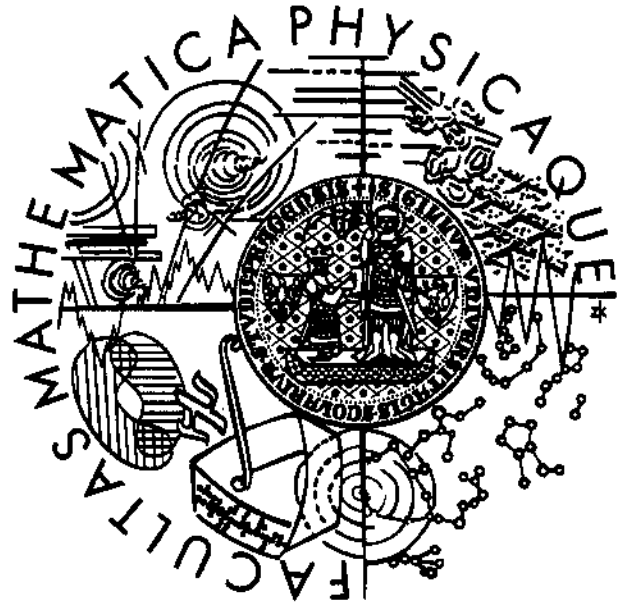
DIPLOMOVÁ PRÁCE



Helena Hajsová

Rekonstrukce liniových prvků v polohově deformovaných binárních obrazech

Katedra softwarového inženýrství

Vedoucí diplomové práce: Dr. Ing. Lubomír Soukup

Studijní program: Informatika

Děkuji Dr. Ing. Lubomíru Soukupovi za pomoc s matematickou stránkou této práce, prof. Ing. Janu Flusserovi, DrSc., za neocenitelné rady, které mě nasměrovaly k úspěšnému řešení, a Ing. Janě Zaoralové za trpělivé zodpovídání mých dotazů.

Prohlašuji, že jsem svou diplomovou práci napsala samostatně a výhradně s použitím citovaných pramenů. Souhlasím se zapůjčováním práce.

V Praze dne 2. dubna 2006

Helena Hajsová

Helena Hajsová

Název práce: Rekonstrukce liniových prvků v polohově deformovaných binárních obrazech
Autor: Helena Hajsová
Katedra: Katedra softwarového inženýrství
Vedoucí diplomové práce: Dr. Ing. Lubomír Soukup, Útia AV ČR
e-mail vedoucího: soukup@utia.cas.cz

Abstrakt: Práce se zabývá návrhem metody na automatické zpracování katastrálních map. Cílem je automaticky vyhledávat význačné body – rámové značky. Důraz je kladen na co nejvyšší úspěšnost na široké kvalitativní škále katastrálních map (včetně skupiny historických map) a rychlost zpracování. Součástí práce je aplikace pro zpracování všech druhů katastrálních map. Program nabízí různé úrovně zpracování včetně zcela automatického.

Klíčová slova: katastrální mapa, zpracování obrazu, obrazová korelace

Title: Reconstruction of line components in positionally deformed binary images
Author: Helena Hajsová
Department: Department of Software Engineering
Supervisor: Dr. Ing. Lubomír Soukup, Útia AV ČR
Supervisor's e-mail address: soukup@utia.cas.cz

Abstract: This thesis proposes a method for automatic processing of cadastral maps. The objective is to automatically search for specific points - frame marks. Two features are emphasized - the highest possible hit rate over a (quality-wise) wide range of maps (including a group of historical maps) and a high processing speed. The work includes application for the processing of all kinds of cadastral maps. The program offers different levels of processing, including a fully automatic one.

Keywords: cadastral map, image processing, matching by correlation

Kapitola 1 Úvod

Tato práce se zabývá řešením dílčího problému v rámci výzkumného záměru Českého úřadu zeměměřického a katastrálního číslo CUZ0002561501: „Výzkum a vývoj v geodezii, katastru a geomatice v letech 2005 – 2009“, v projektu „Vývoj nástroje pro obnovu katastrálního operátu v rámci digitalizace souboru geodetických informací ve vazbě na vývoj informačního systému katastru nemovitostí“. Při mnoha činnostech v rámci vedení a obnovy katastrálního operátu se pracuje s naskenovanými mapami.

Jedná se o velké množství mapových listů, z nichž většina byla před naskenováním skladována dlouhá léta v archivech. Díky tomu došlo k tvarovým deformacím papíru, které ovlivnily původní kresbu.

Proto je nezbytným počátečním krokem oprava těchto deformací. Klíčovou roli v ní hraje nalezení rámových značek. Poloha rámových značek na nezdeformované mapě je totiž známa, a tak je možné z porovnání aktuální a původní polohy vypočítat deformaci a následně ji odstranit. Rámové značky mají dále význam při zanesení mapy do správných souřadnic.

Doposud se vyhledávání značek provádělo ručně. Cílem této práce je zautomatizovat detekci rámových značek, což povede k výrazné úspoře času a zjednodušení práce operátorů, kteří rastrové mapy zpracovávají.

Kapitola 2 Základní pojmy

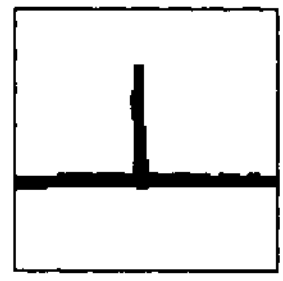
2.1 Katastrální mapa

V následujícím textu se bude vyskytovat slovo „strana“ ve více významech. Abych se vyhnula nejasným formulacím, zavedu pro stranu mapového rámu pojem „hrana rámu“.

Katastrální mapa se skládá z **rámu**, který má obdélníkový tvar. Po jeho obvodu jsou v pravidelných intervalech **rámové značky.** Rozteč těchto značek je jeden vídeňský palec (2,634 cm), a tedy délka každé hrany rámu je násobkem této hodnoty. Uvnitř rámu je kresba. V oblasti rámu se mohou kromě rámových značek vyskytovat i jiné kresby. Vně rámu mapy mohou být dodatečné informace.

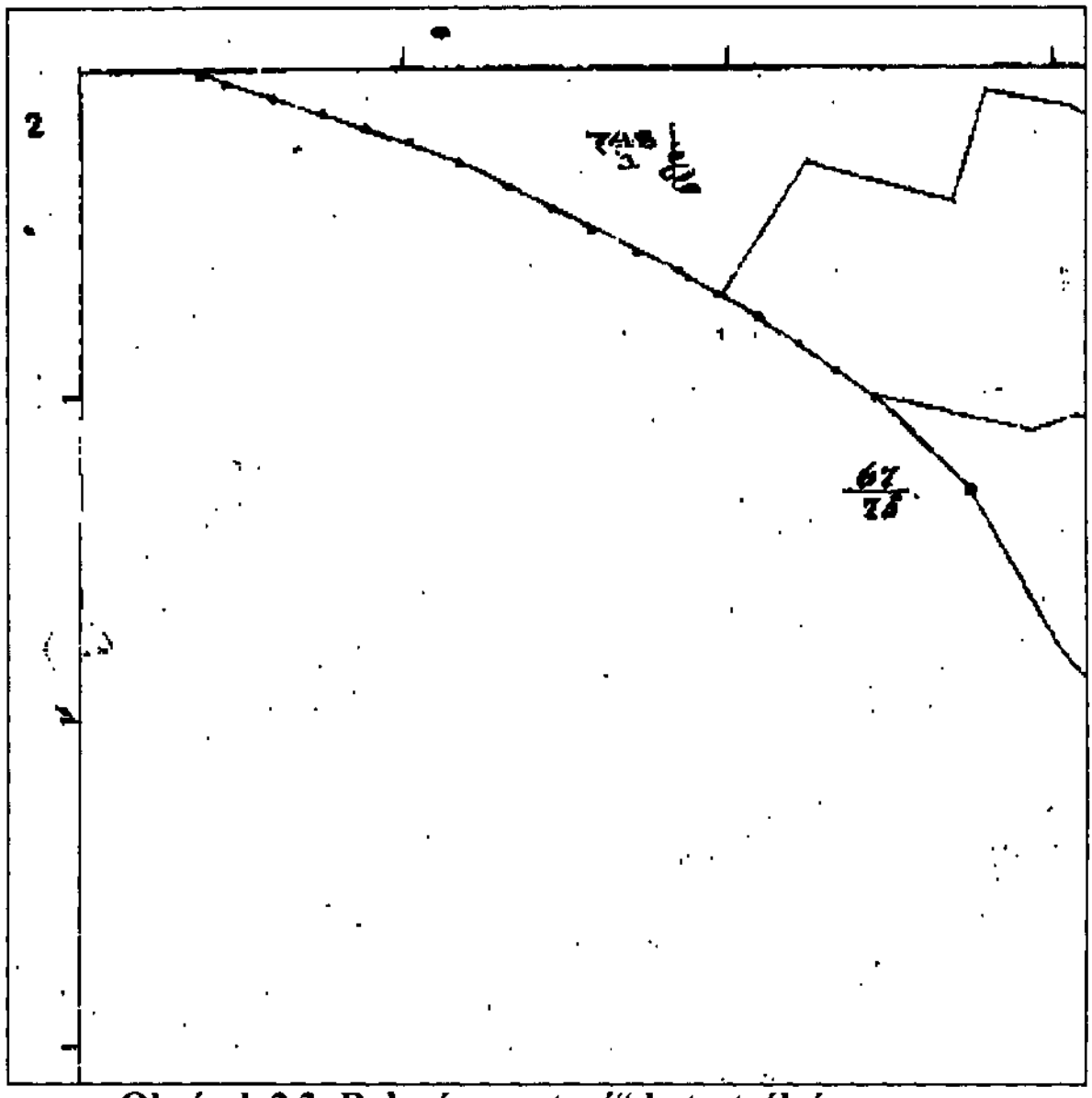
Toto popisuje stav nově narýsované mapy. Díky dlouhému a často nevhodnému skladování, ohýbání papíru a vlivu teplotních změn se mapy zdeformovaly. To způsobilo, že se hrany rámu změnily v křivky, které se pozvolna vlní kolem původní polohy hrany. Rozestupy mezi značkami se v důsledku toho zkrátily či prodloužily. Tvar pozemků se samozřejmě také zdeformoval, a neodpovídá tedy realitě nejen v průběhu jejich hranic, ale také ve výměrách. Proto je nezbytné provést detekci značek, na jejichž základě se zjistí parametry deformace a ta se následně opraví.

Katastrální mapy se dělí na dvě skupiny podle média, a to „nové“ a „staré“. Novější mapy byly skladovány podstatně kratší dobu a také jsou na odolnějším médiu (PET folie), proto je jejich kvalita lepší (méně šumu) a deformace menší.

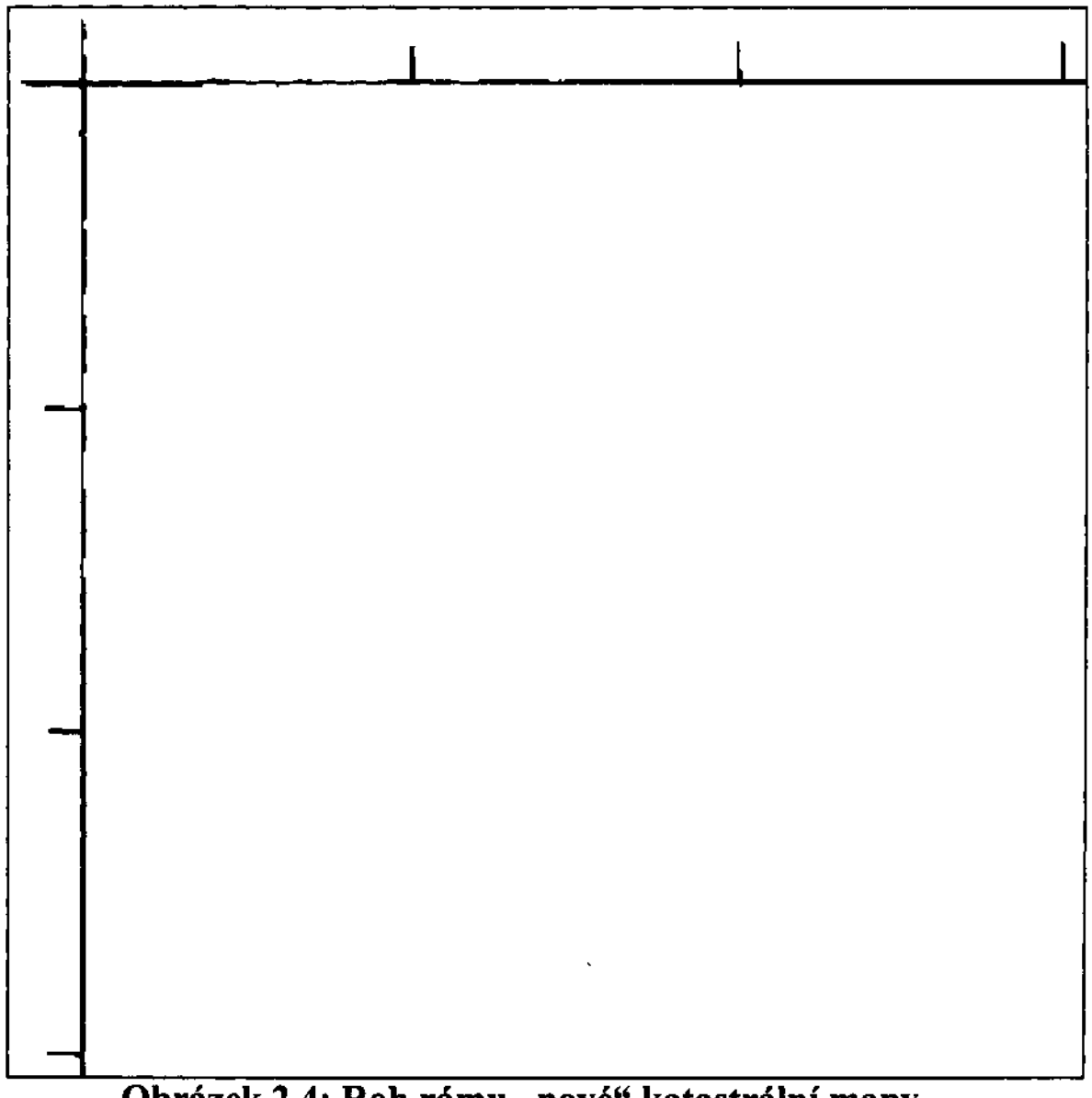


Obrázek 2.1: Rámová značka

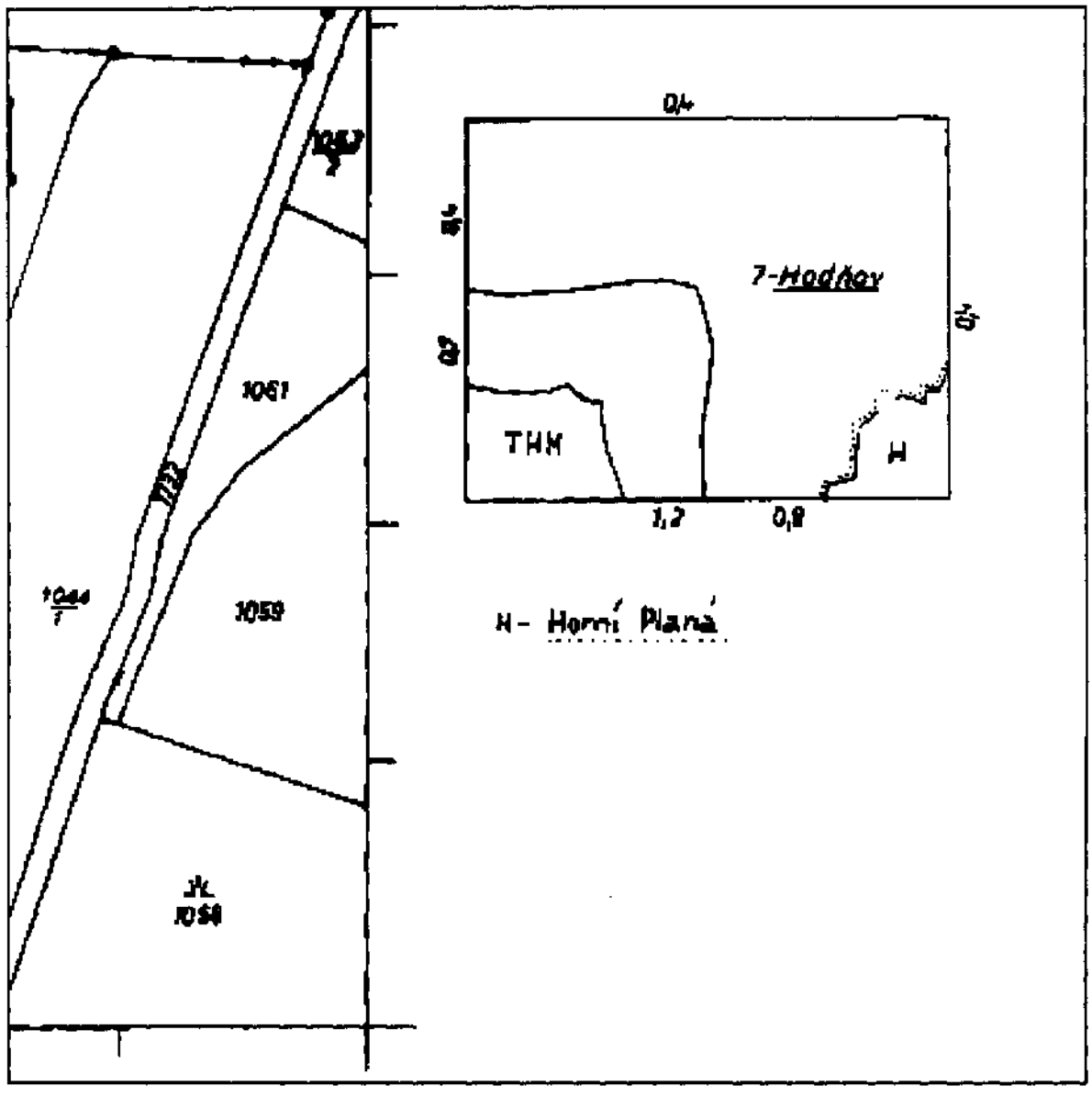
Obrázek 2.2: Ostatní kresby na rámu



Obrázek 2.3: Roh rámu „staré“ katastrální mapy



Obrázek 2.4: Roh rámu „nové“ katastrální mapy



Obrázek 2.5: Dodatečné informace na mapě

2.2 Mapový soubor

Pracuji se zdrojovými soubory v tom tvaru, v jakém byly dodány ze skenovacího pracoviště. Tento soubor obsahuje rastrový obraz mapy a tabulku s informacemi skenovacího pracoviště. Důležitým údajem obsaženým v tabulce je **dpi*** pořízené mapy. Další informace v tabulce se liší v závislosti na konkrétním pracovišti. Velikost strany souboru bývá v průměru 15 000 pixelů, může ale dosahovat až 25 000.

Skenovací pracoviště	V Ú G T K	Skener	EAGLE 2480	Název souboru	kldoks4t.cit
Datum skenování	3.11.1993	Hustota	500 dpi	Kategorie skenovaní	B
Datum a střední chyba transformace	24.11.1993 0.2098	Data jsou majetkem ČÚZK a mohou být šířena jen s jeho souhlasem .			

Obrázek 2.6: Tabulka skenovacích informací

* dpi = dots per inch; jednotka udávající počet pixelů elektronického obrázku připadající na jeden palec (2,54 cm) fyzického obrázku

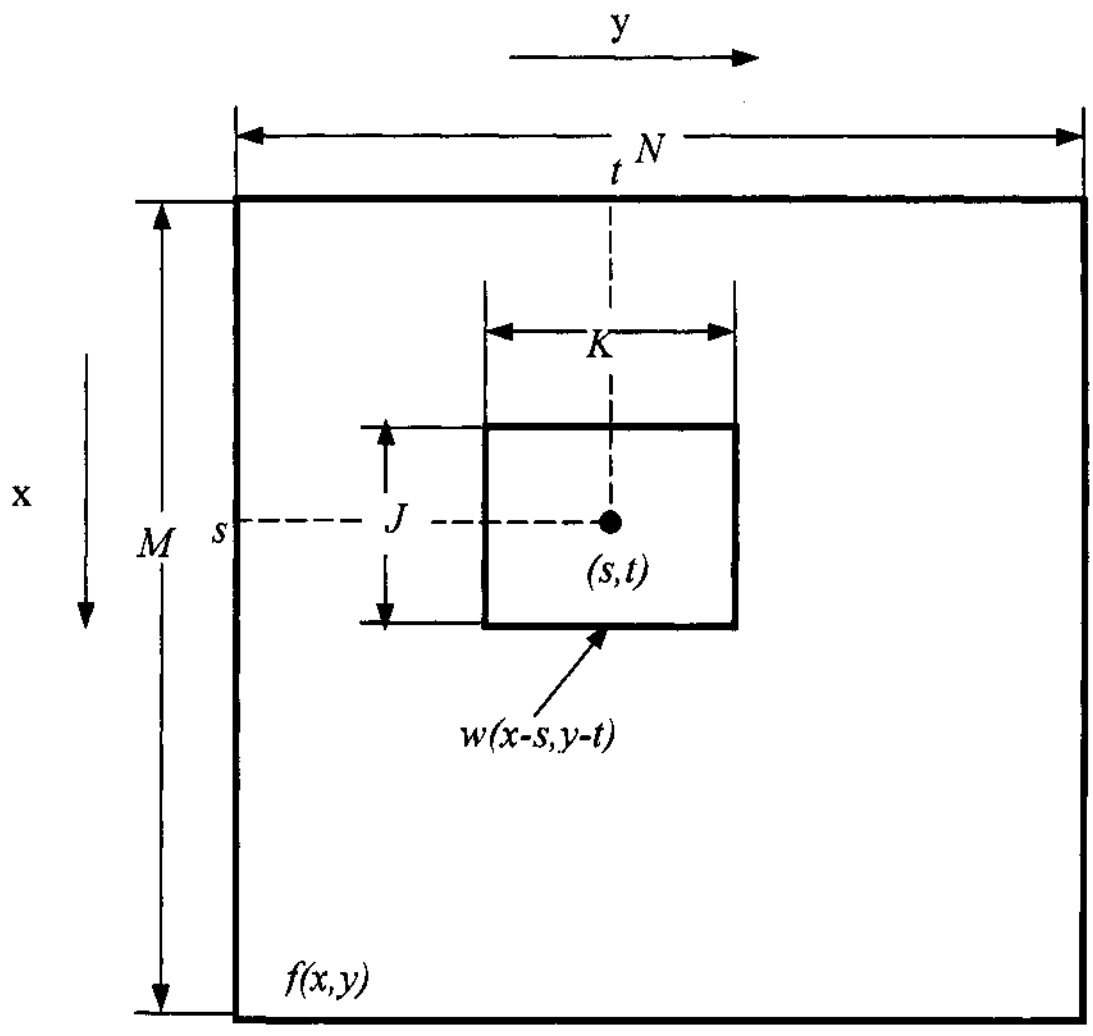
Kapitola 3 Vyhledávání rámových značek

3.1 Obrazová korelace

Jádrem řešení je metoda **vyhledávání pomocí korelace**. Tato metoda, popsána např. v [1] a [2], slouží k vyhledávání podobrázku (masky) *w(x,y)* velikosti $J \times K$ v obraze *f(x,y)* velikosti $M \times N$ za předpokladu, že $J \leq M$ a $K \leq N$. Korelace mezi podobrázkem a obrazem *w(x,y)* a *f(x,y)* je:

$$c(s,t) = \sum_x \sum_y f(x,y)w(x-s,y-t) \qquad (3.1)$$

kde $s = 0,1,2,\ldots,M-1$, $t = 0,1,2,\ldots,N-1$ a sčítá se přes oblast, kde se podobrázek a obraz překrývají.
Pro každou hodnoty *(s,t)* uvnitř *f(x,y)* vede (3.1) na jednu hodnotu *c*. Maximální hodnota udává pozici, kde se podobrázek nejlépe podobá obrazu.
Následující obrázek ilustruje výpočet za předpokladu, že je za počátek obrazu považován levý horní roh a za počátek masky střed (což je nejobvyklejší).



Obrázek 3.1: Schéma výpočtu korelace

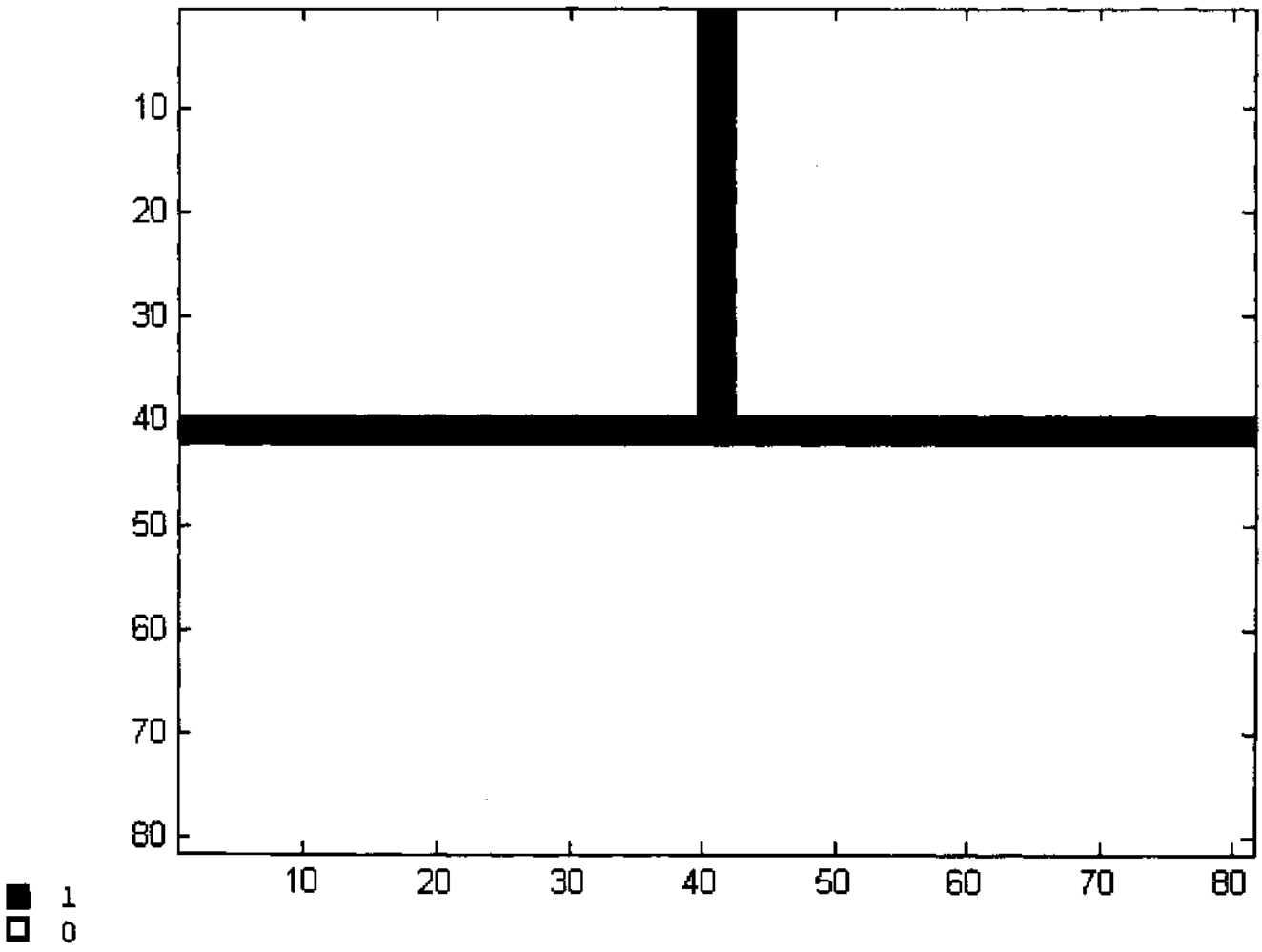
Časová náročnost obrazové korelace pro fixní velikost masky je θ(MN).

Zřejmou nevýhodou této metody je, že vydá výsledek i v případě, že podobrázek v prohledávaném obraze není.

Problém vyhledání rámové značky odpovídá problému, který řeší právě obrazová korelace. Tvar značky je znám a nemění se, a tak na něm může být založena maska pro obrazovou korelaci.

3.1.1 Korelační maska pro detekci značky

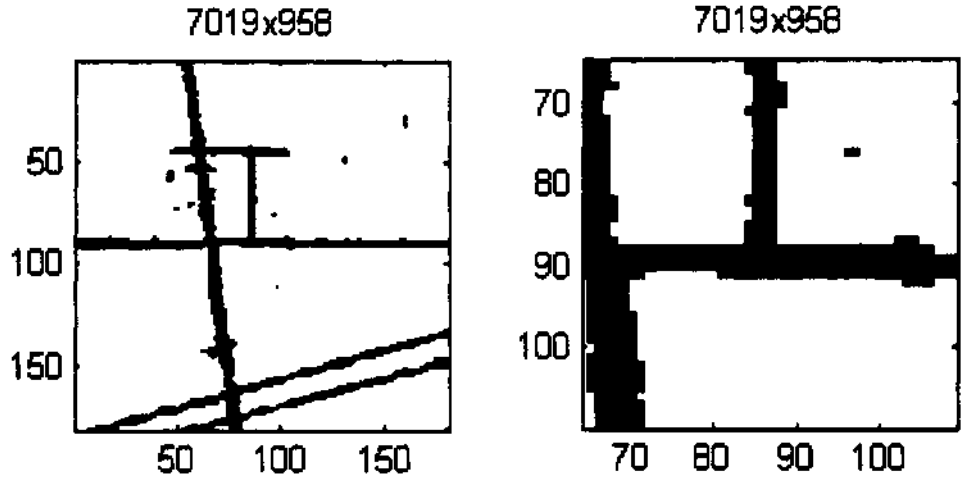
Maska pro výpočet korelace vychází z tvaru rámové značky. Ta připomíná tvarem písmeno T, ovšem délka výstupku se mění podle konkrétní mapy. Masku jsem zvolila čtvercového tvaru o rozměru 81x81 bodů. Stejně jako mapa je i maska binární, hodnoty „1" v oblasti T a „0" všude jinde. Tvar T je patrný z obrázku, průsečík se nachází ve středu masky. Šířka čáry je tři pixely. Ta byla zvolena podle šířky namalované čáry v rozlišení 400 dpi. Velikost masky byla zvolena experimentálně s ohledem na úspěšnost a časovou náročnost.



Obrázek 3.2: Korelační maska pro vyhledání rámové značky

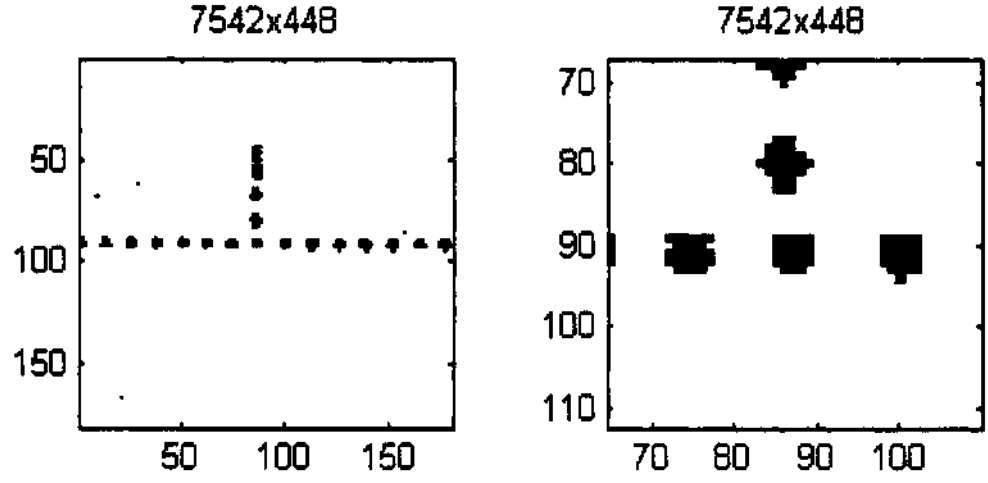
Maska byla navržena tak, aby byla co možná nejvíce odolná proti detekci jiných kreseb vyskytujících se na rámu. Nejčastější kresbou je silná šikmá čára škrtající rám mapy. Detekci této kresby namísto značky zabraňuje dostatečně dlouhý výběžek, díky kterému korelace reaguje jen na kolmé výstupky z rámu mapy.

Červená barva označuje body, které byly korelací označeny jako maximum.

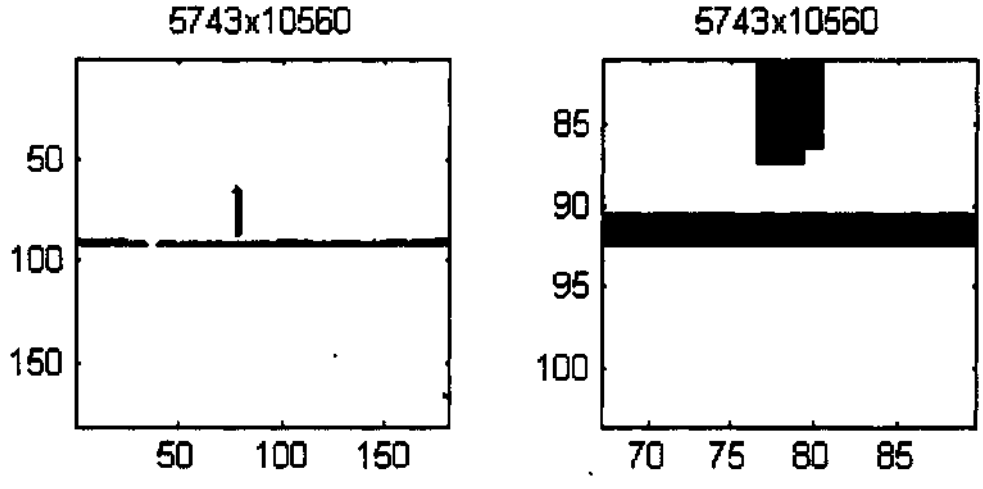


Obrázek 3.3: Správné nalezení značky v okolí výrazné šikmé kresby

Detekce je odolná i vůči poškození rámové značky – není třeba, aby rámová značka byla celistvá. Korelace velice dobře reaguje, i pokud značka jen připomíná tvar T.



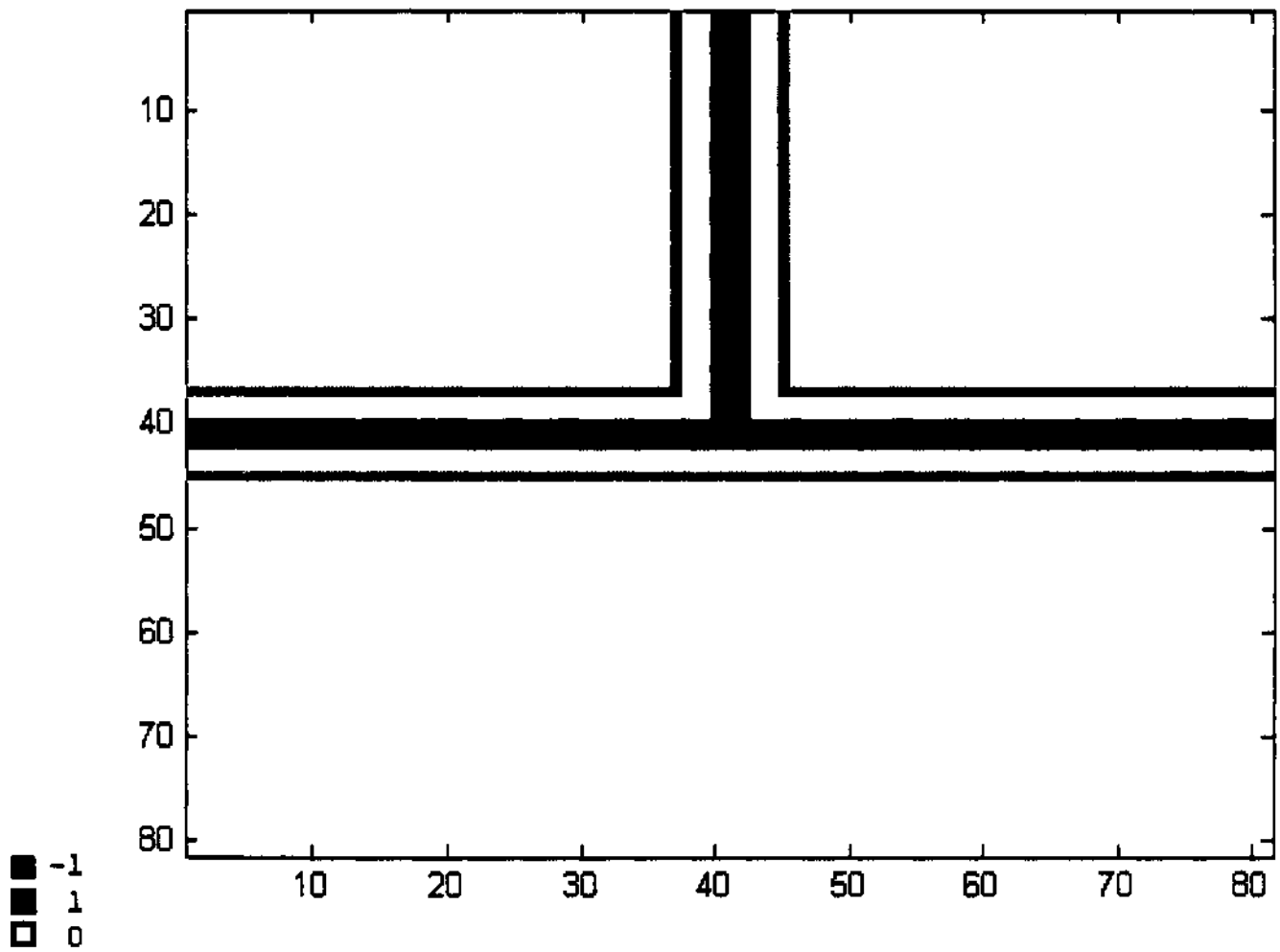
Obrázek 3.4: Správné nalezení značky na přerušované čáře



Obrázek 3.5: Odolnost vůči poškození značky

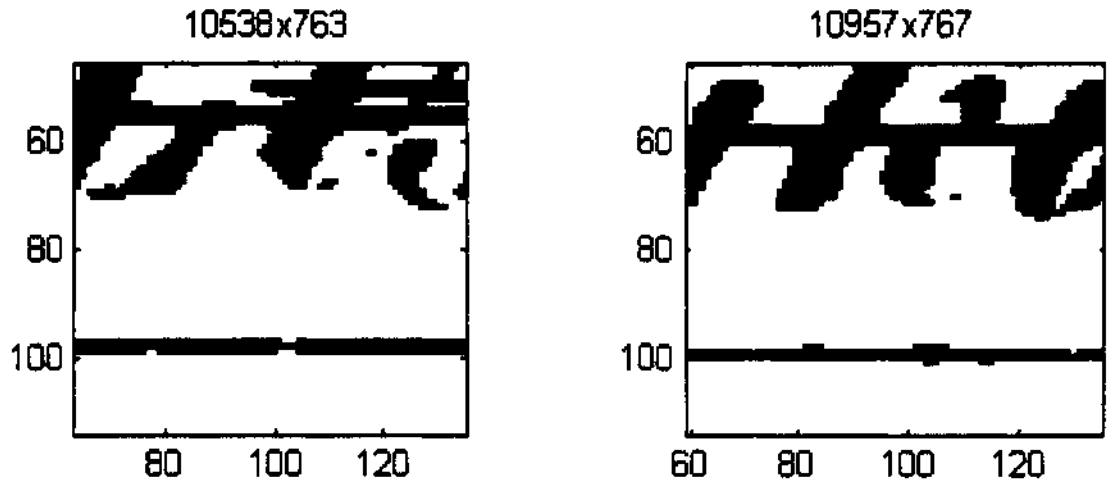
3.1.2 Vylepšení masky

Metoda vyhledávaní pomocí korelace se ukázala jako velice úspěšná, avšak v jednom konkrétním případě selhává – pokud se v prohledávané oblasti nachází vetší černá plocha. Při použití výše popsané masky je ze vzorce (1) zřejmé, že maximum bude nalezeno v oblasti, kde bude maska přiložena největší plochou na černou plochu. Proto jsem tuto masku, původně založenou jen na obrazové podobnosti s hledanou značkou, upravila. Úprava spočívá v tom, že byly do masky přidány pruhy s hodnotou „–1". Protože je mapa binární, v případě, že maska bude přiložena na rámovou značku, padne pruh „–1" do oblasti 0 na mapě a nebude započítán do výsledku. V případě výpočtu korelace s jednolitou plochou bude ale hodnota výsledku snížena o součet všech „–1", které leží v této ploše. Díky tomu bude výsledek na černé ploše menší než u hledané značky. Pruh má šířku jeden pixel a vzdálenost mezi ním a čárou tvaru T je dva pixely.

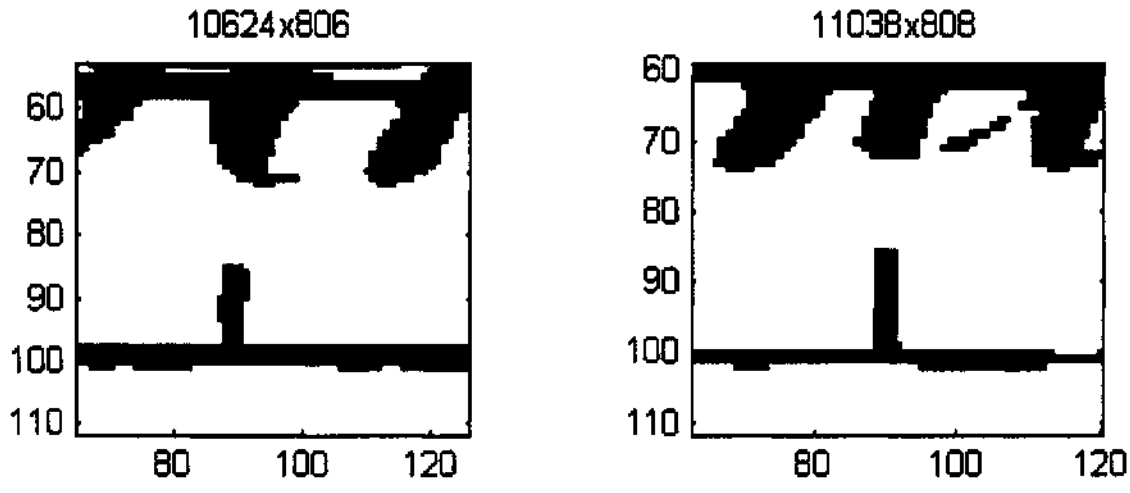


Obrázek 3.6: Výsledná korelační maska pro vyhledání rámové značky

Následující čtveřice obrázků ilustruje nejčastější problém, a to výrazný text v prohledávané oblasti. U první dvojice obrázků se za použití jednoduché masky naleznou oba body špatně, a to v oblasti textu. V druhé dvojici, při použití vylepšení, se oba body naleznou správně.



Obrázek 3.7: Špatně nalezené rámové značky kvůli výraznější kresbě v okolí rámu



Obrázek 3.8: Správné nalezení při použití vylepšené masky

3.2 Výběr bodu rámové značky

Není zaručeno, že korelace vydá jediný bod s maximální hodnotou. K vícenásobnému výsledku může dojít ve dvou případech:

1. korelací prohledávaná oblast rámovou značku neobsahuje nebo obsahuje natolik velké množství šumu, že se nalezne více spolu nesouvisejících bodů se stejnou (maximální) hodnotou
2. mapa má hodně tlusté linie a vzor masky se nalezne v oblasti rámové značky několikrát vedle sebe se stejnou hodnotou

V prvním případě neexistuje žádný postup, jakým určit, který z nalezených bodů je ten správný (pokud některý z nalezených bodů správný je), a tak lze pouze konstatovat, že se značka nenalezla. Tato situace nastává velice zřídka a netvoří významnou část ze špatně nalezených značek.

V druhém případě je výsledkem izolovaná plocha korelačních maxim v oblasti průsečíku rámové značky s rámem. V tomto případě jsem zvolila výběr bodu výpočtem těžiště oblasti korelačních maxim. Vypočtené neceločíselné souřadnice těžiště se zaokrouhlí na nejbližší pixel.

3.3 Vyhledávání rámu

Přestože jsem se úpravou masky byla schopna vyhnout některým konkrétním případům, nevyřešilo to zásadní problém použití korelace na celou mapu. Rámová značka není totiž vzhledem k celé mapě žádným význačným objektem, a tak je třeba aplikaci korelace omezit na malé okolí předpokládané polohy rámové značky. Dalším pádným důvodem pro omezení oblasti je nemalá časová náročnost výpočtu korelace.

Logickým krokem byl pokus o vyhledání rámu mapy, na jehož okolí bych se následně mohla při hledání rámových značek zaměřit. Nejpodstatnější překážkou při řešení tohoto problému je velikost mapy. Vzhledem k tomu, že jediná instance mapy zabírá v paměti v průměru 700 MB (v prostředí MATLAB, které bylo pro realizaci této práce zvoleno), není možné aplikovat žádnou metodu, která je paměťově náročná (V případě, že jsou data větší než operační paměť, začne počítač odkládat data na disk a časová náročnost zpracování mnohonásobně vzroste). Stejně tak není možné s každým prvkem mapy provést časově náročnou matematickou operaci vzhledem k tomu, že mapa má více než 225.10^6 prvků. Proto jsem upustila od metod založených na hranovém detektoru či záplavové výplni, které vyžadovaly dodatečnou alokaci paměti o obdobné velikosti, jako je prohledávaná mapa.

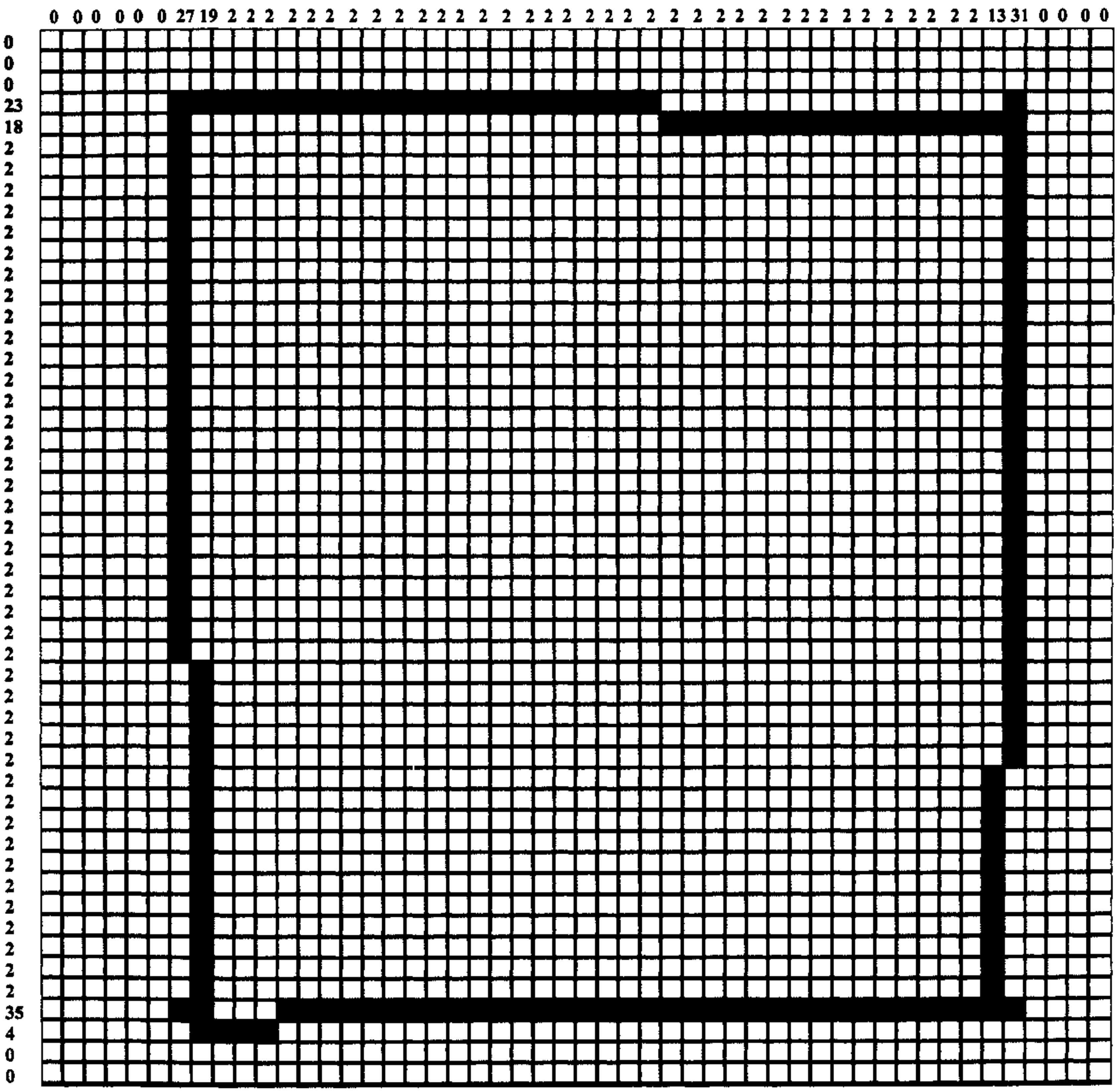
Přesto jsem zaznamenala alespoň částečný úspěch s metodou, která využívá součtu matice mapy do jednoho rozměru. Vzhledem k tomu, že je rám nejdelší čárou (přesněji křivkou) na mapě, která nemá příliš velkou odchylku od směru rovnoběžného s okrajem mapy, objeví se na souřadnici rámu v sečteném vektoru maximum. Vzorce (3.2) a (3.3) ukazují výpočet součtových vektorů pro obrazovou matici n x m.

$$v_i^x = \sum_{j=1}^{m} o_{ij} \qquad (3.2) \qquad\qquad v_j^y = \sum_{i=1}^{n} o_{ij} \qquad (3.3)$$

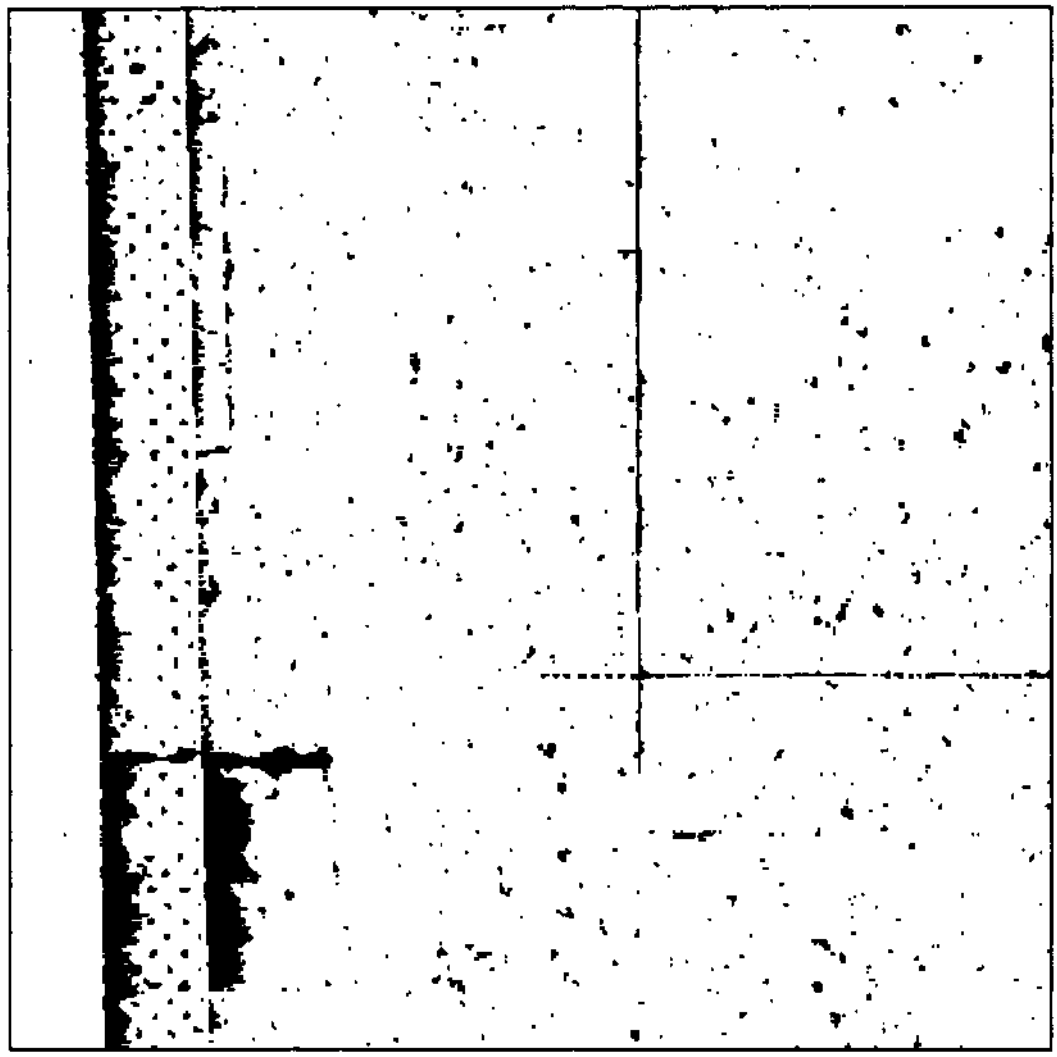
Vzhledem k tomu, že rám mapy je deformován, dá nám součtová metoda jen přibližnou oblast polohy rohů rámu. Přesné nalezení rohů se provádí pomocí výpočtu korelace s příslušnou maskou.

Tato metoda selhává v případech, kdy se v mapě nachází výrazné okraje mapového listu nebo je příliš veliká tabulka skenovacích informací. Pokud je vnější oblast kolem rámu mapy odstraněna, dochází k výraznému nárůstu úspěšnosti. Přestože zadavatel neměl původně v úmyslu tuto úpravu s mapami provádět, nyní o ní uvažuje, protože by to přineslo možnost zpracovávat většinu map zcela automaticky. Konkrétní údaje k úspěšnosti této a ostatních použitých metod jsou uvedeny v kapitole 4.2.

Následující obrázek ilustruje metodu automatické detekce na několikanásobné zvětšenině obrázku o velikosti 50x50 pixelů. Kresba na něm simuluje rám mapy – obdélník, jehož hrany nejsou zcela rovnoběžné s okrajem. Řady čísel jsou součtové vektory v příslušném směru. Na obrázku jsou vyznačeny dvě linky pro každý rozměr, každá je určena maximem z poloviny vektoru.



Obrázek 3.9: Demonstrace automatické detekce rámu

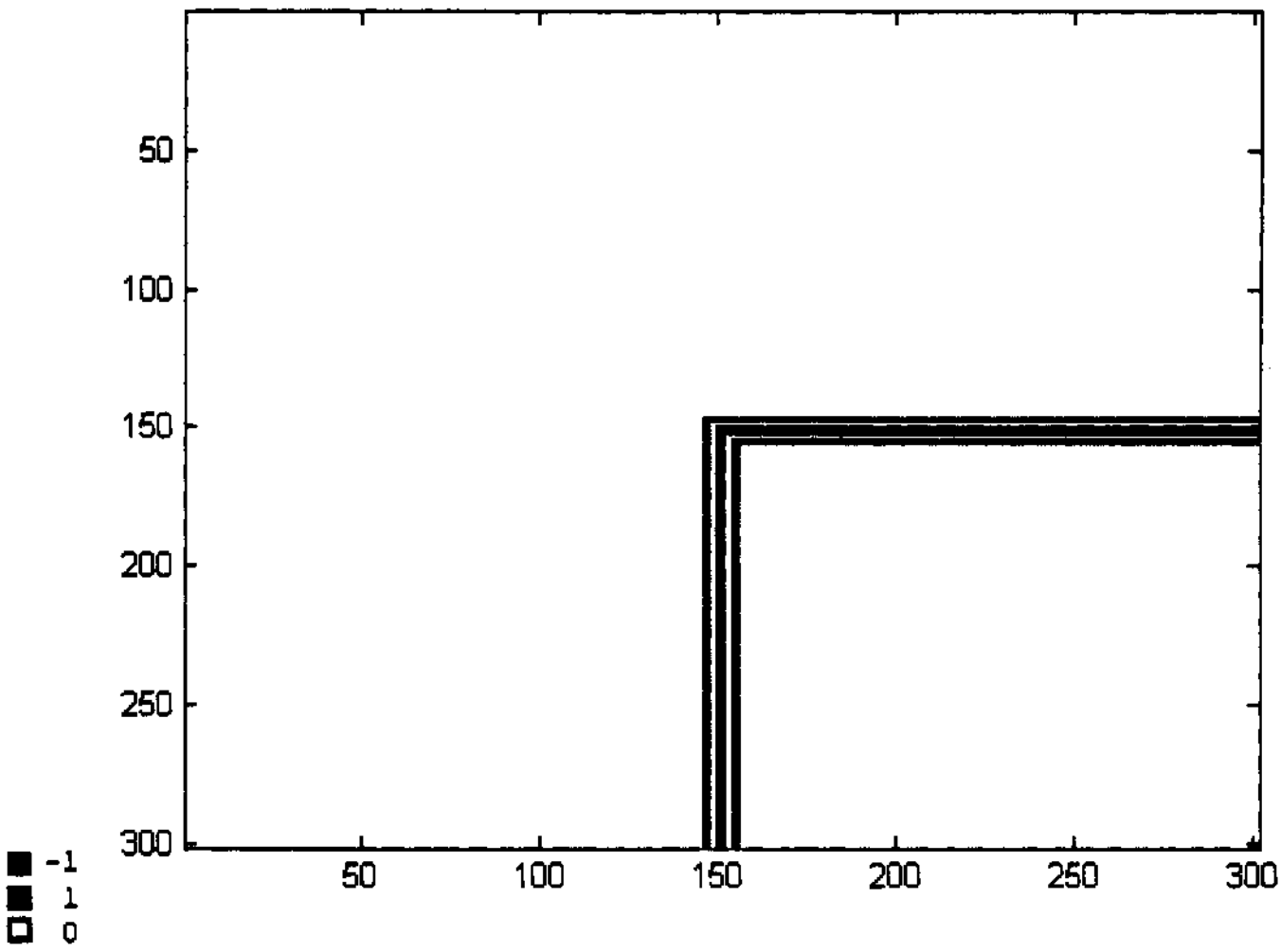


Obrázek 3.10: Šum v obraze rušící metodu automatické detekce rámu mapy

3.4 Nalezení přesné polohy rohů rámu

3.4.1 Korelační maska pro detekci rohů

Tato maska vychází z tvaru rohu obdélníku. Má čtvercový tvar o rozměrech 301x301 pixelů. Vzor má tvar písmene L, spojující středy dvou sousedních stran; šířka hrany je opět tři pixely. I tato maska má heuristické vylepšení využívající pruhu „–1“. Ten má šířku tři pixely, nachází se po obou stranách vzoru a rozestup mezi nimi a vzorem je jeden pixel.



Obrázek 3.11: Korelační maska pro vyhledání rohů rámu

3.4.2 Vyhledání rohů rámu

Detekce značky se provádí korelací se stejně velkou oblastí 301x301 pixelů ležící středem v bodě, který je považován za přibližnou polohu rohu rámu. Tento bod je buď získán automatickou detekcí rámu, nebo je zadán uživatelem. Díky přesnému dohledání není třeba, aby uživatel zadal přesnou polohu rohu, ale jen přibližnou – s tolerancí 50 pixelů v obou rozměrech.

Čím větší je korelační maska, tím větší je šance správného nalezení rohu. Vzhledem k tomu, že roh nesvírá přesně úhel 90°, znamená to také, že čím větší maska, tím menší přesnost máme v označení bodu, kde se obě hrany rámu stýkají.

Kvůli tomu, že se nalezení všech značek odvíjí od správné lokalizace hran rámu, zvolila jsem masku právě takovýchto rozměrů. Díky tomu jsem docílila 100% úspěšnosti, ovšem na úkor jisté ztráty přesnosti.

3.5 Vyhledávání značek v oblasti rámu

Po nalezení rohů mapy se vyhledávání rámových značek pomocí korelace provádí jen kolem hran rámu. Z původní mapy je vyříznut pruh začínající prvním rohem a končící druhým, a to pro každou hranu rámu. Výška pruhu je 141 pixelů se středem v souřadnici prvního rohu. Hrana rámu díky deformaci osciluje kolem spojnice rohů v intervalu ne širším než 80 pixelů, zvolená výška pruhu je tedy dostatečná. Na hraně rámu se krom rámových značek vyskytují i další kresby. Kvůli tomu a také kvůli dalšímu snížení časové náročnosti bylo třeba ještě více omezit oblast pro aplikaci korelace. Vzhledem k tomu, že je známo dpi mapy a reálná vzdálenost mezi značkami, lze vypočítat polohu značky za předpokladu nulové deformace. Rozdíl mezi reálnou polohou a vypočítanou polohou od sousední (již známé) značky pak není příliš velký. V prvním kroku je známa přesná poloha rohu, a tak je možné vypočítat polohu první značky bez deformace. Od této souřadnice se vychází a korelace se aplikuje na oblast o šířce 90 pixelů na každou stranu od této souřadnice. Nalezené přesné souřadnice se použijí pro výběr další oblasti k prohledání atd.

3.5.1 Kompletní algoritmus

V předchozích kapitolách jsem popsala jednotlivé postupy použité v programu pro automatické vyhledávání rámových značek. Nyní je dám do souvislostí

1) **Nalezení rohů rámu.**
- Zcela automaticky pomocí součtové metody a následného upřesnění pomocí korelace.
- Za pomoci přibližného uživatelského vstupu. V tomto případě musí uživatel zadat přibližnou polohu všech čtyř rohů. Poloha je opět upřesněna pomocí korelace.
- Pouze pomocí uživatele. Uživatel zadá přesné polohy všech rohů. Tento mód je jen pro výjimečné případy.
- V případě, že má mapa neúplný rám, je třeba, aby uživatel zadal orientaci konkrétní hrany, na níž se mají značky rozpoznat, a dva body, které ohraničují prohledávaný úsek (bod = roh nebo rámová značka).

2) Pro každou hranu rámu **vyříznutí pruhu** o výšce 141 pixelů se středem v získané souřadnici jednoho z rohů hrany.
3) **Výřez oblastí z pruhu** ve vzdálenosti odpovídající vídeňskému palci za daného dpi od souřadnic poslední rozpoznané značky (v prvním kroku od rohu). Oblast má šířku 181 pixelů.
4) **Vyhledání značky** v oblasti pomocí korelace. V případě vícenásobného výsledku výběr jednoho bodu.

Body 3 a 4 se opakují, dokud rozpoznaná značka není vzdálena od konce pruhu méně nebo právě jeden vídeňský palec.

Kapitola 4 Vlastnosti metody

V této kapitole přiblížím vlastnosti mnou navržené metody a především dokáži, že je metoda použitelná v praxi. K tomu účelu provedu testy na reálných i syntetických datech.

4.1 Syntetická data

Pro potřeby všech testů na syntetických datech byly vygenerovány dvě mapy.

Mapa	generováno v	velikost	tvar rámu	šířka čáry	délka rám. značky	Dpi	počet značek
mapa.m	MATLAB	3500x4000	obdélník	3 pixely	náhodná	400	30
mapa1.pcx	Grafický editor	2600x2800	čtyřúhelník	3 pixely	náhodná	400	16

Tabulka 4.1: Vlastnosti syntetických dat

Syntetické mapy se snaží co nejvěrněji simulovat reálná data s tím rozdílem, že je u nich známa přesná poloha všech hledaných bodů. Vychází z nejobvyklejší hodnoty šířky hrany rámu, a to tří pixelů. I délka rámových značek kopíruje reálná data, ve kterých se setkáváme s nejrůznější délkou. Mapa vytvořená v grafickém editoru simuluje reálná data o něco věrněji, protože žádné dva její rohy rámu nemají společné souřadnice (hrany rámu nejsou rovnoběžné).

4.2 Úspěšnost

V této kapitole stanovím úspěšnosti jednotlivých metod použitých v mém algoritmu. Pro každou skupinu budou úspěšnosti testovány na vzorku 85 map. Jako doplňující test bude ještě proveden pokus se syntetickými daty. Všechny úspěšnosti budou vypočteny následujícím vzorcem:

$$U = \frac{\text{správné detekce}}{\text{všechny detekce}} \cdot 100\ \%$$

V případě rámové značky se za správnou detekci považuje bod, který padne do oblasti průsečíku rámu mapy s rámovou značkou. V případě rohu se za správnou detekci považuje bod, který padne do průsečíku hran rámu mapy.

4.2.1 Úspěšnost automatické detekce rámu

„Staré" mapy
Bez úprav.....**49.4 %**
Po oříznutí.....**98.8 %**

Bez předchozích úprav bylo úspěšně nalezeno 42 map. Po odstranění vnější oblasti rámu (včetně tabulky skenovacích informací) bylo úspěšně nalezeno 84 map. Bylo zachováno to okolí rámu, které obsahovalo informace důležité pro další použití (přesahující kresby, název mapy...).

„Nové" mapy
Bez úprav.....**74.1 %**
Po oříznutí.....**100 %**

Stejně jako v předchozím případě pomohlo oříznutí vnější části rámu ke zvýšení úspěšnosti ze 78 map na plných 85.

4.2.2 Úspěšnost detekce rohů rámu

Úspěšnost přesné detekce rohů byla testována na 340 rozích pro každou skupinu.

„Staré" mapy
Úspěšnost detekce rohů.....**100 %**

„Nové" mapy
Úspěšnost detekce rohů.....**100 %**

4.2.3 Úspěšnost detekce značek

„Staré" mapy
Úspěšnost detekce značek.....**97.9 %**

Z celkového počtu 6639 značek bylo 141 nalezeno špatně.

„Nové" mapy
Úspěšnost detekce značek.....**99.97 %**

Z celkového počtu 7217 značek byly nalezeny 2 špatně.

4.2.4 Úspěšnost na syntetických datech

Úspěšnost automatické detekce rámu.....**100 %**

Úspěšnost detekce rohů rámu.....**100 %**

Úspěšnost detekce značek.....**100 %**

4.3 Přesnost

V této kapitole se pokusím ukázat, že tato metoda má dostatečnou přesnost na to, aby mohla nahradit doposud používané ruční odečty. K tomu použiji experimenty na syntetických mapách a na náhodném vzorku reálných map.

4.3.1 Test na reálných mapách

Vzhledem k tomu, že neexistují žádné jiné údaje o přesné poloze než ruční odečet, budu považovat údaj z ručního odečtu za přesnou polohu rámové značky, tedy za bod, který se snažím svou metodou nalézt, a s ním budu výsledky porovnávat. Srovnání výsledků obou metod budou charakterizovat následující vzorce:

$$\sigma = \sqrt{\frac{\sum_i (\bar{x}_i - x_i)^2 + (\bar{y}_i - y_i)^2}{2n}} \qquad (4.1)$$

$$P = 1 - e^{-\frac{1}{2}\left(\frac{r}{\sigma}\right)^2} \qquad (4.2)$$

kde

$$\bar{x}_i = \frac{\sum_j x_{ij}}{m} \qquad \bar{y}_i = \frac{\sum_j y_{ij}}{m}$$

$i = \{1...n\}$
j = {1...m}
n ..... počet všech značek
m ..... počet ručních odečtů jedné značky
x_i ..... x-ová souřadnice i-té automaticky zjištěné značky (obdobně pro y)
$\bar{x}_{ij}$ ..... x-ová souřadnice j-tého ručního odečtu i-té značky (obdobně pro y)

Vzorec (4.1) je odhadem střední souřadnicové chyby automatického určení vzhledem k ručnímu odečtu. Za hodnotu ručního odečtu se vždy považuje průměr ze všech ručních odečtů pro každou značku. Vzorec (4.2) vyčísluje pravděpodobnost s jakou se vyskytne přesný bod v kruhovém okolí (o poloměru r) automaticky zjištěné značky [7]. Přitom se předpokládá, že nejistota v určení polohy značky je izotropní, tj. nezávislá na směru spojnice správná poloha - určená poloha. Hodnota sigma vypočtená podle vzorce (4.1) představuje velikost průměrné radiální odchylky automaticky určené značky od její teoreticky správné polohy odpovídající ručnímu odečtu.

Hodnoty stanovím pro 10 náhodně vybraných map z každé kvalitativní skupiny. Každá mapa bude devětkrát ručně odečtena. Ze vzorce (4.2) bude za stanovení $P = 0.95$ vypočten poloměr konfidenčního kruhu. Výsledky budou uvedeny v pixelech, milimetrech (odpovídajících rozměrům na listu mapy) a přepočteny v měřítku mapy na centimetry (odpovídající reálnému rozměru na území reprezentovaném mapou).

„Staré" mapy

Mapa číslo	střední souřadnicová chyba (pixely)	střední souřadnicová chyba (mm-mapa)	střední souřadnicová chyba (cm-měřítko)	poloměr konfidenčního kruhu (pixely)	poloměr konfidenčního kruhu (mm-mapa)	poloměr konfidenčního kruhu (cm-měřítko)
1	0.647	0.041	11.830	1.583	0.101	28.957
2	0.646	0.041	11.815	1.581	0.100	28.920
3	0.622	0.040	11.378	1.523	0.097	27.851
4	0.778	0.049	14.230	1.905	0.121	34.831
5	0.660	0.042	12.075	1.616	0.103	29.557
6	0.540	0.034	13.202	1.320	0.084	24.144
7	0.730	0.046	13.354	1.787	0.114	32.688
8	0.736	0.047	13.455	1.801	0.114	32.936
9	0.546	0.035	9.984	1.336	0.085	24.440
10	0.749	0.048	13.691	1.832	0.116	33.516

Tabulka 4.2: Srovnání automatické detekce a ručních odečtů na „starých" mapách

„Nové" mapy

Mapa číslo	střední souřadnicová chyba (pixely)	střední souřadnicová chyba (mm-mapa)	střední souřadnicová chyba (cm-měřítko)	poloměr konfidenčního kruhu (pixely)	poloměr konfidenčního kruhu (mm-mapa)	poloměr konfidenčního kruhu (cm-měřítko)
1	0.642	0.041	11.743	1.572	0.100	28.743
2	0.723	0.046	13.213	1.768	0.112	32.341
3	0.763	0.048	13.950	1.867	0.119	34.147
4	0.716	0.046	13.096	1.753	0.111	32.055
5	0.765	0.049	13.981	1.871	0.119	34.221
6	0.777	0.050	14.209	1.902	0.121	34.780
7	0.698	0.044	12.762	1.708	0.109	31.239
8	0.844	0.054	15.429	2.065	0.131	37.767
9	0.700	0.044	12.798	1.713	0.109	31.327
10	0.760	0.048	13.400	1.860	0.118	34.019

Tabulka 4.3: Srovnání automatické detekce a ručních odečtů na „nových" mapách

Jak je vidět z tabulky, s pravděpodobností 95% se ručně určený bod nachází v kruhu se středem v automaticky určeném bodě a poloměrem pohybujícím se mezi hodnotami 0.08 a 0.13 mm v rozměrech mapy. Přesnost měření rozměrů na mapě je 0.2 mm (podle zpracovávajícího pracoviště), tedy jsou tyto hodnoty na hranici měřitelnosti. V reálných rozměrech odpovídají hodnoty rozmezí 28 až 38 cm, což je vzhledem k rozměrům, se kterými se pracuje v katastrálních mapách, zanedbatelné.

Deformace, k jejichž detekci bude metoda sloužit jsou řádově větší než hodnota konfidenčního kruhu zjištěná v měřeních, a tak je možné tuto metodu použít k náhradě ručních odečtů.

4.3.2 Test na syntetických datech

Na obou synteticky vygenerovaných mapách se všech 46 značek našlo přesně.

4.4 Odolnost vůči šumu

V této podkapitole se budu zabývat vlivem šumu na detekci rámových značek. K tomu použiji mapu generovanou v MATLABu.

4.4.1 Test 1

V prvním testu budu generovat šum „pepř a sůl" následujícím předpisem:

```
x = rand(size(img));
d = find(x < p/2);
img(d) = 0;
d = find(x >= p/2 & x < p);
img(d) = 1;
```

kde *img* je mapa, do které se šum generuje, a *p* zastoupení šumu v obrázku. *P* nabývá v tomto testu hodnot 0 až 1 s krokem 0.05.

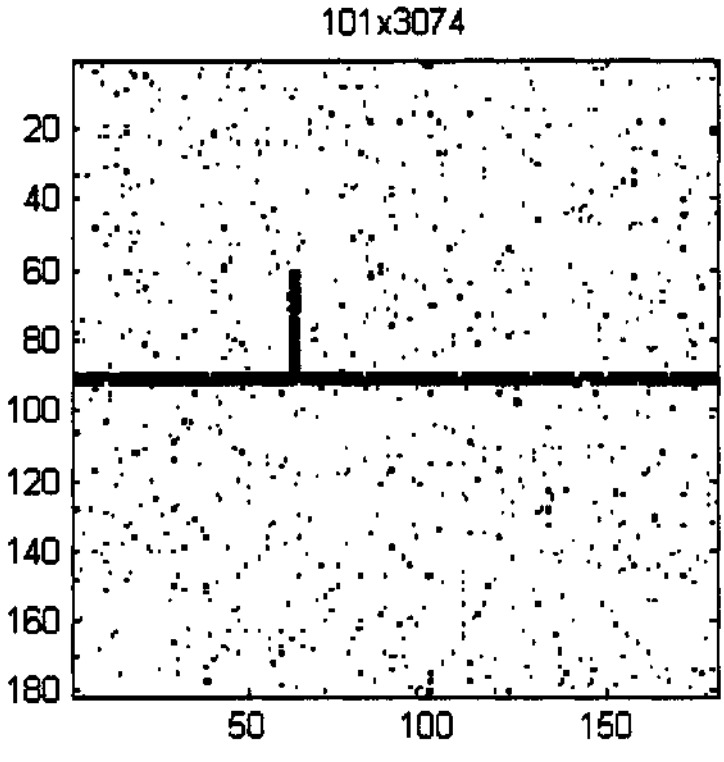
V následující tabulce budou shrnuty výsledky podle procentuelního zastoupení šumu v obraze. Sledované hodnoty budou: úspěšná automatická detekce rámu; úspěšná detekce rohů; počet přesně nalezených značek; počet značek nalezených správně, ale s menší přesností, počet zcela špatně nalezených značek.

hodnota *p*	detekce rámu	detekce rohů	počet přesně nalezených značek*	počet poruch přesnosti	počet špatně nalezených
0	ANO	ANO	30	-	-
0.05	ANO	ANO	30	-	-
0.1	ANO	ANO	30	-	-
0.15	ANO	ANO	30	-	-
0.2	ANO	ANO	30	-	-
0.25	ANO	ANO	30	-	-
0.3	ANO	ANO	30	-	-
0.35	ANO	ANO	30	-	-
0.4	ANO	ANO	30	-	-
0.45	ANO	ANO	30	-	-
0.5	ANO	ANO	29	1	-
0.55	ANO	ANO	28	2	-
0.6	ANO	ANO	24	4	2
0.65	NE	NE	25	1	4
0.7	NE	NE	25	3	2
0.75	NE	NE	3	-	27

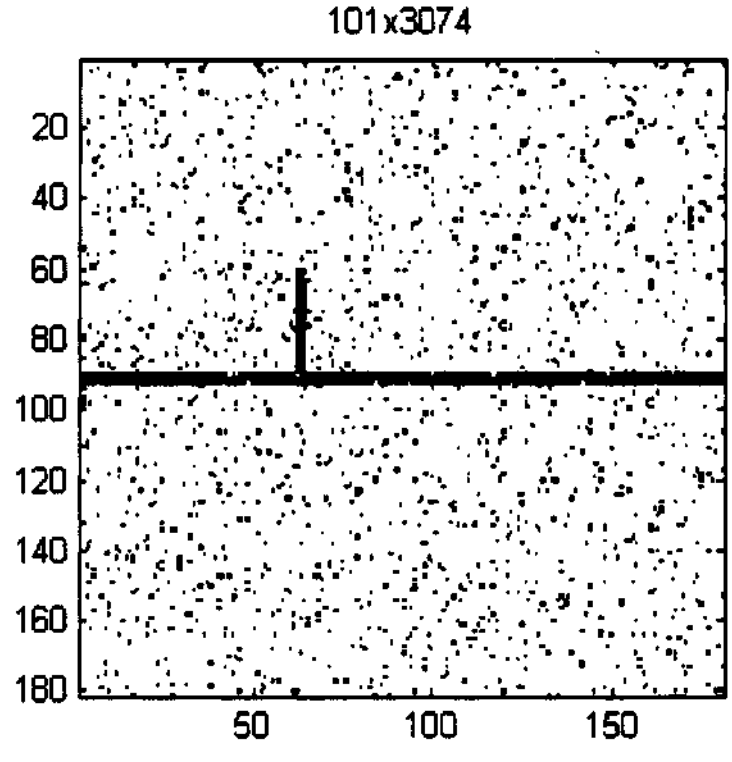
Tabulka 4.4: Odolnost vůči šumu v testu 1

Jak je vidět z tabulky, metoda má vynikající odolnost vůči šumu. Všechny chyby v přesnosti jsou posun v jedné souřadnici o jediný pixel, což je pro praktické použití stále velice dobrý výsledek. Při 75% se počet špatných bodů výrazně zvedl.

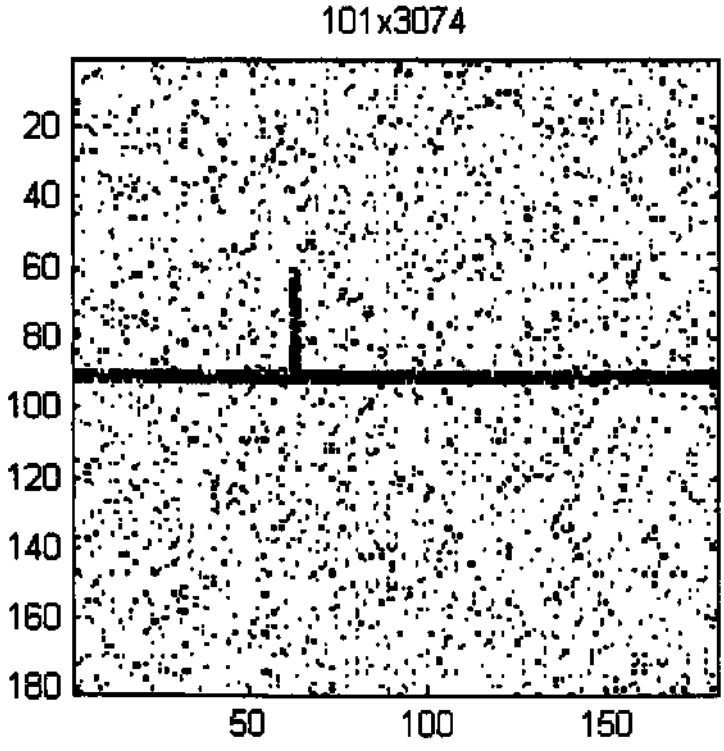
* Počet značek v mapě je 30.



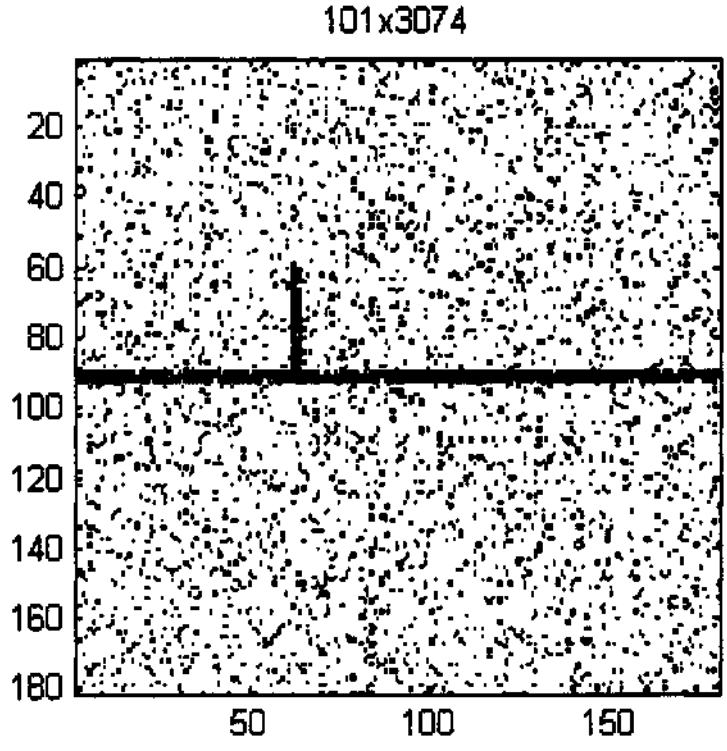
Obrázek 4.1: $p = 0.05$



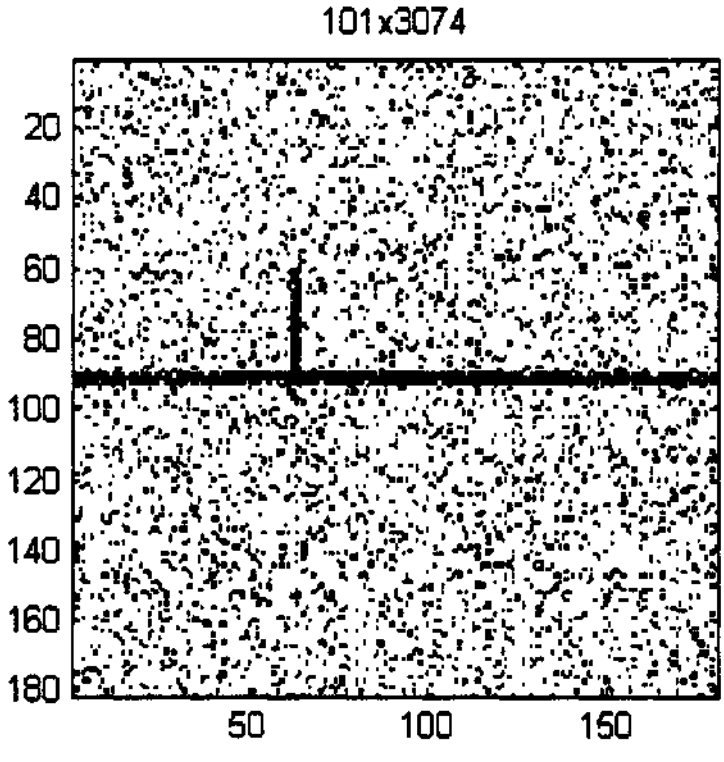
Obrázek 4.2: $p = 0.1$



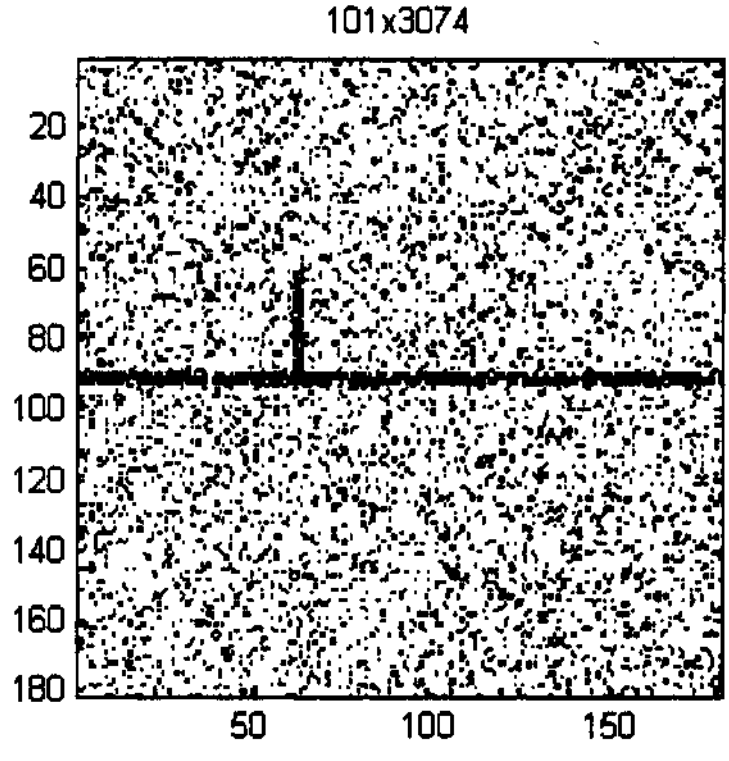
Obrázek 4.3: $p = 0.15$



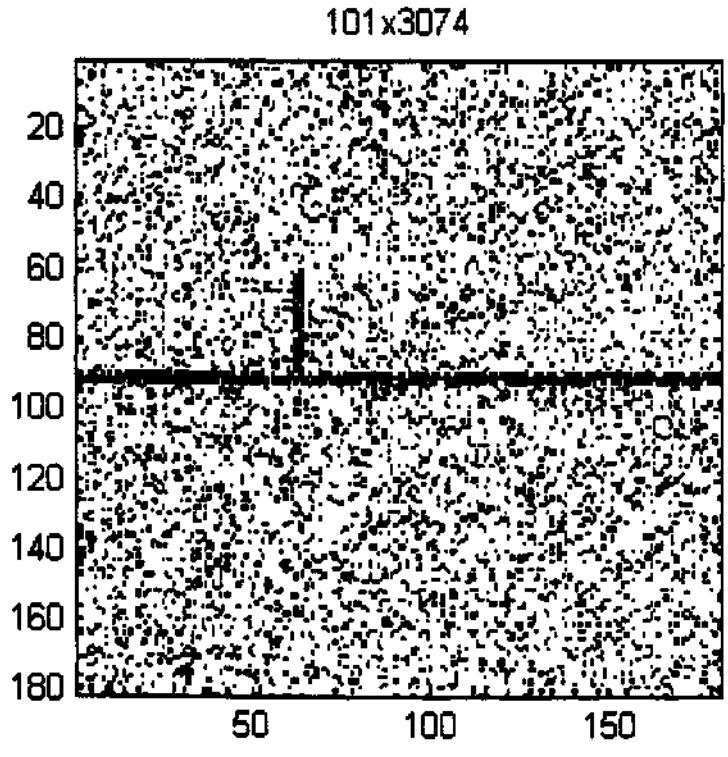
Obrázek 4.4: $p = 0.2$



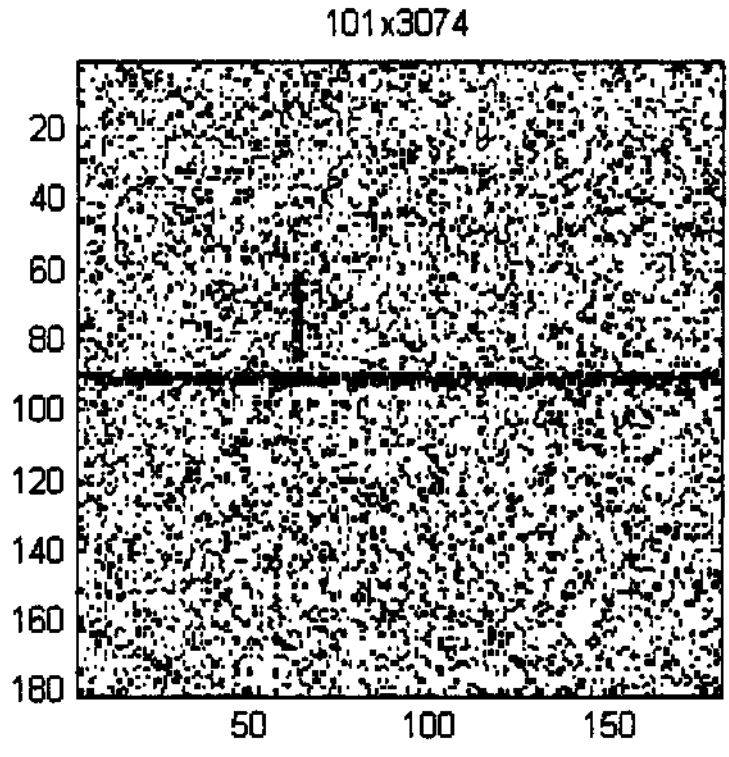
Obrázek 4.5: $p = 0.25$



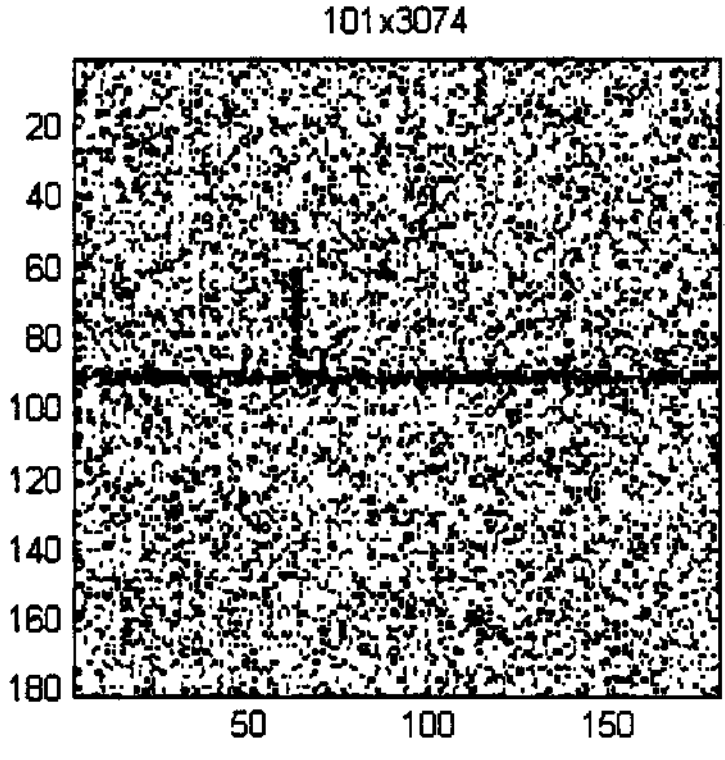
Obrázek 4.6: $p = 0.3$



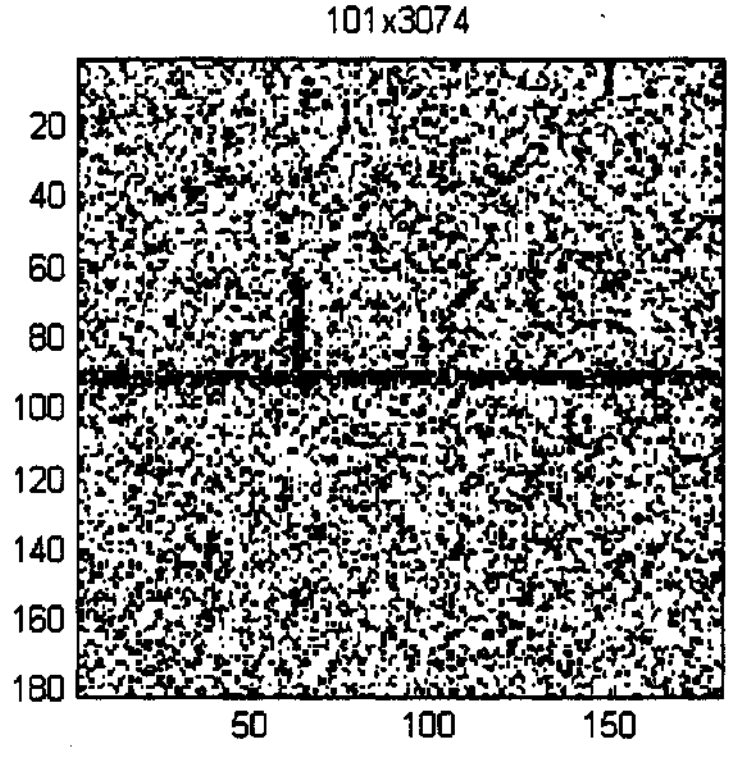
Obrázek 4.7: $p = 0.35$



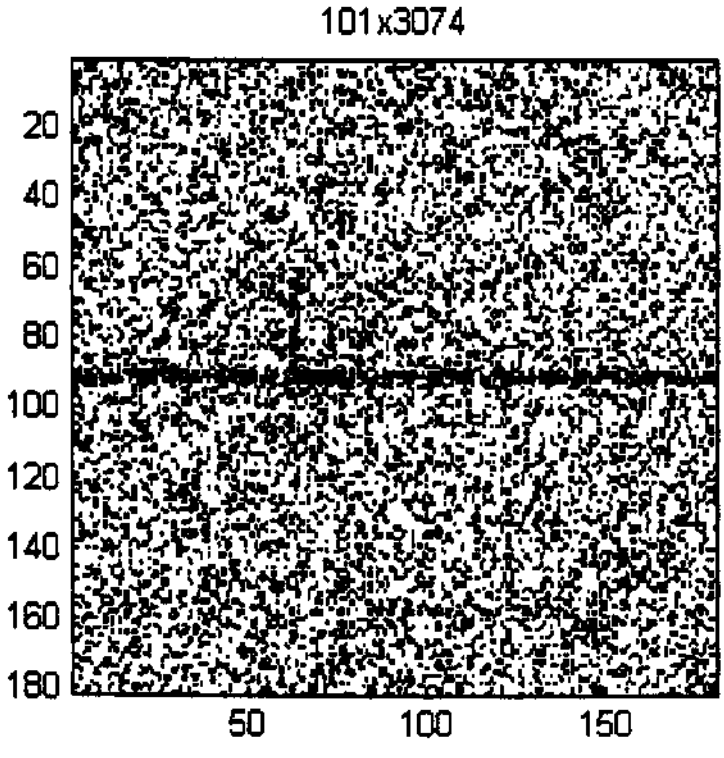
Obrázek 4.8: $p = 0.4$



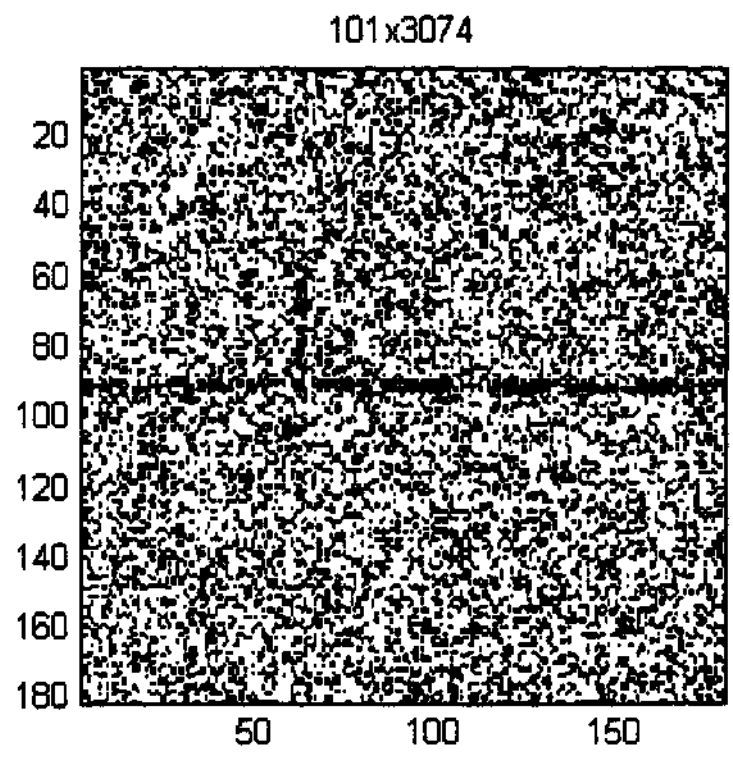
Obrázek 4.9: $p = 0.45$



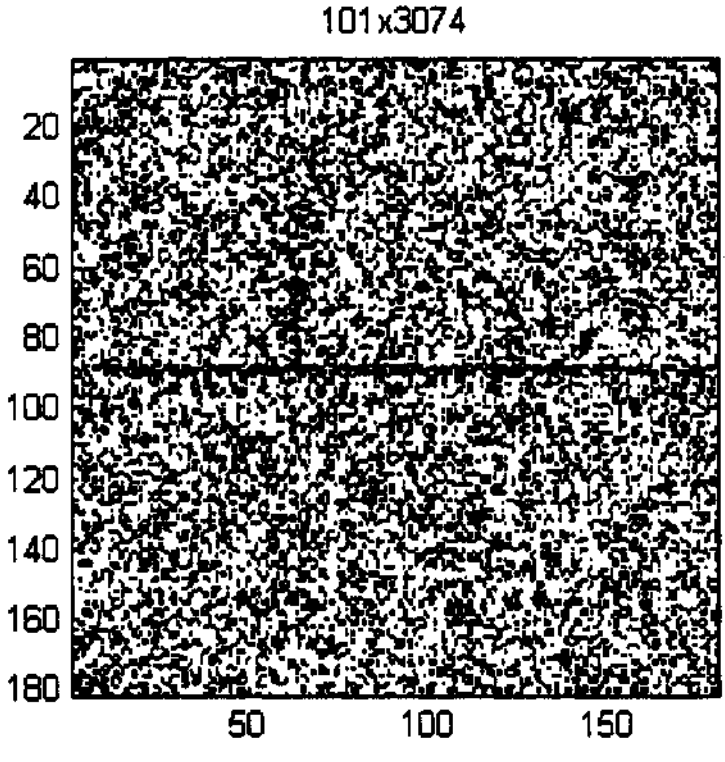
Obrázek 4.10: $p = 0.5$



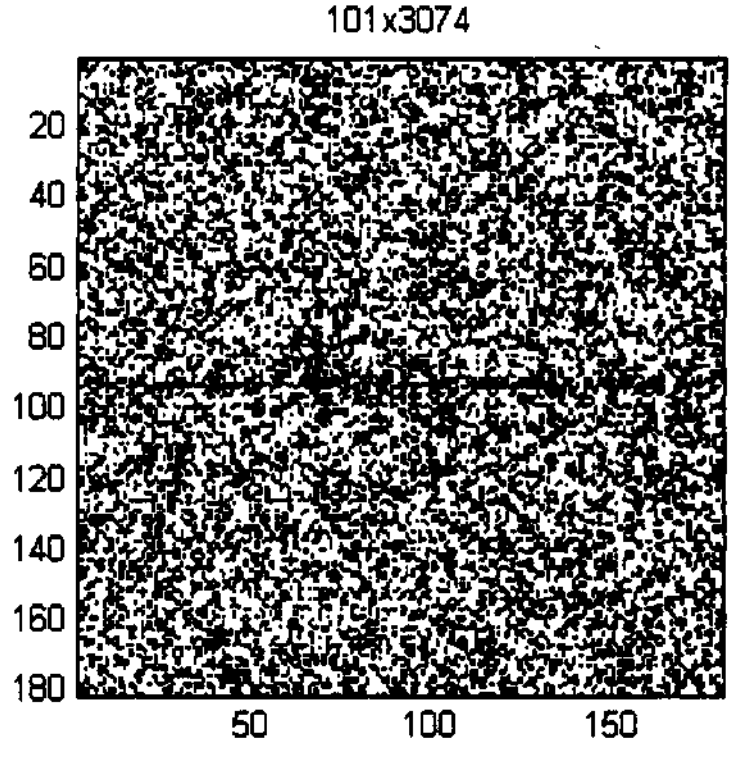
Obrázek 4.11: $p = 0.55$



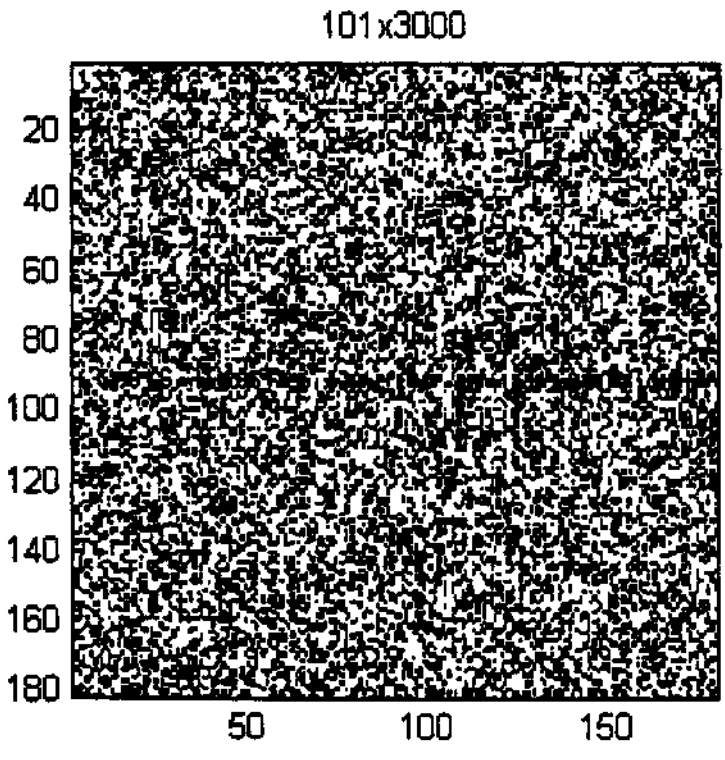
Obrázek 4.12: $p = 0.6$



Obrázek 4.13: $p = 0.65$



Obrázek 4.14: $p = 0.7$



Obrázek 4.15: $p = 0.75$

4.4.2 Test 2

V druhém testu budu generovat poněkud odlišný šum, a to následujícím předpisem:

x = rand(size(img));
d = find(x < p);
img(d) = 1;

Tento šum více odpovídá reálným poruchám, se kterými se setkáváme u starších map, než předchozí případ. Většinou totiž nedochází k výraznému porušení kresby, pouze s časem přibude černých bodů na mapě. Tento šum přidává do obrazu $p \cdot 100\%$ černých bodů.

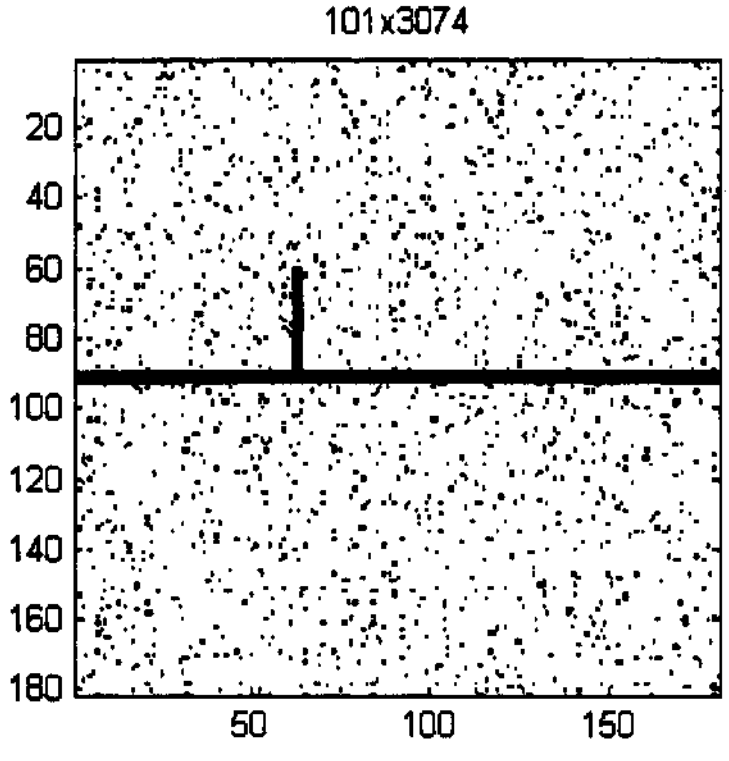
hodnota *p*	**detekce rámu**	**detekce rohů**	**počet přesně nalezených značek***	**počet poruch přesnosti**	**počet špatně nalezených**
0	ANO	ANO	30	-	-
0.05	ANO	ANO	30	-	-
0.1	ANO	ANO	30	-	-
0.15	ANO	ANO	30	-	-
0.2	ANO	ANO	30	-	-
0.25	ANO	ANO	30	-	-
0.3	ANO	ANO	30	-	-
0.35	ANO	ANO	30	-	-
0.4	ANO	ANO	29	1	-
0.45	ANO	ANO	30	-	-
0.5	NE	NE	29	1	-
0.55	NE	NE	30	1	-
0.6	NE	NE	25	4	1
0.65	NE	NE	24	4	2
0.7	NE	NE	21	2	7
0.75	NE	NE	5	-	25

Tabulka 4.5: Odolnost vůči šumu v testu 2

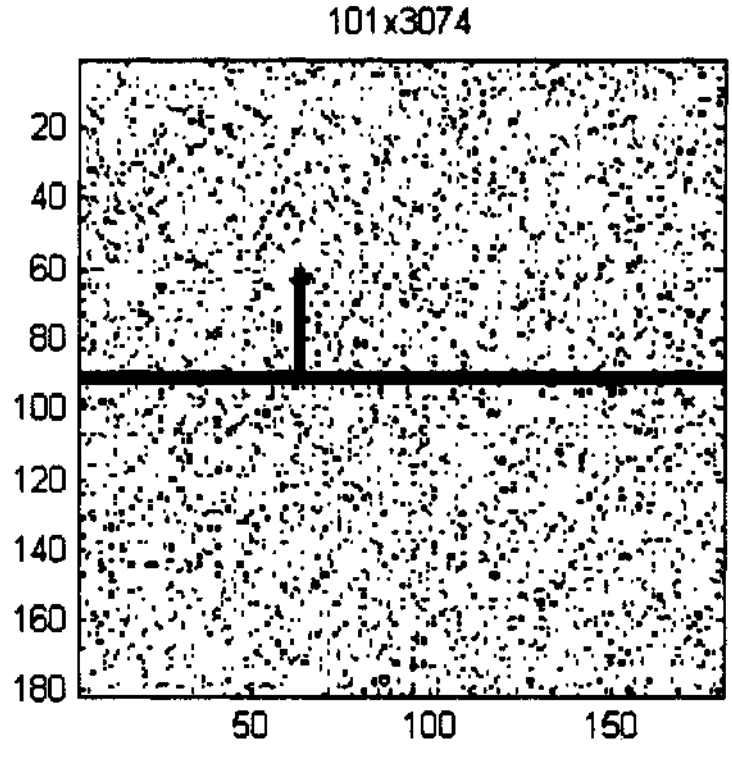
Obdobně jako v předchozím testu i zde jsou všechny poruchy v přesnosti změnou v jedné souřadnici o jediný pixel. Výrazný nárůst chyb opět nastává až při 75% šumu. Na rozdíl od předchozího testu zde dříve selhává automatická detekce rámu a rohů.

Větší citlivost detekce rohů vůči úrovni šumu než u rámových značek je způsobena podstatně větší korelační maskou.

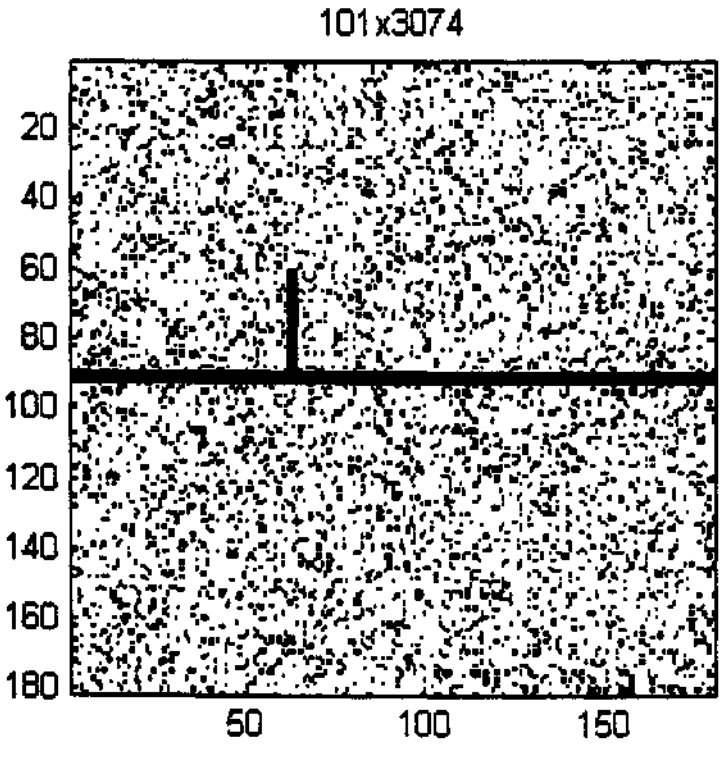
* Počet značek v mapě je 30.



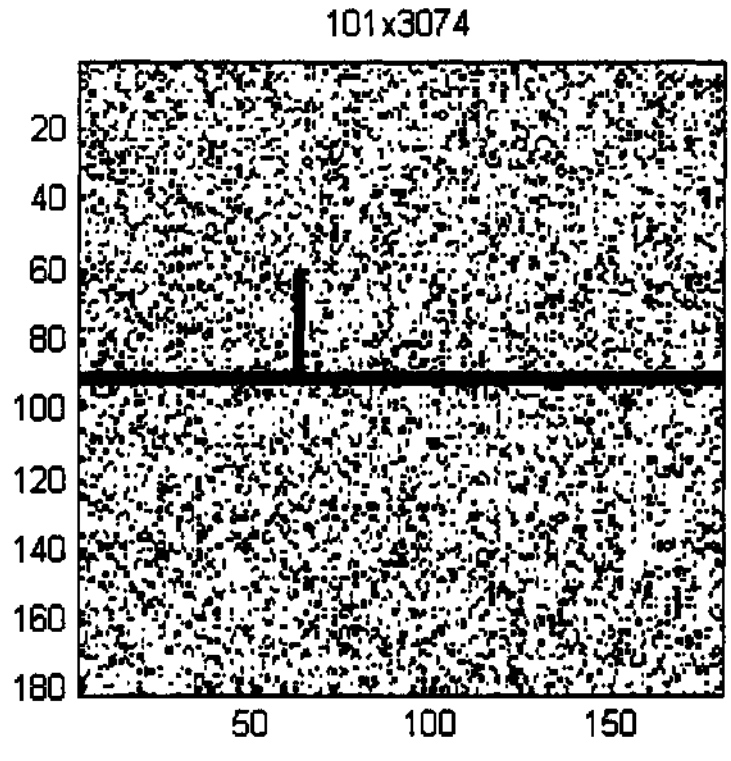
Obrázek 4.16: $p = 0.05$



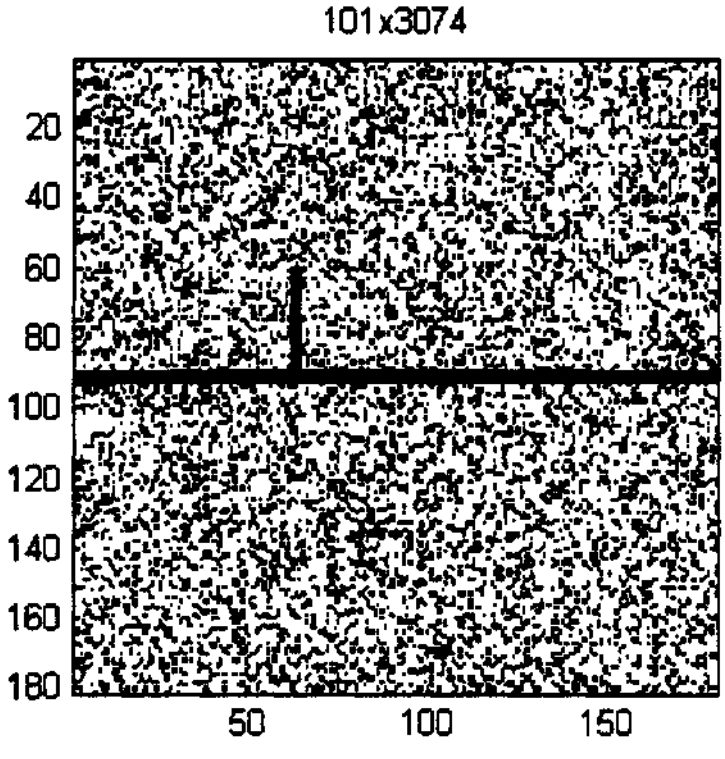
Obrázek 4.17: $p = 0.1$



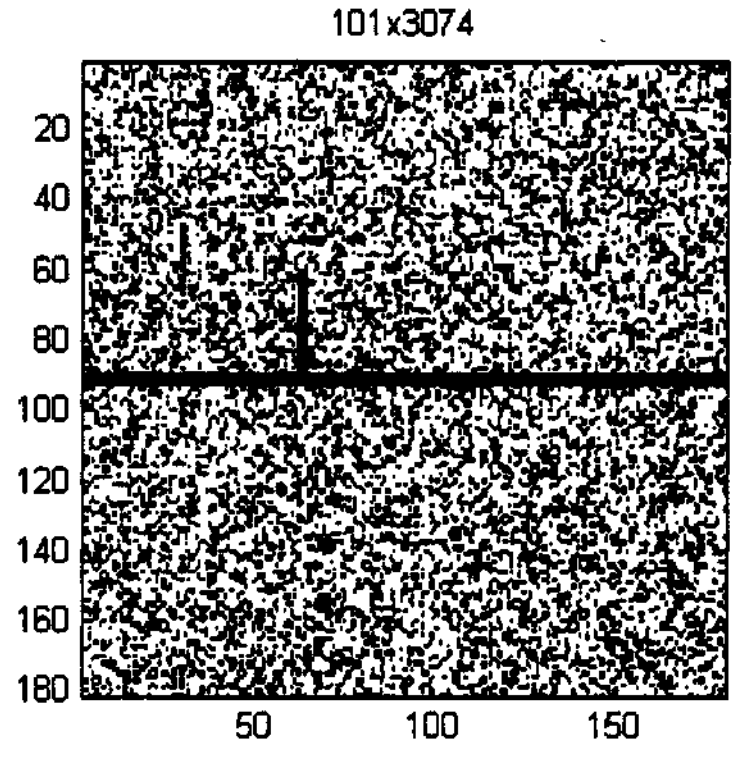
Obrázek 4.18: $p = 0.15$



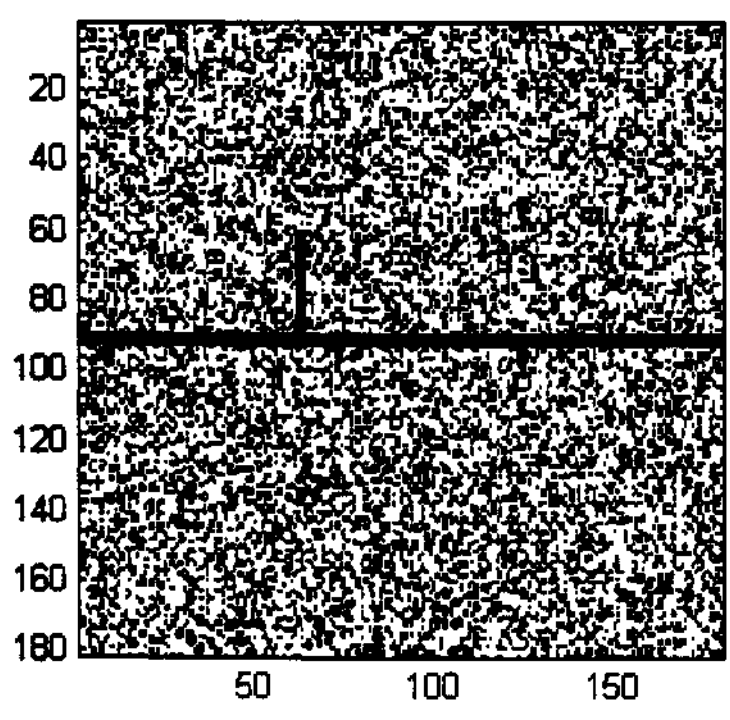
Obrázek 4.19: $p = 0.2$



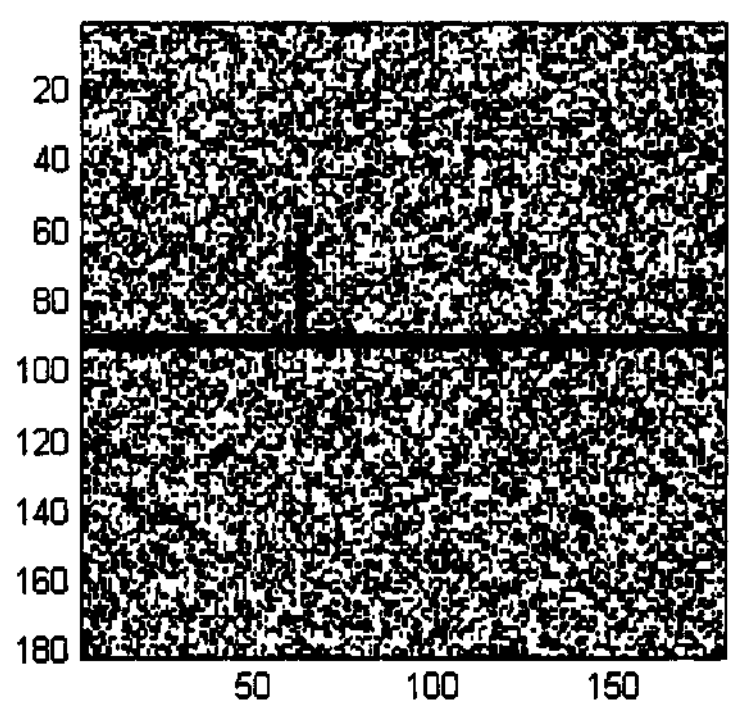
Obrázek 4.20: $p = 0.25$



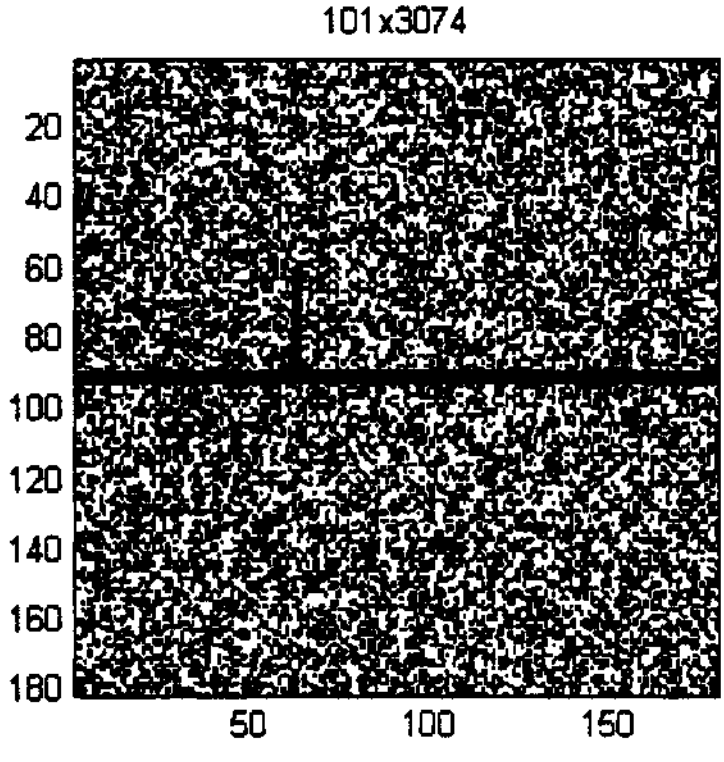
Obrázek 4.21: $p = 0.3$



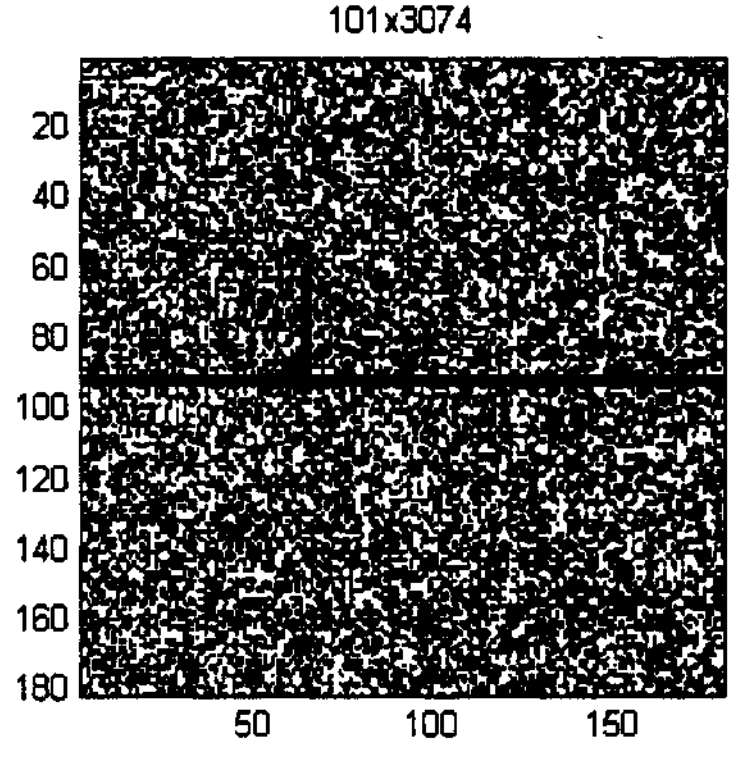
Obrázek 4.22: $p = 0.35$



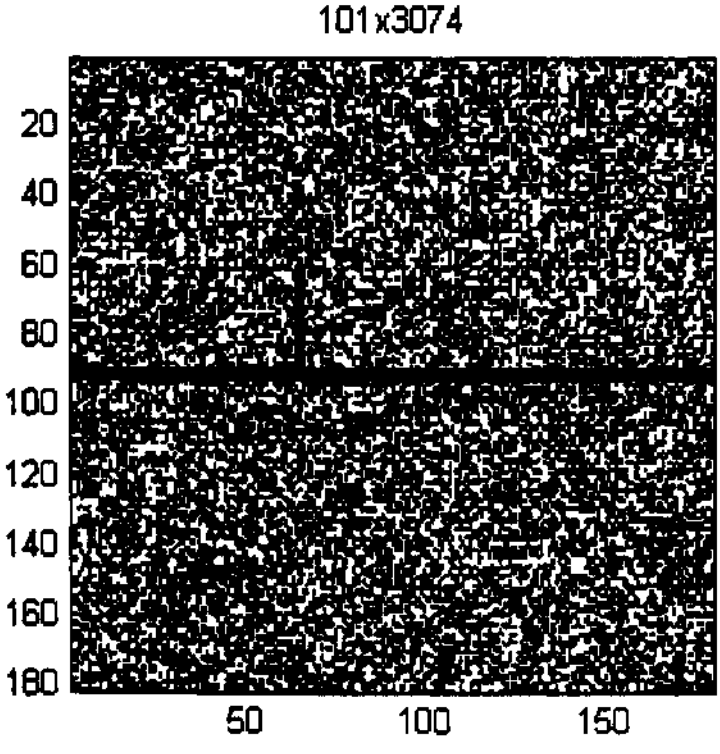
Obrázek 4.23: $p = 0.4$



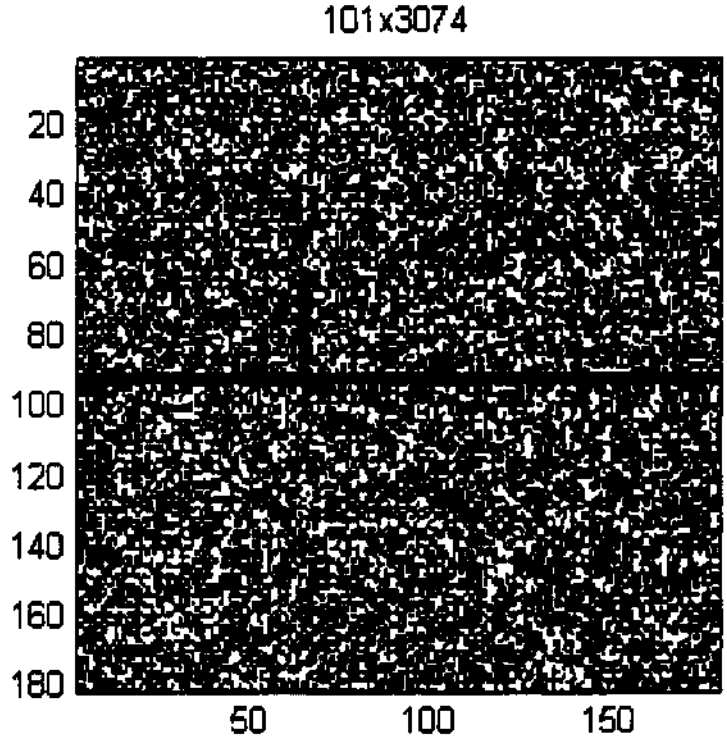
Obrázek 4.24: $p = 0.45$



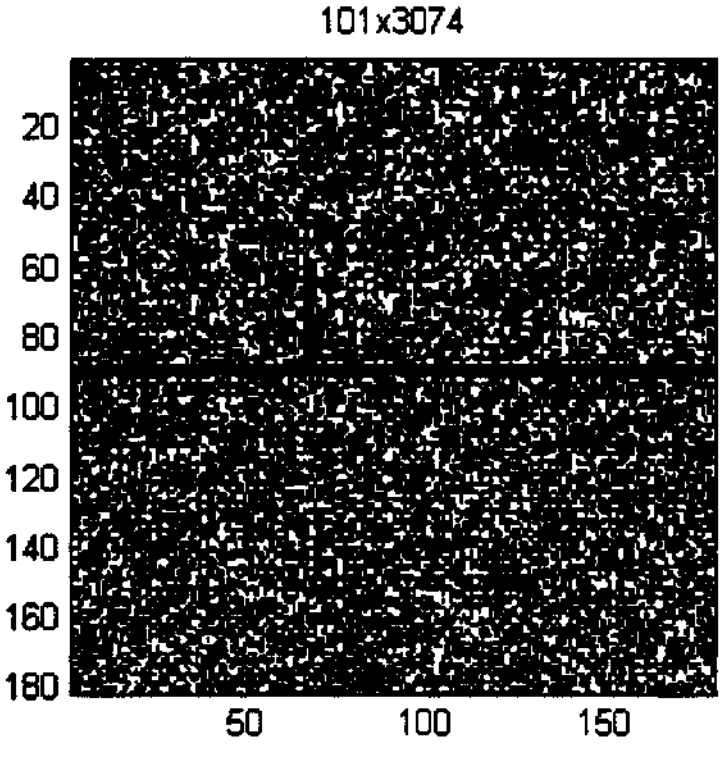
Obrázek 4.25: $p = 0.5$



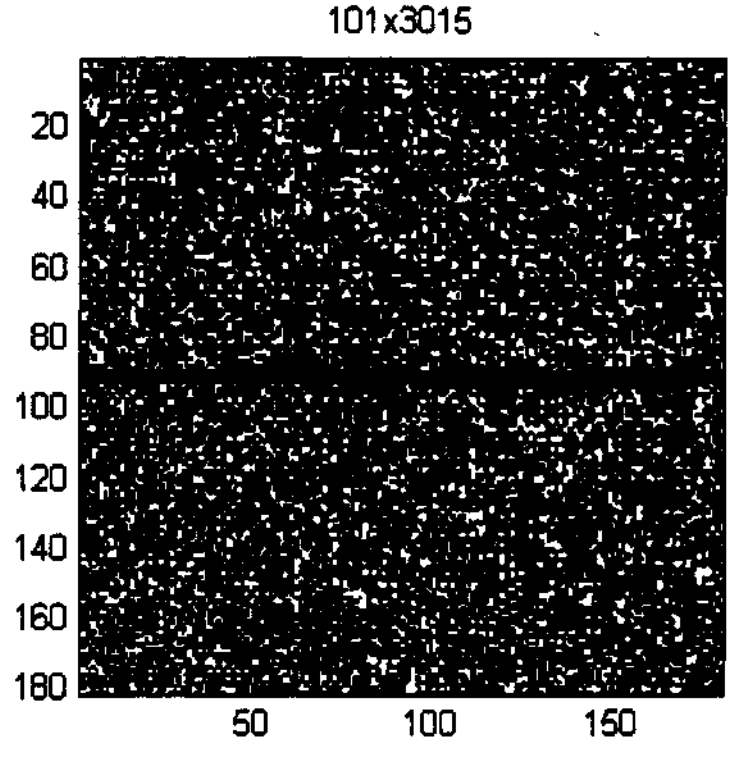
Obrázek 4.26: $p = 0.55$



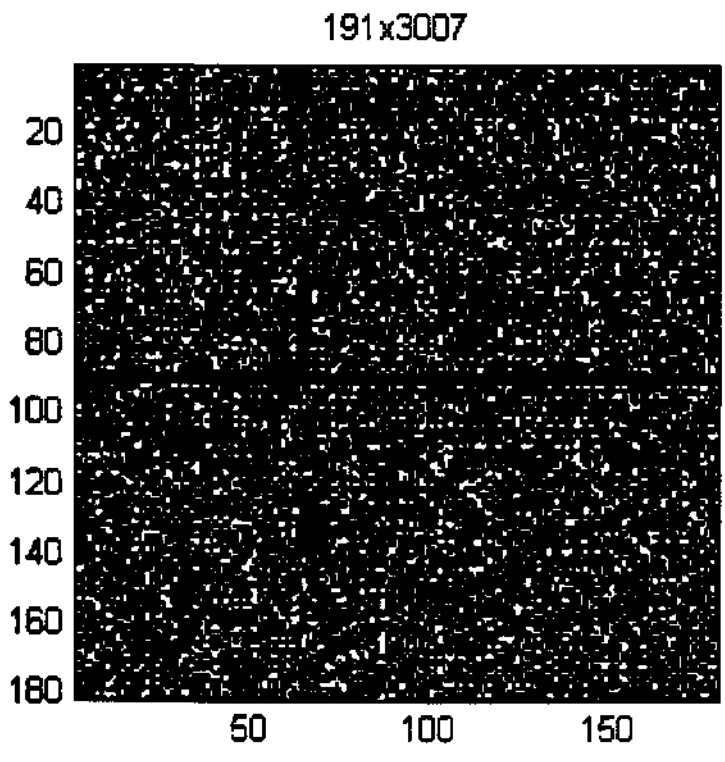
Obrázek 4.27: $p = 0.6$



Obrázek 4.28: $p = 0.65$



Obrázek 4.29: $p = 0.7$



Obrázek 4.30: $p = 0.75$

4.5 Časová úspora při použití automatické detekce

Časovou náročnost automatické detekce nelze vyčíslit vzorcem vzhledem k tomu, že není známo, jakou časovou náročnost mají systémové funkce MATLABu, a protože zpracování zahrnuje interakci s uživatelem. Přesto bych ráda ukázala, jak výraznou časovou úsporu mnou vyvinutá metoda poskytuje. Provedu proto experimentální měření časové náročnosti na vzorku deseti náhodně vybraných map, z nichž každá má 86 rámových značek. Každou z nich zpracuji programem MarkSearch tak, jak by byla zpracována libovolná nově zadaná mapa, a následně provedu ruční odečet všech značek na téže mapě.

U automatického zpracování budu měřit čas od spuštění programu MarkSearch až po vydání výsledku. Toto číslo je ovlivněno výkonem procesoru, množstvím operační paměti a rychlostí reakce uživatele. Sestava, na které byla rychlost automatického zpracování měřena, má procesor **Pentium 4 2.8GHz** a **1 GB RAM**.

Mapa	**Ruční odečet (minuty:sekundy)**	**Automatický odečet (minuty:sekundy)**	**Mód automatického odečtu**
1	9:44	0:50	plně automatický
2	10:48	0:38	plně automatický
3	10:36	0:43	plně automatický
4	12:17	0:36	plně automatický
5	12:00	0:39	plně automatický
6	11:41	0:45	plně automatický
7	12:12	0:40	plně automatický
8	12:14	0:40	plně automatický
9	11:15	0:41	plně automatický
10	12:10	0:38	plně automatický

Tabulka 4.6: Časová úspora při použití automatické detekce

Mimo zjevné **časové úspory** má automatická metoda i další výhody. Na rozdíl od ručního odečtu vydá **vždy stejný výsledek**, protože i v případě jen částečně automatického zpracování nezávisí výsledek na přesném vstupu uživatele. U automatické detekce **nedochází,** na rozdíl od ručních odečtů, **ke změně přesnosti** získaných bodů **v závislosti na čase**, protože člověk neudrží plnou soustředěnost delší časový úsek. Krom toho je **obsluha** automatické detekce mnohem **méně náročná** pro uživatele než provádění ručních odečtů.

Kapitola 5 Program MarkSearch

Program slouží k vyhledávání rámových značek na sáhových katastrálních mapách. Poskytuje grafické rozhraní, které nevyžaduje obsluhu osoby znalé metody vyhledávání ani hlubších znalostí katastrálních map. Je pouze třeba, aby obsluhující osoba byla schopna odečíst hodnotu dpi a případně označit přibližnou polohu rohů rámu.

5.1 Spuštění programu

Program se spouští příkazem marksearch('cesta k souboru mapy') z příkazové řádky programu MATLAB. Parametrem je plná cesta k mapovému souboru, který se má zpracovat.

5.2 Podpora formátů

Vstupní formáty map jsou dány podporovanými formáty programu MATLAB a jsou následující (může se lišit v závislosti na verzi MATLAB): JPEG, TIFF, GIF, BMP, PNG, HDF, PCX, XWD, ICO, CUR, RAS, PBM, PGM, PPM. Výrazně doporučuji používat formát PCX 1-bit.

5.3 Módy programu

Program nabízí několik úrovní zpracování.

Automatický mód

Tento mód zpracuje mapu zcela automaticky. Pokusí se nalézt rám mapy, dohledat přesnou polohu rohů a rozpoznat rámové značky. Jedinou informací, která je od uživatele požadována, je dpi mapy, a to jen v případě, kdy se liší od přednastaveného. Tento mód je používán jako výchozí při spuštění programu.

Přibližné zadání rohů

Pokud neuspěje plně automatická metoda, je potřeba, aby uživatel zadal přibližnou polohu rohů. Je postačující, aby poloha byla přesná zhruba na 50 pixelů od skutečné pozice rohu. Konkrétní polohu si program zjistí sám. Opět je požadováno dpi.

Přesné zadání rohů

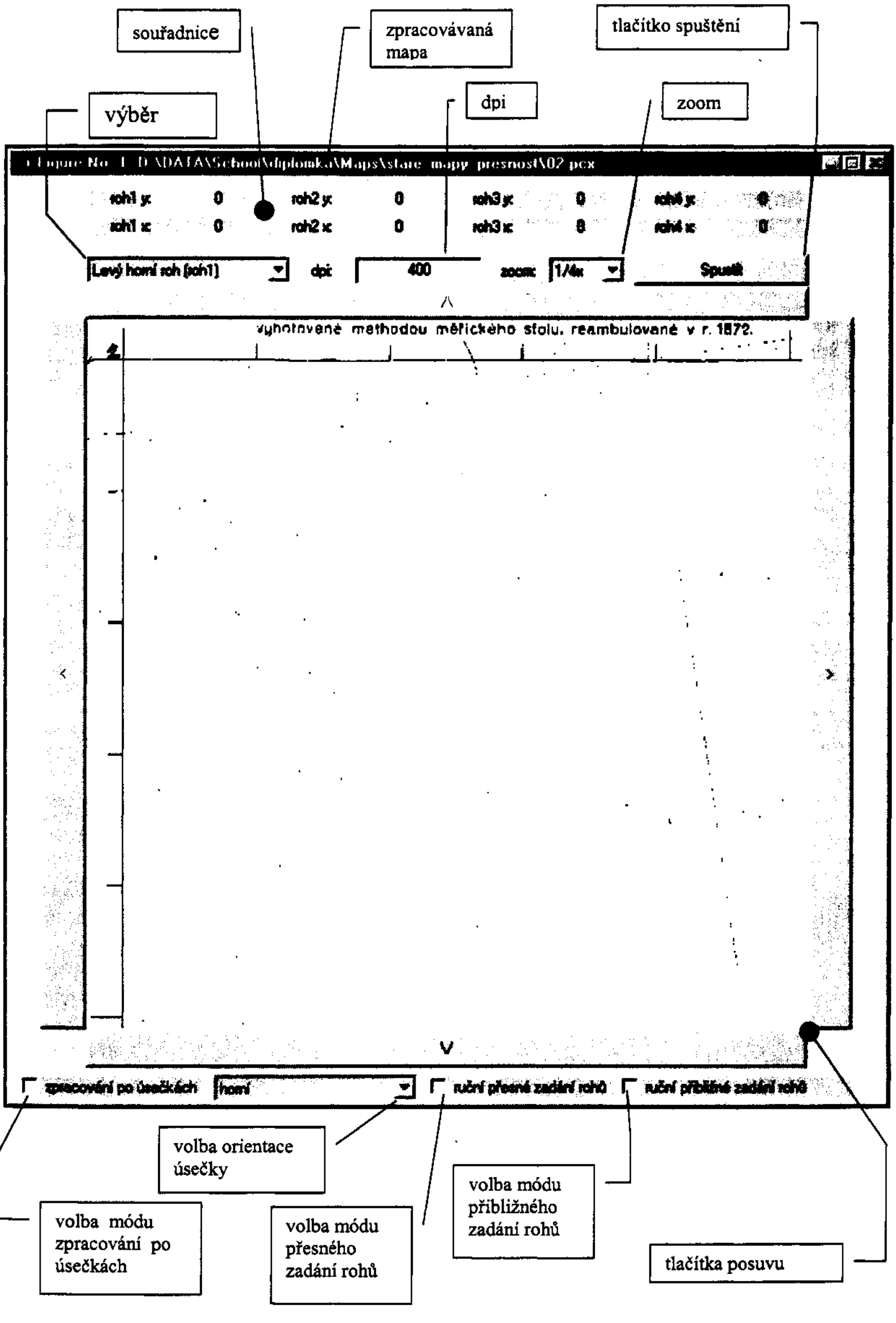
Tento mód nebude obvykle používán. Jde o přesné zadání rohu uživatelem, což může být potřeba, pokud by mapa měla neobvyklé rohy nebo by z nějakých jiných důvodů automatická detekce selhala.

Zpracování po úsečkách

Tento mód byl vytvořen kvůli skupině velice starých map. V té době se kvůli úspoře papíru malé přesahy území zakreslovaly do téže mapy a ne na samostatný arch. V těchto místech je pak rám přerušen (často chybí i více než jedna hrana rámu), a tudíž automatická metoda nefunguje. Program je v tomto případě schopen zpracovat celistvé úseky rámu, je ale potřeba větší spolupráce uživatele.

Je nutné zadat dva krajní body ohraničující úsečku, na níž se mají rámové značky automaticky vyhledat. Za krajní bod je považován roh nebo rámová značka. Mimo to je třeba určit orientaci této úsečky na mapě. Pro orientaci „horní" se berou jako hodnoty krajních bodů hodnoty z polí roh1 a roh2, pro „pravá" z roh2 a roh3, pro „dolní" z roh3 a roh4 a pro „levá" z roh4 a roh1.

5.4 Rozhraní programu



Menu výběr rohu

Toto menu slouží k výběru rohu, pro který je souřadnice momentálně zadávána. Je počítáno s tím, že se rohy budou zadávat po směru hodinových ručiček počínaje levým horním rohem. Vzhledem k tomu, že není třeba se trefit přesně do bodu rohu, obvykle stačí jediný pokus na jeho zadání. Proto jsem zvolila automatickou změnu hodnoty v tomto menu. Ta se automaticky přehodí na další roh v pořadí po každém kliknutí do oblasti mapy. Pro případné opravy je možno hodnotu menu měnit ručně na libovolnou.

Souřadnice

Tato pole zobrazují hodnoty zadané uživatelem jako souřadnice rohů mapového rámu. Umožňují případnou kontrolu zadaných údajů.

Dpi

Toto pole slouží k zadání dpi. Tato informace je nezbytná při všech módech zpracování mapy. Přednastavenou hodnotou je 400 dpi, což je nejčastěji používané rozlišení.

Zoom

Menu zoom slouží ke změně velikosti zobrazení mapy. K dispozici je reálná, poloviční a čtvrtinová velikost. Přednastavenou hodnotou je čtvrtinová velikost. Při tomto zobrazení není detekce polohy tak přesná jako u reálné velikosti, ale u přibližného zadání je postačující. U přesného zadání rohů je vhodnější použití reálné velikosti.

Tlačítko Spustit

Toto tlačítko spouští zpracování mapy. Pokud se ukáže, že zpracování mapy neproběhlo korektně, například bylo špatně zvoleno dpi, stačí změnit parametry a tlačítkem zpracování opětovně spustit.

Tlačítka posuvu

Tato tlačítka slouží k posuvu zobrazovaného výřezu mapy.

Výběr přibližného zadaní

Zaškrtnutím této položky je vybrán mód přibližného zadání rohů.

Výběr ručního zadání

Zaškrtnutím této položky se vybere mód přesného zadání rohů.

Výběr zpracování po úsečkách

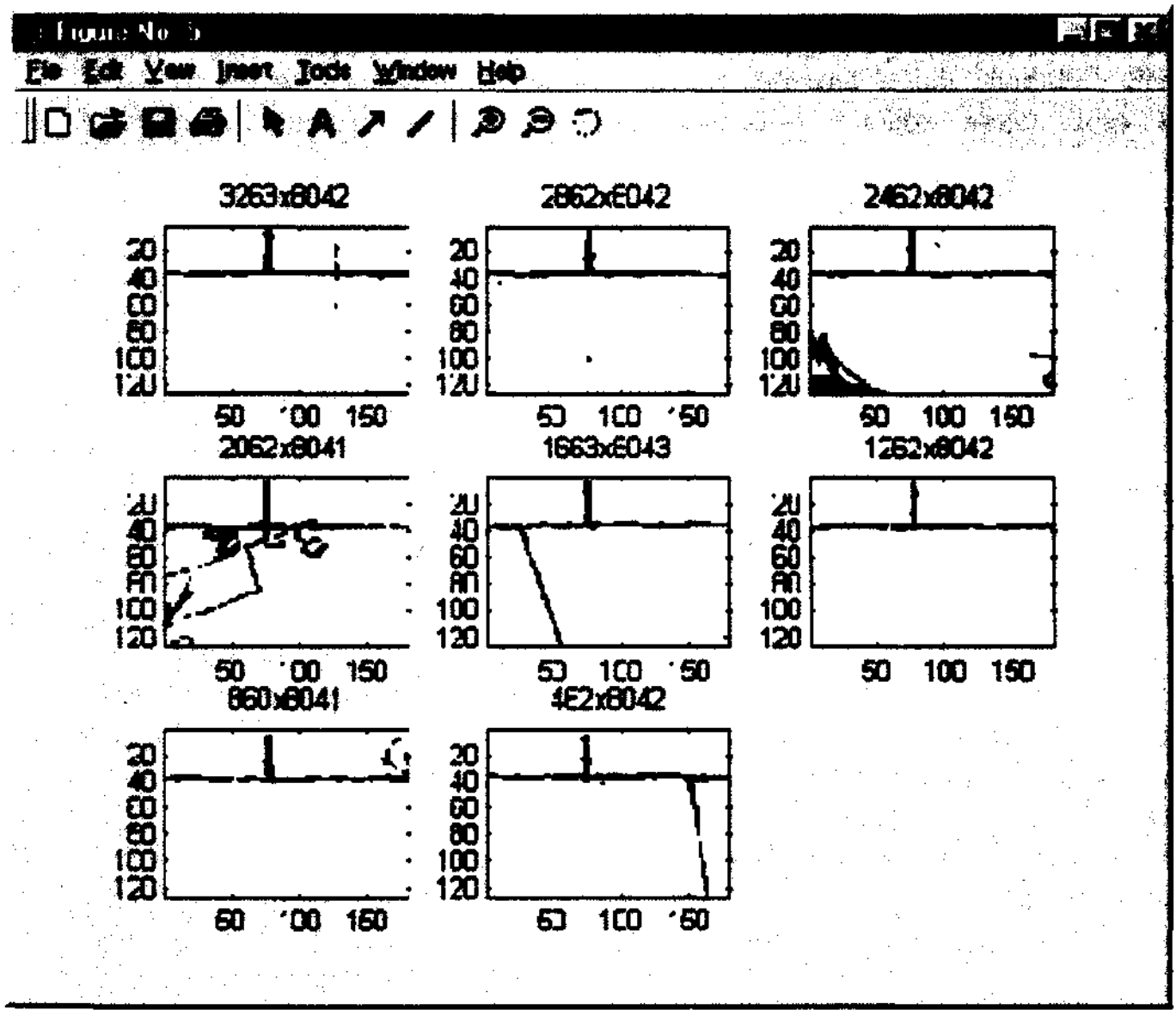
Zaškrtnutím této položky se vybere mód zpracování po úsečkách.

Menu orientace

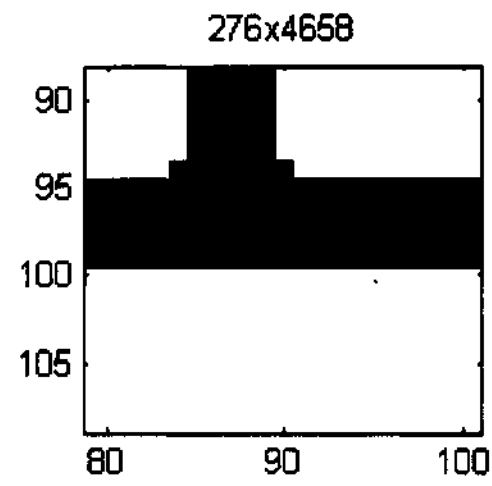
V tomto menu se volí orientace rozpoznávané hrany rámu v módu zpracování po úsečkách.

5.5 Výstup programu

Po zpracování mapy vypíše program do adresáře obsahujícího mapu textový soubor s názvem 'název souboru mapy'.txt obsahující seznam souřadnic rámových značek mapy. V módu zpracování po úsečkách se do souboru ukládají výsledky ze všech zpracovaných úseček od spuštění programu až po jeho ukončení. Pro vizuální kontrolu program vykreslí čtyři okna s obrázky všech prohledávaných oblastí (v módu zpracování po úsečkách jedno okno). V každém okně je červenou barvou zvýrazněn bod nalezený jako správná poloha rámové značky a zeleně jsou označení všichni ostatní kandidáti, ze kterých se správná poloha rámové značky vybírala.



Obrázek 5.1: Výstupní okno pro vizuální kontrolu



Obrázek 5.2: Detail rámové značky

Kapitola 6 Realizace programu MarkSearch

Program je naprogramován v prostředí MATLAB. Toto prostředí je výhodné pro práci s obrazem, protože je MATLAB optimalizován pro maticové a vektorové operace. Mnoho funkcí potřebných pro práci s obrazem je v MATLABu přímo implementováno. Přináší to ale i některé nevýhody. MATLAB neumí pracovat s bitovými proměnnými – každý pixel binární mapy je reprezentován alespoň bajtem. Díky tomu je práce s mapou velice paměťově náročná. Je třeba se vyhnout více než jedné instanci mapy v paměti.

Z důvodu velikosti dat, a také protože MATLAB při volání funkcí předává proměnné hodnotou, jsou v programu MarkSearch použity některé globální proměnné. V MATLABu ale není globální proměnná chápána stejně jako ve většině programovacích jazyků. Jedná se zde o proměnnou sdílenou pouze mezi funkcemi, ve kterých je proměnná uvedena jménem a typem *global*. Pro ostatní funkce a skripty zůstává proměnná skryta.

V MATLABu se s grafickými objekty pracuje pomocí jejich „handle". Přiřazení funkce, například callback tlačítka, probíhá uvedením řetězce názvu této funkce v parametrech objektu. Díky tomu nelze předat takovéto funkci žádnou proměnnou jako parametr. Pokud je tedy třeba pracovat s objekty z jiných funkcí než inicializační, existují dvě možnosti: handle lze uložit do globální proměnné, nebo nastavit objektu unikátní tag, a pak speciální funkcí vyhledat handle objektu, který tento tag vlastní. Stejný postup je třeba použít i pro získávání hodnot uložených v okně programu v textových polích. Jako globální proměnné používám pouze ty proměnné, které jsou potřeba často a ve více funkcích, abych nemusela opakovaně volat zjišťování hodnoty. S grafickým objekty pracuji převážně pomocí tagů.

Nejdůležitější funkce použitá v tomto programu je **Filter2,** což je implementace obrazové korelace v MATLAB. Tato funkce má více módů, na kterých závisí velikost výstupní matice. Používám výchozí mód, který odpovídá popisu korelace z kapitoly 3.1, tedy výstup je stejně velký jako vstupní obraz.

Funkci **Imread** používám pro načtení grafického souboru do programu. Tato funkce podporuje mnoho grafických formátů, avšak implementace jejich podpory se liší. Podpora některých formátů je naprogramována v jazyku programu MATLAB, některé jsou implementovány stejně jako základní sada funkcí MATLAB v jazyku C a některé formáty se načítají pomocí externích modulů naprogramovaných v jazyku Java. U některých formátů se volá na načtenou matici rotace, což MATLAB realizuje alokací další proměnné stejné velikosti jako mapa. Z těchto důvodů a díky tomu, že je nutno pracovat s extrémně velkými daty, je vhodné používat formát PCX. Je implementován v C (díky tomu je načítání rychlé) a neprovádí rotaci načtené matice.

V této kapitole bude použita následující konvence: funkce jsou uvedeny **tučně,** systémové funkce programu MATLAB začínají velkým písmenem; proměnné jsou uvedeny *kurzívou.*

6.1 Řešení dílčích problémů

V této kapitole se pokusím vysvětlit řešení jednotlivých problémů, na které jsem při tvorbě tohoto programu narazila. Tato kapitola vznikla, protože pro budoucí použití bude metoda naprogramována někým jiným v jiném jazyku, než je MATLAB. Proto považuji za důležité, aby bylo dobře zdokumentováno, jaká dílčí řešení byla vybrána a proč.

Grafické rozhraní

Grafické rozhraní programu je realizováno funkcí **gui**. Ta inicializuje okno(figure) a všechny ovládací prvky a nastavuje pro ně obslužné funkce.
Mapa je touto funkcí načtena do globální proměnné *source_img_matrix*. Její výřez je zobrazen při startu rozhraní ve čtvrtinovém zmenšení do připraveného objektu axes. Funkce má tři odlišné módy, první slouží k inicializaci rozhraní při startu programu, další dva reinicializují některé funkce a nastavení po návratu z některé z obslužných funkcí.

Zoom

Změna zoomu je realizována funkcí **zoomchng**. Ta je obslužnou funkcí (callbackem) popup menu zoom. Volá se tedy při změně hodnoty tohoto menu. Funkce provede změnu globální proměnné *skip*. Samotný zoom provede funkce **Imagesc,** která zajistí správné zobrazení výřezu o velikosti *skip* x *skip* do objektu axes, který má pevnou velikost 558x558 pixelů. *Skip* nabývá stejné, dvojnásobné a čtyřnásobné hodnoty jako hrana axes.

Navigace v mapě

Jak již bylo zmíněno, k vykreslování mapy do objektu axes se používá funkce **Imagesc.** Té se jako parametr zadá výřez mapy, jehož velikost určuje proměnná *skip*. Pro posun mapy v prohlížecím okně programu slouží tlačítka. Tato tlačítka mají nastaven callback na příslušnou funkci **skip**_směr. Tyto funkce zajišťují zobrazení nového výřezu, jehož souřadnice začínají (ve směru daném názvem funkce) o polovinu proměnné *skip* dále. V případě, že do konce mapy směrem posunu zbývá méně nebo právě *skip/2*, posune se výřez na konec mapy a tlačítko zajišťující pohyb tímto směrem se zakáže. Na konci funkcí **skip**_směr se volá **gui** s parametrem 2 pro reinicializaci WindowButtonDownFcn na funkci **on_click.**

Odečítání dpi

Dpi se zadává do editboxu s popiskem „dpi", který má tag nastaven na „edit". Přednastavenou hodnotou je 400. Dpi se opětovně získá vyhledáním tagu „edit".

Odečítání souřadnic rohů

Reakci na kliknutí do figure zajišťuje funkce **on_click**, která je nastavena na WindowButtonDownFcn. Protože není možno nastavit tuto funkci pouze pro objekt axes, musí **on_click** kontrolovat, zda bylo kliknuto do oblasti mapy, nebo někam jinam do oblasti figure. To se provádí na základě znalosti velikosti objektu axes a jeho polohy ve figure. Pokud souřadnice kliknutí padne do oblasti mapy, je přepočtena (na základě hodnoty menu zoom a polohy v rámci objektu axes) na globální souřadnici v rámci celé mapy. Následně je zobrazena v okně programu u příslušného popisku, který se volí podle menu výběru rohu. Menu výběru rohu vždy po této operaci změní hodnotu na další roh v pořadí.

Spuštění detekce

Spuštění detekce provádí funkce **call_get** nastavená na Callback tlačítka „Spustit". Ta zavolá funkci **main** v případě módu zpracování po úsečkách nebo **main2** v případě ostatních módů.

Zpracování po úsečkách

Při módu zpracování po úsečkách se provede funkce **main**, která podle hodnoty v menu orientace vezme jako krajní body hodnoty z příslušných polí souřadnic. Na jejich základě vyřízne oblast začínající a končící těmito body o výšce 141 nebo nejvyšší možnou. Změní její orientaci do polohy horní hrany rámu a zavolá na ni funkci **side**, která slouží ke zpracování jednotlivých hran rámu.

Zpracování celého rámu

Při zpracování celého rámu se volá funkce **main2.** Ta provede v případě plně automatické detekce volání funkce **edges** pro zjištění přibližné polohy rohů. Pro módy přibližného ručního zadání a plné automatické detekce provede funkce přesné určení rohů. Na základě přesné polohy rohů vyřízne postupně oblasti kolem všech čtyř hran rámu o výšce 141 pixelů, začínající jedním rohem a končící druhým. Jednotlivé hrany rámu změní do orientace horní hrany a zavolá na ně funkci **side**.

Detekce rámu

Prvním krokem ve zcela automatickém zpracování je přibližná detekce hran rámu. Tu realizuje funkce **edges.** Ta má za úkol sečíst mapu vždy do jednoho rozměru a nadetekovat dvě maxima pro každý rozměr. Součet do jednoho rozměru provádí funkce **Sum.** Pokud bych nenarážela na paměťové problémy, provedlo by se **Sum** pro každý rozměr zvlášť. Výsledné vektory by se rozdělily na polovinu a v každé polovině by se nalezlo maximum. Protože funkce **Sum** selhává na nedostatek paměti, bylo třeba výše uvedený postup pozměnit. **Sum** se aplikuje v cyklu na posunující se pruh o šířce 1000 prvků, dokud se koncem pruhu nepřesáhne polovina hrany mapy. Při tom se uchovává souřadnice maxima přes celý tento cyklus. Cyklus se provede pro oba rozměry od obou konců mapy, tedy čtyřikrát.

Zpracování po úsečkác

Při módu zpracování po úsečkách se provede funkce **main**, která podle hodnoty v menu orientace vezme jako krajní body hodnoty z příslušných polí souřadnic. Na jejich základě vyřízne oblast začínající a končící těmito body o výšce 141 nebo nejvyšší možnou. Změní její orientaci do polohy horní hrany rámu a zavolá na ni funkci **side**, která slouží ke zpracování jednotlivých hran rámu.

Zpracování celého rámu

Při zpracování celého rámu se volá vždy funkce **main2.** Poté, co zjistí přesné polohy rohů, provede vyříznutí příslušných pruhů, začínajících jedním rohem a končících druhým, o výšce 141 nebo největší možné, pokud je okraj mapy blíže. Na každý z pruhů je zavolána funkce **side** poté, co je pruh zorientován jako horní linka.

Detekce přesné polohy rohů

Nezávisle na způsobu zjištění přibližné polohy se provede detekce přesné polohy rohů rámu pomocí obrazové korelace. Toto realizuje funkce **main2.** Ta zkontroluje, zda nebyla zaškrtnuta volba přesného zadání rohů. Pokud tomu tak není, vyřízne pro každou přibližnou souřadnici rohu oblast 301x301 prvků se středem v tomto bodu, zavolá se funkce **makemask** pro vygenerování masky pro obrazovou korelaci a provede se funkce **Filter2**, která provádí obrazovou korelaci. Pro každý roh se maska před korelací rotuje, aby souhlasila orientace s rozpoznávaným rohem. Funkce **Filter2** vydá bod, který je přesnou polohou rohu.

Vyhledání rámových značek

Pro vyhledání rámových značek se vždy volá funkce **side**. Ta předpokládá vstup v podobě výřezu obsahujícího hranu rámu, ohraničenou z obou konců rohy nebo koncovými body (v módu zpracování po úsečkách), v orientaci horní hrany rámu. Ten je uložen v proměnné *stripe.* Pro vygenerování masky se volá **makemask_t(**81,81**).** Na základě proměnné *step* (původní vzdálenosti mezi značkami přepočítané na pixely) se skáče po *stripe* od začátku do konce. Začíná se jeden *step* od začátku *stripe.* Vyřízne se oblast o výšce *stripe*, středem v *step* a šířce 181. Pro detekci značky se volá **Filter2** s parametrem vyříznuté oblasti a vygenerované masky. Na základě výsledku se poupraví skok na další předpokládanou pozici a vše se opakuje, dokud není do konce *stripe* méně nebo právě *step.* Každá oblast zpracovaná funkcí **Filter2** se vykreslí na výstup s vyznačením všech bodů, ve kterých bylo označeno maximum, a bodem, který byl vybrán jako finální poloha rámové značky. Ta se spočítá jako zaokrouhlení souřadnic těžiště oblasti maxim s tou zvláštností, že pro hodnotu „končící na 0.5" se souřadnice zaokrouhluje vždy směrem „dolů". Do textového souboru se vypíše pro každou prohledanou oblast jen jeden nalezený bod přepočítaný do globálních souřadnic mapy.

6.2 Seznam funkcí a významných proměnných programu MarkSearch

Každá funkce je samostatným .m-souborem se stejným názvem jako funkce, kterou obsahuje. Program MarkSearch se sestává z následujících souborů-funkcí:

Funkce: marksearch(*path*)
Parametry:
path = cesta k souboru mapy
Účel:
Spouštěcí funkce.

Funkce: gui(*image,mode*)
Parametry:
image cesta k souboru mapy
mode mód volání funkce
Významné proměnné:
global source_img_matrix matice mapy
global heigth_of_picture výška mapy
global width_of_picture šířka mapy
global skip velikost hrany zobrazovaného výřezu mapy v okně programu
global begin_x x-ová souřadnice (prvek matice) počátku zobrazovaného výřezu mapy
global begin_y y-ová souřadnice (prvek matice) počátku zobrazovaného výřezu mapy
global end_x x-ová souřadnice (prvek matice) konce zobrazovaného výřezu mapy
global end_y y-ová souřadnice (prvek matice) počátku zobrazovaného výřezu mapy
global lst_h handle menu zoom
global fig_h handle figure
global popup_h handle menu výběru rohu
global a_h handle objektu axes
Účel:
Funkce inicializující grafické rozhraní. V módu 1 vytvoří okno programu, v módu 2 reinicializuje funkci **WindowButtonDownFcn** reagující na stisk tlačítka, v módu 3 nastavuje po změně zoomu prohlížecí okno programu do výchozího stavu.

Funkce: skip_down
skip_left
skip_right
skip_up
Významné proměnné:
disable určuje, zda se po skoku ve směru pohybu dosáhlo konce mapy a tlačítko pro skok tímto směrem má být zakázáno
Účel:
Tyto funkce obsluhují posun zobrazovaného výřezu mapy. Jsou nastaveny jako **Callback** tlačítek pro pohyb po mapě. Mění globální proměnné *begin_x*, *begin_y*, *end_x* a *end_y*, do kterých ukládají aktuální pozici výřezu v mapě.

Funkce: zoomchng
Účel:
Je **Callback** menu zoom. Reaguje na změnu hodnoty. Podle ní nastaví hodnotu proměnné *skip*. Nakonec volá **gui** v módu 3 pro nastavení výchozího stavu zobrazení.

Funkce: on_click
Významné proměnné:
coor souřadnice kliknutí, jak je uložena v *Currentpoint* proměnné figure
zoom aktuální hodnota v menu zoom
Účel:
Tato funkce je nastavena jako **WindowButtonDownFcn** objektu figure, tedy okna programu. Ve funkci se otestuje, zda kliknutí myši padlo do oblasti výřezu mapy. Pokud se tak stalo, přepočítá se lokální poloha kliknutí do výřezu na globální souřadnici v rámci celé mapy a uloží do pole příslušného rohu. Roh je vybrán podle hodnoty v popup menu s handlem *popup_h*.

Funkce: *X* **= makemask(***size_x, size_y***)**
Významné proměnné:
size_x šířka masky
size_y výška masky
X výstupní matice masky
Účel:
Generuje masku pro detekci rohů o vstupních rozměrech.

Funkce: *X* **= makemask_t(***size_x, size_y***)**
Významné proměnné:
size_x šířka masky
size_y výška masky
X výstupní matice masky
Účel:
Generuje masku pro detekci rámových značek o vstupních rozměrech.

Funkce: call_get
Účel:
Tato funkce spouští samotné zpracování mapy. V případě zpracování po úsečkách volá funkci **main**, v ostatních **main2**.

Funkce: main
Významné proměnné:
dpi dpi zpracovávané mapy
orientation orientace zpracovávané úsečky
corner1[x,y] počáteční bod zpracovávané úsečky
corner2[x,y] koncový bod zpracovávané úsečky
stripe výřez pro funkci **side**, která hledá rámové značky
sheight výška vyřezávaného pruhu směrem od středu k okraji mapy
sheight2 výška vyřezávaného pruhu směrem od středu dovnitř mapy
Účel:
Tato funkce zpracovává jednotlivé úsečky rámu, ohraničené dvěma krajními body. Podle orientace se z příslušných polí vyberou hodnoty krajních bodů zadaných uživatelem, uloží se do proměnných *corner1* a *corner2* a použijí se k vyříznutí *stripe*. Výšku vyřezávané

oblasti určují proměnné sheight a sheight2. Pokud je dostatečně velký okraj, mají obě hodnotu 70, pokud ne, *sheight* nabývá nejvyšší možné.

Funkce: main2
Významné proměnné:
corner1_coordinates[x,y], corner2_coordinates[x,y], corner3_coordinates[x,y], corner4_coordinates[x,y] přibližné souřadnice rohů rámu
corner1[x,y], corner2[x,y], corner3[x,y], corner4[x,y] přesné souřadnice rohů rámu
msk maska pro detekci rohů
corner_diff maximální povolený rozdíl mezi stejnými souřadnicemi dvou sousedních rohů
stripe výřez pro funkci **side**, která hledá rámové značky
sheight výška vyřezávaného pruhu směrem od středu k okraji mapy
sheight2 výška vyřezávaného pruhu směrem od středu dovnitř mapy
Účel:
Tato funkce zpracovává celý rám. Zvolí podle módu zadání rohů buď přímý přepis uživatelem zadaných hodnot do proměnných *corner** (mód přesného zadání rohů), nebo zadání hodnot do *corner*_coordinates*. Jejich hodnota se bere z uživatelem zadaných hodnot v případě módu přibližného zadání, při výchozím nastavení – plné automatické detekci – se volá funkce **edges**. Pokud se nejedná o mód přesného zadání rohů, provede funkce volání **makemask**(301,301) pro vygenerování masky. Poté se volá **Filter2 postupně** na všechny čtyři vytipované oblasti obsahující rohy rámu. Maska se rotuje vždy do souhlasné orientace s hledaným rohem. Po stanovení hodnot *corner** se vyříznou postupně čtyři pruhy začínající jedním rohem a končící druhým. Každý z nich je zorientován do pozice horní oblasti a na každý je zavolána funkce **side** pro jejich zpracování.

Funkce: edges
Významné proměnné:
width šířka pruhu, přes který se sčítá
Účel:
Provádí automatickou detekci přibližné polohy rámu. Tvoří ji čtyři cykly, které provádějí sčítání od každého okraje mapy směrem do středu. Provádí se tak v pruzích o šířce 1000 pixelů, a to z důvodů paměťové náročnosti. Samotné sečtení provádí funkce **Sum.**

Funkce: side(*orient, r_x, r_y ,slength , dpi, stripe, sheigth, sheigth2*)
Významné proměnné:
orient orientace pruhu, nutné pro přepočet do globálních souřadnic
r_x, r_y globální souřadnice levého horního rohu oblasti předávané do funkce
slength délka oblasti
stripe předávaná oblast
step skok o vídeňský palec přepočítaný na pixely
Účel:
Zpracuje oblast, která obsahuje jednu hranu rámu a je ohraničena konci hrany v orientaci horní hrany. Pohybuje se po oblasti po skocích o velikosti „*step* = round((dpi/2.54)*2.634)“ a na daných pozicích vyřízne výřez o šířce 181 pixelů. Na ten volá **Filter2**, který vydá polohu značky. Před dalším skokem se hodnota skoku zkoriguje o zjištěný rozdíl mezi reálnou polohou značky a předpokládanou polohou značky založenou na *step*. Každá prohledávaná oblast se vykreslí do figure. Podle hodnoty *orient* a *r_x,r_y* se přepočtou polohy zjištěných značek do globálních souřadnic a uloží do souboru.

6.3 Schéma volání funkcí

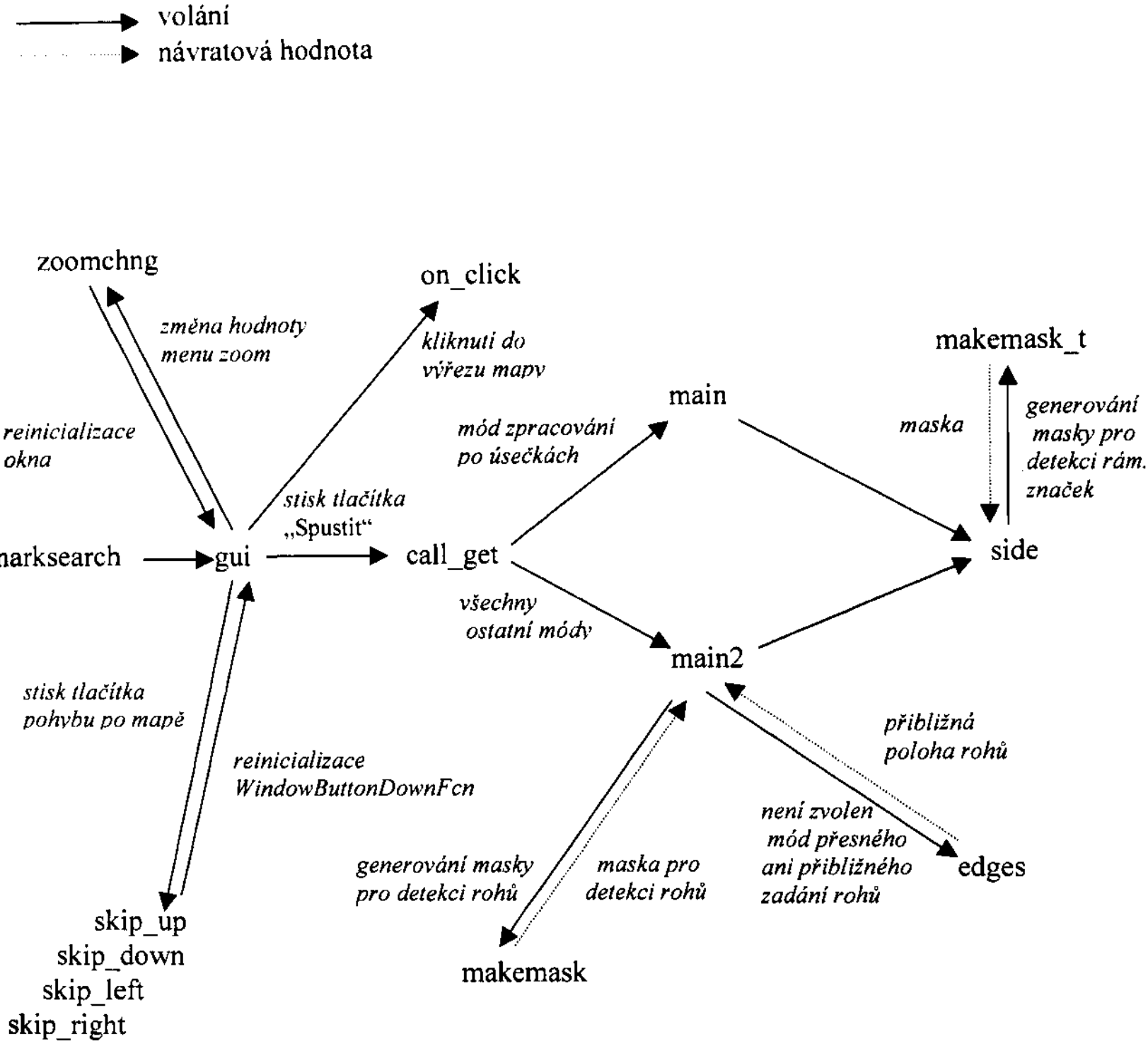


Schéma 6.1: Schéma volání funkcí a předávání dat

Kapitola 7 Závěr

Cílem práce bylo vyvinout metodu, která nahradí doposud ručně prováděné odečty rámových značek na sáhových katastrálních mapách. Účelem bylo urychlit a zjednodušit práci a hlavní důraz byl kladen na co největší úspěšnost detekce a minimalizaci práce uživatele.

Jednalo se o zcela nový problém, jehož řešením se doposud nikdo nezabýval. Neměla jsem k dispozici ani žádnou literaturu týkající se přímo tohoto problému.

Podařilo se mi vyvinout metodu pro všechny známé typy sáhových katastrálních map. Základem metody je obrazová korelace, která používá masku se speciální heuristickou úpravou. Díky ní se mi podařilo dosáhnout 98% úspěšnost v detekci rámových značek na nejstarší skupině map a téměř 100% úspěšnosti na ostatních katastrálních mapách. Metoda založená pouze na obrazové korelaci vyžadovala spolupráci uživatele, který musel zadat přibližnou polohu rohu mapového rámu. Toto jsem zautomatizovala pomocí metody, která součtem mapy do jednoho rozměru nalezne přibližnou polohu rámu. Díky tomu je možné zpracovat až 99% map zcela automaticky.

Práce obsahuje aplikaci pro zpracování všech typu sáhových katastrálních map vyvinutou v prostředí MATLAB.

Literatura

[1] Pratt W. K.: Digital Image Processing (3rd ed.), John Wiley, New York, 2001

[2] Gonzales R. C., Woods R. E., Digital Image Processing (2nd ed.), Prentice Hall, 2002

[3] Devijver Pierre A.; Kittler Josef: Pattern Recognition: A Statistical Approach

[4] Schlesinger, Michail I. - Hlaváč Václav: Deset přednášek z teorie statistického a strukturního rozpoznávání ČVUT Praha, 1999

[5] Young, Tzay Y.: Handbook of Pattern Recognition and Image Processing, Academic Press 1994

[6] Vybrané články z časopisu Image and Vision Computing, Computer Graphics and Image Processing.

[7] http://slon.fsv.cvut.cz/~soukup/kontidel.pdf

[8] Karel Zaplatílek – Bohuslav Doňar: MATLAB - tvorba uživatelských aplikací, BEN 2004

Dodatek A – Obsah CD

\namerena_data\data_pro_presnost data pro výpočet přesnosti metody
\namerena_data\data_sum data obou testu odolnosti vůči šumu
\program_marksearch program pro zpracování map
\syntetické_mapy syntetická testovací data
\utility\manualni_odecet utilita pro manuální odečet značek
\utility\vypocet_presnosti program pro výpočet statistických vzorců
\vzorova_data vzorek map z každé kvalitativní skupiny