



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

# ROZPOZNÁVÁNÍ ZOUBKOVÁNÍ POŠTOVNÍCH ZNÁMEK

POST STAMP PERFORATION RECOGNITION

BAKALÁŘSKÁ PRÁCE  
BACHELOR'S THESIS

AUTOR PRÁCE  
AUTHOR

VEDOUCÍ PRÁCE  
SUPERVISOR

VLADIMÍR KONÍČEK

Ing. MICHAL ŠPANĚL,

BRNO 2008

Zadání Bakalářské práce a licenční smlouva jsou uvedeny v archivním výtisku uloženém v knihovně FIT VUT v Brně.

## **Abstrakt**

Zoubkování poštovních známek slouží k oddělování jednotlivých známek z archu. Existuje několik rozměrů a druhů zoubkování, podle nichž je možno poštovní známky nejen oceňovat, ale také je podle nich možno určovat pravost poštovních známek. Tato práce se zabývá návrhem a realizací jednoduchého detektoru zoubkování poštovních známek.

## **Klíčová slova**

Poštovní známka, zoubkování, detekce.

## **Abstract**

Post stamp perforation was used for dividing particular post stamps from sheet. There exists a few sizes and kinds of perforation, from which post stamps can be not only assessed, but also post stamp perforation is used for verifying authenticity of post stamps. This text is about suggestion and realisation of simple post stamp perforation detector.

## **Keywords**

Post stamp, perforation, detection.

## **Citace**

Vladimír Koníček: Rozpoznávání zoubkování poštovních známek, bakalářská práce, Brno, FIT VUT v Brně, 2008

# Rozpoznávání zoubkování poštovních známek

## Prohlášení

Prohlašuji, že jsem tuto bakalářskou práci vypracoval samostatně pod vedením Ing. Michala Španěla.

Uvedl jsem všechny literární prameny a publikace, ze kterých jsem čerpal.

.....  
Vladimír Koníček  
12. května 2008

© Vladimír Koníček, 2008.

*Tato práce vznikla jako školní dílo na Vysokém učení technickém v Brně, Fakultě informačních technologií. Práce je chráněna autorským zákonem a její užití bez udělení oprávnění autorem je nezákonné, s výjimkou zákonem definovaných případů.*

# Obsah

<b>1</b>	<b>Úvod</b>	<b>2</b>
<b>2</b>	<b>Zpracování obrazu</b>	<b>4</b>
2.1	Obraz . . . . .	4
2.2	Detekce hran . . . . .	4
2.3	Bresenhamův algoritmus pro kresbu kružnice . . . . .	8
2.4	Semínkové vyplňování . . . . .	10
<b>3</b>	<b>Druhy zoubkování poštovních známek</b>	<b>12</b>
3.1	Řádkové zoubkování . . . . .	12
3.2	Hřebenové zoubkování . . . . .	13
3.3	Rámcové zoubkování . . . . .	15
<b>4</b>	<b>Návrh detektoru zoubkování známek</b>	<b>16</b>
4.1	Odstranění nežádoucího šumu . . . . .	16
4.2	Detekce středu perforací . . . . .	16
4.3	Rozlišení středů perforací . . . . .	16
4.4	Konečná klasifikace zoubkování . . . . .	17
<b>5</b>	<b>Implementace</b>	<b>18</b>
5.1	Odstranění nežádoucího šumu . . . . .	18
5.2	Detekce středu perforací . . . . .	19
5.3	Rozlišení středů perforací . . . . .	20
5.4	Konečná klasifikace zoubkování . . . . .	21
<b>6</b>	<b>Výsledky</b>	<b>22</b>
<b>7</b>	<b>Závěr</b>	<b>26</b>
	<b>Literatura</b>	<b>27</b>
	<b>Seznam použitých zkratk a symbolů</b>	<b>28</b>

# Kapitola 1

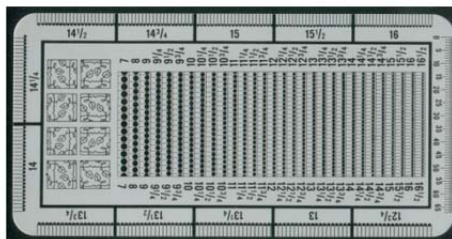
## Úvod

Zoubkování poštovní známky je označení pro průměrný počet perforačních otvorů, který připadá na smluvně určenou délku perforace 2 cm. Zoubkování poštovních známek slouží nejen k jejich oceňování, ale také například k určování jejich pravosti. Pokud je u zoubkování uvedeno pouze jedno číslo, platí toto zoubkování pro všechny čtyři strany poštovní známky. U smíšeného zoubkování, kdy mají vodorovné strany poštovní známky jiné zoubkování, než svislé strany, udává první číslo průměrný počet otvorů na vodorovných stranách a druhé na svislých stranách známky. Pokud je rozměr zoubkování udáván třemi nebo čtyřmi čísly, provádí se měření ve směru postupu hodinových ručiček počínaje horní vodorovnou řadou v přirozené poloze známky.

V současné době je zoubkování poštovních známek možno určit několika způsoby:

- zoubkoměrem
- elektronickými přístroji

Zoubkoměr je většinou kovová destička (obr. 1.1), na které je předtištěno několik druhů zoubkování poštovních známek. Určování zoubkování tímto způsobem se provádí pro každou stranu poštovní známky zvlášť a probíhá postupným přikládáním stran poštovní známky k jednotlivým druhům zoubkování. Pokud se zoubkování poštovní známky shoduje s některým z předtištěných zoubkování, zkoumaná strana poštovní známky má právě toto zoubkování. Jak si můžeme domyslet, je tento způsob určování zoubkování poštovních známek nejen docela pomalý, protože se provádí pro každou stranu poštovní známky zvlášť, ale také při tomto způsobu určování záleží na subjektivním pocitu člověka, který tuto kontrolu provádí. Toto je způsobeno hlavně kvalitou zoubkování, kdy u starších známek mohou být některé zoubky poškozeny, případně mohou zcela chybět.



Obrázek 1.1: Zoubkoměr

Dalším způsobem určování poštovních známek jsou elektronické přístroje. Tyto přístroje (například na obr. 1.2) již určují zoubkování poštovních známek elektronicky, takže jsou o něco rychlejší, než předchozí způsob pomocí zoubkoměru. Ale i při tomto způsobu určování zoubkování se jednotlivá měření provádí pro každou stranu známky zvlášť, což měření zpomaluje.



Obrázek 1.2: Perfotronic - přístroj pro měření zoubkování poštovních známek, obrázek převzat z [www.infofila.cz](http://www.infofila.cz)

Protože současné určování zoubkování poštovních známek pomocí zoubkoměru je docela pomalé, rozhodl jsem se pokusit se vytvořit program, který by z obrázku poštovní známky zjistil její zoubkování na všech stranách najednou.

## Kapitola 2

# Zpracování obrazu

### 2.1 Obraz

Ve 2D počítačové grafice, popsané např. v publikaci [3] se můžeme setkat se dvěma základními přístupy - s vektorovou a s rastrovou grafikou. Vektorová grafika ukládá přesná geometrická data, topologii a styl, například souřadnice bodů, propojení jednotlivých bodů, jejich vybarvení, vyplnění a tloušťku. Vektorová grafika také umožňuje obraz reprezentovat nejen pomocí jednotlivých bodů, ale také třeba pomocí kružnic, čtverců a jiných geometrických útvarů. Aby ale bylo většinou možno vektorový obraz v počítači zobrazit, je nutné jej většinou převést na rastrový obraz. Rastrová grafika je pravidelná síť pixelů, která je organizována jako dvourozměrná matice bodů. Každý pixel v rastrové grafice obsahuje specifické informace, například o jasu, barvě, nebo průhlednosti bodu. Obraz v rastrové grafice má omezené rozlišení, které se udává počtem řádek a sloupců.

### 2.2 Detekce hran

Lidské vnímání je založeno na rozpoznávání hran. Jednou z možností, jak zvýraznit nějaký obraz, abychom ho vnímali jako ostřejší, je zvýraznit v něm hrany.

Hranu (edge) v diskrétním obraze vnímáme tam, kde dochází k výrazné změně sousedních pixelů. Hrana je vysokofrekvenční informace, a proto je její zvýraznění inverzní operací k odstranění šumu. Hrana je určena *gradientem*, tj. velikostí a směrem. Směr lze popsat vektorovým operátorem nábla  $\nabla$

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

a velikost gradientu je tedy určena jako délka vektoru:

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

Výše uvedené funkce platí pro spojitě funkce. V diskrétním obraze gradient odhadujeme.

Ostření obrazu je pak založeno na následujícím postupu. Označíme si  $s(i, j)$  jako funkci, která reprezentuje velikost gradientu obrazu  $f$  v bodě  $[i, j]$ . Výsledný obraz  $g(i, j)$  získáme z obrazu  $f(i, j)$  ostřením pomocí koeficientu  $c$ .

$$g(i, j) = f(i, j) + c \cdot s(i, j)$$



Funkce  $s(i,j)$  vrací velikost gradientu a o její patřičný násobek se zvýší intenzita pixelu v odpovídajícím bodě. Pro určení gradientu se používají postupy založené na analýze okolí pixelu s použitím konvolučních nebo jiných operátorů.

## Robertsův operátor

Pravděpodobně nejjednodušší metodou určení velikosti gradientu pixelu je použití takzvaného *Robertsova operátoru*. Tento operátor není založen na konvoluci a jeho implementace je snadná. Robertsův operátor používá k výpočtu pixel a tři sousední pixely tohoto pixelu a má tvar:

$$|\nabla f(i, j)| = |f(i, j) - f(i + 1, j + 1)| + |f(i, j + 1) - f(i + 1, j)|$$

Velikost gradientu touto metodou se tedy určí jako součet absolutních hodnot změn ve směru hlavní a vedlejší diagonály obrázku. Robertsův operátor se používá především pro detekci hran s sklonem  $45^\circ$ .

Robertsův operátor můžeme rozdělit na dvě složky, z nichž každá detekuje hrany v jednom ze dvou na sebe kolmých směrech.

## Sobelův operátor

Na podobném principu jako Robertsův operátor založen Sobelův operátor (obr. 2.1). Tento operátor je směrově orientovaný, aproximuje první derivaci a pracuje s okolními pixely právě zkoumaného pixelu podle použité dvojice komplementárních konvolučních masek. Vždy je složen z dvojice komplementárních konvolučních masek označených jako  $h$  a  $\bar{h}$ . Komplementární maska se získá z původní masky rotací o devadesát stupňů kolem středu. Protože se v dalším výpočtu vypočítá druhá mocnina nebo absolutní hodnota, je nepodstatné, zda masku otáčíme doleva či doprava.

Jako komplementární konvoluční masky můžeme použít například tyto matice:

$$h = \begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix}, \bar{h} = \begin{bmatrix} 1 & 2 & 1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{bmatrix},$$

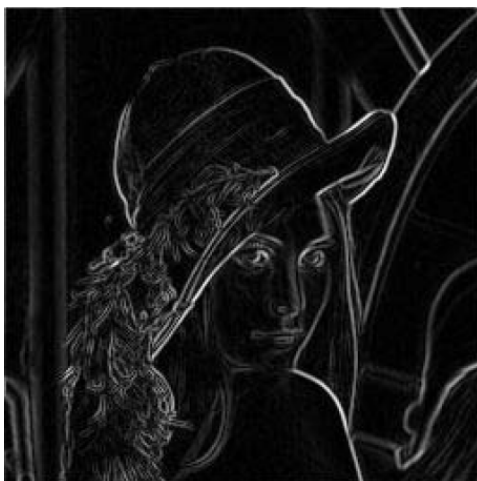
$$h = \begin{bmatrix} -2 & -1 & 0 \\ -1 & 0 & 1 \\ 0 & 1 & 2 \end{bmatrix}, \bar{h} = \begin{bmatrix} 0 & 1 & 2 \\ -1 & 0 & 1 \\ -2 & -1 & 0 \end{bmatrix}.$$

Absolutní velikost gradientu je potom získána dvojnásobnou aplikací konvoluce, nejprve pro  $h$  a poté pro  $\bar{h}$ , a součtem:

$$|G| = \sqrt{h^2 + \bar{h}^2}.$$

Součet je tedy možno zjednodušit na:

$$|G| = |h| + |\bar{h}|.$$



Obrázek 2.1: Lena – Sobelův operátor, převzato z [2]

### Laplaceův operátor

Další možnou metodou pro detekci hran je *Laplaceův operátor*. Výpočet pomocí tohoto operátoru je na rozdíl od předešlých případů založen na konvoluci. Laplaceův operátor se označuje jako  $\Delta$  a pro výpočet ze čtyřokolí pixelu ve směru kolmém na souřadnicové osy má jeho konvoluční jádro tvar

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

Varianta, která k výpočtu velikosti gradientu využívá hodnot z osmiokolí pixelu, má tvar

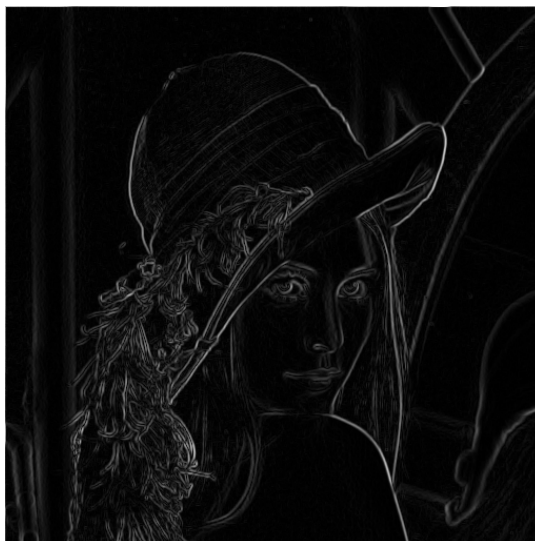
$$h = \begin{bmatrix} 1 & 1 & 1 \\ 1 & -8 & 1 \\ 1 & 1 & 1 \end{bmatrix}$$

Laplaceův operátor je invariantní k otáčení o násobky  $45^\circ$ .

### Cannyho hranový detektor

Další možností pro detekci hran v dvourozměrném obraze je Cannyho hranový detektor [5] (obr. 2.2). Cannyho hranový detektor je algoritmus, který v sobě zahrnuje několik kroků pro získání co nejlepšího výsledku při detekci hran v dvourozměrném diskrétním obraze. Postup detekce hran v obraze pomocí tohoto detektoru by se měl skládat z následujících částí:

1. eliminace šumu Gaussovým filtrem
2. určení gradientu(první derivace)
3. nalezení lokálních maxim
4. eliminace nevýznamných hran



Obrázek 2.2: Lena – Cannyho hranový detektor, vytvořeno použitím programu IrfanView

### Eliminace šumu Gaussovým filtrem

Dvourozměrná varianta Gausova normálního rozložení je dána vztahem

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

kde  $x, y$  jsou souřadnice pixelu v obraze s  $\sigma$  je standardní odchylka rozdělení (běžně  $\sigma = 1 - 1.4$ ).

Výpočet je vhodné realizovat pomocí konvoluce. Tímto vzorcem se vypočítá pouze konvoluční maska, která se pak aplikuje na celý obraz.

### Velikost a směr gradientu

V této části algoritmu je nejvhodnější použít Sobelův operátor 2.2. Sobelův operátor totiž vrací nejen velikost gradientu hrany, ale také její směr. Směr hrany totiž potřebujeme v dalších krocích.

### Nalezení lokálních maxim (thining)

Úkolem této funkce je vybrat z hodnot gradientů (stanovených v předchozím kroku) jen lokální maxima. Respektive odebrat body, které nejsou maximem. Tím zajistíme, že hrana bude detekována v místě největšího gradientu.

Toto znamená najít pixely, jejichž okolí je ve směru a proti směru gradientu nižší. Mámeli například pixel, jímž prochází svislá hrana, musí být jeho levý a pravý soused nižší hodnoty (hodnota jeho gradientu) aby byl určen jako skutečná hrana. Pokud podmínku nesplňuje, není označen z hranu. Které dva okolní pixely zahrnout do porovnávání je dáno směrem gradientu (určeno v předchozím kroku).

## Eliminace nevýznamných hran (prahování)

V předchozím kroku jsme určili kde přesně leží hrany, ale doposud jsme se nezabývali významem hran. V tuto chvíli jsou označeny i ty nejmenší hrany, protože i ty mají své lokální maximum. Není vhodné určit jeden práh nad kterým budeme gradient považovat za významný, protože hodnota může kolísat například vlivem šumu.

Zvolíme si tedy minimální (T1) a maximální (T2) hodnotu (prahy) mezi kterými může gradient kolísat. Pokud hodnota gradientu daného pixelu leží nad vyšším prahem T2 je přímo označen jako hranový. Pokud posuzujeme bod, jehož hodnota leží mezi T1 a T2 pak je jako hrana označen jedině pokud sousedí s bodem který už byl jako hrana označen dříve.

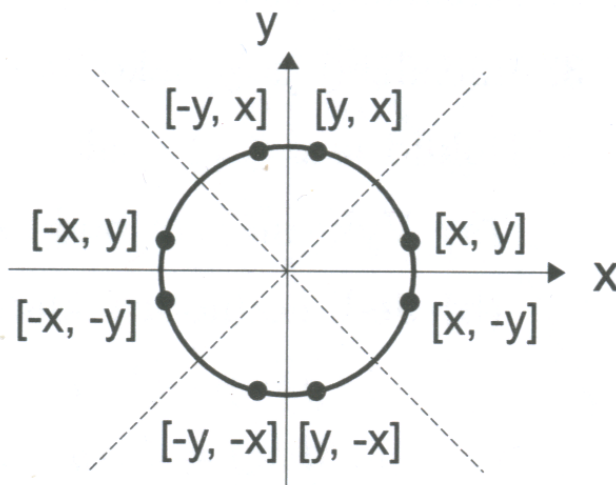
Cannyho hranový detektor je například pro detekci hran použit v programu IrfanView.

Operátory, které zvýrazňují hrany, zvýrazňují bez rozdílu všechny vysoké frekvence a tedy i šum. Operátory, které pracují s větší konvoluční maskou (s větším okolím pixelu), zvýrazňují šum méně.

Součet hodnot konvolučních masek pro detekci hran musí být roven nule. Toto zaručí, že v oblastech s konstantní hodnotou bude i odezva konvoluce nulová.

## 2.3 Bresenhamův algoritmus pro kresbu kružnice

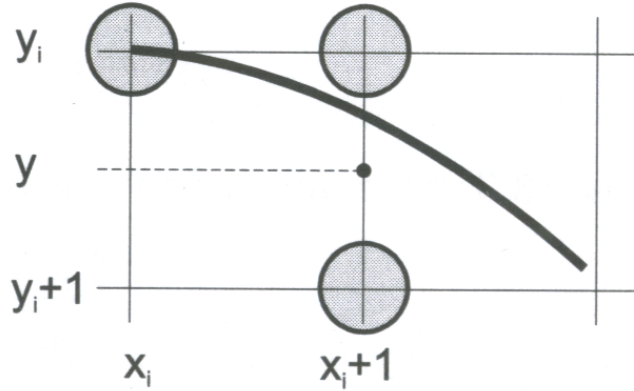
Též známý jako Midpoint algoritmus, popsáný např. v [3], je algoritmus pro kresbu kružnice, který při rasterizaci nachází body ležící nejbližše skutečné kružnici pouze pomocí celočíselné aritmetiky. V případě vykreslování kružnice je možno s výhodou využít skutečnosti, že kružnice je středově symetrická (obr. 2.3). Z jediného vypočítaného bodu kružnice lze tedy odvodit dalších sedm bodů pouhou záměnou souřadnic a změnou jejich znaménka. Pro vykreslení celé kružnice tedy stačí vypočítat hodnoty souřadnic bodů ležících v jednom oktantu, například v úseku od  $x = 0$  do  $x = y$ .



Obrázek 2.3: Symetrické body na kružnici, převzato z [3]

Pro popis postupu výpočtu jednotlivých bodů kružnice využijí obrázek 2.4, kde je zobrazena část rastru, který leží v jednom oktantu kružnice. V tomto oktantu se souřadnice  $x$

lišší sousedních bodů vykreslované kružnice liší právě o jeden pixel. Krok v ose  $x$  je tedy konstantní, vedlejší osou je osa  $y$ . Algoritmus začíná v bodě  $[0, r]$  a končí v průsečíku kružnice s hlavní diagonálou, kdy  $x = y$ .



Obrázek 2.4: Část kružnice v rastru, převzato z [3]

Pro body na kružnici platí implicitní rovnice  $x^2 + y^2 - r^2 = 0$ , kterou zapíšeme jako funkci:

$$F(x, y): x^2 + y^2 - r^2 = 0 \quad (2.1)$$

Znaménko funkce určuje polohu bodu  $[x, y]$  vůči kružnici. Funkční hodnota pro body uvnitř kružnice je záporná, pro body vně je kladná. Funkce  $F$  je proto vhodným kritériem při zavedení rozhodovacího členu. Předpokládejme, že bod  $[x_i, y_i]$  byl určen jako bod nejbližší skutečné kružnici. Následující bod může tedy mít buď souřadnice  $[x_i + 1, y_i]$ , nebo  $[x_i + 1, y_i - 1]$ . Na obrázku můžeme dále vidět naznačený bod ležící v polovině mezi dvěma uvedenými kandidáty (takzvaný midpoint). Dosadíme-li jeho souřadnice  $[x_i + 1, y_i - \frac{1}{2}]$  do funkce 2.1, znaménko výsledku určí, zda tento bod leží vně nebo uvnitř kružnice. Výslednou hodnotu budeme opět nazývat rozhodovacím členem  $p_i$ :

$$\begin{aligned} p_i &= F(x_i + 1, y_i - \frac{1}{2}) \\ &= (x_i + 1)^2 + (y_i - \frac{1}{2})^2 - r^2 \end{aligned} \quad (2.2)$$

Je-li znaménko  $p_i$  záporné, bude pro další kresbu vybrán bod se stejnou souřadnicí  $y_i$ , jinak bude vykreslen bod ležící o jeden pixel níže.

Hodnotu iteračního členu budeme určovat během iteračního výpočtu z předcházející hodnoty. Po úpravě tedy získáme výsledný vzorec:

$$p_{i+1} = p_i + 2x_i + 3 + (y_i - \frac{1}{2})^2 + (y_{i+1} - \frac{1}{2})^2 \quad (2.3)$$

Vzorec 2.3 můžeme tedy vyčíslit podle znaménka  $p_i$ :

1. Inicializuj pomocné proměnné:  $dvox = 3$ ,  $dvey = 2r - 2$
2. Inicializuj rozhodovací člen  $p$  na hodnotu  $1 - r$
3. Inicializuj  $[x, y]$  jako  $[0, r]$
4. Dokud je  $x \leq y$ , opakuj:
  - (a) Vykresli osm bodů symetrických s bodem  $[x, y]$
  - (b) Je-li hodnota  $p$  kladná, pak
    - i.  $p = p - dvey$
    - ii.  $dvey = dvey - 2$
    - iii.  $y = y - 1$
  - (c)  $p = p + dvox$
  - (d)  $dvox = dvox + 2$
  - (e)  $x = x + 1$

Algoritmus 2.1: Bresenhamův algoritmus pro kresbu kružnice

$p_i \leq 0$	$p_{i+1} = p_i + 2x_i + 3$
$p_i > 0$	$p_{i+1} = p_i + 2x_i + 5 - 2y_i$

I přesto, že vzorce pro aktualizaci rozhodovacího členu obsahují násobení, lze je implementovat jen pomocí sčítání a odečítání. Pro členy  $2x_i$  a  $2y_i$  zavedeme pomocné proměnné  $dvox$  a  $dvey$ , obsahující dvojnásobek příslušné souřadnice. Když se hodnota  $x$ , respektive  $y$  změní o jedničku, aktualizujeme příslušnou pomocnou proměnnou přičtením, respektive odečtením dvojky. Tyto proměnné inicializujeme tak, abychom se ve výše uvedené tabulce zbavili konstant tři a pět.

## 2.4 Semínkové vyplňování

Semínkové vyplňování je metoda určená pro vyplňování hranice nakreslené v rastru, popsána v [3]. Nezbytným parametrem pro semínkové vyplňování je souřadnice semínka – vybraného vnitřního bodu oblasti, od kterého se s vyplňováním začne. Vzhledem k tomu, že hranice vyplňované oblasti není geometricky jasně definována, informace o vyplňované oblasti je nutné získávat z obrazové (rastrové) paměti.

Při semínkovém vyplňování postupujeme od zadaného semínka a zkoumáme, zda jeho sousední body patří k vnitřní části oblasti. O příslušnosti bodu k oblasti rozhodujeme podle nějaké testované vlastnosti, například barevné intenzity. Existují dvě základní varianty:

1. Hraniční vyplňování - testovaný bod je vnitřní, má-li testovanou vlastnost odlišnou od vlastnosti hranice, například má-li jinou barvu.
2. Záplavové vyplňování - testovaný bod je vnitřní, má-li shodnou vlastnost jako zadané semínko. Tato metoda se také nazývá lavinové vyplňování či přebarvování.

1. UmístiSemínko  $(x, y)$
2. Pokud je bod  $[x, y]$  vnitřním bodem oblasti a dosud nebyl obarven, pak
  - (a) Obarvi bod  $[x, y]$  požadovanou barvou
  - (b) UmístiSemínko  $(x + 1, y)$
  - (c) UmístiSemínko  $(x - 1, y)$
  - (d) UmístiSemínko  $(x, y + 1)$
  - (e) UmístiSemínko  $(x, y - 1)$

Algoritmus 2.4: Semínkové vyplňování rekurzivním postupem

Nejjednodušší metoda semínkového vyplňování je popsána rekurzivním algoritmem 2.4. Semínkové vyplňování pro své šíření využívá buď čtyřokolí právě zpracovávaného bodu - další semínka generuje pouze ve vodorovném a svislém směru od právě zpracovávaného bodu - a nebo využívá osmiokolí bodu - další semínka generuje ve všech osmi směrech od právě zpracovávaného bodu. To, zda je vhodnější použít pro generování dalších semínek čtyřokolí a nebo osmiokolí bychom si měli rozmyslet podle toho, jak asi bude zpracovávaná oblast vypadat. Při použití osmiokolí bodu, totiž může dojít ke zpracování (přebarvení) bodů, které již ve zpracovávané oblasti nejsou.

## Kapitola 3

# Druhy zoubkování poštovních známek

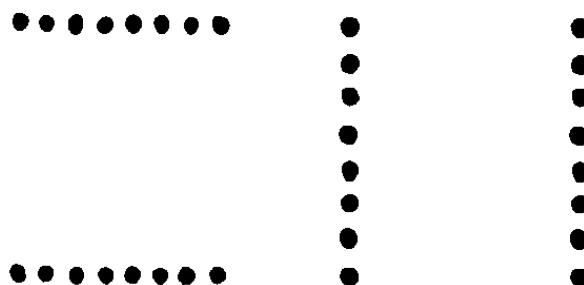
Zoubkování je způsob oddělování poštovních známek pomocí perforačních otvorů vyraženým do tiskového archu perforačním zařízením. Podle uspořádání jehel v perforačním zařízení rozlišujeme zoubkování

1. řádkové,
2. hřebenové,
3. rámcové

Kombinací těchto základních druhů pak může vzniknout zoubkování sdružené, kombinací rozměrů pak zoubkování smíšené.

### 3.1 Řádkové zoubkování

Název tohoto zoubkování je odvozen od obrazu, který vytvoří jehly po prvním úderu držáku perforačky do archu poštovních známek. K zoubkování stokusového archu známek (10 x 10 známek) ve vodorovném směru je zapotřebí jedenácti úderů. Po pootočení archu o 90° se provede zoubkování ve svislém směru (k poloze známky v archu).

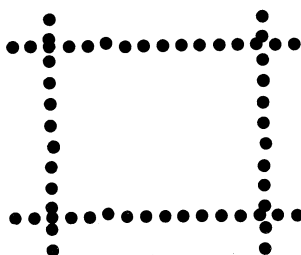


Obrázek 3.1: Rozložení jehel při tvorbě řádkového zoubkování



Pro řádkový způsob zoubkování je charakteristické

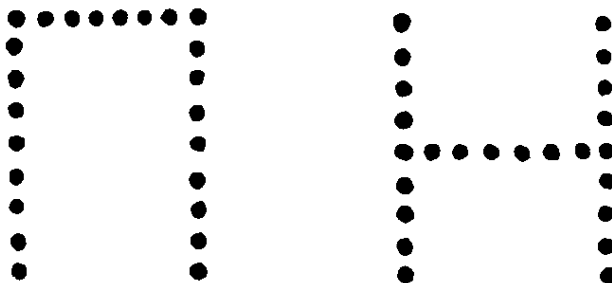
1. Při pohledu na celý arch poštovních známek můžeme vidět, že řady perforačních otvorů procházejí až do krajů, a to nahore, dole, na levé i na pravé straně archu.
2. Perforační otvory se na průniku svislých a vodorovných řad setkávají zcela nahodile, jak je vidět na obr. 3.2. Jen opravdu výjimečně se stává, že na průniku svislých a vodorovných os, tedy přesně ve stejném místě, došlo k průniku jehly při perforaci ve vodorovném i ve svislém směru. Ještě méně častá je možnost, že by tímto způsobem jehly pronikly archem ve dvou, třech nebo dokonce čtyřech rozích poštovní známky.



Obrázek 3.2: Znamka s řádkovým zoubkováním

### 3.2 Hřebenové zoubkování

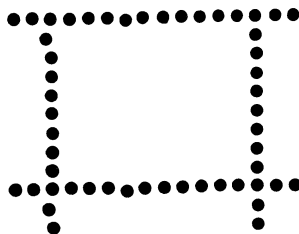
Hřebenové zoubkování bylo zavedeno k zrychlení práce při zoubkování poštovních známek. Název má opět podle obrazu, který při prvním úderu do archu vytváří soustava jehel zasažená do perforačního držáku. Prvním úderem jsou ozoubkovány známky na horním okraji první řady a dále je provedeno svislé zoubkování první řady známek. Spodní okraje první řady známek nejsou po prvním úderu ozoubkovány. K perforaci celého archu známek je zapotřebí jedenácti úderů. Při posledním úderu je využita pouze vodorovná řada jehel. Jehly ve svislém směru při posledním úderu prorazí pouze spodní okraj archu (bez známek). Na obr. 3.3 je vlevo standardní uspořádání jehel, vpravo tzv. H modifikace.



Obrázek 3.3: Rozložení jehel při tvorbě hřebenového zoubkování

Pro hřebenový způsob zoubkování je charakteristické:

1. Při pohledu na celý arch poštovních známek můžeme vidět, že řady otvorů procházejí až do kraje pouze na spodním kraji archu. Nikoli však nahoře a na levém a pravém okraji archu. Obvykle, ne však vždy, přesahují vodorovné řady o jeden otvor. Nahoře však nikoli, jestliže nejde o vodorovně postupující hřebenové zoubkování (ležmý hřeben).



Obrázek 3.4: Znamka s hřebenovým zoubkováním

2. Na průniku os svislých a vodorovných řad otvorů je centrován otvor. Ten také vymezuje charakteristický obraz rohu známky, viz. obr. 3.4. To platí u hřebenového způsobu zoubkování postupujícího shora dolů u levého i pravého horního rohu známky. U dolního levého a pravého rohu je otvor přesně na průniku svislých a vodorovných os otvorů jenom tehdy, pracuje-li perforační stroj dokonale. To se ale moc často nestává, většinou totiž perforační stroj pracuje nepřesně. Existuje devět možností (variant) při průniku jehel upevněných v držáku do archů poštovních známek. Pouze jedna z nich vzniká dokonalou prací perforačního stroje.
  - (a) Další úder dopadl proti předchozímu naprosto dokonale. V tomto případě je ve všech rozích známky perforační otvor vždy přesně na průniku svislých a vodorovných řad otvorů. To znamená, že obraz všech čtyř rohů známky je shodný, má pouze jinou polohu podle toho, o který roh jde
  - (b) Nastává v případě, když další úder perforačního držáku dopadl proti předchozímu úderu co do výšky přesně, ale vychýlil se nalevo.
  - (c) Je shodná s druhou možností, ale další úder se oproti předchozímu vychýlil vpravo.
  - (d) Další úder oproti předchozímu nedopadl co do potřebné výšky známky přesně, ale poněkud výš, a fakticky známku zkrátil. Úder se však směrově, to znamená vlevo ani vpravo, nevychýlil.
  - (e) Je shodná se čtvrtou možností, avšak další úder se oproti předchozímu vychýlil vlevo.
  - (f) Je také shodná se čtvrtou možností, avšak v tomto případě se úder oproti předchozímu vychýlil vpravo.
  - (g) Další úder perforačního držáku dopadl oproti předchozímu úderu níže, než bylo potřebné a tím došlo k prodloužení známky. Ve vodorovném směru (vlevo nebo vpravo) k vychýlení úderu nedošlo.
  - (h) Je shodná se sedmou možností, avšak další úder se oproti předchozímu vychýlil vlevo.

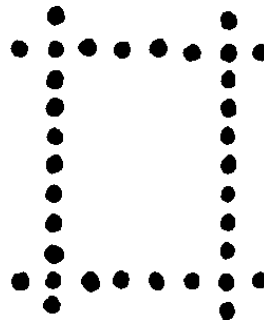
- (i) Je také shodná se sedmou možností, avšak další úder se oproti předchozímu vychýlil vpravo.

Pro možnosti 2b až 2i platí, že obraz levého i pravého horního rohu je stejný (má jen jinou polohu). Rohový otvor je tedy přesně na průniku svislých a vodorovných perforačních otvorů. Obrazy dolního levého a pravého rohu jsou u každé 2b až 2i možnosti zcela odlišné. žádná další varianta u svisle postupujícího hřebenového zoubkování neexistuje. Svislý postup u hřebenového zoubkování může jít shora dolů (jako v těchto uváděných příkladech), avšak arch mohl být vložen do perforačního stroje pootočený o  $180^\circ$ . V tomto případě se jedná o svisle postupující hřebenové zoubkování zdola nahoru.

Je však ještě nutno upozornit, že může nastat určitá modifikace popsaných zásad, a to v případě, pokud byla jehla, upevněná v průniku vodorovných a svislých řad jehel, vychýlena a vytváří tomuto vychýlení odpovídající obraz rohu známky.

### 3.3 Rámcové zoubkování

Druh zoubkování, při němž se perforačními otvory opatří celý arch najednou, jediným úderem perforačního zařízení. Použitím tohoto způsobu zoubkování lze dosáhnout pravidelně uspořádaných otvorů po celém obvodu všech známek, lepší kvality zoubkování a současně i zrychlení výroby známek. Používá se hlavně při zoubkování jednotlivých aršíků a tiskových archů s malým počtem známek.



Obrázek 3.5: Rozložení jehel při tvorbě rámcového zoubkování

O problematice popsané v této kapitole se můžete dozvědět více například v publikaci [4] nebo v [1].

## Kapitola 4

# Návrh detektoru zoubkování známek

Program, který jsem se pokusil vytvořit, načte obrázek známky ve formátu bmp a určí zoubkování poštovní známky na všech jejích stranách.

Ještě před zpracováním známky pomocí programu, je třeba obrázek upravit tak, aby na něm byly zvýrazněny hrany známky. Tuto operaci je možné provést nejlépe některým hranovým detektorem, viz. 2.2. Tyto hranové detektory jsou většinou obsaženy v programech, které slouží k prohlížení nebo ke zpracovávání obrázků. Můžeme použít například hranový detektor, který je obsažený v programu IrfanView (příkaz Image – Effects – Edge Detection). Dále je ještě nutné obrázku nastavit rozlišení, pokud toto již nastaveno není, ve kterém byl naskenován. V IrfanView je rozlišení obrázku možno nastavit pomocí příkazů Image – Information... – Resolution. Takto upravený obrázek již můžeme v programu použít.

### 4.1 Odstranění nežádoucího šumu

Protože známka většinou není na ideálním pozadí, hranový detektor nám v obrázku vytvoří šum. Protože by nám takto vzniklý šum při dalším zpracovávání obrázku vadil, je nutné se ho nějakým způsobem zbavit. V programu provádím vyhledání všech oblastí, které vznikly pomocí hranového detektoru a nežádoucí oblasti odstraňuji.

### 4.2 Detekce středů perforací

Po odstranění nežádoucího šumu v okolí známky můžeme tedy přistoupit k další operaci, kterou je detekce středů perforací na okrajích známky. Jelikož poštovní známky mají různé průměry perforačních otvorů, je nutné tuto operaci provést pro každý testovaný průměr zvlášť. Dále se také mohou objevit poštovní známky, které mají na různých stranách různé zoubkování. Tuto možnost jsem v programu také ošetřil.

### 4.3 Rozlišení středů perforací

Algoritmus pro detekci středů perforací použitý v předchozím kroku pracuje s celým obrázkem a středy perforací tak hledá i na vnitřní straně známky. Proto je v tomto kroku nutno rozlišit, zda nalezené středy patří k vnější (s těmi budeme dále pracovat) nebo k vnitřní

(s těmi již pracovat nebudeme) straně hrany známky. Dále v této části programu také rozlišuji, ke kterým stranám známky jednotlivé nalezené středy patří.

#### **4.4 Konečná klasifikace zoubkování**

Poslední věc, kterou v programu potřebujeme zjistit před konečným výpočtem zoubkování, je měřítko obrázku. Pokud předpokládáme, že byla známka naskenována a obrázek má nastaveno rozlišení, je jeho měřítko (počet pixelů na metr) uloženo ve struktuře, která obsahuje načtený obrázek. Počet pixelů získaný tímto způsobem je nutno přepočítat, aby jsme získali počet pixelů, který odpovídá skutečné vzdálenosti 2 cm.

## Kapitola 5

# Implementace

Program pracuje s obrázky získanými pomocí skeneru, nejlépe v rozlišení 300 DPI, s bitovou hloubkou 24 bitů na pixel.

Program jsem psal v programovacím jazyce C++, v programu Dev-cpp, pod operačním systémem Windows XP. Jako základní kostru programu jsem použil program pánů Igora Potůčka a Michala Seemana z druhého cvičení předmětu Základy počítačové grafiky z akademického roku 2006/2007.

### 5.1 Odstranění nežádoucího šumu

Pro odstranění nežádoucího šumu v obrázku používám následující algoritmus. Použitím hranového detektoru nám z původně barevného obrázku vznikl obrázek ve stupních šedi, případně pouze černobílý. Abych zajistil, že následující algoritmy vždy naleznou hledané hrany známky, použiji prahování obrázku, čímž ho převedu na černobílý, kde bíle jsou označeny hrany známky a případný šum a černě je označeno vše ostatní.

Postupně tedy procházím jednotlivé pixely obrázku a porovnávám, jestli jsem náhodou nenarazil na bílý pixel (hrana známky nebo šum). Pokud jsem na bílý pixel narazil, zapamatuji si souřadnice tohoto pixelu a uložím jej na zásobník. Souřadnice tohoto bodu poté využiji jako semínko pro semínkové vyplňování, které použiji tímto způsobem:

Pomocnou proměnnou, kterou používám pro uchování informace o rozloze zkoumané bílé oblasti nastavím na 0.

Modrou složku barevné informace právě zkoumaného pixelu nastavím na 0, aby nedocházelo k opětovnému zkoumání již jednou procházených pixelů.

Pomocnou proměnnou zvětším o 1. Na zásobník nagentuju čtyřkolí právě zpracovávaného bodu.

Ze zásobníku vyberu další bod a zjistím, zda se jedná o pixel bílé barvy, nebo zda má již nastavenou modrou složku barevné informace na 0.

Pokud se jedná o bílý pixel, zvětším pomocnou proměnnou o 1 a do zásobníku vygeneruji čtyřkolí právě zkoumaného bodu a pokračuji výběrem dalšího bodu ze zásobníku.

Pokud má již zkoumaný pixel nastavenou modrou složku barvy na 0, nagentuju žádné další body, pouze vyberu ze zásobníku další bod a opět ho testuji. Toto provádím, dokud není zásobník prázdný.

Poté porovnám zjištěnou velikost oblasti s referenční hodnotou. Pokud je zjištěná hodnota velikosti oblasti menší než referenční hodnota, narazil jsem na šum a je potřeba jej odstranit. To provedu následujícím způsobem:

Na zásobník opět uložím dříve zapamatovaný bod (počáteční semínko z předchozího algoritmu), ale tentokrát již nezkoumám velikost oblasti, nýbrž pouze nastavuji již prozkoumané pixely na barvu pozadí (černou). Poté na zásobník nagentruji čtyřokolí právě zpracovávaného bodu. Tento postup opakuji, dokud není opět zásobník prázdný.

## 5.2 Detekce středu perforací

Pro detekci středů perforací používám algoritmus, který využívá Bresenhamův algoritmus pro kresbu kružnice.

Porotože poštovní známky mohou mít různé průměry perforačních otvorů na různých stranách, je nejprve nutné pro každou stranu zvlášť zjistit, jaký průměr perforačních otvorů jí odpovídá.

Na počátku algoritmu vždy nastavím modrou složku barevné informace obrázku na 0. Tuto barevnou složku budu využívat k vyhledávání středů perforací v následujících krocích.

Obrázek procházím po jednotlivých pixelech. Pokud narazím na pixel, který označuje hranu známky, pomocí Bresenhamova algoritmu kolem tohoto pixelu vytvořím kružnici, jejíž poloměr odpovídá právě zkoumanému poloměru perforace. U pixelů, které Bresenhamův algoritmus určí jako kružnici, zvýším modrou složku barevné informace o 1.

Poté, co tímto způsobem projdu celý obrázek, pokračuji dalším krokem, ve kterém zjišťuji průměrnou intenzitu modré složky pixelu pro jednotlivé strany poštovní známky. Nyní je postup algoritmu rozdělen na procházení obrázku z každé strany. Všechny tyto části algoritmu jsou totožné, proto zde popíši postup algoritmu při hledání středů perforací na spodní straně známky.

Obrázek procházím z levého spodního rohu postupně po jednotlivých sloupcích směrem doprava a zjišťuji maximální intenzitu (nejvyšší složku modré barvy pixelu), kterou na této straně vytvořil Bresenhamův algoritmus. Pokud při procházení jednotlivých sloupců narazím na hranu známky, prohledávání aktuálního sloupce přeruším a přejdu na sloupec další. Takto nalezenou maximální intenzitu modré složky barvy si poznamenám.

Poté opět procházím obrázek z levého dolního rohu po jednotlivých sloupcích a kontrolovuji, zda není intenzita modré složky barvy vyšší, než polovina maximální nalezené intenzity. Pokud takovýto pixel naleznu, je pravděpodobné, že jsem našel střed perforace, ale v jeho okolí mohou být také další pixely s podobnou intenzitou a střed perforace je nutné určit co nejpřesněji.

K tomuto účelu opět využiji semínkové vyplňování. Při prohledávání oblasti pomocí semínkového vyplňování v tomto případě kontrolovuji, zda je intenzita modré barvy pixelu v rozmezí maximální intenzity až poloviny maximální intenzity. U pixelů, které jsem již algoritmem prošel, nastavím intenzitu modré složky na 0, aby nedocházelo k cyklování algoritmu.

Při procházení jednotlivých stran si zaznamenávám počet pixelů, u kterých se intenzita modré složky pixelu pohybuje v daném rozmezí a také počet oblastí, které jsem takto prošel.

Počítadlo oblastí zvýším vždy, když dojde k vyprázdnění zásobníku se semínky. Po průchodu obrázkem (při nalezení hrany známky opět přecházím na další sloupec) vydělím počet nalezených pixelů počtem oblastí, čímž zjistím průměrnou velikost oblasti, ve které se středy perforací při zkoumaném poloměru nachází.

Tuto verzi jsem zvolil z důvodu, že není zaručeno použití obrázku, na kterém je zadní strana známky, ale mohl být použit obrázek s přední stranou známky. Pokud by byl použit obrázek s přední stranou známky, mohlo by vlivem obrazu na známce dojít ke zkreslení středů perforací.

Toto zkreslení se použitím průměrné velikosti jednotlivých oblastí částečně eliminuje. Zjištěné průměrné hodnoty pro jednotlivé strany poštovní známky a pro jednotlivé průměry perforací si zaznamenám do tabulek.

Nyní již tedy známe průměrné obsahy oblastí pro jednotlivé strany poštovní známky i pro jednotlivé poloměry perforací. Je tedy nutné určit, který poloměr perforace nejlépe odpovídá které straně známky. Z tabulek, které jsme si v předchozí části algoritmu vytvořili pro jednotlivé strany známky, zjistíme, kterému poloměru odpovídá nejmenší číslo. Čím více totiž některý poloměr svou velikostí odpovídá hledanému poloměru perforace, tím je jeho průměrná velikost oblasti menší. Tímto způsobem tedy zjistíme, které straně nejvíce odpovídá který poloměr perforace.

Poté vždy postupně pro jednotlivé strany nageneryjeme pomocí Bresenhamova algoritmu pro kresbu kružnic opět intenzitu a zjistíme maximální intenzitu, kterou se nám podařilo vytvořit. Poté postupně procházíme zkoumanou stranu známky.

Pokud narazíme na intenzitu v daném rozmezí, použijeme opět semínkové vyplňování tímto způsobem:

Intenzitu modré složky barvy pixelu a souřadnice pixelu si zaznamenáme a pokud při procházení semínkovým vyplňováním v této nalezené oblasti nalezneme pixel, jehož intenzita je stejná, jako již nalezená, pomocí průměru souřadnic si vypočítáme pravděpodobný střed perforace.

Pokud v oblasti narazíme na pixel s vyšší intenzitou, předešlou intenzitu nahradíme nalezenou a předešlý průměrný střed perforace přepíšeme souřadnicí právě nalezeného pixelu.

Vždy když dokončíme prohledávání jedné oblasti (když dojde k vyprázdnění zásobníku se semínky), pixel na souřadnicích vypočítaných během prohledávání oblasti označíme jako *střed perforace*.

Tímto způsobem vždy pro každou stranu známky nalezneme středy perforací o poloměrech, které jsme si v předešlém kroku určili.

### 5.3 Rozlišení středů perforací

Algoritmus se aplikuje na každou stranu obrázku zvlášť. Jako příklad si proto opět uvedeme rozlišování středů perforací pro spodní stranu známky.

V tomto případě postupuje algoritmus od levého spodního rohu obrázku po jednotlivých sloupcích směrem k pravé straně.

Tento algoritmus je také ošetřen pro případ, že bude známka mírně natočena.

Pokud algoritmus při procházení jednotlivých pixelů obrázku narazí na pixel označený jako *hrana známky*, prohledávání aktuálního sloupce ukončí a přejde na následující sloupec. Pokud algoritmus narazí na pixel označený jako *střed perforace*, může se stát několik věcí:

1. Pokud je střed nalezen jako první na procházené straně známky, je jeho poloha pouze zaznamenána, střed je označen jako *nalezený* a algoritmus pokračuje dále.
2. Pokud je střed nalezen jako druhý v pořadí na procházené straně známky, jsou zde 2 možnosti, jak může algoritmus postupovat:
  - Střed je v rozmezí poloměru kružnice, která byla použita pro vytvoření středů perforací. Pro středy hledané na horizontálních stranách se kontroluje jejich vertikální souřadnice, pro středy na vertikálních stranách se kontroluje jejich horizontální souřadnice. V tomto případě se střed zaznamená jako *nalezený* a algoritmus pokračuje dále.



- Střed není v rozmezí poloměru kružnice, která byla použita pro vytvoření středů perforací. V tomto případě se předchozí nalezený střed označí opět jako *střed perforace*, protože s největší pravděpodobností patří k jiné, než právě zkoumané straně známky. Právě kontrolovaný střed se nastaví jako *nalezený* a počet nalezených středů se nastaví na 1.
3. Pokud se další nalezený střed nachází příliš blízko předchozímu nalezenému středu (u horní a spodní strany se kontroluje horizontální a u levé a pravé strany se kontroluje vertikální souřadnice), nalezený střed se nezapočítá, protože pravděpodobně není pravým středem perforace, ale může se jednat o zdvojení středů perforací. Možnost zdvojení středů perforací se vyskytuje nejvíce v rozích poštovní známky, hlavně proto, že algoritmus pro hledání středů perforací pracuje s každou stranou známky zvlášť.

V průběhu tohoto algoritmu si zaznamenáváme veškeré přiřazené středy perforací do tabulek odpovídajícím jednotlivým stranám známky. Tyto tabulky poté využijeme v následující části.

## 5.4 Konečná klasifikace zoubkování

Nyní již známe počet i souřadnice všech středů, které patří k jednotlivým stranám známky. Je tedy již možno určit zoubkování pro jednotlivé strany známky.

Nejprve však musíme zjistit, kolik pixelů vlastně odpovídá straně o délce 2 centimetry. To zjistíme ze struktury obsahující zpracovávaný obrázek za předpokladu, že má tento obrázek nastaveno rozlišení. Tato hodnota však není počet pixelů na 2 cm, ale počet pixelů na 1 metr délky. Proto je nutné tuto hodnotu přepočítat.

Jako další krok si pro jednotlivé strany známky určíme průměrnou vzdálenost mezi jednotlivými nalezenými středy. K tomuto účelu využijeme tabulky vytvořené v předchozí části. Postup zjišťování průměrné vzdálenosti je následující:

1. Zjistíme počet nalezených středů na jednotlivých stranách známky.
2. Zjišťujeme vzdálenost vždy mezi prvním středem a středem, který je první v druhé polovině tabulky, mezi druhým středem a středem, který je druhý v druhé polovině tabulky, . . . .
3. Zjištěné vzdálenosti poté zprůměrujeme a ještě vydělíme polovinou nalezených středů tak, abychom dostali průměrnou vzdálenost sousedních středů.
4. Zoubkování jednotlivých stran známek poté zjistíme, když vydělíme počet pixelů, který odpovídá dvěma centimetrům, vypočítanou průměrnou vzdáleností mezi nalezenými středy.

Rozměry zoubkování zjištěné pomocí výše popsaného postupu jsou poté postupně vypsány na standardní výstup.

## Kapitola 6

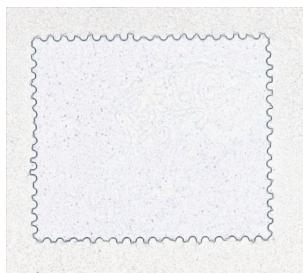
# Výsledky

V této kapitole uvedu několik příkladů funkčnosti programu v porovnání s klasickým určováním zoubkování poštovních známek pomocí zoubkoměru.

Obrazy známek, které jsem získal pomocí skeneru (např. 6.1), je před zpracováním pomocí programu nutné upravit některým hranovým detektorem. Obrázek po aplikaci hranového detektoru (6.2) poté již můžeme zpracovat pomocí programu. Výsledný obrázek, který vznikne po použití programu (6.3) obsahuje zvýrazněné jednotlivé nalezené středy perforací (světle modré pixely). V obrázku se v okolí skutečných detekovaných středů nachází také šedé pixely, které označují středy perforací se stejnou intenzitou modré složky barevné informace jako skutečně detekovaný střed. Důvod, proč byly za střed perforací určeny zrovna světle modré pixely a ne některý z ostatních šedých pixelů v jejich okolí je vysvětlen v kapitole 5.2.

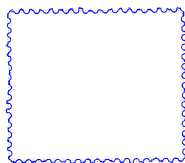


Obrázek 6.1: Znamka s malým úhlem natočení



Obrázek 6.2: Znamka po aplikaci Cannyho hranového detektoru

Úkolem programu je však kromě detekce jednotlivých středů perforací známky také určení zoubkování zobrazené známky na všech stranách. Pro porovnání výsledků zde uvedu



Obrázek 6.3: Výsledný obrázek, který vznikne po použití programu

nejen zoubkování zjištěné pomocí programu, ale také zoubkování zjištěné pomocí zoubkoměru. Pro obrázek 6.1 jsou zjištěné rozměry uvedené v tabulce 6.1.

strana známky	Zoubkoměr	program
dole	$11\frac{1}{2}$	11.48
nahoře	$11\frac{1}{2}$	11.44
vlevo	$11\frac{3}{4}$	11.82
vpravo	$11\frac{3}{4}$	11.81

Tabulka 6.1: Výsledky zjišťování zoubkování pro obrázek 6.1

V tomto případě, kdy hrany známky mají vodorovný a svislý směr, program dosahuje velice dobrých výsledků.

Na dalších příkladech si ukážeme, že program nemusí vždy pracovat tak, jak bychom to od něj čekali. Například pokud je známka na obrázku výrazně natočena (obr. 6.4)

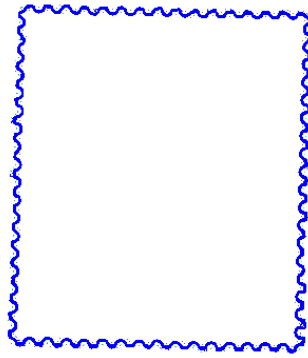


Obrázek 6.4: Znamka s větším úhlem natočení

Obrázek vzniklý po použití programu je v tomto případě 6.5. Zde si můžeme všimnout, že program při tomto natočení známky nevyhledává středy na horní straně známky, ale na této straně prohledávání ukončil při nalezení prvního středu, který k horní straně známky nepatřil.

Zoubkování vypočítané na zbylých stranách známky však opět odpovídá zoubkování zjištěnému pomocí zoubkoměru (viz. tabulka 6.2)

Pokud se nejedná o obrázek získaný pomocí skeneru, ale například o obrázek získaný pomocí fotoaparátu (obr. 6.6), může se stát, že hrana známky nebude dostatečně dobře



Obrázek 6.5: Výsledný obrázek, který vznikne po použití programu

strana známky	Zoubkoměr	program
dole	$13\frac{3}{4}$	13.76
nahoře	$13\frac{3}{4}$	undefined
vlevo	14	13.95
vpravo	14	13.89

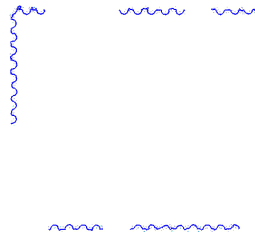
Tabulka 6.2: Výsledky zjišťování zoubkování pro obrázek 6.4

rozpoznána a některé její části mohou být proto odmazány algoritmem pro odstranění šumu (obr. 6.7). V tomto případě zobrazuje program výrazně rozdílné výsledky od výsledků získaných pomocí zoubkoměru. Při použití fotoaparátu dochází také ke zkreslení přímek, takže fotografie známek nejsou moc vhodné pro zjištění rozměrů zoubkování.



Obrázek 6.6: Znáмка se špatně zaostřenými hranami, převzato z internetové aukce filatelie Burda

Nejlepších výsledků v rozpoznávání zoubkování poštovních známek program dosahuje, pokud je na obrázku zadní strana známky, pokud je na obrázku přední strana známky, není u programu zaručeno, že středy perforací detekuje správně. Přední strana známky může obsahovat razítko pošty, které se tiskne přes okraj známky, čímž nám může detekci zoubkování znehodnotit.



Obrázek 6.7: Výsledný obrázek, který vznikne po použití programu

Pokud by bylo zoubkování známky nějak výrazně poškozeno, program nedokáže najít všechny středy perforací a tím určí jiné zoubkování, než známka ve skutečnosti má.

# Kapitola 7

## Závěr

Úkolem práce bylo navrhnout a vytvořit jednoduchý program, který by byl schopen z obrázku poštovní známky určit zoubkování. Maximální velikost obrázku, který je program schopen zpracovat je 800 x 700 pixelů. Z tohoto důvodu se jeví jako nejvhodnější použití obrázků s rozlišením 300 DPI. Pro dosažení obstojných výsledků zjišťování zoubkování poštovních známek je také vhodné použití obrázků, na nichž je známka co nejméně potočena. Pokud je známka na obrázku potočena o větší úhel, algoritmy použité v programu nemusejí vykazovat stejné výsledky, jaké bychom mohli získat při měření zoubkování poštovních známek pomocí zoubkoměru. Také v případě, že je zoubkování známky nějak výrazněji poškozeno, není programem zaručeno, že zoubkování zjištěné pomocí programu bude odpovídat skutečnosti.

Tento program by bylo možno dále rozšiřovat v rámci diplomové práce. Bylo by možno například implementovat grafické uživatelské rozhraní aplikace. Dále by bylo možno algoritmus pro detekci hran implementovat přímo v aplikaci, případně implementovat algoritmus, který by zajišťoval doplnění středů perforací na jednotlivých stranách známky, pokud by zoubkování známky bylo výrazněji poškozeno.

# Literatura

- [1] Filatelistická encyklopedie. online, [cit. 2008-05-08].  
URL [http://www.informace-scf.cz/index.php?option=com\\_content&task=category&sectionid=15&id=33&Itemid=47](http://www.informace-scf.cz/index.php?option=com_content&task=category&sectionid=15&id=33&Itemid=47)
- [2] HÝNA, P.: *Detekce rohů v obraze*. bakalářská práce, FIT VUT v Brně, Brno, 2007.
- [3] ŽÁRA, J.; aj.: *Moderní počítačová grafika*. Computer Press, 2004, ISBN 80-251-0454-0.
- [4] ŠRÁMEK, F.: *Filatelisté a filatelie*. Nakladatelství dopravy a spojů, 1988, oS-31-021-88-09-11.
- [5] Wikipedie: Cannyho hranový detektor — Wikipedie: Otevřená encyklopedie. online, 2008, rev. 18. ledna 2008, [cit. 2008-05-08].  
URL [http://cs.wikipedia.org/w/index.php?title=Cannyho\\_hranov%C3%BD\\_detektor&oldid=2166951](http://cs.wikipedia.org/w/index.php?title=Cannyho_hranov%C3%BD_detektor&oldid=2166951)

# Seznam použitých zkratek a symbolů

BMP – Microsoft Windows Bitmap

DPI – dots per inch