y$ -nových souřadnic překrývá s aktuální oblastí
- Následuje řada testů, které upřesňují, zda by se mohlo jednat o text.
- Pokud byla oblast uznána, následuje sloučení s oblastí aktuální. Nově přidaná oblast je prohlášena za aktuální a dokud není dosaženo další oblasti nebo poslední dílčí oblasti, pokračuje algoritmus druhým bodem.
- Pokud (dle předchozího bodu) bylo dosaženo poslední dílčí oblasti, je prohlášena za aktuální dílčí oblast ta, která je opět nejvíce vlevo a zároveň ještě nebyla tímto procesem zpracovávána.
- Jestliže již ve vstupních datech není oblast, která nebyla zpracovávána, nastává ukončení tohoto kroku.

Test, který zpřesňuje rozhodnutí, zda se jedná o text či nikoli, je tvořen několika stupni. Nejdříve je testována výška. Jedná se o test, zda oblast, která má být přidána, má výšku větší než polovinu a menší než dvojnásobek průměrné výšky dílčích oblastí, které patří do stejného shluku jako oblast aktuální. V dalším kroku je testováno, zda se oblasti v  $y$ -ových souřadnicích dostatečně překrývají. Poslední krok není povinný a jedná se o test, zda není příliš velká vzdálenost mezi slučovanými oblastmi. Tento poslední test nemá význam u předloh, ve kterých se nachází pouze jeden řádek v rámci jedné hodnoty  $y$ -ové souřadnice. Pokud je tento test použit, tak poskytuje nejlepší výsledky při hledání další dílčí oblasti do vzdálenosti 1,3 násobku výšky oblasti.

## 7.5 DRUHOTNÁ FILTRACE

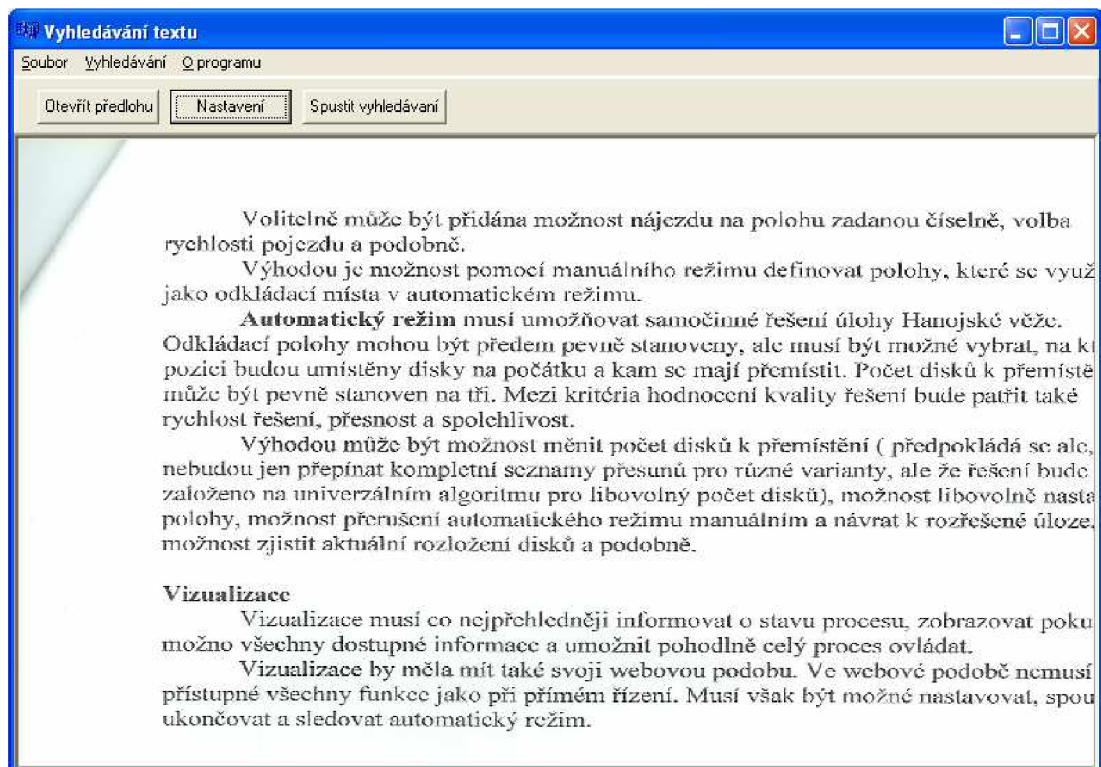
Druhotná filtrace tvoří poslední krok samotného vyhledávání textu v obraze. Do druhotné filtrace vstupuje soubor oblastí, které vznikly v předchozím kroku sloučením dílčích oblastí. Druhotná filtrace pracuje s omezeními, která byla definována v kapitole 5.2. Jedním z nejsilnějších nástrojů je výběr oblastí, které vznikly sloučením dostatečného počtu dílčích oblastí. Za dostatečný počet lze považovat už dvě oblasti. Nepříjemným důsledkem tohoto řešení je možnost odstranění jednotlivých písmen či číslic umístěných dostatečně daleko od ostatního textu. Tento problém lze částečně zredukovat u předloh, které obsahují na stejné výškové úrovni maximálně jeden řádek. Tím pádem není třeba během slučování dílčích oblastí sledovat jejich horizontální vzdálenost, čímž je možno zredukovat počet samostatných znaků. Výsledek po této filtraci však stále ještě obsahuje příliš mnoho chybných oblastí. Následují tedy další filtrace. Jednou z nich je test překrytí oblastí, co se týče  $x$ -ových souřadnic. Tento filtr není vhodný pro předlohy, které obsahují kurzívu. Poslední filtr funguje tak, že vypočte v obdélníku opsaném oblastí poměr mezi počtem bodů, které nebyly uznány jako možný text a počtem bodů, které jako možný text uznány byly. Pokud je jeho hodnota větší než hodnota přednastavená, není celá oblast uznána jako text.

## 7.6 REPREZENTACE VÝSLEDKU

Reprezentace výsledku byla navržena pro text, jehož řádky jsou téměř rovnoběžné s horním okrajem obrázku, jako výsledek lze považovat právě opsané obdélníky (souřadnice jejich vrcholů) daným oblastem. Tyto oblasti je ještě vhodné rozšířit směrem nahoru a doprava, čímž do oblasti opět vrátíme ztracenou diakritiku a interpunkci. Dále je možno uložit jejich obsah do samostatného souboru (nebo paměťové struktury), ze kterého může být oblast převzata systémem pro rozpoznávání znaků.

## 8. APLIKACE VT

Aplikace VT slouží jako uživatelské rozhraní pro ukázkou použití tříd, jejichž princip popisuje předchozí kapitola.

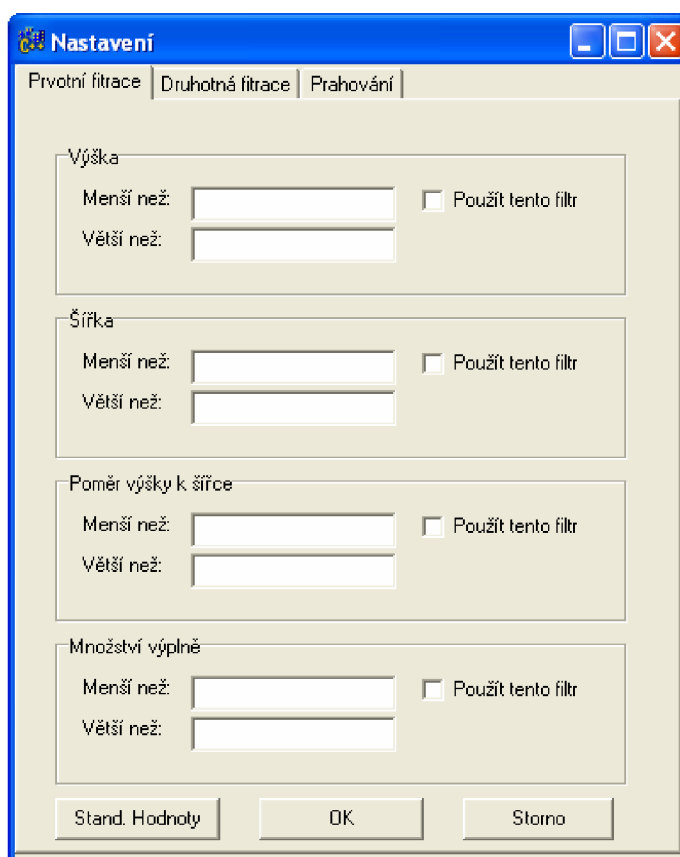


**Obrázek 8.1 Hlavní okno aplikace**

Aplikace se spouští pomocí souboru VT.exe. V adresáři, ve kterém se nachází tato aplikace, musí být také přítomny soubory z knihovny OpenCV. Po spuštění aplikace se otevře okno, jež je zobrazeno na obrázku Obrázek 8.1. Přes jeho roletové menu může uživatel provést otevření obrazové předlohy v některém z formátů JPEG, BMP, ICO, EMF a WMF. Dále umožňuje uživateli provést nastavení všech výše popsaných filtrů a místo zobrazení výsledku.

Formulář pro nastavení filtrů se vyvolává přes volbu Vyhledávání → Nastavení převodu. Formulář pro nastavení filtrů je znázorněn na obrázku Obrázek 8.2. Formulář obsahuje záložky Prahování, Prvotní filtrace a Druhotná filtrace. Záložka Prahování slouží k nastavení okolí pro prahování a nastavení charakteru

předlohy (světlejší text, tmavší text). Záložka Prvotní filtrace obsahuje nastavení všech filtrů (viz. kap. 7.3) včetně jejich povolení resp. zakázání. Záložka Druhotná filtrace obsahuje také všechny možnosti nastavení filtrů druhotné filtrace (viz. kap. 7.5) včetně jejich povolení resp. zakázání. Formulář obsahuje také tlačítko Stand.Hodnoty, které vrátí do všech nastavovacích polí experimentálně zjištěné optimální hodnoty (pro testované předlohy).



**Obrázek 8.2 Formulář pro nastavení aplikace**

Formulář pro nastavení filtrů se vyvolává přes volbu Vyhledávání → Zobrazení nabízí uživateli volbu, zda chce zobrazit výsledek na šedotónovou předlohu nebo na vyprahovanou předlohu.

Po všech případných nastaveních je možno spustit proces vyhledávání stiskem tlačítka Spustit vyhledávání. Po spuštění vyhledávání se zobrazí formulář, který informuje uživatele o průběhu vyhledávání. Toto okno se zavře po dokončení vyhledávání.



## 9. ZÁVĚR

Tato práce řeší úlohu vyhledávání textu v obraze. Jejím cílem bylo navrhnout knihovnu, pomocí níž by bylo možné provést vyhledání textu v obraze a možnost dalšího zpracování.

Práce vychází z teoretického popisu diskrétní obrazové funkce a pojmů s ní spojených (histogram, dvourozměrná diskrétní konvoluce, ...) a popisuje fundamentální operace s obrazem, využitelné při řešení dané problematiky. V části věnované textu jsou obecně definovány pojmy spojené s textem, důkladně jsou rozebrány vlastnosti znaků, slov a řádků, které tvoří základ každého textu. a jsou zde obsaženy také definice kategorií, díky kterým lze zvýšit přesnost a spolehlivost převodu. Poslední část je věnována samotnému programovému řešení zvolené problematiky a také důkladnému popisu použitých metod.

Řešení problematiky vyhledávání textu je úlohou značně složitou pro její obecnost. Jako jedna z neúčinnějších metod filtrace je filtrace na základě toho, zda se dílčí oblast (znak) podařilo sloučit s dalším znakem. Navržené řešení se snaží získat z obrázku většinu textu, chybu však mohou vytvořit jednotlivé osamocené znaky, které jsou příliš vzdáleny od ostatních. Největším problémem se ukázalo, jak je „strojově“ velmi složité (podle čeho) rozhodnout, zda se v dané oblasti nachází text či nikoli. Na rychlost zpracování má zase velký vliv obsah šumu v předloze.

Samotné jádro výsledné knihovny pro vyhledávání textu bylo vytvořeno pomocí čtyř tříd (shluk, subshluk, obr a hist). Ne všechny realizované filtry jsou samozřejmě ideální a v praxi může výjimečně dojít i k označení oblasti, která textem není, snahou však bylo, aby počet těchto oblastí byl co možná nejmenší.

Práce je doplněna doprovodným CD, které obsahuje kromě jádra knihovny také referenční manuál k realizovaným třídám a množství obrazových předloh, které byly vybrány tak, aby pokryly v co největší míře všechny zmiňované kategorie textu v obraze.

## 10. LITERATURA

- [1] Beneš, Žára, Sochor, Felkel: *Moderní počítačová grafika*, Computer Press, Brno, 2004, ISBN 80-251-0454-0
- [2] Genčúr: *Nástroj na zpracování fotografovaného textu*, bakalářská práce, Brno, Vysoké učení technické v Brně, Fakulta informačních technologií, 2007
- [3] Hlaváč, Šonka: *Počítačové vidění*, Grada, Praha 1992, ISBN 80-85424-67-3
- [4] Horák, Kalová: *Počítačové vidění*, Brno, Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií, 2006
- [5] Mařík, Štěpánková, Katanský: *Umělá inteligence. 2*, Academia, Praha, 1997, ISBN 80-200-0504-8
- [6] Rektoris: *Přehled užité matematiky, 3. vydání*, SNTL, Praha, 1973
- [7] Smrž, Hrbáček, Martinásek: *Off-line Recognition of Cursive Handwritten Czech Text*, Brno, Masarykova univerzita, Fakulta informatiky, 1998, dostupné na <http://www.fi.muni.cz/reports/files/older/FIMU-RS-98-02.pdf>
- [8] Špatněl, Beran: *Obrazové segmentační techniky – Přehled existujících metod*, Brno, Vysoké učení technické v Brně, Fakulta informačních technologií, 2005, dostupné na <http://www.fit.vutbr.cz/~spanel/segmentace/.cs.iso-8859-2>

