

Západočeská univerzita v Plzni

Fakulta aplikovaných věd

Katedra matematiky

Diplomová práce

Automatické hodnocení geometrické přesnosti skenerů

Vložené zadání

Prohlášení

Předkládám tímto k posouzení a obhajobě diplomovou práci zpracovanou na závěr studia na Fakultě aplikovaných věd Západočeské univerzity v Plzni.

Prohlašuji, že jsem diplomovou práci vypracovala samostatně a výhradně s použitím zdrojů, jejichž úplný seznam je její součástí.

V Plzni dne 29. května 2013

.....

Poděkování

Na tomto místě bych chtěla poděkovat Ing. Radku Fialovi, Ph. D., vedoucímu diplomové práce, především za poskytnuté rady a věcné připomínky, materiály, věnovaný čas a zejména za zařízení zhotovení nových testovacích předloh a také za velkou dávku trpělivosti.

Děkuji také pracovníkům Zeměměřického úřadu, panu Ing. Gašparu Péchymu a paní Ivetě Kvasnicové, a pracovníkům Katastrálního úřadu pro Plzeňský kraj, panu Richardu Kcholovi a panu Ing. Stanislavu Žofkovi, za poskytnuté informace a za umožnění naskenování testovacích předloh.

Dále bych chtěla poděkovat celé své rodině za mnoho odpolední strávených hlídání dcery, za velkou podporu a pomoc po celou dobu studia. Velký dík také patří především manželovi za podnětné nápady, věcné připomínky a zejména za pomoc s kódem aplikace, především pak za pomoc s optimalizací algoritmů a za drobné korektury textu.

Abstrakt

Diplomová práce popisuje metodiku testování geometrické přesnosti skenerů dle Pokynů č. 32 ČÚZK. Zabývá se popisem algoritmů použitých pro automatickou analýzu skenovaného obrazu testovací předlohy. Algoritmy byly implementovány v aplikaci (v programovacím jazyce JAVA), jejímž výstupem jsou charakteristiky geometrické přesnosti skeneru. Práce se dále zabývá návrhem nové testovací předlohy a zhodnocením její použitelnosti. Teoretická část práce popisuje některé metody zpracování digitalizovaného obrazu. Na základních principech některých z popsáných metod zpracování obrazu jsou založeny použité algoritmy.

Klíčová slova

Analýza, aplikace, atest, geometrická přesnost, obraz, obrazová data, skenování, testovací předloha, testování přesnosti.

Abstract

Diploma thesis describes methodology of the testing geometric accuracy of scanners according to the Directive no. 32 ČÚZK. Thesis deals with describing algorithms, which are used for automatic analysis of the scanned image of the testing pattern. Algorithms were implemented in an application (which was written in the JAVA programming language). Outputs from application are characteristics of geometric accuracy of a scanner. Furthermore thesis deals with a design of a new testing pattern and discusses its use. Theoretical part of the thesis describes some of the image processing methods. Some of the used algorithms are based on basic principles from some of described image processing methods.

Keywords

Analysis, application, atest, certification, geometric accuracy, image, image data, scan, testing pattern, accuracy testing.

Obsah

Úvod	8
1 Zpracování digitalizovaného obrazu	10
1.1 Digitalizace obrazu	11
1.1.1 Vzorkování	12
1.1.2 Kvantování	14
1.2 Předzpracování	14
1.2.1 Lokální operace předzpracování	15
1.2.2 Matematická morfologie	19
1.3 Segmentace	21
1.3.1 Prahování	22
1.3.2 Srovnávání se vzorem	23
1.4 Popis objektů	24
2 Metodika testování geometrické přesnosti skenerů	25
2.1 Skenování mapových podkladů	26
2.2 Testování skenerů	27
2.2.1 Metodika průběhu testování	30
3 Metoda nejmenších čtverců	32
3.1 Odvození řešení pro soustavu lineárních rovnic pomocí MNČ	32
3.2 Aplikace MNČ – aproximace přímkou	33
3.3 Odvození afinní transformace používané při testování + aplikace MNČ u afinní transformace	34
4 Analýza skenovaného obrazu testovací předlohy	37
4.1 Nalezení skeletu	39
4.1.1 Algoritmus nalezení skeletu	42
4.2 Nalezení středů křížků	43
4.2.1 Algoritmus nalezení středů křížků	46
4.2.2 Optimalizace algoritmů	47
4.3 Nalezení orientace naskenované testovací předlohy	49
4.4 Výpočet klíče afinní transformace, charakteristiky geometrické přesnosti	51
5 Popis aplikace	55
5.1 Načtení obrázku	59

5.1.1	Prohlížení obrázku	59
5.1.2	Přibližování a oddalování obrázku	60
5.2	Analýza obrazu předlohy	61
5.3	Export a vizualizace výsledků	64
5.3.1	Vizualizace nalezených středů průsečíků mřížky	66
6	Návrh nové testovací předlohy	68
6.1	Výroba předlohy	71
6.2	Skenování předloh	71
6.3	Zhodnocení použitelnosti předloh	74
6.4	Doporučení pro nové předlohy	76
6.4.1	Vyhodnocení jedné z navržených předloh aplikací	78
7	Závěr	81
	Seznam zdrojů	83

Úvod

Dnešní technicky zaměřená doba budí dojem, že pokud data nejsou uchovávána v digitální podobě, tak jakoby téměř ani neexistovala (obzvláště ve spojení s publikací dat na Internetu). Valná většina profesí se neobejde bez použití počítače a kupříkladu výkon státní správy bez počítačů by byl dnes zřejmě velmi obtížný. Také katastr nemovitostí se neobejde bez počítačových prostředků a digitálních dat. Část území České republiky stále ještě nemá převedeny katastrální mapy do podoby digitální vektorové mapy a na těchto územích je katastrální mapa vedena v analogové podobě. Převod analogových map do podoby rastrového obrazu probíhá jejich skenováním. Český úřad zeměměřický a katastrální vydal Pokyny č. 32 pro skenování katastrálních map a grafických operátů dřívějších pozemkových evidencí [15] a právě tyto pokyny obsahují informace, jaké parametry musí splňovat skenery aby výsledky skenování mohly být použity pro účely katastru.

V pokynech je uvedeno, jakým způsobem se testují parametry skenerů (geometrická přesnost) – je použita testovací předloha s křížky (u které známe souřadnice středů křížků) a tyto středy se ručně vektorizují na naskenovaném obraze předlohy a jsou porovnány s referenčními souřadnicemi (výsledkem jsou hodnoty geometrické přesnosti skeneru).

Cílem práce je vytvoření aplikace, která (polo)automaticky nalezne středy křížků na naskenovaném obraze předlohy a vypočte statistiky geometrické přesnosti. S tímto zároveň budou navrženy algoritmy zpracovávající předlohu.

Požadavky na aplikaci jsou následující:

- analýza naskenovaného obrazu testovací předlohy bude probíhat v reálném čase (v řádu 15-30 minut; nyní trvá ruční vektorizace jednoho naskenovaného obrazu předlohy jeden pracovní den),
- aplikace bude spouštěna přes příkazový řádek s parametry vstupu a výstupu a po dokončení analýzy se na obrazovku vypíše statistiky týkající se geometrické přesnosti,

- vstupem bude černobílý obraz naskenované předlohy,
- výstupem budou kromě statistik geometrické přesnosti i souřadnice středů křížků s jejich odchylkami od referenčních souřadnic.

Dalším cílem je navržení nové testovací předlohy a její ověření založené na zkušenostech, získaných při jejím skenování. Stávající předloha je relativně řídká, křížky jsou od sebe vzdáleny 50 mm a mají velikost 10 mm, tzn. mezi křížky se nachází „hluchý prostor“. V tomto volném prostoru neodhalíme případné nepřesnosti snímače skeneru a proto by nová předloha měla obsahovat více zájmových bodů (např. středů křížků) aby se zvýšil počet porovnávaných souřadnic a detailněji se tak popsaly chyby skeneru.

Pokud se předloha osvědčí a bude předpoklad, že by v budoucnu mohla nahradit stávající (jedinou!) testovací předlohu pak by bylo vhodné, aby aplikace uměla nalézt středy zájmových bodů i na této nové předloze.

1 Zpracování digitalizovaného obrazu

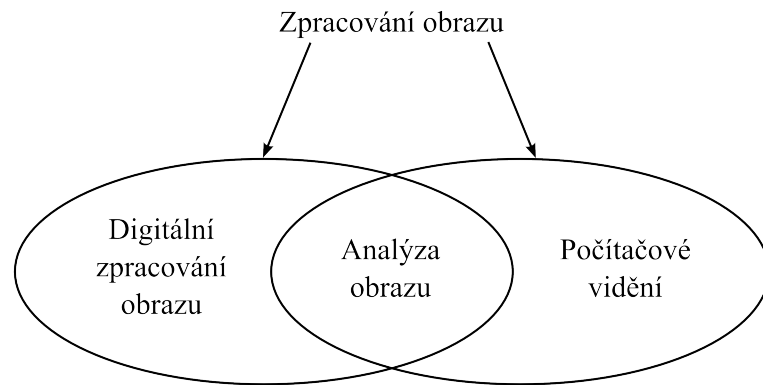
Produkty počítačové grafiky a práce s obrazem nás provázejí na každém kroku a v mnoha odvětvích jsou metody zpracování obrazu důležitým nástrojem, ať už se jedná o zpracování lékařských snímků, meteorologických snímků, obrazů pořízených při pozorování vesmíru či zpracování fotografií.

Zpracování (zkoumání) obrazu lze dělit na dvě úrovně – na digitální zpracování obrazu (tzv. nižší úroveň) a na počítačové vidění (vyšší úroveň). Počítačové vidění se snaží co nejvíce napodobit lidské vidění, snaží se o porozumění a vyhodnocení vlastností dané scény předkládané pomocí obrazu (či série obrazů). U člověka hraje při vnímání a vyhodnocování vizuální informace velkou roli inteligence a nabyté zkušenosti a právě inteligence nám umožňuje vnímání, interpretování a porozumění okolního světa. V počítačovém vidění se vytváří modely (reprezentace) reálného světa a to na základě interpretace obrazových dat, kdy je využíváno znalostí, cílů a zpětné vazby v konkrétní oblasti vidění. Obecně je počítačové vidění velice složitá úloha (ztráta informací z 3D scény při převodu do 2D, velké množství dat apod.) a využívají se techniky umělé inteligence. Nižší úroveň – čili zpracování obrazu – oproti vyšší úrovni neinterpretuje obrazová data, velmi málo využívá apriorní znalosti o obsahu obrazu a výstupem je opět obraz případně jeho jednoduché charakteristiky. Zpracování obrazu zahrnuje například kompresi obrazu, filtraci šumu či ostření obrazu. [2], [6], [10]

Společným prvkem obou úrovní zpracování je analýza obrazů, jež je zodpovědná za jejich rozvoj a napomáhá řešení problému. Zahrnuje průzkum obrazových dat pro specifickou oblast použití. Analýza zahrnuje interpretaci obrazu, která přináší další informace pro řešení dané úlohy a spadají do ní např. metody segmentace, transformace obrazu či rozpoznávání objektů. [7]

Provázanost jednotlivých disciplín zpracování obrazu je znázorněna na obr. 1.1. Průběh zpracování a rozpoznávání obrazu reálného obvykle rozdělujeme do několika kroků (záleží na jednotlivých případech zda se provedou všechny), zde je jejich výčet dle [3]:

- snímání a digitalizace obrazu
- předzpracování
- segmentace obrazu
- popis objektů
- klasifikace



Obrázek 1.1: Provázanost disciplín zpracování obrazu. Zpracováno dle [7]

1.1 Digitalizace obrazu

Obecně lze obraz chápat např. jako obraz na sítnici lidského oka, fotografii, obraz získaný kamerou či lékařskými přístroji (rentgenový snímek atd.). Pokud bychom chtěli obraz zapsat jako matematický model pak se používá spojité funkce dvou nebo tří proměnných, která se označuje jako obrazová funkce. U obrazové funkce dvou proměnných $f(x, y)$ je obraz popsán pomocí dvou plošných souřadnic. Tři proměnné jsou použity pokud se obraz mění v čase t , tzn. obrazová funkce je pak vyjádřena v následujícím tvaru: $f(x, y, t)$, případně se může jednat o objemový obraz $f = (x, y, z)$. Hodnotou obrazové funkce jsou fyzikální veličiny vyjádřené reálným číslem (teplota, vzdálenost od pozorovatele, jas, data z tomografu apod.). Nejčastější hodnotou obrazové funkce je jas a pro obraz dvou proměnných nesoucích informací o něm se používá termín intenzitní obraz.

Obraz zpracováváný na počítači je reprezentován diskretní funkcí, která má nejčastěji tvar matice. Prvky matice se nazývají obrazové elementy, neboli pixely (z anglického picture element). Pixel je prvek o konečném rozměru, je dále nedělitelný a často se můžeme setkat s pojmem bod, který se používá ve smyslu pixelu.

Snímaný obraz je vyjádřen spojitou funkcí, kterou ale pro zpracování na počítačích musíme digitalizovat do podoby konečného počtu pixelů obsahujících hodnoty jasu. Digitalizace obrazu tedy znamená převedení spojitě funkce na diskretní a probíhá ve dvou krocích – vzorkování a kvantování. [2], [3], [6], [9]

1.1.1 Vzorkování

Spojitý obraz je vzorkován v určitých intervalech do mřížky, obvykle do matice $M \times N$ bodů. Získaný vzorek odpovídá obrazovému prvku (elementu) – pixelu ve 2D digitálním obraze (ve 3D obraze se pro obrazový element používá termín voxel [6]). Při vzorkování je nutné vyřešit dvě otázky – určit interval vzorkování a vybrat vzorkovací mřížku.

U intervalu vzorkování se jedná o určení vzdálenosti mezi nejbližšími vzorkovacími body v obraze, odvíjí se od velikosti detailů v obraze (je potřeba zvolit takový interval vzorkování, abychom zachytili nejmenší detaily v obraze). Vzdálenost (krok) si označíme jako Δx a pokud si představíme, že např. v polovině kroku došlo ke změně vzorkované spojitě funkce pak tato hodnota není ve vzorcích zaznamenána. Převrácená hodnota vzdálenosti se nazývá vzorkovací frekvence f_s [9]:

$$f_s = \frac{1}{\Delta x}, \quad (1.1)$$

přičemž její jednotkou je buď počet vzorků za jednotku času [Hz] anebo na jednotku vzdálenosti [DPI].

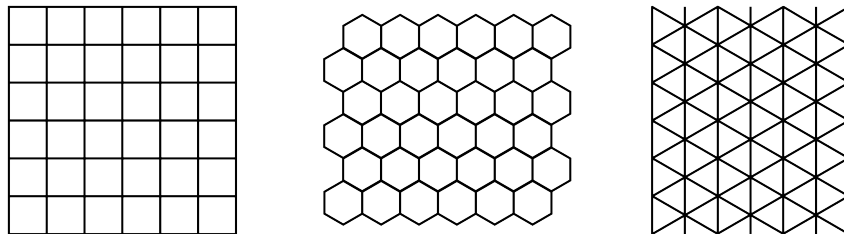
Volbu vzorkovací frekvence řeší Shannonova vzorkovací věta (též se používá termín vzorkovací teorém), která je uvedena pro jednorozměrné signály v [2], přičemž ji lze aplikovat i na dvojrozměrný signál (viz [9]). Shannonova věta zní takto: *Pokud signál $x(t)$ spojitý v čase obsahuje pouze frekvenční složky s frekvencemi menšími než*

f_{max} , pak $x(t)$ může být jednoznačně rekonstruován z posloupnosti ekvidistantních vzorků $x(n\Delta x)$, pokud vzorkovací frekvence uvedená v (1.1) je větší, než $2f_{max}$, čili

$$f_s > 2 f_{max}. \quad (1.2)$$

Vzorkovací teorém říká, že vzorkovací frekvence musí být alespoň dvakrát větší než nejvyšší frekvence v signálu. U obrazového signálu tak pak pro interval vzorkování Δx platí, že se volí jako menší nebo rovný polovině rozměru nejmenších detailů v obrazu. Nejmenšími detaily mohou být například tenké čáry či body. Při vzorkování ve skutečnosti požadujeme, aby tloušťka čáry (detailu) byla 4 - 5 pixelů [10], tzn. aby byl obraz vzorkován s ještě vyšší frekvencí než jak je uvedeno v teorému. Samozřejmě, že s vyšší vzorkovací frekvencí roste počet pixelů (výsledných vzorků), tj. narůstá tak rozlišení obrazu a paměťová náročnost reprezentace [9].

Vzorkovací mřížka určuje rozložení bodů v ploše při vzorkování. Používá se mřížka složená z pravidelných mnohoúhelníků, obvykle se volí mezi třemi mnohoúhelníky. Těmito třemi obvyklými pravidelnými mnohoúhelníky mohou být čtverce, pravidelné šestiúhelníky a rovnostranné trojúhelníky (ukázka vzorkovacích mřížek na obr. 1.1.1). [2] Čtvercová mřížka je nejčastěji používaná, ovšem vyvstává u ní problém s definicí vzdálenosti jednotlivých bodů, který např. u hexagonální mřížky odpadá (středů sousedících šestiúhelníků jsou od sebe konstantně vzdáleny). Pravidelná hexagonální a trojúhelníková mřížka není příliš rozšířená reprezentace, nicméně pro digitální model terénu či povrchu se používá nepravidelné trojúhelníkové sítě.



Obrázek 1.1.1: Vzorkovací mřížky

1.1.2 Kvantování

Spojité obrazové funkce se kromě vzorkování ještě kvantuje a to tak, že se kvantuje spojitá jasová úroveň každého vzorku do k intervalů a jasová funkce tím nabývá celočíselných hodnot. Většinou se používá kvantování do k stejných intervalů (konstantní délka intervalů). Jestliže se pro vyjádření jasu pixelu používá b bitů, pak je počet jasových úrovní $k = 2^b$. U barevných obrazů se obvykle používá 8 bitů na pixel pro jeden kanál (červený, zelený a modrý kanál), mohou se vyskytnout i systémy s jinými počty bitů (12, 6, 4 či jedno bitová reprezentace bodu u binárních obrazů).

Hodnota v intervalu se obvykle určuje jako průměr celého intervalu (lze ji určit ještě např. jako medián, vážený průměr intervalu atd.), tzn. nahrazení množiny hodnot jednou hodnotou má za následek ztrátu informací a tato ztráta se označuje jako kvantizační chyba. Projevuje se jako náhlý skok barvy, na což je lidské oko citlivé a působí rušivě. Více se projevuje u uniformního kvantování (tj. stejné intervaly), přičemž omezit tuto chybu umožňuje neuniformní kvantování (proměnná délka intervalu). Lidské oko vnímá falešné obrysy (náhlý skok v barvě) pokud je počet úrovní jasu menší než 50 (tj. oko je schopno rozlišit okolo 50 úrovní jasu – počet kvantovacích úrovní by měl vyšší než tato hodnota). [2], [6], [9]

1.2 Předzpracování

Operace s obrazy na nízké úrovni abstrakce se nazývají předzpracování. Vstupem je intenzitní obraz $f(x, y)$ a výstupem je taktéž intenzitní obraz, který budeme značit $g(x, y)$. Předzpracování nepřináší nárůst informace kterou obraz obsahuje, naopak informace klesá a proto se také můžeme setkat s tvrzením: „*Nejlepší předzpracování je žádné předzpracování.*“ [6], [10]

Cílem předzpracování může být odstranění šumu, potlačení zkreslení či zvýraznění některých rysů obrazu, které jsou důležité pro další zpracování. Metody předzpracování se dělí do čtyř kategorií podle velikosti okolí zpracovávaného bodu, pro který se vypočítává nový jas. Jedná se o:

- bodové jasové transformace,
- geometrické transformace,
- lokální operace (filtrace, ostření),
- frekvenční analýzu (do ní patří restaurace obrazu). [10]

Do metod předzpracování se řadí i matematická morfologie, jež je relativně samostatnou částí analýzy obrazu. V následujících dvou kapitolách budou popsány principy lokálních operací předzpracování a matematická morfologie.

1.2.1 Lokální operace předzpracování

Lokální operace používají ke zkoumání bodu jeho malé okolí (nejčastěji se jedná o posun čtvercové matice – konvoluční masky – po jednotlivých pixelech obrazu a nová hodnota jasu pixelu se počítá jako funkce okolních pixelů). Lokální operace předzpracování se dělí podle účelu zpracování obrazu na dvě skupiny – vyhlazování (filtraci) a detekci hran, přičemž metody mohou být lineární či nelineární.

Vyhlazování se používá k odstranění šumu a malých fluktuací v obraze. Pokud máme několik obrazů téže předlohy, použije se průměrování přes více snímků, často ale máme k dispozici jen jediný obraz a předpokládáme, že sousední pixely mají převážně stejnou nebo blízkou hodnotu jasu a hodnoty pixelů zkreslených šumem se pak opravují na základě analýzy pixelů v jejich okolí.

Lokální průměrování v daném okolí se řadí mezi základní lineární metodu filtrace. Filtrace se provede tak, že nový jas bodu je vypočten jako vážený průměr původních jasů ve zvoleném okolí. Matematický operátor vyjadřující tento postup se nazývá diskrétní konvoluce. Okolí a váhy jsou definovány konvoluční maskou. Nejčastěji se používá konvoluční maska h o velikosti 3×3 (ve vztazích v 1.3 je ukázána konvoluční maska pro průměrování a další dvě masky pro průměrování, u kterých je větší váha středového bodu případně i některých sousedních bodů). Nevýhodou průměrování je rozostření hran.

$$h = \frac{1}{9} \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix} \quad h = \frac{1}{10} \begin{bmatrix} 1 & 1 & 1 \\ 1 & 2 & 1 \\ 1 & 1 & 1 \end{bmatrix} \quad h = \frac{1}{16} \begin{bmatrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{bmatrix} \quad (1.3)$$

Problém s rozostřením hran částečně eliminují nelineární metody filtrace. Jedním ze zástupců těchto metod je průměrování používající rotující masku. Kolem okolí bodu o velikosti 5×5 rotuje malá maska o velikosti 3×3 , celkem je tedy možných 9 poloh masky (jedna poloha je se zájmovým bodem na masce uprostřed, u ostatních je pak zájmový bod na okraji masky), přičemž pro každou polohu masky se spočítá rozptyl jasů a vybere se ta maska, ve které je rozptyl jasů nejmenší.

Další metodou nelineární filtrace je filtrace mediánem. Princip výpočtu mediánu je jednoduchý – vzestupně se uspořádají hodnoty obrazové funkce v lokálním okolí a mediánem je hodnota nacházející se uprostřed (příklad určení mediánu viz 1.4). Pro snadné určení se používá okolí, jehož rozměry jsou liché (3×3 , 5×5 apod.). U filtrace mediánem je výhodou to, že není ovlivněna jednou či více vychýlenými hodnotami, nevýhodou ovšem je, že porušuje tenké čáry a ostré rohy v obraze.

$$f(i, j) = \begin{bmatrix} 50 & 46 & 53 \\ 37 & 55 & 49 \\ 71 & 40 & 44 \end{bmatrix} \quad (1.4)$$

37 40 44 46 49 50 53 55 71

Detekce hran je jednou z metod lokálního předzpracování obrazu. Hrany buď chceme zvýraznit (zaostřit) či detekovat. Hranou je takové místo v obraze, kde dochází k náhlé změně obrazové funkce (dochází k nějaké nespojitosti – nejčastěji v jase). Hrana je určena gradientem – jeho velikostí a směrem.

Ve spojitém případě se velikost gradientu pro obrazovou funkci $f(x, y)$ vypočte jako:

$$|\nabla f(x, y)| = \sqrt{\left(\frac{\partial f(x, y)}{\partial x}\right)^2 + \left(\frac{\partial f(x, y)}{\partial y}\right)^2} \quad (1.5)$$

a směr gradientu jako:

$$\psi = \operatorname{arctg} \left(\frac{\frac{\partial f(x, y)}{\partial y}}{\frac{\partial f(x, y)}{\partial x}} \right). \quad (1.6)$$

U směru gradientu je nutno určovat kvadrant dle znamének u čitatele a jmenovatele, protože funkce arctg je prostá (totéž platí i níže pro gradient v diskretním tvaru). Pro digitalizovaný obraz (a tedy diskretní případ) se parciální derivace aproximují diferencemi:

$$\begin{aligned} \Delta_x f(x, y) &= f(x, y) - f(x - 1, y), \\ \Delta_y f(x, y) &= f(x, y) - f(x, y - 1). \end{aligned} \quad (1.7)$$

Velikost a směr gradientu pro diskretní případ má pak následující podobu:

$$|\operatorname{grad} f| = \sqrt{(\Delta_x f)^2 + (\Delta_y f)^2}, \quad (1.8)$$

$$\psi = \operatorname{arctg} \left(\frac{\Delta_y f}{\Delta_x f} \right). \quad (1.9)$$

Gradientní operátory neboli hranové detektory můžeme rozdělit do tří skupin:

1. operátory aproximující derivaci diferencemi,
2. operátory pracující na základě průchodu nulou druhé derivace obrazové funkce,
3. operátory založené na srovnávání obrazové funkce s parametrickým modelem hran.

Ad 1. Aproximace derivací diferencemi

- Robertsův operátor – jeden z nejjednodušších gradientních operátorů, používá okolí o velikosti 2×2 a proto je citlivý na šum. Jeho konvoluční masky vypadají takto:

$$h_1 = \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix}, \quad h_2 = \begin{bmatrix} 0 & -1 \\ 1 & 0 \end{bmatrix}. \quad (1.10)$$

Velikost gradientu se vypočítá takto:

$$|\nabla f(x, y)| = |f(x, y) - f(x + 1, y + 1)| + |f(x, y + 1) - f(x + 1, y)|. \quad (1.11)$$

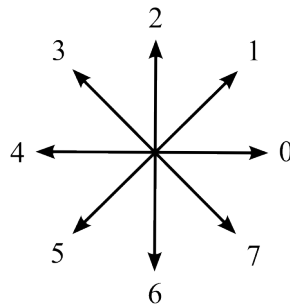
- Laplaceův operátor – aproximuje druhou derivaci, jeho označení je ∇^2 a udává pouze velikost hrany. Laplaceián je definován jako:

$$\nabla^2 f(x, y) = \frac{\partial^2 f(x, y)}{\partial x^2} + \frac{\partial^2 f(x, y)}{\partial y^2}. \quad (1.12)$$

V digitálním obraze se Laplaceián aproximuje diskrétní konvolucí. Používané konvoluční masky pro 4-okolí a 8-okolí mají následující podobu:

$$h = \begin{bmatrix} 0 & 1 & 0 \\ 1 & -4 & 1 \\ 0 & 1 & 0 \end{bmatrix}, \quad h = \begin{bmatrix} 1 & 1 & 1 \\ 1 & -8 & 1 \\ 1 & 1 & 1 \end{bmatrix}. \quad (1.13)$$

- Operátor Prewittové – aproximuje první derivaci, gradient je vypočten v osmi směrech (směry viz obrázek 1.2.1) a vybrána je ta maska, kde je maximum gradientu (a k němu příslušný směr):



Obrázek 1.2.1: Osm směrů

Ukázka tří konvolučních masek pro gradientní operátor Prewittové (ostatní se odvodí jednoduše – rotací o 45°):

$$h_0 = \begin{bmatrix} 1 & 1 & 1 \\ 0 & 0 & 0 \\ -1 & -1 & -1 \end{bmatrix}, \quad h_1 = \begin{bmatrix} 1 & 1 & 0 \\ 1 & 0 & -1 \\ 0 & -1 & -1 \end{bmatrix}, \quad h_2 = \begin{bmatrix} 1 & 0 & -1 \\ 1 & 0 & -1 \\ 1 & 0 & -1 \end{bmatrix}. \quad (1.14)$$

- Sobelův operátor – směrově závislý, aproximuje první derivaci a často se používá pro detekci horizontálních a svislých čar. Opět ukázka masek pro tři směry (další vzniknou pootočením):

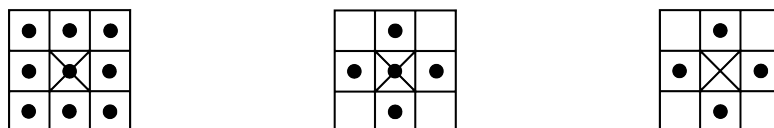
$$h_0 = \begin{bmatrix} 1 & 2 & 1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{bmatrix}, \quad h_1 = \begin{bmatrix} 2 & 1 & 0 \\ 1 & 0 & -1 \\ 0 & -1 & -2 \end{bmatrix}, \quad h_2 = \begin{bmatrix} 1 & 0 & -1 \\ 2 & 0 & -2 \\ 1 & 0 & -1 \end{bmatrix}. \quad (1.15)$$

Další metody filtrace a detekce hran je možno nalézt v [2], [6], [7] a [10] – z těchto materiálů bylo čerpáno při psaní této kapitoly (tzn. kap. 1.2.1).

1.2.2 Matematická morfologie

Matematická morfologie představuje relativně samostatnou oblast analýzy obrazu. Používá se pro předzpracování ale i pro segmentaci objektů od pozadí či popis objektů číselnými charakteristikami. Morfologie se vztahuje ke struktuře či formě objektů. Matematická morfologie se opírá o teorii bodových množin – binární obraz je množina dvojic celých čísel ve 2D eukleidovském prostoru E^2 , obraz s více úrovněmi obrazové funkce (např. šedotónový) je bodová množina v E^3 . Dále se budeme zabývat pouze binární matematickou morfologií.

Morfologická transformace je dána relací mezi obrazem (bodová množina objektů X) a obvykle menší bodovou množinou nazývanou strukturní element B . Strukturní element reprezentuje tvar, může mít jakoukoliv velikost a počátek nemusí být bodem strukturního elementu (příklady strukturního elementu se nachází na obr. 1.2.2, přičemž počátek je u každého elementu označen křížkem, tzn. u třetího strukturního elementu počátek není bodem elementu). Strukturní element se systematicky pohybuje po celém obraze a výsledek relace se zapíše do okamžitého bodu obrazu (okamžitý bod je bod obrazu, který se shoduje s počátkem strukturního elementu). Mezi základní transformace patří posunutí, dilatace, eroze, otevření a uzavření. [5], [10]

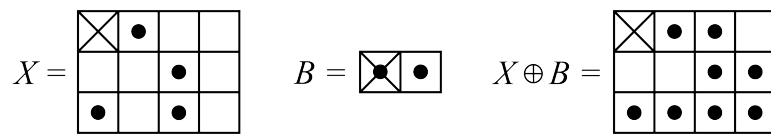


Obrázek 1.2.2: Strukturní elementy

Dilatace \oplus skládá body dvou množin pomocí vektorového součtu (Minkowského množinový součet). Objekty se rozrostou, dochází k zaplnění malých děr a spojení blízkých objektů. [6], [10]

$$X \oplus B = \{p \in E^2 : p = x + b, x \in X \text{ a } b \in B\} \quad (1.16)$$

Př.: $X = \{(0, 1), (1, 2), (2, 0), (2, 2)\}$
 $B = \{(0, 0), (0, 1)\}$
 $X \oplus B = \{(0, 1), (0, 2), (1, 2), (1, 3), (2, 0), (2, 1), (2, 2), (2, 3)\}$

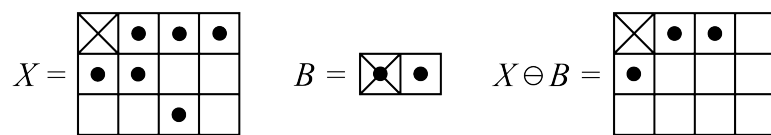


Obrázek 1.2.3: Ilustrace příkladu dilatace

Eroze \ominus skládá dvě bodové množiny s využitím rozdílu vektorů (Minkowského rozdíl). Nejedná se o inverzní transformaci k dilataci. Zmizí objekty a čáry menší než strukturální element a osamělé body. [6], [10]

$$X \ominus B = \{p \in E^2 : p = x + b \in X \text{ pro každé } b \in B\} \quad (1.17)$$

Př.: $X = \{(0, 1), (0, 2), (0, 3), (1, 0), (1, 1), (2, 2)\}$
 $B = \{(0, 0), (0, 1)\}$
 $X \ominus B = \{(0, 1), (0, 2), (1, 0)\}$



Obrázek 1.2.4: Ilustrace příkladu eroze

Skelet neboli kostra se definuje pomocí pojmu maximální kruh. Skelet $S(X)$ je množina bodů (středů kružnic), které jsou obsaženy v X a dotýkají se hranice X alespoň ve dvou bodech [10] (kruh se dotýká hranice X na dvou či více místech a již jej nelze zvětšit).

Skelet lze vytvořit pomocí dilatací a erozí, problémem je, že takovýto skelet může být z čar tlustších než jeden bod. Skelet se často nahrazuje množinou zpracovanou sekvenčním homotopickým zpracováním (sekvenční ztenčování či zesilování, kde

strukturní element B byl získán pomocí transformace tref či miň – anglicky *hit or miss*, více o sekvenčním zpracování v [6] či [10]).

1.3 Segmentace

Segmentace představuje jeden z nejdůležitějších kroků při analýze a zpracování obrazových dat. Jejím cílem je rozčlenění obrazu na části, které úzce souvisí s objekty reálného světa obsažené v obraze. Vstupem je intenzitní obraz, výstupem jsou oblasti, které mohou odpovídat objektům. Segmentace může být kompletní či částečná. Kompletní znamená, že oblasti jednoznačně korespondují s objekty ve vstupním obraze kdežto u částečné segmentace oblasti ne zcela souvisí s objekty obrazu a navíc se mohou překrývat (vytvořené oblasti jsou homogenní vzhledem k určitým vlastnostem jako je např. barva, jas či textura). Zejména při kompletní segmentaci se pokud možno co nejvíce využívá znalostí o řešeném problému jako je např. tvar objektu, poloha či vztah k ostatním objektům.

Výsledky segmentace se používají v další analýze a slouží ke změně reprezentace dat (pixely jsou organizovány do vyšších jednotek, které mají větší význam či jsou efektivnější pro další analýzu).

Segmentační metody mohou být rozděleny do tří skupin. První skupinu představují metody pracující na základě globální znalosti obrazu či jeho částech (obvykle se jedná o informace obsažené v histogramu jasu obrazu). Druhou skupinu segmentačních metod tvoří metody založené na hledání hranic a třetí skupina je založena na vytváření oblastí (metoda narůstání oblasti). [3], [5], [6], [10]

1.3.1 Prahování

Prahování je nejjednodušší a nejstarší metoda segmentace. Často se aplikuje na šedotónové obrazy, kdy chceme vytvořit z původního obrazu binární obraz. Prahování je velmi rychlá segmentační metoda proveditelná v reálném čase. Problémem je volba prahu a omezenost použití na obrazy, ve kterých jsou jasně jasově odlišitelné objekty od pozadí.

Prahování spočívá v rozdělení jasové úrovně obrazu na dvě části, kdy jedna i druhá část je nahrazena jednou hodnotou dle následujícího předpisu [3], [6], [9], [10]:

$$g(i, j) = \begin{cases} 1 & \text{pro } f(i, j) \geq T \\ 0 & \text{pro } f(i, j) < T \end{cases}, \quad (1.18)$$

kde T představuje práh (předem určenou konstantu), hodnota $g(i, j) = 1$ náleží objektům a $g(i, j) = 0$ patří pozadí (případně obráceně pokud by zájmové objekty neměly vyšší hodnotu jasu než pozadí).

Někdy nelze použít jen jedinou hodnotu prahu pro celý obraz (např. při nerovnoměrném osvětlení) a používá se modifikace prahování či se prahuje pomocí více prahů, případně se aplikuje poloprahování.

Práh lze určovat manuálně či automaticky. Lze například využít apriorní znalosti o tom, kolik procent plochy obrazu zaujímají objekty a nastavit práh právě tak, aby se tyto objekty odlišily od pozadí (aby měly jinou barvu než pozadí) – toto se nazývá procentní prahování. Většinou se práh určuje z histogramu obrazu. **Histogram** představuje vektor četností hodnot obrazové funkce v obraze (četnost bodů s daným jasem). Hodnota histogramu pro index i říká, kolik pixelů v obraze má danou intenzitu, čili získáváme představu o rozložení jasových úrovní v obraze. Obrazy s více barevnými kanály mají pro každý kanál zvláštní histogram, šedotónový obraz má jediný histogram, přičemž tmavé odstíny se nacházejí blízko nulové hodnoty a spolu s rostoucí hodnotou ve vodorovném směru roste i intenzita (až k bílé barvě) [9]. Ideálním histogramem pro prahování je bimodální histogram, kde práh určíme například jako lokální minimum mezi dvěma největšími dostatečně vzdálenými lokálními maximy [10] (u bimodálního ideálního histogramu odpovídá jeden vrchol objektům a druhý pozadí).

1.3.2 Srovnávání se vzorem

Srovnávání se vzorem patří k metodám segmentace, u nichž předem známe nějaké informace o obrazu. Úkolem je nalézt známé objekty či např. texturu nebo geometrický útvar, přičemž obvykle vzory ke srovnání mají podobu obrazu. Srovnávání se vzorem je vhodné i pro srovnávání stereoskopických snímků scény, kdy srovnáváme dva snímky z různého místa a hledáme stejný objekt či může být použito pro dynamické snímky, u kterých se určuje relativní pohyb objektů (např. snímky z kamery sledující provoz, případně snímky pohybujících se mraků).

Přesnou kopii hledaného vzoru nemůžeme v obraze nalézat protože obraz může být zašuměn, zkreslen či jinak poškozen. Proto nehledáme absolutní souhlas ale hledáme jen maximální míru souhlasu podle nějakého kritéria (maximální korelaci mezi vzorem a obrazem). Příklady kritérií používajících jako míru souhlasu vzájemnou korelaci [6], [10]:

$$\begin{aligned}
 C_1(u, v) &= \frac{1}{\max_{(i, j) \in V} |f(i + u, j + v) - h(i, j)|}, \\
 C_2(u, v) &= \frac{1}{\sum_{(i, j) \in V} |f(i + u, j + v) - h(i, j)|}, \\
 C_3(u, v) &= \frac{1}{\sum_{(i, j) \in V} (f(i + u, j + v) - h(i, j))^2},
 \end{aligned} \tag{1.19}$$

kde f představuje zpracováváný obraz u kterého testujeme souhlas se vzorem h umístěným v poloze (u, v) na množině všech pixelů zpracováváného obrazu. Kritéria popisují míru shody vzoru s danou částí obrazu pro každou polohou vzoru h v obraze f . Lokální maxima, která jsou větší než určené kritérium reprezentují polohu v obraze. Problémem je, pokud se vzor v obraze vyskytuje natočený, s jinou velikostí nebo zkreslením – museli bychom testovat míru souhlasu pro všechna možná natočení, zkreslení atd [10].

1.4 Popis objektů

Cílem popisu objektů je určit buď číselný vektor příznaků či nečíselný syntaktický popis. Nečíselný syntaktický popis objektů popisuje relace mezi objekty a jejich tvarové vlastnosti. Patří sem např. popis tvaru na základě hranice oblastí pomocí Freemanova řetězového kódu [6], [10]. Číselný vektor příznaků vyjadřuje určité vlastnosti objektů, většinou metrické povahy, příznaků je celá řada, některé si krátce představíme [3], [6], [10]:

- Velikost oblasti (objektu) představuje počet pixelů, které jsou obsaženy v oblasti. Jedná se přirozenou a nejjednodušší vlastnost, z geometrického hlediska je tomuto popisu nejbližší termín plocha oblasti.
- Eulerovo číslo vyjadřuje počet souvislých částí oblastí, od kterých jsou odečteny díry (díra = obklopená ze všech stran oblastí).
- Výstřednost je charakteristika poměru mezi délkou nejdelší tětivy A a na ní kolmé nejdelší tětivy B (poměr na sebe kolmých nejdelších tětiv).
- Podlouhlost představuje poměr mezi délkou a šířkou pravoúhelníku opsaného oblastí, který má nejmenší plochu ze všech pravoúhelníků, které lze různými natočeními opsat oblastí.
- Pravoúhlost je maximum z čísel, vypočtených jako poměr mezi velikostí oblasti a plochou opsaného pravoúhelníka pro různá natočení. Dokonale pravoúhlá oblast je když se poměr mezi velikostí oblasti a plochou opsaného pravoúhelníka v nějakém směru rovná 1.
- Nekompaktnost je vypočtena jako druhá mocnina délky hranice oblasti vydělené velikostí oblasti. Čím delší hranici (obvod) má objekt tím větší je jeho nekompaktnost. Nejkompaktnějším objektem v Eukleidově prostoru je kruh.

2 Metodika testování geometrické přesnosti skenerů

V následujících podkapitolách bude vysvětleno a popsáno jak a proč se skenují mapové podklady, jak se testují skenery, jaké druhy atestů lze na základě provedeného testování získat a k čemu tyto atesty opravňují. Výchozím legislativním pramenem jsou Pokyny č. 32 Českého úřadu zeměměřického a katastrálního (dále jen „ČÚZK“) ze dne 28. dubna 2004 č.j. 1014/2004-22 pro skenování katastrálních map a grafických operátů dřívějších pozemkových evidencí, které ČÚZK jednak zpracoval a i vydal. [15] Pokyny nabyly účinnosti dnem 15. května 2004 a nachází se v nich ustanovení, na jehož základě se zrušují Prozatímní pokyny pro skenování katastrálních map a map dřívějších pozemkových evidencí ČÚZK č.j. 4669/1993-22 ze dne 1. prosince 1993. Dále byly pokyny č. 32 čtyřikrát změněny a to následujícími dodatky:

1. Dodatkem č. 1 ze dne 15. února 2005 č.j. 613/2005-22, který nabył účinnosti dnem 22. února 2005.
2. Dodatkem č. 2 ze dne 8. března 2005 č.j. 1503/2005-22, který nabył účinnosti dnem 15. března 2005.
3. Dodatkem č. 3 ze dne 7. dubna 2006 č.j. ČÚZK 1223/2006-22, který nabył účinnosti dnem 15. dubna 2006.
4. Dodatkem č. 4 ze dne 16. května 2006 č.j. ČÚZK 2321/2006-22, který nabył účinnosti dnem 1. června 2006.

Skenování mapových podkladů a testování skenerů se tedy provádí na základě Pokynů č. 32 ČÚZK ze dne 28. dubna 2004 č.j. 1014/2004-22 ve znění dodatku č. 1 ze dne 15. února 2005 č.j. 613/2005-22, dodatku č. 2 ze dne 8. března 2005 č.j. 1503/2005-22, dodatku č. 3 ze dne 7. dubna 2006 č.j. ČÚZK 1223/2006-22 a dodatku č. 4 ze dne 16. května 2006 č.j. ČÚZK 2321/2006-22. Pokyny jsou rozděleny na tři části, přičemž testováním skenerů se zabývá druhá část.

2.1 Skenování mapových podkladů

Protože ještě nebyl dokončen proces převodu katastrálních map do digitální podoby, je část území pokryta analogovou katastrální mapou, která je po skenování k dispozici v rastrovém podobě.

První část pokynů se věnuje skenování, konkrétně v odst. 1 najdeme okruh jejich vymezení: „*Pokyny se týkají vyhotovování kopií mapových podkladů v digitální formě jako rastrových počítačových souborů (dále jen „rastrové soubory“) pro účely archivace, vedení a obnovy katastru nemovitostí a pro jiné účely, např. pro využití v dalších informačních systémech jiných správců.*“ [15].

Dále se v odst. 2 skenování dělí na dva druhy – na kartometrické a orientační skenování. Kartometrické skenování znamená, že se mapové podklady skenují pro účely katastru tak, aby byla zachována jejich kartometrická hodnota (odst. 2.1 pokynů). Pod orientační skenování patří skenování pro jiné účely bez vysokých nároků na zachování přesnosti, přičemž rastrová data pořízená orientačním skenováním nelze použít jako podklad pro práce v katastru nemovitostí (odst. 2.2 a odst. 12.1 písm. b). Požadavky na rastrová data pořízená kartometrickým nebo orientačním skenováním jsou uvedeny v tabulce 2.1, jedná se vlastně o hodnoty chyb a odchylek, které musí skenery, na nichž se skenuje, splňovat.

Tabulka 2.1: Požadavky na rastrová data

	Kartometrické skenování	Orientační skenování
Střední chyba transformačního klíče	$\leq 0,07$ mm	$\leq 0,12$ mm
Střední souřadnicová chyba	$\leq 0,10$ mm	$\leq 0,15$ mm
Maximální odchylka v poloze	pro válcové skenery: $\leq 0,30$ mm pro ostatní skenery: $\leq 0,20$ mm	pro všechny skenery: $\leq 0,40$ mm
Hodnota rozlišení	minimálně 400 DPI	minimálně 400 DPI

Zdroj [15].

Skenery jsou podrobovány testování geometrické přesnosti, podle jehož výsledků je udělen atest buď právě pro kartometrické nebo pro orientační skenování (podrobnosti o testování skenerů je možno nalézt v následující kapitole č. 2.2).

Skenovací práce řídí, organizuje a koordinuje ČÚZK (odst. 4.1 pokynů), ovšem na základě odst. 4.2 je provádí Zeměměřický úřad (dále jen „ZÚ“). Dále ještě mohou skenovací práce provádět katastrální úřady (dále jen „KÚ“) a případně i externí skenovací pracoviště (samozřejmě skener musí mít udělen platný atest).

Mapové podklady v originálu se poskytují pro kartometrické skenování a používají se skenery, kterým byl udělen atest pro tento druh skenování (podrobnosti viz odst. 5.1 a odst. 7.1 a další pokynů). Kartometrickou kvalitu rastrových kopií mapových podkladů zajišťuje konstrukce skeneru a platný atest pro kartometrické skenování (odst. 8.1 pokynů). Grafická kvalita se posuzuje pohledovou kontrolou (odst. 8.2). Pro orientační skenování se poskytují mapové podklady, které musí být po celou dobu skenovacích prací umístěny v transparentním krytu (odst. 12.1 písm. a) a rastrová data jsou pořízena skenerem, který má udělen platný atest.

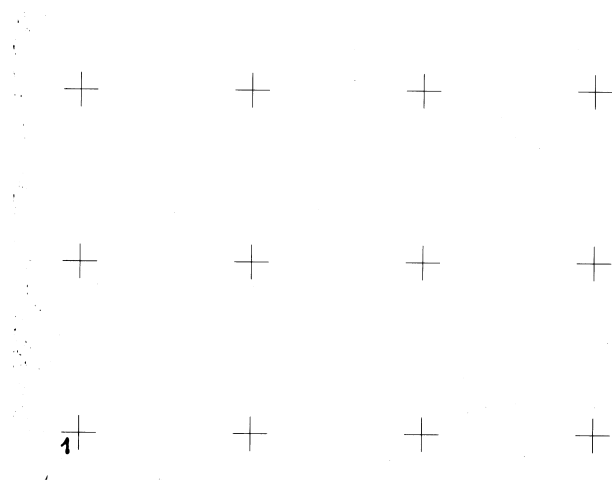
2.2 Testování skenerů

Testování skenerů se věnuje druhá část Pokynů č. 32 ČÚZK. V odst. 13.1 je uvedeno, že pracoviště pro kartometrické nebo orientační skenování mapových podkladů musí splnit technické podmínky zařízení a vybavení a dále se vyžaduje, aby provozovatel skeneru byl oprávněn vykonávat zeměměřické činnosti ve smyslu zákona o zeměměřictví (toto oprávnění doloží provozovatel kopií živnostenského listu popřípadě dokladem o vzdělání odborně způsobilé osoby).

Dle odst. 13.2 skenery testuje ZÚ, přičemž testování se provádí na základě písemné žádosti provozovatele skeneru. Po vyhodnocení testu skeneru postoupí ZÚ žádost spolu s protokolem o testování geometrie skeneru ČÚZK, který rozhodne o udělení atestu.

Pokud ČÚZK rozhodne o udělení atestu pak je atest udělen na určitou dobu. Poprvé je tato doba čtyři měsíce. Doba platnosti atestu může být prodloužena až na jeden rok, pokud se se opakovanými testy potvrdí stabilita vnitřní přesnosti skeneru. Před vypršením platnosti atestu je provozovatel skeneru povinen požádat o nové testování. Pokud provozovatel skeneru shledá zhoršení technických vlastností skeneru přeruší skenovací práce a oznámí tuto skutečnost ZÚ, který to oznámí ČÚZK a ten zruší platnost atestu skeneru a po opravě provozovatel požádá o nové testování (tento postup vyplývá z odst. 9.7 v první části pokynů).

Testování skeneru se podle odst. 13.5 zaměřuje především na určení geometrické přesnosti rastrového obrazu skenované předlohy a na zhodnocení grafické kvality rastrového obrazu (čitelnosti kresby) dosažené při optimálním nastavení prahových hodnot. V dalším odstavci je pak uvedeno, jakým způsobem se určuje geometrická přesnost skeneru. Používá se kontrolní mřížka o rozměrech 700 × 550 mm, na níž je síť po 50 mm (fólie, na které je vzor vyryt má rozměry o něco větší než nanesený vzor; ukázka části mřížky na obr. 2.2.1). Kontrolní mřížka byla vyrobena rytinou na plastové fólii kreslicím stolem CORA II v ZÚ. Materiál fólie se nazývá astralon a kontrolních fólií bylo pořízeno více (okolo roku 1991) ale časem se poškozovaly až zůstala pouze jedna. Problémem je, že dnes již neexistuje kreslicí stůl, na kterém byla fólie vyryta a navíc použité černidlo bylo agresivní a poškozovalo životní prostředí a proto byla tato technologie zakázána.



Obrázek 2.2.1: Výřez z kontrolní mřížky

Souřadnice průsečíků mřížky jsou proměřeny na digitalizátoru s přesností 0,02 mm, uloženy jsou v podobě seznamu souřadnic kontrolních bodů v referenčním souboru. Neměnnost souřadnic se kontroluje ve Výzkumném ústavu geodetickém, topografickém a kartografickém ve Zdíbech a z kontrol vyplývá, že je fólie rozměrově stabilní.

Skenerem je pořízen rastrový obraz kontrolní mřížky, přičemž parametry skenování u testovaného skeneru se nastavují tak, aby hodnota rozlišení byla alespoň 400 DPI (a zároveň blízká této hodnotě) a aby nastavená hodnota prahu poskytovala tloušťku čáry mřížky 3 až 6 pixelů (odst. 14 pokynů). Na rastrovém obraze kontrolní mřížky jsou na programovém vybavení IRAS/B digitalizovány souřadnice průsečíků mřížky. Ze 4 rohových bodů mřížky se určí transformační klíč afinní transformace pomocí něhož se transformuje zbylých 156 průsečíků mřížky do souřadnicového systému digitalizátoru a vypočtou se posuny dx , dy vzhledem k souřadnicím v referenčním souboru a určí se střední chyby v souřadnicích x , y a v poloze.

Dle odst. 15.2 se grafická kvalita posuzuje pohledovou kontrolou černobílé rastrové kopie barevné kopie mapového listu mapy bývalého pozemkového katastru, která reprezentuje horší průměr opotřebení. Vzorový list musí obsahovat dostatečné množství červených změnových zákresů. Posuzuje se plynulost, spojitost a tloušťka čar kresby (odst. 15).

V odst. 16.1 je uvedeno, že v protokolu testování jsou shrnuty výsledky testu, jsou v něm uvedeny údaje o skeneru, nastavené parametry skenování při testu, výsledky testování a stanovisko ZÚ týkající se použití testovaného skeneru pro kartometrické či orientační skenování. Přílohou protokolu jsou výsledky transformace a pohledové kontroly (odst. 16.2 pokynů). Protokol musí být podepsán pracovníkem ZÚ jenž test prováděl a jeho vedoucím (odst. 16.3).

2.2.1 Metodika průběhu testování ¹

1. Před samotným skenováním je nutno připravit fólii s kontrolní mřížkou. Fólie s křížky (180 křížků, rastr 15×12 , vzdálenost 50 mm) se položí na rovnou podložku (nejčastěji stůl), kde dojde k jejímu vyrovnání a srovnání teploty, což trvá 1 – 2 hodiny podle okolní teploty. Fólie se přepravuje stočená v roli a během přepravy je vystavena střídání teplot. Pokud by se fólie s křížky nenechala dostatečnou dobu srovnat, mohlo by dojít k lokální deformaci.
2. Během doby přípravy fólie je provedeno černobílé skenování opotřeбенé barevné mapy (pro přiblížení reálným podmínkám) s rozlišením 400 DPI. K tomuto se přistupuje z důvodu ověření schopnosti skeneru věrně převést různobarevná data na černobílou reprezentaci (může totiž dojít k problémům se skenováním části barevného spektra). Skenování se provede v případě nutnosti několikrát, dokud není vhodným nastavením prahu dosaženo optimální kvality. Sleduje se zejména kvalita skenování červených čar, protože jeden z testovaných skenerů měl potíže s jejich skenováním.
3. Po vyrovnání fólie (tzn. po 1 – 2 hodinách) se provede vlastní skenování. Skenuje se opět černobíle při rozlišení nejméně 400 DPI. Volí se dva prahy a to tak, aby linky naskenovaných křížků měly tloušťku 3 až 6 pixelů. Po naskenování se kontroluje, zda tloušťka 4 rohových křížků a několika dalších náhodně zvolených je v požadovaných mezích. Po nalezení vhodných prahů se skenování provádí celkem čtyřikrát: pro dva různé prahy (kdy pro jeden práh má hodnotu takovou, aby tloušťka křížků byla zhruba na spodní hranici požadované tloušťky a druhý práh se volí tak, aby tloušťka křížků po naskenování byla přibližně na horní hranici meze) a pro dvě orientace fólie (otočení o 180°). V praxi se ukazuje, že hodnoty prahů pro jednotlivé přístroje se nemění, čili pokud se po nějakém časovém období provádí testování na stejném přístroji odpadá nutnost hledání optimálních prahových hodnot.

¹ Převzato a upraveno z [8].

4. V naskenovaných rastrových datech se najdou středy křížků – ručně se vektorizují jednotlivé úsečky tvořící křížky a z nich se vypočtou průsečíky – čili středy křížků. Tato metoda je mnohem přesnější, než kdyby se přímo určovaly středy jako jeden bod křížku.
5. Nalezené souřadnice středů křížků se poté transformují na místní (referenční) souřadnice fólie za použití afinní transformace. Nejprve je ze 4 rohových křížků metodou nejmenších čtverců spočítán transformační klíč. S jeho pomocí se poté transformují i ostatní křížky. Použití afinní transformace, s klíčem vypočítaným ze 4 rohů, vychází z praxe, kde se stejná metoda používá i pro transformaci map. Afinní transformace dokáže potlačit chybu rotováním, zkosením, posunem a změnou měřítko. Chyby, které mohou být způsobeny vnějšími vlivy (obsluhou skeneru) a nemají tedy vliv na geometrickou přesnost samotného skeneru, spadají do kategorie chyb potlačitelných afinní transformací.
6. Výše uvedeným postupem se získají souřadnice křížků naskenovaných dat ve stejném souřadnicovém systému jako referenční souřadnice vzorových křížků. Očíslují-li se křížky postupně po řádcích (12 bodů na řádek), pak se srovnáním jednotlivých párů získají souřadnicové odchylky a z nich se vypočítá chyba transformačního klíče podle vzorce:

$$\mathcal{E}_k = \sqrt{\frac{1}{4} \sum (\delta_{xi}^2 + \delta_{yi}^2)} \quad \text{pro } i = \{1, 12, 169, 180\} \quad (2.1)$$

a směrodatná odchylka (střední souřadnicová chyba) podle vzorce:

$$\mathcal{E}_s = \sqrt{\frac{1}{180} \frac{1}{2} \sum (\delta_{xi}^2 + \delta_{yi}^2)} \quad \text{pro } i = \{1, \dots, 180\} \quad (2.2)$$

přičemž platí, že δ_{xi} a δ_{yi} jsou souřadnicové chyby mezi skutečnými a naměřenými souřadnicemi po transformaci (i je číslo příslušného křížku). Tyto hodnoty potom představují chybový vektor:

$$\vec{\epsilon}_i = (\delta_{xi}, \delta_{yi}) \quad (2.3)$$

Ze získaných souřadnicových odchylek se dále sestaví vektorová mapa odchylek.

3 Metoda nejmenších čtverců

Metoda nejmenších čtverců (dále „MNČ“) je jednou ze statických matematických metod. Často se snažíme najít parametry funkční závislosti dvou veličin x a y a to tak, že chceme vytvořit model, kterým bychom závislost veličin popsali (závislost $y = f(x)$). Dá se říci, že hledáme nejlepší aproximační funkci pro dané hodnoty a chceme zjistit parametry (koeficienty) této funkce. Tato nejlepší aproximace má velmi dobré statistické vlastnosti (vyrovnává vliv náhodných chyb v zadaných hodnotách).

MNČ je tedy aproximační metoda, která spočívá v nalezení takového řešení (parametrů) funkce f , aby součet druhých mocnin chyb nalezeného řešení byl minimální (tj. aby součet čtverců odchylek vypočtených hodnot byl nejmenší od známých hodnot). [1]

3.1 Odvození řešení pro soustavu lineárních rovnic pomocí MNČ

Jelikož se v diplomové práci aplikuje metoda nejmenších čtverců hned několikrát, bude v následujícím textu ukázáno odvození pro jednu z častých aplikací MNČ, kterou je řešení soustavy lineárních rovnic.

Uvažujme tedy soustavu lineárních rovnic ve tvaru [4]:

$$Ax = l, \tag{3.1}$$

kde A je daná matice soustavy o m řádcích a $(n + 1)$ sloupcích, předpokladem je, že matice $A^T A$ je symetrická, pozitivně definitní [4]. l je vektor pravé strany (známé hodnoty) a x je hledaný vektor (hledané řešení). Po převodu si na pravou stranu zavedeme vektor oprav neboli reziduum:

$$Ax - l = r. \tag{3.2}$$

Při metodě nejmenších čtverců nám jde o nalezení vektoru hodnot, pro který požadujeme, aby kvadrát součtů reziduí byl co nejmenší, čili minimalizujeme normu

rezidua:

$$\|r\|^2 = r^T r \rightarrow \min. \quad (3.3)$$

Výraz (3.3), který minimalizujeme lze zapsat takto:

$$\|r\|^2 = \sum r_i^2 = r^T r = (Ax - l)^T (Ax - l) = \phi. \quad (3.4)$$

Ve výrazu (3.4) se nám objevil tzv. funkcionál ϕ (kriteriální funkce), přičemž nabývá minima v bodech, kdy je derivace rovna nule – tzn. nutná podmínka existence minima má následující tvar:

$$\phi = \frac{\partial \phi}{\partial x} = 0. \quad (3.5)$$

$$\begin{aligned} \phi &= [(Ax - l)^T (Ax - l)]' = 0, \\ [x^T A^T Ax - x^T A^T l - l^T Ax + l^T l]' &= 2A^T Ax - A^T l - l A^T = 0. \end{aligned} \quad (3.6)$$

Výsledný výraz po derivování upravíme do tvaru tzv. soustavy normálních rovnic:

$$A^T Ax = A^T l, \quad (3.7)$$

přičemž řešením této soustavy je hledaný vektor řešení x :

$$x = (A^T A)^{-1} A^T l. \quad (3.8)$$

3.2 Aplikace MNČ – aproximace přímkou

Mějme N bodů, kterými chceme proložit přímkou a to tak, aby součet druhých mocnin odchylek jednotlivých bodů od přímky (tj. čtverců vzdáleností od přímky) byl minimální. Úloha tedy představuje aproximaci bodů přímkou metodou nejmenších čtverců. Protože se jedná o lineární model, lze úlohu přeformulovat maticově a využít tak postupů uvedených v kap. 3.1, jelikož hledání řešení přejde na výpočet přeurlčené soustavy rovnic.

Rovnice přímky má následující tvar [4]:

$$y = kx + q, \quad (3.9)$$

přičemž koeficienty k a q jsou naše neznámé, které chceme nalézt.

Protože chceme úlohu řešit maticově a převést jí na vztah soustavy lineárních rovnic (3.1) tak hledané koeficienty umístíme do vektoru z :

$$z = \begin{bmatrix} k \\ q \end{bmatrix}. \quad (3.10)$$

Vztah (3.9) pak po převedení do maticového tvaru vypadá následovně:

$$\begin{bmatrix} x & 1 \end{bmatrix} \begin{bmatrix} k \\ q \end{bmatrix} = [y], \quad (3.11)$$

respektive jej také můžeme zapsat následujícím způsobem:

$$\underbrace{\begin{bmatrix} x_1 & 1 \\ x_2 & 1 \\ x_3 & 1 \\ \vdots & \vdots \\ x_n & 1 \end{bmatrix}}_A \underbrace{\begin{bmatrix} k \\ q \end{bmatrix}}_x = \underbrace{\begin{bmatrix} y_1 \\ y_2 \\ y_3 \\ \vdots \\ y_n \end{bmatrix}}_l. \quad (3.12)$$

Úlohu jsme postupně přepsali na maticový výpočet, převedli jsem ji tedy do požadovaného tvaru – na soustavu lineárních rovnic uvedenou ve vzorci (3.1). Odvození řešení takovéto soustavy se nachází v kap. 3.1, tudíž řešení pro úlohu aproximaci bodů přímkou má pak již známý tvar:

$$z = (A^T A)^{-1} A^T l. \quad (3.13)$$

3.3 Odvození afinní transformace používané při testování + aplikace MNČ u afinní transformace²

Definujme jednotlivé vektory jako souřadnice bodů, které reprezentují:

\vec{x}_i – středy křížků získané z naskenovaných dat,

\vec{y}_i – středy křížků získané transformací naskenovaných dat,

\vec{z}_i – referenční souřadnice křížků,

kde i je číslo příslušného křížku, které nabývá diskretních hodnot $i = \{1, \dots, 180\}$.

² Převzato a upraveno z [8].

Dále řekněme, že data jsou naskenována s chybou $\vec{\epsilon}_{Ci}$:

$$\vec{z}_i = \vec{x}_i + \vec{\epsilon}_{Ci}. \quad (3.14)$$

Chybu $\vec{\epsilon}_{Ci}$ lze vyjádřit jako rozdíl:

$$\vec{\epsilon}_{Ci} = \vec{\epsilon}_{Li} - \vec{\epsilon}_i, \quad (3.15)$$

kde $\vec{\epsilon}_{Li}$ je lineární chyba prvního řádu a $\vec{\epsilon}_i$ reprezentuje ostatní chyby. Protože je chyba $\vec{\epsilon}_{Li}$ lineární prvního řádu lze ji rozepsat jako součin:

$$\vec{\epsilon}_{Li} = \mathbf{\epsilon}_A \vec{x}_i + \vec{\epsilon}_b, \quad (3.16)$$

kde $\mathbf{\epsilon}_A$ je bazová matice chyby a $\vec{\epsilon}_b$ chybový posun.

Dosazením vztahu 3.15 do 3.14 získáme:

$$\vec{z}_i = \vec{x}_i + \vec{\epsilon}_{Li} - \vec{\epsilon}_i \quad (3.17)$$

Chceme, aby transformace na naskenovaných datech potlačila lineární chybu prvního řádu, proto definujeme vztah mezi \vec{x}_i a \vec{y}_i , z něhož vyplyne vztah mezi \vec{y}_i a \vec{z}_i jako:

$$\vec{y}_i = \vec{x}_i + \vec{\epsilon}_{Li} \text{ pak } \vec{z}_i = \vec{y}_i - \vec{\epsilon}_i. \quad (3.18)$$

Potom můžeme dosadit do prvního vztahu v 3.18 chybu vyjádřenou v 3.16:

$$\vec{y}_i = \vec{x}_i + \mathbf{\epsilon}_A \vec{x}_i + \vec{\epsilon}_b, \text{ po úpravě dostaneme } \vec{y}_i = (\mathbf{I} + \mathbf{\epsilon}_A) \vec{x}_i + \vec{\epsilon}_b. \quad (3.19)$$

Definujme substitute:

$$\begin{aligned} \mathbf{I} + \mathbf{\epsilon}_A &= \mathbf{A}, \\ \vec{\epsilon}_b &= \vec{b}. \end{aligned} \quad (3.20)$$

Potom platí:

$$\vec{y}_i = \mathbf{A} \vec{x}_i + \vec{b}, \quad (3.21)$$

což je předpis pro *afinní transformaci*. Matice \mathbf{A} a vektor \vec{b} pak tvoří klíč afinní transformace.

Klíč afinní transformace vypočteme následujícím způsobem – nejprve dosadíme druhý vztah z 3.18 do 3.21:

$$\mathbf{A} \vec{x}_i + \vec{b} - \vec{z}_i = \vec{\epsilon}_i \quad (3.22)$$

Výsledek převedeme na soustavu dvou rovnic:

$$\begin{aligned} a_1 x_{1i} + a_2 x_{2i} + b_1 - z_{1i} &= \varepsilon_{1i} \\ a_3 x_{1i} + a_4 x_{2i} + b_2 - z_{2i} &= \varepsilon_{2i}, \end{aligned} \quad (3.23)$$

kterou následně převedeme do maticového tvaru:

$$\begin{bmatrix} x_{1i} & x_{2i} & 0 & 0 & 1 & 0 \\ 0 & 0 & x_{1i} & x_{2i} & 0 & 1 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \end{bmatrix} \begin{bmatrix} a_1 \\ a_2 \\ a_3 \\ a_4 \\ b_1 \\ b_2 \end{bmatrix} - \begin{bmatrix} z_{1i} \\ z_{2i} \\ \vdots \end{bmatrix} = \begin{bmatrix} \varepsilon_{1i} \\ \varepsilon_{2i} \\ \vdots \end{bmatrix} \quad (3.24)$$

Z maticového zápisu je vidět, že při výpočtu řešíme soustavu rovnic o 6 neznámých. Každý bod, který bereme v úvahu „přidá“ do matic dvě řádky. Z toho plyne, že pro řešení je třeba alespoň 3 bodů (uvažujeme-li nulovou chybu). Při použití více bodů je soustava přeurčena – při jejím řešení je potřeba použít aproximační metodu. Soustavu převedeme do vektorového zápisu:

$$\mathbf{M} \vec{p} - \vec{c} = \vec{\varepsilon}, \quad (3.25)$$

kde \mathbf{M} je matice složená z prvků vektorů \vec{x}_i , vektor \vec{c} je složen z prvků vektorů \vec{z}_i , vektor $\vec{\varepsilon}$ reprezentuje chybu aproximace a vektor \vec{p} reprezentuje neznámé parametry. Úlohu máme tedy v požadovaném tvaru a odvození řešení takovéto soustavy je popsáno v kap. 3.1 – tudíž vektor parametrů má následující podobu:

$$p = (\mathbf{M}^T \mathbf{M})^{-1} \mathbf{M}^T c \quad (3.26)$$

a lze z něj sestavit vypočtený transformační klíč: matici \mathbf{A} a vektor \vec{b} .

4 Analýza skenovaného obrazu testovací předlohy

Vstupní data (ČB obraz) jsou získávána dle Pokynů č. 32 Českého úřadu zeměměřického a katastrálního [15], jejichž obsahu se podrobněji věnuje kapitola 2.

Předloha, na které jsou vyryty křížky (180 křížků, 15×12 , křížky jsou od sebe vzdáleny 50 mm) se naskenuje černobíle s rozlišením nejméně 400 DPI. Při skenování se volí dva prahy tak, aby tloušťka linky křížku na naskenovaném obraze byla mezi 3 až 6 pixely. Namátkovou pohledovou kontrolou se ověří, zda-li mají křížky požadovanou tloušťku. Předloha se skenuje čtyřikrát, tzn. pro dva různé prahy a orientace (otočení o 180°), přičemž se ale v současné době pro testování přesnosti používají pouze dva takto získané rastry (ty se stejným prahem otočené o 180°) a to kvůli náročnosti ruční vektorizace (náročnost získání středů křížků).

Máme tedy naskenovanou předlohu, na které se nachází 180 pravoúhlých křížků (15×12). Cílem úlohy je jakoukoli metodou nalézt středy křížků – nalezení souřadnic středu pro každý křížek. Naskenovaná data (dále jen „obraz“) jsou černobílá a jejich rozměr je cca $10\,000 \times 12\,000$ pixelů (120 Mpix). Pro výpočet geometrické přesnosti skeneru se poté nalezené středy křížků (jejich souřadnice) porovnájí s referenčními souřadnicemi středů křížků a vypočtou se příslušné charakteristiky přesnosti.

Předpokládáme, že pro účely testování geometrické přesnosti skenerů budou získaná data dostatečně kvalitní. Při specifikaci požadavků na vstupní data vycházíme z informací a postupů při skenování poskytnutých pracovníkem ZÚ panem Ing. Bohumilem Michalíkem, Csc. Požadavky jsou založeny zejména na základě pečlivé pohledové kontrole po naskenování předlohy. Můžeme proto specifikovat požadavky, které budou bez problémů při skenování splněny, ačkoliv v jiných oblastech zpracování obrazu by byly velmi omezující.

Požadavky na vstupní data jsou následující:

- maximální natočení mřížky a tím i jednotlivých křížku je menší než 5° ,
- data jsou postižena šumem pouze minimálně a lokalizovaně nebo vůbec,
- mřížka je zkosená pouze minimálně nebo vůbec.

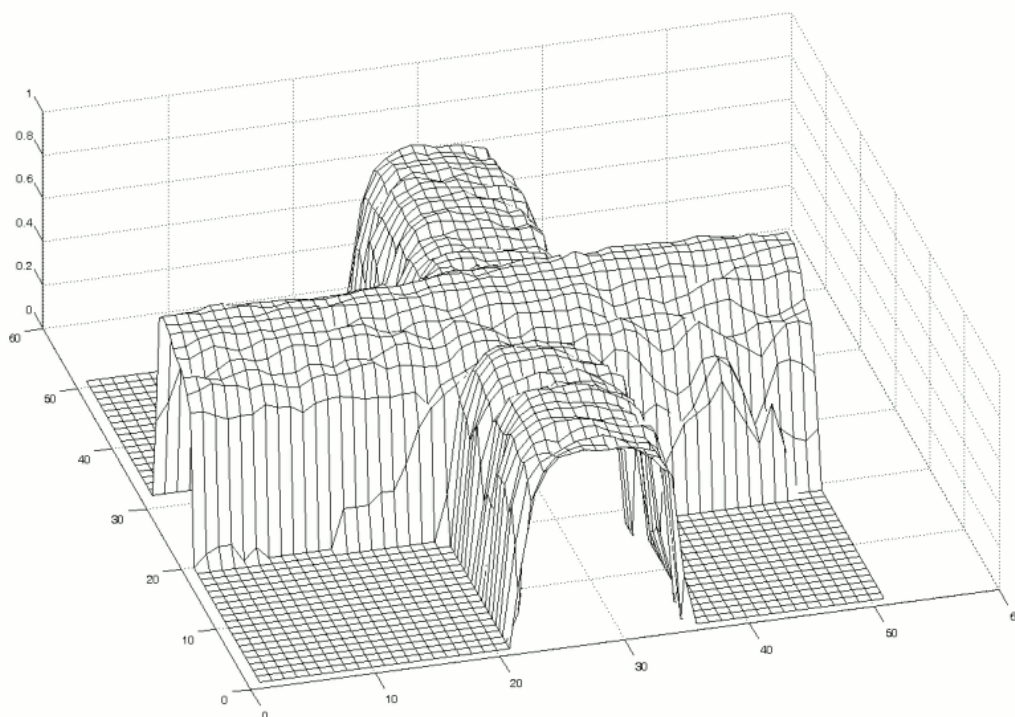
Algoritmus pro řešení úlohy se může rozepsat do čtyř kroků:

1. **Nalezení skeletu** – naskenované křížky mají určitou nenulovou tloušťku (počet pixelů na šířku, kterými je úsečka křížku reprezentována). Cílem je nalézt pro každý křížek bezrozměrnou reprezentaci. Jako reprezentaci jsme zde zvolili dvě množiny bodů, které definují dvě úsečky, kterými je křížek tvořen. Cílem je nalézt tyto body tak, aby po proložení přímkami byl průsečík přímek co nejbližší skutečnému středu křížku. Tyto body označujeme jako body skeletu. Skelet je v tomto případě chápán jinak než jak byl popsán v teoretických pasážích práce.
2. **Nalezení středů křížků** – provedeme analýzu dat získaných v předchozím kroku. Nejprve vybereme body skeletu příslušející jednomu křížku. Protože může dojít k chybnému nalezení provedeme ještě kontrolu, zda body skeletu opravdu tvoří křížek. V případě že ano, proložíme body skeletu přímkami a vypočteme jejich průsečík – střed křížku.
3. **Určení orientace a očíslování** – vzhledem k tomu, že předloha může být naskenována s různou orientací, musíme tuto orientaci nejprve zjistit. To nám umožní určit čísla křížků v obrazu a následné porovnání vypočtených souřadnic křížků v místní soustavě s referenčními souřadnicemi křížku. Při určení orientace předlohy se vychází z teoretické části práce zabývající se srovnáváním se vzorem.
4. **Výpočet výsledků** – v tomto kroku spočítáme transformační klíč (matici transformace), vypočítáme polohu středů křížků v milimetrech a dopočítáme ostatní požadované výsledky – střední chybu transformačního klíče, průměrnou, maximální a střední souřadnicovou chybu jak pro transformační klíč tak pro vlastní souřadnice.

4.1 Nalezení skeletu

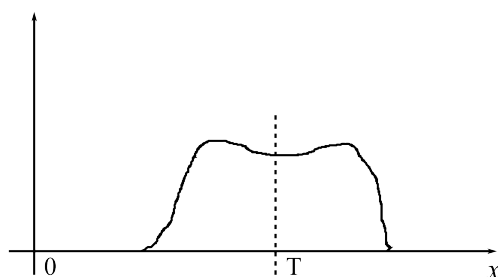
Abychom mohli vyhodnotit geometrickou přesnost skeneru srovnáním referenčních a vypočtených souřadnic středů křížků, musíme nejprve definovat, co je střed křížku na předloze. Proto si ještě definujeme rovinu, ve které se nachází předloha a na této rovině pak v souladu s předlohou definujeme pravoúhlo souřadnicovou soustavu, jež má počátek v křížku č. 1 (levý dolní roh), osa y vede od počátku ke křížku č. 169 (pravý dolní roh), osa x je kolmá na osu y (vede do bodu č. 12 – levý horní roh).

Úsečky, tvořící křížky, mají na předloze nenulovou tloušťku a hloubku. Celou předlohu tak můžeme definovat jako spojitou funkci $f(x, y)$, která má obor hodnot $H(f) = \langle 0, 1 \rangle$ v závislosti na hloubce rytí na předlohu (0 = čistá předloha, 1 = nejhlubší rytí). Křížek je reprezentován dvěma vyrytými úsečkami a střed křížku se nalézá na průsečíku podélných os těchto úseček majících nenulovou tloušťku. Naším cílem je tedy nalézt dvě úsečky, odpovídající těmto osám a z nich vypočítat střed křížku. Ilustračně si můžeme ukázat okolí středu křížku spojitě funkce na obr. 4.1.1.



Obrázek 4.1.1: Ilustrační příklad spojitě funkce $f(x, y)$ v okolí středu křížku

Z požadavků na naskenování vychází, že úsečky jsou téměř svislé nebo vodorovné. Dále definujeme řez obrazovou funkcí jako fixaci jednoho parametru funkce na konstantu. Je zřejmé, že je možné mít dvě orientace řezů vůči obrazové funkci: „horizontální“ – pokud zafixujeme parametr y a „vertikální“ – pokud zafixujeme parametr x (ukázka řezu níže na obr. 4.1.2). Pro zjednodušení budeme dále pracovat s obrazovou funkcí obsahující pouze jeden křížek (nebude-li psáno jinak). Definujeme těžiště T řezu úseček křížku jako bod, který rozděluje řez na dvě stejně velké části, přičemž podélná osa úsečky křížku prochází těžištěm daného řezu. Můžeme proto zadefinovat jednotlivé polohy těžišť řezů podél přímek rovnoběžných s jednou resp. s druhou souřadnicovou osou jako polohy bodů skeletu. V této fázi zpracování obrazu se zaměřujeme na vyhledání množiny těžišť, která je dále označována jako **skelet** a jednotlivá těžiště jsou nazývána body skeletu.



Obrázek 4.1.2: Řez obrazovou funkcí

Řez může procházet přes několik křížků, respektive úseček, kterými jsou křížky tvořeny. Jednotlivé úsečky jsou odděleny oblastmi obrazové funkce, kde má funkce hodnotu rovnou nule. Matematická definice těžiště T pak vypadá následovně pro horizontální řezy (vertikální řezy analogicky):

$$\forall z: f(z - \Delta, y_f) = 0; f(z + \Delta, y_f) = 0;$$

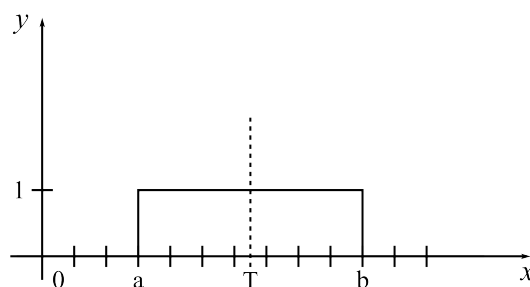
kde Δ je vždy nejmenší možné

y_f je zafixovaný parametr

(4.1)

$$T: \int_{z-\Delta}^T f(x, y_f) = \int_T^{z+\Delta} f(x, y_f)$$

V našem případě pracujeme s naskenovaným obrazem testovací předlohy, což je ve skutečnosti obrazová funkce s omezeným diskrétním definičním oborem (= matice) i omezeným diskrétním oborem hodnot. Těžiště pro binární obrazovou funkci v našem případě můžeme zobrazit takto – obr. 4.1.3:



Obrázek 4.1.3: Těžiště pro binární obrazovou funkci

Vzhledem k tomu, že při počítačovém zpracování nebudeme pracovat se spojitými funkcemi, je potřeba pro náš případ vzorec pro těžiště diskretizovat:

$$T: \sum_{z-\Delta}^T f(i, y_f) = \sum_T^{z+\Delta} f(i, y_f). \quad (4.2)$$

Obor hodnot navíc obsahuje pouze hodnoty 0 a 1 (černobílý obrázek). Proto lze vzorec dále upravit do zjednodušené formy (v kontextu s obr. 4.1.3):

$$(T - a) = (b - T) \quad (4.3)$$

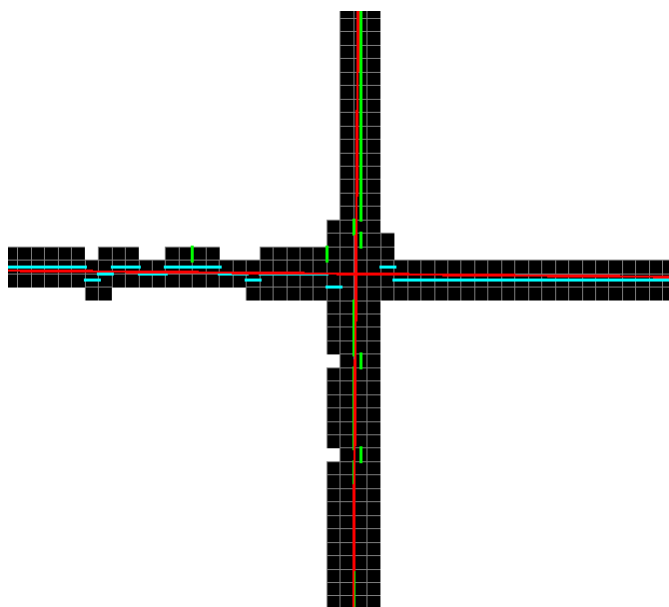
$$T = \frac{a + b}{2}$$

Těžiště se tedy nachází uprostřed řezu úsečky křížku.

Protože definiční obor funkce je diskrétní a omezený, počet všech řezů je konečný. Při vyhledávání bodů skeletu tedy nalezneme všechna těžiště ve všech řezech obrazové funkce. Nalezená těžiště odpovídají bodům skeletu, přičemž platí, že jedna souřadnice bodu skeletu je poloha těžiště a druhá souřadnice je fixovaný parametr pro řez funkce, ve kterém bylo těžiště nalezeno. Mezi body skeletu mohou být přítomny i takové, které nepatří žádnému křížku (šum při skenování, smítka na předloze apod.). Tyto body jsou při další krocích zpracování rozpoznány a odfiltrovány. Vzhledem

k tomu, že obrazová funkce je diskrétní, existuje minimální vzdálenost kterou mezi sebou každé dva body skeletu mají. Tato minimální vzdálenost je stejná jako je rozměr jednotlivých pixelů. Hustota nalezených bodů skeletu je tedy stejná jako hustota pixelů (hustota představuje frekvenci nejmenších prvků; minimální vzdálenost mezi jednotlivými prvky). Ukázka nalezených bodů skeletu se nachází na obr. 4.1.4.

V práci budou dále použity termíny: horizontální bod skeletu a vertikální bod skeletu. Toto označení říká, v kterém typu řezu byl bod nalezen. Máme tedy dvě podmnožiny bodů skeletu: horizontální skelet a vertikální skelet.



Obrázek 4.1.4: Ukázka nalezených bodů skeletu (modrá a zelená barva)

4.1.1 Algoritmus nalezení skeletu

Obrazová data (matici) procházíme nejprve po řádcích a potom po sloupcích (řádky a sloupce představují řezy funkcí, prvky matice jsou jednotlivé pixely). Postupujeme od začátku do konce řádky/sloupce a hledáme prvky takové, že prvek je roven 1 (= černá), přičemž předchozí prvek je roven 0 (= bílá) – hledáme náběžnou hranu. Pozice tohoto prvku v řádce/sloupci je parametr a (viz vzorec 4.3). Pro každý takto nalezený prvek se nalezne nejbližší následující za ním, pro který platí, že prvek je roven 0 (= bílá),

přičemž předchozí prvek je roven 1 (= černá) – hledáme sestupnou hranu. Pozice tohoto prvku v řádce/sloupci je parametr b (opět viz vzorec 4.3). Pro dvojici nalezených prvků se vypočítá těžiště mezi nimi dle vztahu pro jeho výpočet uvedeného v 4.3. Pokud je vzdálenost pozic a a b větší než definovaná maximální tloušťka křížku, vypočítané těžiště se pro další výpočet nepoužije. Poloha těžiště a řádkového/sloupcového indexu (řezu) jsou souřadnicemi bodu skeletu, které jsou uloženy pro výpočet v dalších krocích algoritmu. Poté se algoritmus opakuje pro další řádky/sloupce. Definiční obor obrazové funkce je diskrétní a omezený (jak již bylo zmíněno výše). Nejmenší dílek tohoto oboru (hustota) má velikost jeden pixel. Definiční obor takto nalezených bodů (námi nazývaných skeletem) je dvojnásobně větší na ose řezu (dáno dělením 2 při výpočtu těžiště). Na druhé ose je stejný jako definiční obor obrazové funkce. Pokud známe DPI, je možné přepočítat velikost v pixelech na centimetry a opačně – tato skutečnost je důležitá pro další výpočty.

Algoritmus byl zvolen kvůli jeho výborné časové složitosti $O(n)$, kde n je počet pixelů. Další výhodou je jeho snadná implementace. Algoritmus má poměrně velkou paměťovou náročnost, protože data obrazové funkce jsou rozsáhlá. Tato nevýhoda je ale potlačena díky optimalizaci, která bude popsána dále.

4.2 Nalezení středů křížků

Při skeletizaci jsme obrazovou funkcí, na které byly křížky definovány úsečkami s nenulovou tloušťkou, převedli na jinou funkci, kde křížky mají bezrozměrnou reprezentaci. V tuto chvíli tedy máme body skeletu a stále ještě nevíme, kde se křížky nacházejí, musíme je nejprve nalézt. Každý křížek je tvořen dvěma úsečkami, které jsou na sebe kolmé – svislou a vodorovnou. Způsob, jakým jsme našli body skeletu garantuje, že body skeletu náležející svislé úsečce jsou v podmnožině horizontálního skeletu a body náležející vodorovné úsečce jsou v množině vertikálního skeletu. Nejprve jsou tedy v datech nalezeny všechny svislé a vodorovné přímky.

Definujme funkci $h(x, y)$, pro kterou platí, že v bodě $[x, y]$ nabývá hodnoty 1 v případě, že existuje horizontální bod skeletu se souřadnicemi $[x, y]$ a hodnoty 0 v opačném případě. Analogicky definujme funkci $v(x, y)$ pro vertikální skelet. Pokud podmnožina bodů skeletu tvoří přímku, budou rozdíly jedné jejich souřadnice oproti této průměrné souřadnici minimální (viz požadavky – maximální natočení 5°). Definujme funkci reprezentující množství bodů skeletu pro jednotlivé vodorovné a svislé přímky:

$$\begin{aligned} h_s(y) &= \sum_x h(x, y) \\ v_s(x) &= \sum_y v(x, y) \end{aligned} \quad (4.4)$$

Parametry funkcí jsou koeficienty posunu přímek vůči počátku souřadnic (bod, ve kterém přímka protíná osu souřadnic). Hodnoty funkcí reprezentují množství bodů skeletu pro jednotlivé svislé/vodorovné řezy. Protože úsečky křížku mohou být natočeny, přidáme toleranci Δ vyjadřující, jak maximálně může být bod vzdálen od přímky aby ještě přímce náležel. Tolerance Δ zohledňuje maximální natočení předlohy o α stupňů (dle požadavků skenování $\alpha = \max. 5^\circ$), velikost křížku v centimetrech a počet pixelů na centimetr. Tolerance má takovýto tvar:

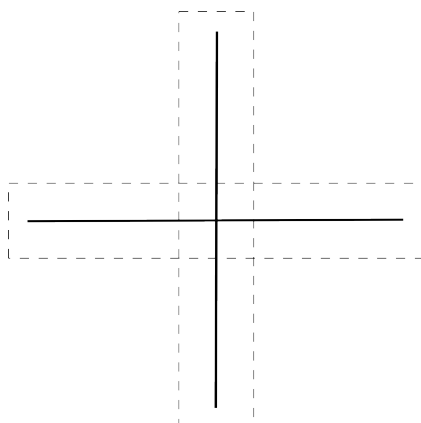
$$\Delta = \sin \alpha \cdot \text{velikost křížku v cm} \cdot \text{počet pixelů na centimetr} \cdot 0,5 \quad (4.5)$$

Dále definujme funkce s tolerancí:

$$\begin{aligned} h_t(y) &= \sum_{-\Delta}^{\Delta} h_s(y + i) \\ v_t(x) &= \sum_{-\Delta}^{\Delta} v_s(x + i) \end{aligned} \quad (4.6)$$

Známe délku úseček, které tvoří křížek. Délka úsečky v pixelech odpovídá počtu bodů skeletu. Najdeme všechna x a y na výše zmíněných funkcích, pro které funkce nabývají vyšších hodnot než je délka úsečky vyjádřená v pixelech. Toto můžeme udělat, protože hodnoty mají stejný rozměr (hustotu). Takto získáme množinu x -ových a množinu y -ových souřadnic, na kterých se potenciálně nachází křížek.

Pro každou kombinaci (kartézský součin) souřadnic z nalezených množin určíme, zda se na daných souřadnicích nachází křížek. Pro účely nalezení křížku budeme definovat úsečku, která je kratší či delší než normální délka úsečky křížku – takováto úsečka bude kratší či delší o 10% (empiricky zjištěný údaj). Definujeme dvě obdélníková okna, jejichž rozměry jsou: [zvětšená délka úsečky v pixelech, $2 \times \text{tolerance } \Delta$] a [$2 \times \text{tolerance } \Delta$, zvětšená délka úsečky v pixelech]. Tato okna „umístíme“ tak, aby jejich středem byly testované souřadnice. Při umístění oken přes sebe se využívá toho, že se úsečky křížků navzájem půlí (okna se v překrytu také navzájem půlí - viz obrázek 4.2.1).



Obrázek 4.2.1: Obdélníková okna

Pro obě okna vybereme body skeletu, které se v příslušném okně nacházejí. Prohlásíme, že křížek se na daných souřadnicích nachází, pokud je v každém z oken alespoň tolik bodů skeletu, kolik je délka úsečky v pixelech. I zde mají hodnoty stejný rozměr a stejnou hustotu.

Pokud je potvrzeno, že na souřadnicích se nachází křížek, body skeletu v jednotlivých oknech se pomocí metody nejmenších čtverců proloží přímkou. Pro vodorovnou přímkou se použije předpis $y = kx + q_i$, pro svislou $x = k_j y + q_j$. Tyto přímky se použijí jako první aproximace a vypočítá se průsečík přímek, který je první aproximací středu křížku.

Máme první aproximaci přímek a jejich průsečíku (středu křížku) – s jejich pomocí se z množiny bodů skeletu odstraní body skeletu, které do výpočtu přímek

(potažmo průsečíku přímek) nepatří. Odstraňují se (filtrují se) nalezené body skeletu, které jsou příliš vzdálené na to, aby byly považovány jako příslušné dané přímce a dále se ještě odstraňují body skeletu v okolí středu křížku.

Filtrace se provede tak, že body skeletu vzdálené o δ pixelů od příslušné přímky (horizontální či vertikální body skeletu) se odstraní. Vzdálenost δ od přímky se odvíjí od maximální tloušťky křížku v pixelech (tento parametr je možno při analýze v aplikaci změnit; výchozí hodnotu je 10 pixelů) a výchozí vzdálenost jsou 2 pixely od přímky (na každou stranu přímky):

$$\delta = \text{maximální tloušťka křížku v pixelech} / 5 \quad (4.7)$$

Filtrace bodů skeletu okolo středu křížku se provádí tak, že se okolo vypočtené aproximace středu křížku vytvoří kružnice, jejíž poloměr r má výchozí hodnotu 5 pixelů. Výpočet poloměru opět vychází z maximální tloušťky křížku v pixelech. Body skeletu, nacházející se uvnitř této kružnice jsou odstraněny.

$$r = \text{maximální tloušťka křížku v pixelech} / 2 \quad (4.8)$$

Po filtraci se již provede finální aproximace přímek dle předpisů uvedených výše a jejich průsečík – střed křížku. Filtrací došlo k odstranění takových bodů skeletu, které byly příliš vzdáleny úsečkám křížku (např. vlivem šumu) a bodů okolo středu křížku. Do výpočtu tak vstupují vyfiltrované body horizontálního a vertikálního skeletu.

4.2.1 Algoritmus nalezení středů křížků

Algoritmus pro nalezení středů křížků je implementován tak, jak je popsán výše. Nejprve se spočítají všechny hodnoty pro funkce h_s a v_s , z nich se vypočítají všechny hodnoty funkcí h_t a v_t , na kterých se naleznou množiny souřadnic x a y . Pro každou kombinaci souřadnic se ověří, zda se jedná o křížek. Pro nalezené křížky se z příslušných bodů skeletu vypočítají pomocí MNC přímky, ze kterých se určí střed křížku. Ten se potom zapamatuje pro další použití.

Algoritmus byl zvolen kvůli jeho časové složitosti $O(n)$, kde n je počet bodů

skeletu. Tento algoritmus má také poměrně snadnou implementaci. Algoritmus má poměrně malou paměťovou náročnost, zapamatovávají se křížky, kterých je malý počet. I tento krok je optimalizován, což bude popsáno dále.

4.2.2 Optimalizace algoritmů

Prohledávaná obrazová data jsou příliš velká. Současná testovací předloha, která má po naskenování s rozlišením 400 DPI velikost cca 120 Mpix, je po plné expanzi na hloubku 32 bitů/pixel ($3 \times 8\text{bit}$ kanály barev + 8bit alfa kanál) čtyřnásobně velká – 480MB. To je ovšem zbytečné, protože obrazová data mají hloubku 1 bit/pixel. Při použití knihoven JAI [14], se toto optimalizuje automaticky a pro uložení obrázku se vybere nejlepší datový model automaticky. Uložení obrázku v paměti je záležitost interní implementace použitých knihoven.

Problém ovšem nastává, pokud potřebujeme přistupovat k obrazovým datům přímo (naš případ). Načítat data pixel po pixelu je příliš časově náročné, především kvůli režii interního datového modelu. Taktéž načtení všech pixelů najednou je náročné na paměť, protože při načtení dojde k expanzi modelu na větší hloubku. Více obsluhované paměti je potom také i náročné časově. Standardním řešením při zpracovávání obrazových dat je použití tzv. dlaždic.

Dlaždice je obdélníkový výřez původního obrázku. Výhodou tohoto přístupu je omezení časové náročnosti na režii přístupu k datům – získáváme více pixelů najednou. Zároveň dojde k omezení paměťové náročnosti, protože nemusíme držet celý expandovaný model v paměti. Tím se omezí časová náročnost obsluhy paměti.

Algoritmus nalezení skeletu i algoritmus nalezení středu křížku lze použít na celý obrázek i na jednotlivé dlaždice. Protože výpočty pro jednotlivé dlaždice jsou na sobě nezávislé, můžeme provádět tyto výpočty paralelně v několika vláknech. Jedno vlákno je dedikováno pro vytvoření seznamu všech dlaždic obrázku a poskytování identifikátorů dlaždic ostatním vláknům (dlaždice je identifikována pozicí v obrázku a rozměry). Ostatní vlákna pak zpracovávají jednotlivě dlaždice, dokud nejsou všechny zpracovány.

Algoritmus lze rozepsat do několika kroků:

- získání identifikátoru dlaždice,
- získání obrazových dat dlaždice,
- nalezení skeletu,
- nalezení středů křížků.

Všechny operace prováděné ve vláknech jsou pouze čtení. Jedinou výjimkou, kterou je potřeba synchronizovat, je zápis nalezených středů křížků.

Použití dlaždic přináší problém, pokud křížek leží na rozhraní dlaždic. Takový křížek by nebyl nalezen ani v jedné dlaždici. Abychom tomuto problému předešli, je potřeba upravit způsob jakým jsou dlaždice vybírány. Pokud má být střed křížku nalezen vždy, musí se dlaždice překrývat minimálně o velikost křížku. Při tomto způsobu výběru dlaždic ale může nastat případ, kdy je křížek nalezen ve více dlaždicích. Tento problém odstraníme filtrací vícenásobných nalezení po ukončení výpočtu.

Algoritmus nalezení křížků je nejrobustnější pokud je v dlaždici nejvýše jeden křížek. Toho by bylo dosaženo pokud by dlaždice měla rozměr stejný, jako je velikost křížku. Nicméně to není možné, protože by poté kvůli nutnosti překryvu bylo nutné vyhodnotit nekonečný počet dlaždic. Proto jsme rozhodli, že použijeme čtvercovou dlaždici o hraně velikosti dvojnásobku velikosti křížku s překryvem o velikosti křížku (kromě těch dlaždic, které by přesáhly rozměry obrázku – tyto koncové dlaždice jsou zmenšené tak, aby nikdy nepřesahovaly rozměr původního zpracovávaného obrázku). Při volbě této kombinace zůstane algoritmus nalezení křížku robustní (pro současnou testovací předlohu bude v dlaždici vždy nejvýše jeden křížek). Celková plocha všech dlaždic bude cca 4x větší než plocha celého obrázku. Časová a paměťová úspora je i tak velmi výrazná oproti zpracování celého obrázku.

Díky optimalizaci jsme byli schopni dosáhnout velmi krátkých časů vyhodnocení. Pro v současnosti používanou testovací předlohu je čas kompletního vyhodnocení méně než 2 sekundy (na počítači se 4-jádrovým procesorem o taktu 3GHz

s přidělenou pamětí 256 MB; při přidělení větší kapacity paměti by se vyhodnocení mělo zrychlit). Dle dostupných informací trvá ruční vyhodnocení jednoho naskenovaného obrazu testovací předlohy cca 8 hodin a je potřeba vyhodnotit dva obrazy (otočené o 180°).

4.3 Nalezení orientace naskenované testovací předlohy

Předpokládáme, že v předchozím kroku byly nalezeny všechny křížky. Cílem tohoto kroku je určit orientaci s jakou byla data naskenována a křížky očíslovat tak, aby čísla nalezených křížků odpovídala číslům křížků na testovací předloze.

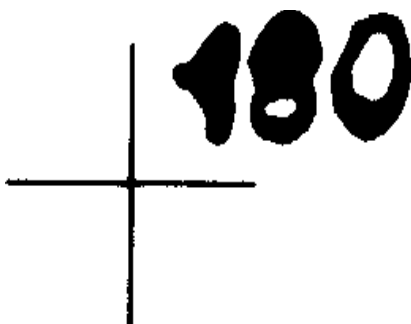
Nejprve nalezneme rohové křížky. Protože dle požadavku je natočení mřížky maximálně 5° (viz požadavky – druhá strana v kap. 4), můžeme definovat kriteriální funkce:

$$\begin{aligned}k_1(x, y) &= +x + y \\k_2(x, y) &= -x + y \\k_3(x, y) &= +x - y \\k_4(x, y) &= -x - y\end{aligned}\tag{4.9}$$

Spočítáme hodnoty kriteriálních funkcí pro každý křížek. Pro každou kriteriální funkci potom vybereme křížek, který má pro danou funkci největší hodnotu. Tyto vybrané čtyři křížky jsou rohové křížky.

Víme, že čtyři rohové křížky jsou v testovací předloze označeny (v současně používané předloze jsou označeny čísly 1, 12, 169, 180). Dále víme, že předloha je ve většině případů skenována „na šířku“ (kdyby nebyla bylo by nutné provést její otočení před načtením) a z vrchní strany (předloha je průhledná fólie – je možné ji naskenovat z obou stran). Z toho plyne, že existují pouze dvě možné varianty orientace. Abychom určili orientaci, stačí zjistit pro jeden nalezený rohový křížek jeho číslo. To lze pomocí metody srovnání se vzorem (více o metodě srovnávání se vzorem v teorii v kapitole 1.3.2). Zvolili jsme křížek č. 180, protože jeho popis má v předloze nejvíce černých pixelů a je nejkompaktnější – bude vykazovat největší rozdíly mezi jednotlivými testy

orientace (v praxi je kterýkoliv ze čtyř rohových křížků dostačující pro určení orientace). Máme dán vzor, který má stejné rozlišení (DPI) jako naskenovaná předloha. Na vzoru je křížek, jehož střed je nalezen (stejně, jako u jednotlivých dlaždic) a významný grafický útvar (ručně napsané číslo 180).



Obrázek 4.3.1: Vzor rohového křížku

Testujeme dva rohové křížky, které mohou být křížkem č. 180. Z obrazových dat vybereme výřez rohového křížku, který je rozměrově stejný jako vzor tak, aby středy rohového a vzorového křížku ležely na sobě. Pro testování druhého rohového křížku otočíme vzor o 180° tak, abychom zachovali shodnou orientaci vzoru a výřezu. Metoda srovnání se vzorem vrátí míru shody – čím větší míra shody, tím podobnější je rohový křížek vzoru. Křížek s největší mírou shody je křížek č. 180. Ostatní rohové křížky se označí podle křížku 180.

Abychom očíslovali ostatní křížky, spočítáme parametry afinní transformace souřadnic nalezených rohových křížků na virtuální souřadnice – celočíselné souřadnice odpovídající pořadí řádku a sloupce křížku v mřížce. Pro rohové křížky původní testovací předlohy jsou to souřadnice: $[1,1]$, $[1,12]$, $[15,1]$, $[15,12]$. Výpočtem získáme klíč afinní transformace s jehož pomocí pro každý nalezený křížek získáme jeho virtuální souřadnice. Tyto virtuální souřadnice poté přepočítáme na číslo křížku odpovídající číslu křížku předlohy:

$$\text{číslo křížku} = y_p - y_i + 1 + y_p(x_i - 1) \text{ kde}$$

$$y_p = \text{počet křížků ve sloupci (výška mřížky)} \quad (4.10)$$

a $[x_i, y_i]$ jsou virtuální souřadnice zaokrouhlené na celé číslo.

4.4 Výpočet klíče afinní transformace, charakteristiky geometrické přesnosti

Máme k dispozici identifikovanou množinu souřadnic středů křížků z naskenované předlohy v místní soustavě. Dostáváme se tedy k výpočtu hodnot geometrické přesnosti dle protokolu.

Abychom mohli vypočítat charakteristiky geometrické přesnosti musíme nejprve množinu souřadnic středů křížků transformovat do soustavy referenčních souřadnic. Transformace se provádí afinní transformací ze čtyř rohových křížků (může být provedena i ze všech křížků). Pomocí klíče afinní transformace se transformuje zbylých 176 průsečíků mřížky. Odvození a popis výpočtu matice afinní transformace (matice transformace) a její klíč se nachází v kapitolách 2.2 a 3.3. Vztah pro výpočet klíče afinní transformace může být zapsán tímto způsobem:

$$x = (A^T A)^{-1} A^T l. \quad (4.11)$$

Dále se vypočte střední chyba transformačního klíče m_k :

$$m_k = \sum \sqrt{\frac{(\delta_{xi}^2 + \delta_{yi}^2)}{4}} \quad \text{pro } i = \{1, 12, 169, 180\} \quad (4.12)$$

Pro tento vzorec a i pro následující platí, že δ_{xi} a δ_{yi} jsou vzdálenosti mezi referenčními a vypočtenými souřadnicemi po transformaci (souřadnicové odchylky vzniklé srovnáním odpovídajících párů souřadnic; i je číslo příslušného křížku, které nabývá diskretních hodnot $i = \{1, \dots, 180\}$).

Pro transformační klíč se ještě dále vypočte střední souřadnicová chyba m_{ks} , střední (průměrná) chyba v poloze d_{kxy} i pro oba směry x a y (d_{kx} , d_{ky}) a maximální odchylka v poloze max_{kxy} taktéž i pro oba směry (max_{kx} , max_{ky}). Vzorce jsou uvedeny níže.

Střední souřadnicová chyba transformačního klíče:

$$m_{ks} = \sqrt{\frac{1}{4} \sum \frac{(\delta_{xi}^2 + \delta_{yi}^2)}{2}} \quad \text{pro } i = \{1, 12, 169, 180\} \quad (4.13)$$

Střední (průměrná) chyba v poloze transformačního klíče (i pro oba směry):

$$\begin{aligned} d_{kxy} &= \sum \sqrt{\frac{(\delta_{xi}^2 + \delta_{yi}^2)}{4}} \quad \text{pro } i = \{1, 12, 169, 180\} \\ d_{kx} &= \frac{\sum \delta_{xi}}{4} \quad \text{pro } i = \{1, 12, 169, 180\} \\ d_{ky} &= \frac{\sum \delta_{yi}}{4} \quad \text{pro } i = \{1, 12, 169, 180\} \end{aligned} \quad (4.14)$$

Maximální odchylky transformačního klíče + průměrná maximální polohová odchylka transformačního klíče:

$$\begin{aligned} \max_{kx} &= \max(d_{kx}) \\ \max_{ky} &= \max(d_{ky}) \\ \max_{kxy} &= \sqrt{\max(d_{kx})^2 + \max(d_{ky})^2} \end{aligned} \quad (4.15)$$

Ve chvíli, kdy jsou všechny souřadnice středů křížků z naskenovaného obrazu transformovány do stejného souřadnicového systému jako referenční souřadnice křížků, se může přistoupit k výpočtu dalších charakteristik geometrické přesnosti. Výše v textu jsou uvedeny chyby transformačního klíče, nyní budou uvedeny chyby souřadnic.

Střední souřadnicová chyba m_{ss} se vypočte dle následujícího vzorce:

$$m_{ss} = \sqrt{\frac{1}{180} \sum \frac{(\delta_{xi}^2 + \delta_{yi}^2)}{2}} \quad \text{pro } i = \{1, \dots, 180\} \quad (4.16)$$

Dále je vypočtena střední (průměrná) chyba v poloze d_{sxy} spolu s maximální odchylkou v poloze \max_{sxy} . Obě tyto odchylky jsou vypočteny i pro oba směry souřadnic (d_{sx} , d_{sy} respektive \max_{sx} , \max_{sy}).

$$\begin{aligned}
 d_{sxy} &= \sqrt{\frac{(\delta_{xi}^2 + \delta_{yi}^2)}{180}} \quad \text{pro } i = \{1, \dots, 180\} \\
 d_{sx} &= \frac{\sum \delta_{xi}}{180} \quad \text{pro } i = \{1, \dots, 180\} \\
 d_{sy} &= \frac{\sum \delta_{yi}}{180} \quad \text{pro } i = \{1, \dots, 180\}
 \end{aligned}
 \tag{4.17}$$

$$\begin{aligned}
 \max_{sx} &= \max(d_{sx}) \\
 \max_{sy} &= \max(d_{sy}) \\
 \max_{sxy} &= \sqrt{\max(d_{sx})^2 + \max(d_{sy})^2}
 \end{aligned}
 \tag{4.18}$$

Všechny výše uvedené chyby se zapisují do textového protokolu, ve kterém je ještě uveden začátek zpracování obrazu, název zpracovávaného obrazu, transformační klíč (matice transformace) a souřadnice středů křížků jak referenční v milimetrech tak v obraze v pixelech a transformovaných do referenční soustavy opět v milimetrech. Dále jsou ještě u každého středu křížku vypsány chyby (odchyly) v souřadnicích oproti referenčním a střední (průměrná) polohová odchylka (označována jako d). Ukázka části protokolu se nachází na obr. 4.4.1, ukázka celého protokolu se nachází v Příloze A.

```

----- Protokol -----
Začátek zpracování: 2013-05-27 10:20:15
Zpracování souboru: zupl_1.tif
Transformační klíč (matice transformace) (px -> mm):
-----
0.06348,    0.00051,   -51.08242
0.00047,   -0.06341,    604.91769
0.00000,    0.00000,    1.00000

Chyby transformačního klíče [mm]:
-----
Střední chyba transf. klíče:    0.116
Počet vzorků:                    4
Střední souřadnicová chyba:    0.082
Střední chyba v poloze:
      x      y      d
Maximální odchylka v poloze:  0.018  0.114  0.116

Chyby souřadnic [mm]:
-----
Počet vzorků:                    180
Střední souřadnicová chyba:    0.093
Střední chyba v poloze:
      x      y      d
Maximální odchylka v poloze:  0.070  0.086  0.119

Souřadnice křížků:
-----

```

	na předloze [mm]		v obraze [px]		transformované [mm]		chyby [mm] (x, y, d)		
1:	0.000	0.000	728.607	9543.092	0.018	0.114	0.018	0.114	0.116
2:	-0.053	49.990	735.118	8756.521	0.031	49.995	0.084	0.005	0.084

Obrázek 4.4.1: Ukázka části protokolu

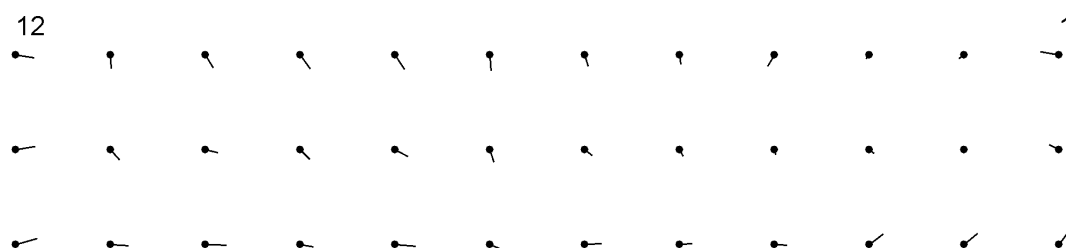
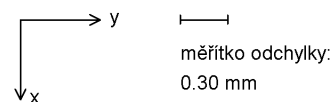
Kromě protokolu je charakteristikou geometrické přesnosti i vektorová mapa odchylek, která je sestavena ze získaných souřadnicových odchylek δ_{xi} a δ_{yi} . V mapě odchylek jsou vykresleny referenční středy křížků tečkou a u každého křížku je úsečkou vykreslena odchylka polohy středu křížku naskenované předlohy (chybový vektor). Rohové křížky v mapě odchylek jsou označeny příslušnými čísly. Orientace mapy odchylek vychází z orientace obdobné mapy obsažené v protokolech o atestu vyhotovených ZÚ (atest možno nalézt na přiloženém DVD v adresáři „Atest“).

Na mapě odchylek je v záhlaví uvedeno jméno testovaného obrazu, hodnota chyby transformace, střední a maximální chyby v poloze, ve směru x a ve směru y . Dále pak orientace souřadnicových os předlohy (dle definice na začátku kapitoly 4.1) a měřítko odchylky. Měřítko odchylky se vykresluje podle maximální chyby v poloze (zaokrouhluje se na dvě desetinná místa buď na 0 či 5). Část vektorové mapy odchylek se ukazuje na obr. 4.4.2, ukázka celého obrazu vektorové mapy odchylek je uvedena v Příloze B.

Testovaný soubor: zup1_1.tif

Chyba transformace: 0.116

[mm]	střední chyba	maximální chyba
v poloze	0.119	0.293
ve směru x	0.070	0.228
ve směru y	0.086	0.204



Obrázek 4.4.2: Výřez z vektorové mapy odchylek

5 Popis aplikace

Aplikace, nalézající v obraze středy průsečíků mřížky spolu s výpočtem charakteristik geometrické přesnosti, byla zpracována v programovací jazyce *Java* [14], který byl vyvinut společností *Sun Microsystems* (nyní *Oracle*). Tento programovací jazyk byl zvolen zejména proto, že se má všeobecnou veřejnou licenci GNU, je multiplatformní a v neposlední řadě byl v seznamu literatury navržen v zadání diplomové práce. Jako vývojové prostředí byl použit *Eclipse* [12]. Pro práci s maticemi se využívá balíček *JAMA* [13], který umožňuje základní operace s maticemi (toto bylo např. potřeba u metody nejmenších čtverců k výpočtu inverzní matice) a dále jsou ještě využívány funkce z balíku *Java Advanced Imaging* (dostupné z [14]).

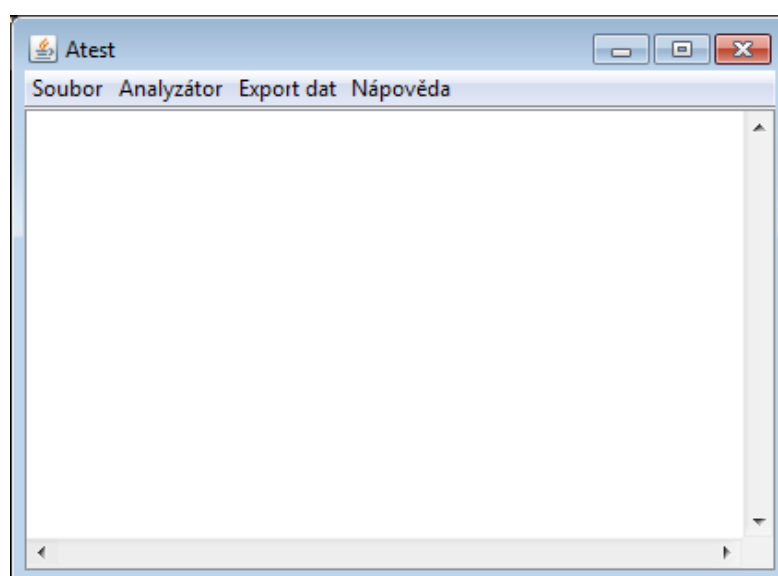
Aplikace se nachází na přiloženém DVD v adresáři nazvaném „*Aplikace*“ a je nejprve nutné tento adresář s veškerým jeho obsahem překopírovat na disk na místo, kam máme oprávnění zapisovat (k tomuto účelu bohatě postačí dnes hojně rozšířený flash disk). Zápis na disk je nutný kvůli ukládání vlastní konfigurace aplikace, konfigurace analyzátoru (při změně těchto konfigurací uživatelem) a pro otevření reportu po dokončení analýzy obrazu. Pokud oprávnění zapisovat není, pak aplikaci nelze spustit přímo a není zaručena plná funkčnost (např. se automaticky neotevře report).

Aplikace se spustí poklepnáním na soubor *Atest.bat*, což je dávkový soubor obsahující příkazy pro spuštění souboru *Atest.jar* přes příkazovou řádku, případně by se na některých systémech mohla aplikace spustit rovnou poklepnáním na soubor *Atest.jar* (tento způsob nedoporučujeme, neboť aplikace může postrádat nějaká nastavení, která jsou jinak obsažena v dávkovém souboru).

Dávkový soubor obsahuje, kolik paměti se má aplikaci přidělit. Pro současnou předlohu stačí 256MB, pokud bude zpracovávána hustější předloha či předloha s vyšším DPI je nutno přidělit více paměti - nutno v *Atest.bat* přepsat „-*Xmx256M*“ na „-*Xmx1024M*“. Jako poslední parametr je v dávkovém souboru toto: „*1>Atest.log 2>&1*“ – jedná se o nastavení zachycování textového výstupu do souboru.

Pro korektní běh aplikace je doporučeno mít nainstalované běhové prostředí (*JRE – Java Runtime Environment*) *Java* od firmy *Oracle* ve verzi 6 nebo vyšší (ke konci května 2013 byla na stránkách uvedených v [14] dostupná verze *Java 7 Update 21*, instalační soubory *JRE* verze 6 i 7 jak pro 32-bit tak i pro 64-bit operační systémy *Windows* se nachází na přiloženém DVD v adresáři „*Instalacni_soubory*“).

Po spuštění se zobrazí standardní okno aplikace, se kterým je možno operovat stejným způsobem jako s většinou jakýchkoli jiných programů, dle zažitých zvyklostí. Okno je tedy možno libovolně přesouvat, zmenšovat a roztahovat jak nám vyhovuje a samozřejmě jsou v záhlaví okna vpravo přítomny ikony umožňující minimalizovat okno na lištu, maximalizovat jej a ukončit aplikaci (zavřít okno).



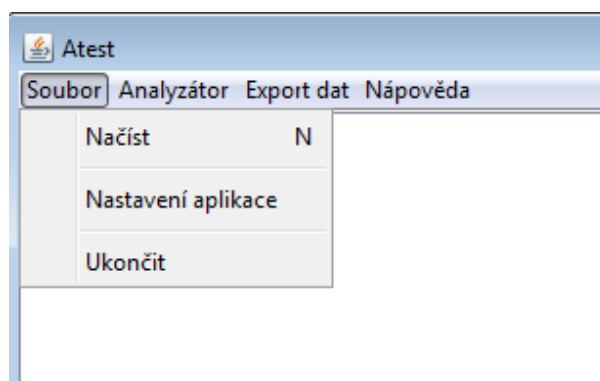
Obrázek 5.1: Okno aplikace

Jak je vidět na obr. 5.1, v hlavní nabídce se nachází 4 položky:

- Soubor,
- Analyzátor,
- Export dat,
- Nápověda,

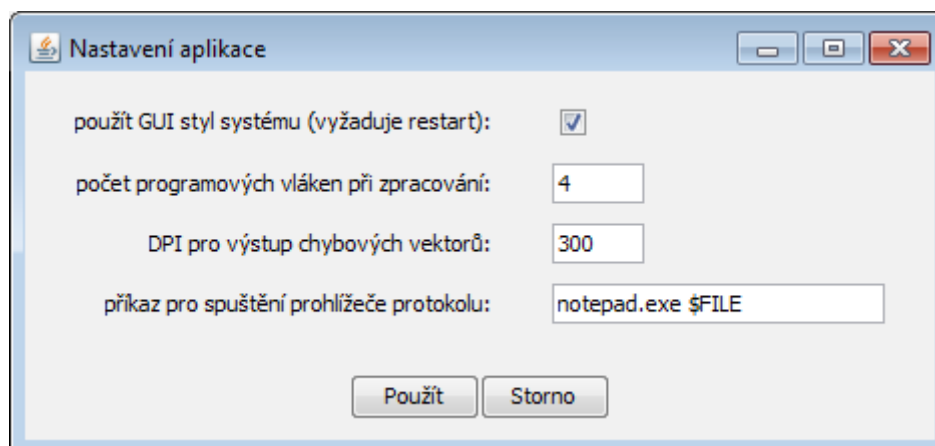
a postupně bude při popisu práce s aplikací vysvětleno, co se pod kterou položkou hlavní nabídky nalézá.

Pod záložkou *Soubor* se nachází tři položky (obr. 5.2), přičemž poslední z těchto položek s názvem *Ukončit* představuje alternativu ukončení aplikace (na předchozí stránce bylo vysvětleno, že aplikaci lze ukončit stisknutím obvyklého tlačítka křížku na záhlaví okna). Než budou vysvětleny ostatní položky ještě se podíváme na menu *Nápověda*: obsahuje informace o programu a nápovědu k programu, která se otevírá ve formátu **.pdf* a dále ještě obsahuje informace o programu.



Obrázek 5.2: Nabídka položky *Soubor*

Po klepnutí na druhou položku v tomto menu se otevře okno *Nastavení aplikace* (na obr. 5.3 lze spatřit výchozí nastavení pro aplikaci), ve kterém je možno změnit grafický vzhled aplikace, pokud by stávající vzhled (používající styl systému) nevyhovoval. Změnu vzhledu je nutné potvrdit klepnutím na tlačítko *Použít* a je nutné restartovat aplikaci. Po spuštění má pak aplikace výchozí grafický vzhled, který nepřebírá nastavení vzhledu oken systému. Na obr. 5.4 si můžeme prohlédnout jak tento vzhled vypadá. Vracení aplikace do původního stavu vzhledu se provede zaškrtnutím příslušné volby a dále se již postupuje jak je popsáno výše.

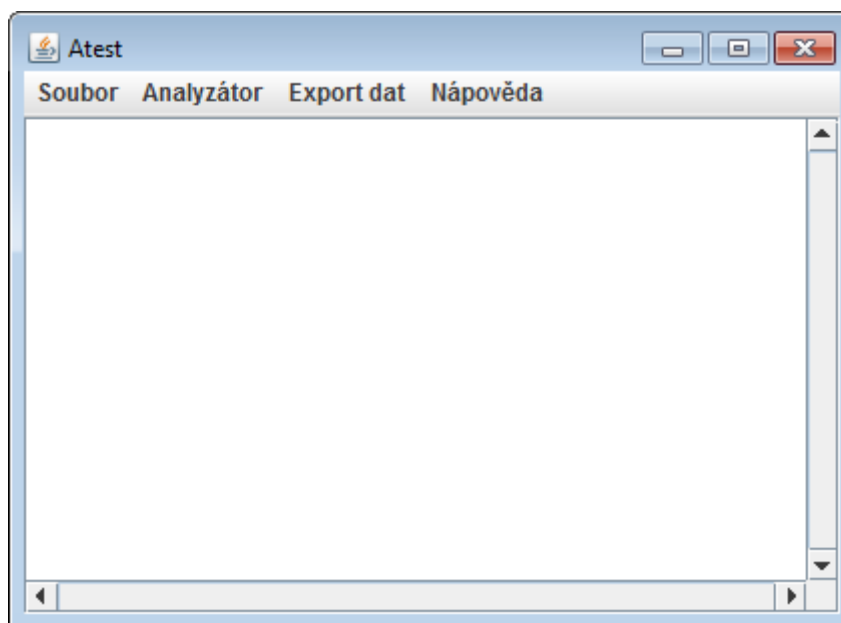


Obrázek 5.3: Nastavení aplikace

V okně *Nastavení aplikace* lze dále ještě nastavit počet programových vláken, ve kterých bude probíhat výpočet (počet programových vláken při zpracování určuje rychlost zpracování analýzy; výchozí hodnotou je 20 vláken a při vyšším počtu již rychlost zpracování významně neklesá).

Další položku tvoří možnost nastavení rozlišení obrázku vektoru odchylek (bližší popis uložení tohoto obrázku je možno nalézt v kap. 5.3).

Poslední položkou v *Nastavení aplikace* se zadává, v jakém programu se má otevřít report s výsledky analýzy. Je nutno vložit do pole celou cestu k požadovanému programu a za cestu zadat mezeru a napsat tyto znaky: „*\$FILE*“ (ekvivalentní příkazové řádce), přestože na obr. 5.3 se u výchozího prohlížeče reportu nenachází celá cesta – toto je způsobeno tím, že cesta k programu *notepad.exe* se nachází uvnitř aplikace. Ukázka vyplnění pole pro otevření reportu v Internet Exploreru (cesta se může u jednotlivých počítačů lišit): *"c:\Program Files\Internet Explorer\iexplore.exe" \$FILE*.



Obrázek 5.4: Druhý možný grafický vzhled okna aplikace

5.1 Načtení obrázku

Volba načtení obrázku naskenované předlohy se nachází v menu *Soubor* → *Načíst* (obr. 5.2) a má přiřazenu klávesovou zkratku „N“, tzn. po stisku této klávesy dojde k otevření dialogového okna. V dialogovém oknu si vybereme, v jakém adresáři se nachází data, která chceme zpracovat (obrázek ve formátu *.gif, *.tif, *.tiff či *.png) a vybraný obrázek otevřeme. Aplikace si pamatuje poslední navštívený adresář a pokud uchovááme data v jednom adresáři nemusíme se k nim znovu proklikávat po příštím spuštění aplikace.

Aplikace umí vyhodnotit pouze obrázky, které jsou „na šířku“, tzn. pokud byla předloha skenována „na výšku“ je nutné obraz před načtením otočit v jiném programu.

5.1.1 Prohlížení obrázku

Po načtení obrázku do aplikace je možno si jej v okně aplikace libovolně prohlížet. Pohyb a posun obrázku lze realizovat několika způsoby:

1. Držením levého tlačítka myši a uchopením obrázku přičemž posunujeme myš v libovolném směru. Tímto způsobem se lze posunout maximálně o šířku (výšku) okna a to za předpokladu, že chytíme obrázek blízko okraje okna.
2. Přidržením pravého tlačítka myši spustíme kontinuální posun. Rychlost posunu můžeme ovlivnit vzdáleností kurzoru od bodu, kde jsem tlačítko stiskli. Rychlost a směr posunu je zobrazena jako červená šipka.
3. Poslední možností jak si prohlédnout obrázek jsou posuvné lišty nacházející se v okně aplikace vpravo a dole.

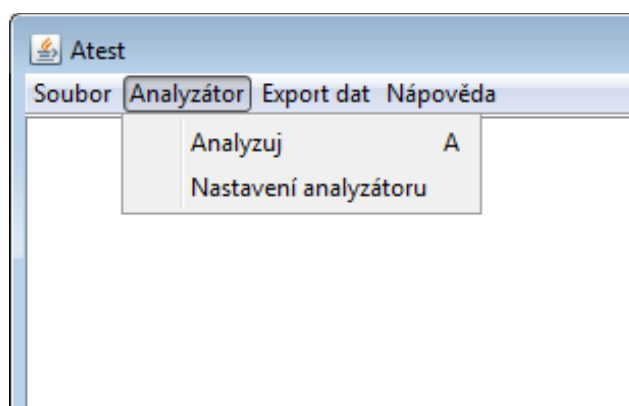
5.1.2 Přiblížování a oddalování obrázku

Stejně jako je při prohlížení obrázku více způsobů realizace jak si obrázek prohlédnout tak i u přiblížování/oddalování máme více možností. Výchozí úroveň přiblížení (čili zobrazení obrázku v okně) je taková, že jednomu pixelu na obrazovce odpovídá jeden naskenovaný pixel obrázku. Pro přiblížení/oddálení lze použít následujících metod:

1. Nejpřirozenějším pro přiblížení či oddálení je pohyb kolečkem myši.
2. Dále je možné zvětšit si (přiblížit si) obrázek kliknutím na prostřední tlačítko myši (tzn. na většině druhů myši tato akce znamená stisknutí kolečka) a po dobu jeho držení si pohybem myši vybrat obdélníkovou oblast, kterou chceme přiblížit.
3. Totožnou akci jako v předchozím bodu vykoná i kombinace stisku klávesy CTRL a přidržení levého tlačítka myši (opět si vybereme obdélníkovou oblast, kterou chceme přiblížit).
4. Kombinací stisku klávesy CTRL a pravého tlačítka myši dojde k vyresetování nastavení přiblížení na výchozí úroveň.
5. Kombinací stisku nejprve pravého tlačítka a poté současně kliknutím levého tlačítka myši taktéž dojde k vrácení přiblížení na výchozí úroveň.

5.2 Analýza obrazu předlohy

Provedení analýzy naskenovaného obrazu předlohy lze spustit pouze pokud máme načtený naskenovaný obraz testovací předlohy a to buď přes nabídku v hlavním menu *Analyzátor* → *Analyzuj* (obr. 5.2.1) anebo stiskem klávesy „A“. Analýza spočívá v nalezení středů křížků (středů průsečíků mřížky) na naskenovaném obraze předlohy a výpočtu charakteristik geometrické přesnosti skeneru.



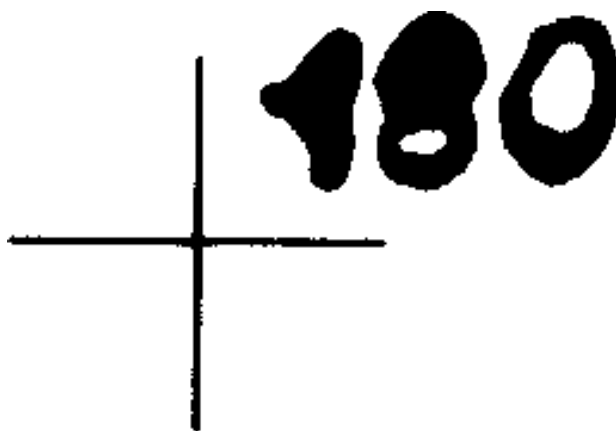
Obrázek 5.2.1: Nabídka položky *Analyzátor*

Jak je vidět na obrázku výše je možné otevřít si okno s nastavením analyzátoru přes menu *Analyzátor* → *Nastavení analyzátoru* (okno nastavení analyzátoru se nachází na obr. 5.2.3). Nastavení je nutné provést ještě před započítím analýzy, např. pokud došlo ke změně rozlišené naskenovaného obrázku či je potřeba změnit referenční souřadnice středů mřížky.

V pravé části okna nastavení analyzátoru se nachází tabulka se souřadnicemi středů průsečíků mřížky, přičemž jestliže nad kterýmkoli řádkem tabulky klikneme pravým tlačítkem myši objeví se nabídka přidání řádku před či za řádek nad kterým jsme provedli klik myši, případně můžeme vybraný řádek odstranit. Do tabulky se souřadnicemi středů průsečíků mřížky lze tedy libovolně přidávat další středy a také lze přepisovat jejich souřadnice. Zkušený uživatel může provést výměnu souřadnic ve složce *analyzer.ap*, což je archiv, který nemá standardní příponu *.zip* (aby pro uživatele nebylo „lákavé“ soubory dekomprimovat), ale po změně přípony *.ap* na *.zip* lze přepsat

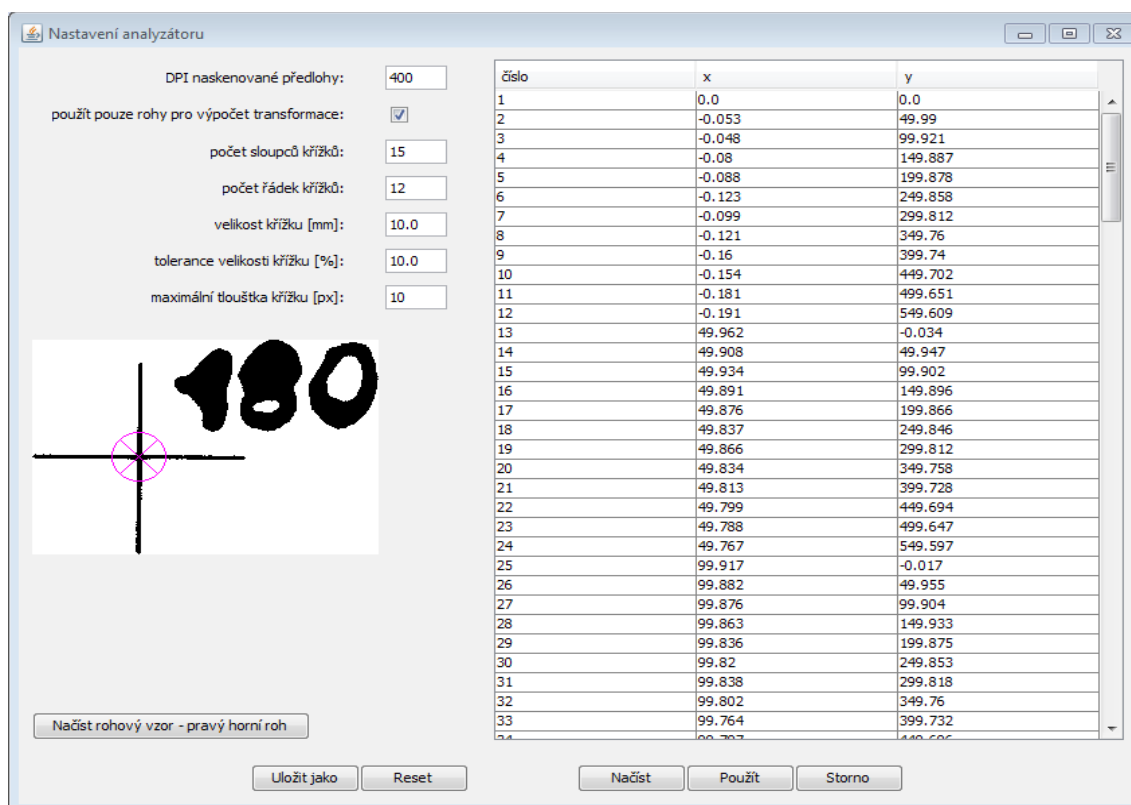
soubor se souřadnicemi *raster.csv*. Je nutné dodržet následující formát souřadnic: desetinné místo oddělené tečkou, mezi souřadnicemi je bez mezery čárka (příklad souřadnice: 99.917, -0.017)

V levé části okna analyzátoru se nastavuje rozlišení naskenované předlohy (počet DPI) či zda chceme pro výpočet klíče afinní transformace naskenovaného obrazu předlohy použít pouze rohové anebo všechny křížky. Dále je možno nastavit počet sloupců a řádků křížení (tzn. kolik středů křížení mřížky se nachází ve sloupci a řádku předlohy), velikost křížku v milimetrech, toleranci velikosti křížku a maximální tloušťku křížku v pixelech. Poslední parametr nastavitelný před započítím analýzy představuje načtení rohového vzoru – vždy se musí jednat o pravý horní roh předlohy při standardní orientaci (tzn. když se díváme na v současnosti používanou předlohu máme levý dolní roh označený jako č. 1, levý horní roh má č. 12, pravý dolní roh má č. 169 a pravý horní roh je označen číslem 180, viz obrázek rohového vzoru 5.2.2). Pokud se změní rozlišení při skenování (dnes se používá hodnota 400 DPI), je nutné vytvořit si z naskenovaného obrazu příslušný výřez pravého horního rohu a před jeho načtením je potřeba nastavit příslušné DPI (rohový vzor musí mít stejné rozlišení jako analyzovaný obraz předlohy.), eventuálně i velikost či maximální tloušťku křížku. Na přiloženém DVD se v adresáři „*Rohove_vzory*“ nachází rohový vzor pro současnou testovací předlohu v rozlišení 400 DPI a také rohový vzor pro možnou potenciální předlohu v rozlišení 600 DPI. Načtení rohového vzoru je možné po kliknutí na stejně pojmenované tlačítko v levé části okna nastavení analyzátoru – v dialogovém okně si vybereme cestu k rohovému vzoru a otevřeme jej (formát obrázku vzoru musí být buď **.gif*, **.tif*, **.tiff* či **.png*). Na rohovém vzoru mřížky se poté automaticky zaměří střed průsečíku (pozor na nutnost stejného DPI vzoru rohového křížku a testované předlohy).



Obrázek 5.2.2: Vzor rohového křížku č. 180

Na snímku okna nastavení analyzátoru (obr. 5.2.3) jsou nastaveny výchozí hodnoty pro analýzu současné testovací předlohy, přičemž tato konfigurace je uložena v souboru aplikace *analyzer.backup.ap*, který se nachází v adresáři „*config*“ v adresáři s aplikací. Pokud změníme hodnoty v nastavení analyzátoru a rozmyslíme si jejich použití můžeme si tlačítkem *Reset* zobrazit původní hodnoty, které byly zadány v nastavení analyzátoru při jeho otevření. Veškeré změny, které v nastavení analyzátoru provedeme, můžeme rovnou použít (stiskem tlačítka *Použít*) anebo si můžeme celou konfiguraci analyzátoru uložit do libovolného adresáře pomocí tlačítka *Uložit jako*, přičemž jestliže bychom pak chtěli načíst nějakou uloženou konfiguraci tak klepneme na tlačítko *Načíst* a příslušný soubor s konfigurací analyzátoru si načteme. Vždy po spuštění aplikace jsou v nastavení analyzátoru ty hodnoty, které se v něm nacházely při posledním použití, aplikace si je ukládá do souboru *analyzer.ap*.



Obrázek 5.2.3: Okno nastavení analyzátoru

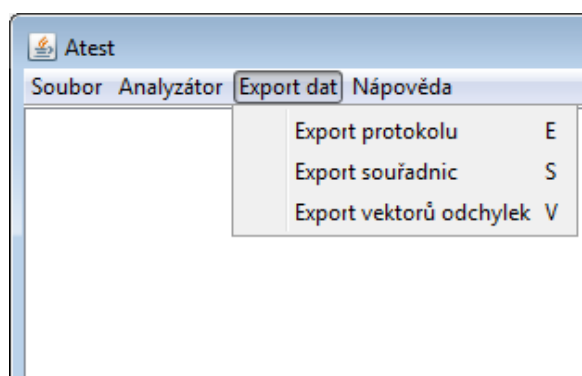
5.3 Export a vizualizace výsledků

Okamžitě po dokončení analýzy obrazu naskenované předlohy se v předem zvoleném programu otevře protokol s názvem *_report.tmp* (výchozím programem ve kterém se report otevírá je poznámkový blok, který je součástí operačního systému *Windows*, nastavení výchozího programu je konfigurovatelné v okně *Nastavení aplikace*). V protokolu jsou vypsány výsledky testování, konkrétně se jedná o:

- transformační klíč afinní transformace,
- chyby transformačního klíče,
- chyby souřadnic.

Dále se v protokolu ještě nachází seznam souřadnic průsečíků mřížky s jejich chybami. Jsou uvedeny jak referenční souřadnice průsečíků v [mm], místní souřadnice v [px] tak i transformované souřadnice v [mm].

Protokol je možné rovnou uložit v poznámkovém bloku přes volbu *Soubor* → *Uložit jako*, pojmenovat jej můžeme libovolně a zvolíme si místo pro uložení. Pokud si protokol neuložíme tímto způsobem a zavřeme si poznámkový blok nemusíme se obávat, že bychom ztratili výsledky testování, protože soubor s názvem *_report.tmp* se automaticky ukládá do adresáře aplikace *app* a je tedy možno si jej přejmenovat a překopírovat kam potřebujeme. Poslední možností jak si uložit protokol je přes hlavní nabídku menu aplikace *Export dat* → *Export protokolu* (obr. 5.3.1) či stisknutím klávesy „E“ (další z několika klávesových zkratk). V tomto případě je pojmenování automatické podle názvu analyzovaného obrazu, přípona je pak **.txt* a je nabídnuto uložení do adresáře kde se nachází testovaný obraz



Obrázek 5.3.1: Nabídka položky *Export dat*

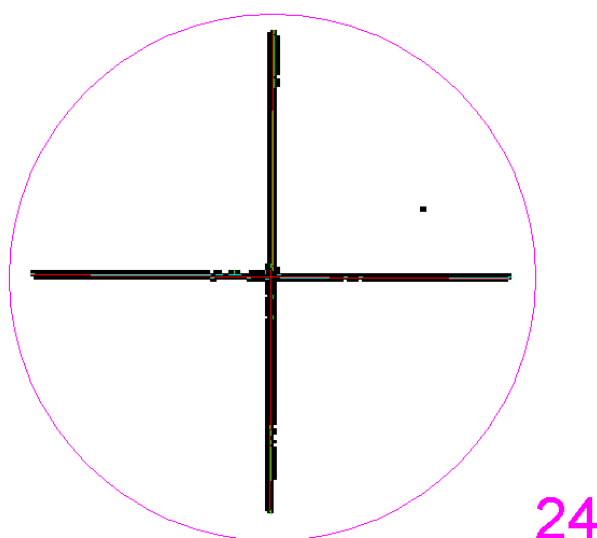
Dále je možné exportovat výsledné transformované souřadnice průsečíků mřížky a to opět buď přes nabídku hlavního menu *Export dat* → *Export souřadnic* anebo stiskem klávesové zkratky „S“. Název i místo pro uložení souboru se nabídnou automaticky dle názvu a uložení testovaného obrazu. Export má podobu souřadnic průsečíků (středů) oddělených čárkou a na každém řádku se nachází souřadnice jednoho středu (formát **.csv*).

Poslední položkou v menu *Export dat* je *Export vektorů*, což představuje možnost uložit ve formátu **.png* vektorovou mapu odchylek (obrázek vektorové mapy

odchylek jde taktéž uložit po stisku klávesové zkratky „V“). Název souboru a místo pro uložení se opět automaticky nabídnou. Vektorová mapa odchylek představuje vizualizaci rozdílu polohy transformovaných souřadnic z naskenovaných dat oproti referenčním souřadnicím. Vektory odchylek představují směr a velikost posunu souřadnice průsečíku mřížky (čím větší posun vzdálenosti tím větší vektor u souřadnice). Obrázek vektorové mapy odchylek se nachází v příloze A.

5.3.1 Vizualizace nalezených středů průsečíků mřížky

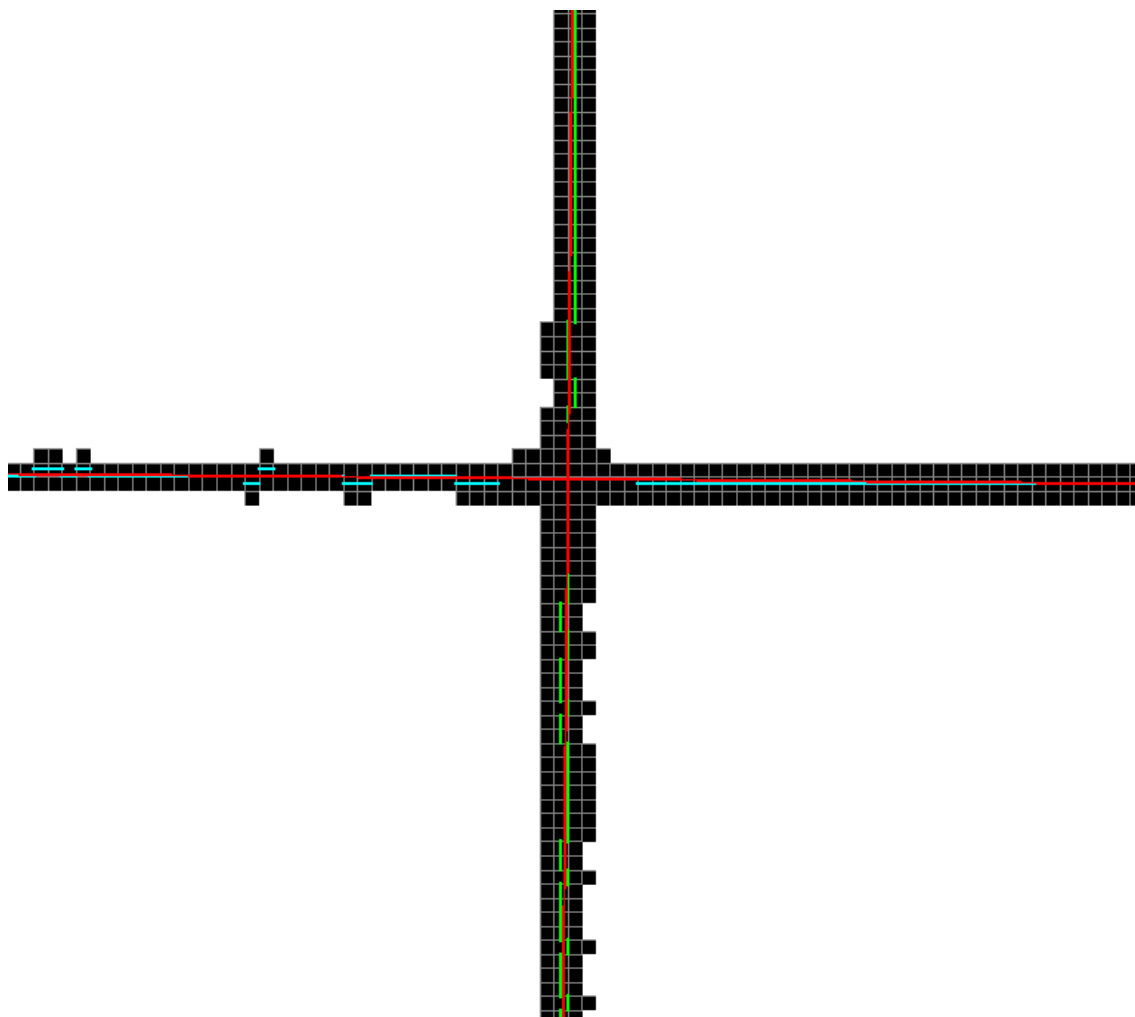
V hlavním okně aplikace se po provedení analýzy vykreslí okolo každého nalezeného středu průsečíku mřížky kružnice s číslem křížku (obr. 5.3.2). Číslo středu křížku se vykreslí jen pokud byly spočteny transformované souřadnice středu z místní soustavy do referenční.



Obrázek 5.3.2: Vykreslení čísla a kružnice okolo středu

Zároveň jsou ještě vykresleny body horizontálního a vertikálního skeletu jako barevné čárky (zeleně jsou vykresleny body horizontálního skeletu a modře jsou vykresleny body vertikálního skeletu). Červeně jsou pak vykresleny přímky vzniklé

proložením bodů skeletu pomocí MNČ, přičemž v místě jejich průsečíku se nalézá střed křížku. Ukázka vykreslení bodů skeletu a přímek se nachází na obr. 5.3.3, kdy se v aplikaci použilo velkého přiblížení, při kterém jsou pak velmi dobře viditelné jednotlivé pixely, jež jsou odděleny od sebe světlejší čarou.



Obrázek 5.3.3: Detail vykreslení

6 Návrh nové testovací předlohy

V současnosti se pro testování skenerů používá testovací předloha o rozměrech 700×550 mm, na které je síť křížků, jejichž středy jsou od sebe vzdáleny 50 mm. Souřadnice středů křížků jsou pravidelně proměřovány a z kontrol vyplývá, že je testovací předloha rozměrově stálá. Problémem ovšem je, že Zeměměřický úřad má již jen jednu testovací předlohu (jak již bylo zmíněno v kap. 2.2) a bylo by vhodné navrhnout předlohu novou.

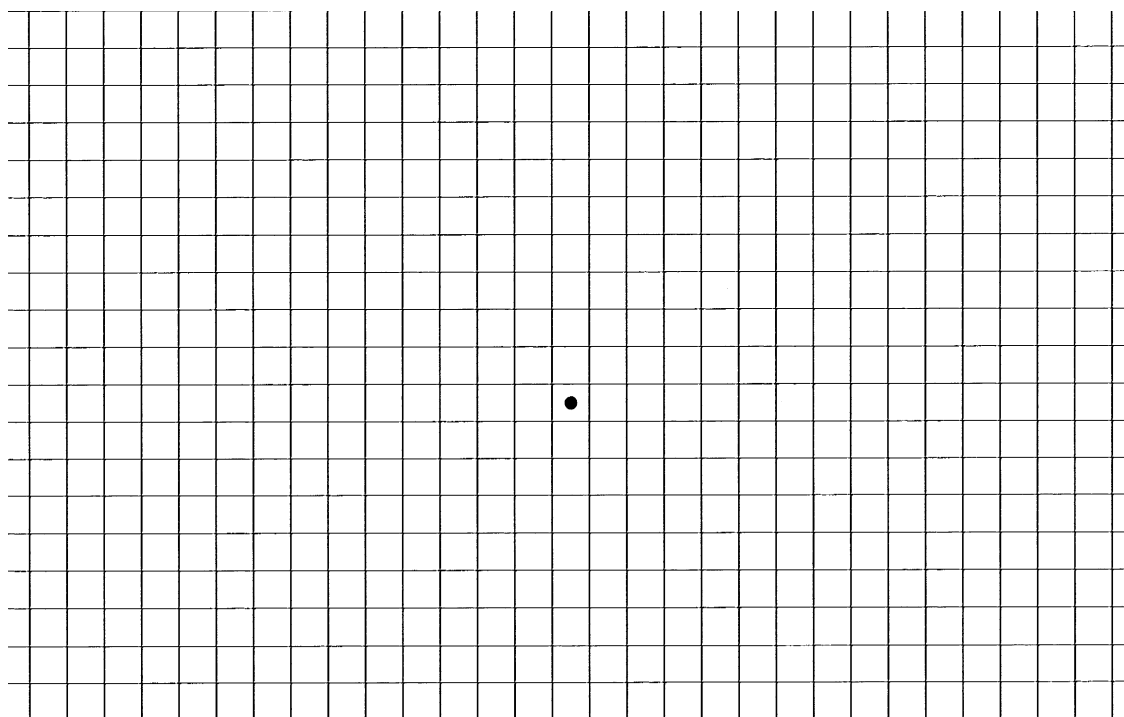
Při návrhu bylo zejména potřeba brát v úvahu rozměrovou stálost materiálu (hlavní kritérium), na kterém bude vzor nanesen, snadnost přepravy s předlohou (ne vždy je k dispozici automobil) a dále bylo potřeba navrhnout takový vzor, který bude mít body zájmu rozvrženy hustěji než jak je má původní předloha. Hustější mřížka je žádoucí z hlediska detailnějšího popisu chyb skeneru, zároveň je ale logické, že při zvýšeném počtu porovnávaných souřadnic by zpracování muselo probíhat automaticky (časově náročná je zejména vektorizace naskenované předlohy – viz úvod práce – dnes vektorizace jedné naskenované předlohy trvá jeden den).

Při úvahách o materiálu předlohy byl samozřejmě okamžitě zavržen papír, neboť nedosahuje požadované rozměrové stálosti, nehledě na to, že je náchylný na jakékoliv poškození (ať již kapalinou či mechanicky).

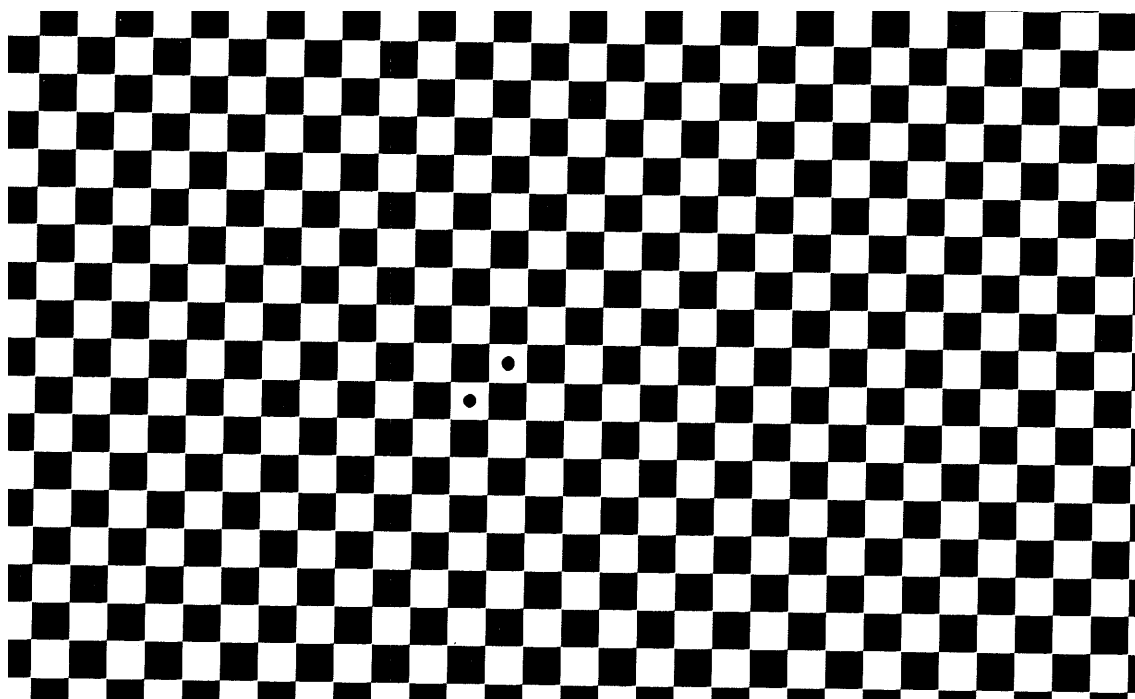
Dále bylo uvažováno o tom, že by vzor mohl být nanesen na PET fólii (podobně jako byly dříve katastrální mapy). Problémem by v dnešní době nebyl tisk na PET fólii (původní předloha je vyryta) ani jeho přesnost a trvanlivost. Problematická bude nejdůležitější vlastnost předlohy – rozměrová stálost. PET fólií existuje mnoho druhů, v ZÚ ji zkoušeli a dospěli k závěru, že fólie nevykazovala dostatečnou rozměrovou stálost. Bohužel dříve používaný materiál Astralon (vinylchloridmethylmethakrylátový kopolymer [18]), který měl vysokou rozměrovou stabilitu při zachování flexibility a odolnosti vůči světlu [11], se již nevyrábí.

Po úvaze jsme dospěli k závěru, že z dostupných materiálů se jako nejvhodnější jeví tenká hliníková fólie, která splňuje nároky na rozměrovou stálost (nepodléhá srážce působením vlhkosti a je jen minimálně ovlivněna změnami teplot).

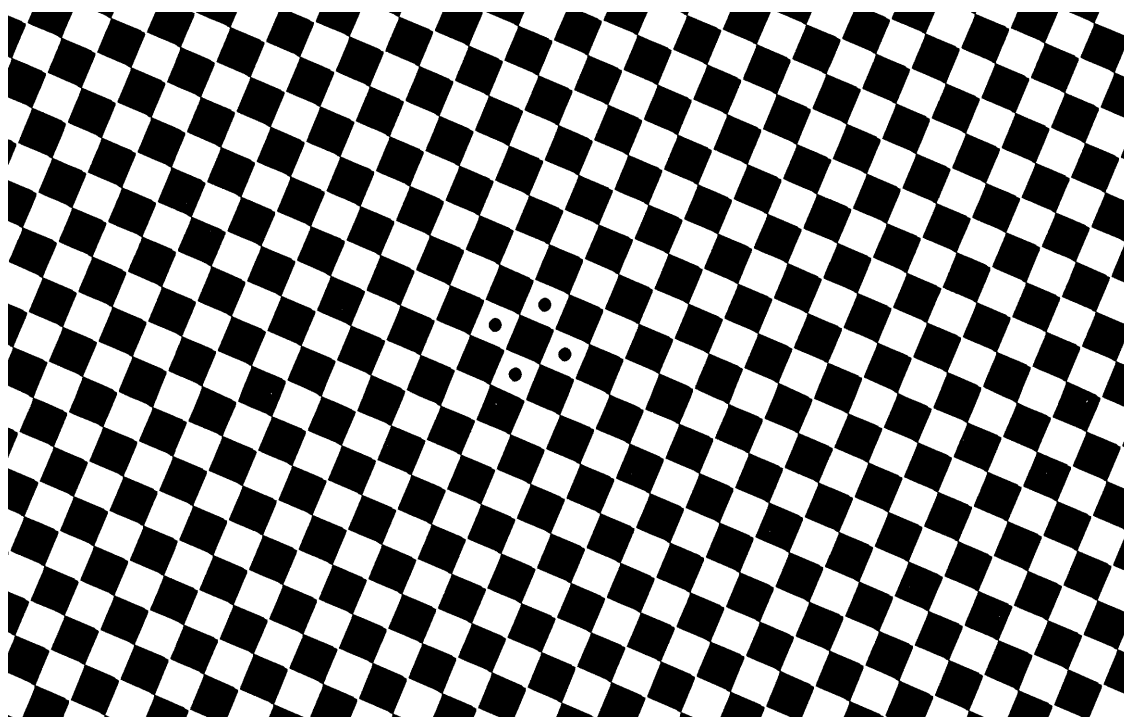
Rozměry předlohy měly být dle zadání ZÚ 750×600 mm s přesahem asi 5 mm (příčemž zájem by byl o síť úseček maximálně po 25 mm). Vzory byly navrženy vedoucím diplomové práce tři, prvním z nich je síť úseček po 5 mm, druhým je šachovnicový vzor s velikostí čtverců šachovnice 5 mm a třetím vzorem je opět šachovnice se stejným rozměrem čtverců ovšem s natočením o 70° . Pro orientaci vzoru byly ke třem rohům požadovaných rozměrů umístěny tečky (do levého horního rohu jedna tečka, do pravého horního dvě a do levého spodního čtyři tečky). Ukázka vzorů předloh s tečkami je na obrázcích 6.1, 6.2 a 6.3.



Obrázek 6.1: Výřez vzoru sítě úseček s rohovým identifikátorem jedné tečky



Obrázek 6.2: Výřez šachovnicového vzoru s rohovým identifikátorem dvou teček



Obrázek 6.3: Výřez vzoru šikmé šachovnice s rohovým identifikátorem čtyř teček

6.1 Výroba předlohy

Přenesení vzoru na hliníkovou fólii (desku) proběhlo v Plzni v tiskařských závodech TYPOS [17]. Vzor byl na fólii přenesen litograficky (tzn. tiskem z desky), přičemž deska byla vyrobena na osvitové jednotce přístroje KODAK TRENDSETTER VLF Platesetter, který exponuje s velkou přesností a používá laserový systém expozice nazývaný KODAK SQUARESPOT [16].

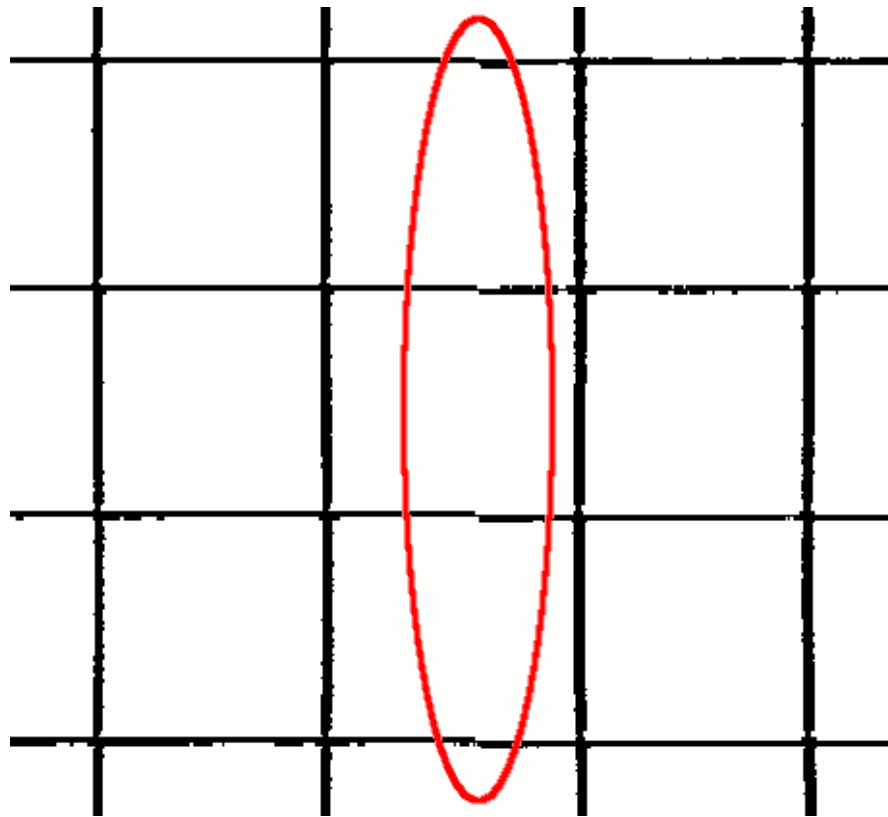
Předloha byla na hliníkovou fólii nanášena po celé její šířce a téměř po celé její výšce (na dolní straně fólie se nacházejí informace potřebné k tisku), vzor má tedy větší přesah než jaký byl požadován (rozměry desky jsou cca 1030 × 800 mm).

Takto jsme nechali zhotovit všechny tři vzory – výsledkem jsou tedy tři hliníkové fólie, každá s jiným vzorem.

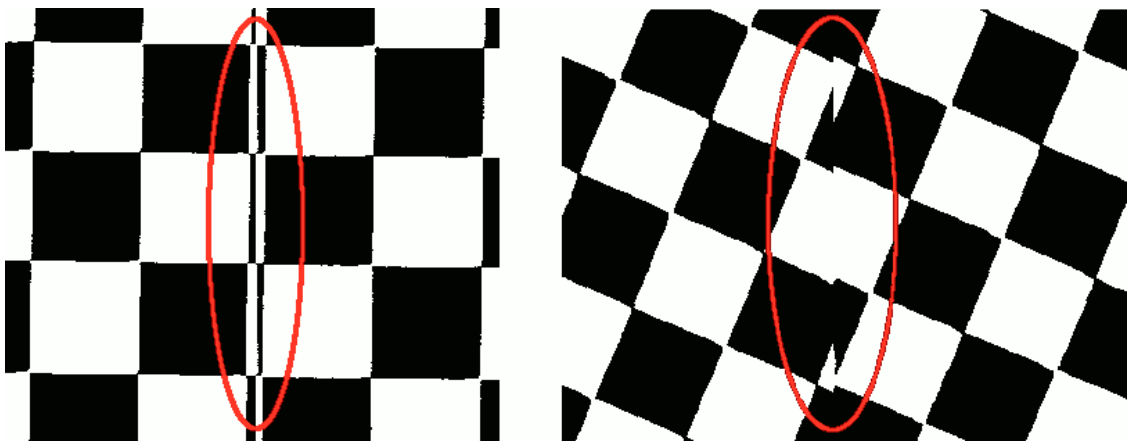
6.2 Skenování předloh

Pro zjištění, zda jsou nové předlohy použitelné v praxi (zejména zda by bylo možno alespoň materiálem nahradit stávající předlohu) bylo nejprve provedeno jejich skenování. Následující kapitola 6.3 se pak věnuje shrnutí zkušeností ze skenování předloh a jsou hodnoceny i další aspekty práce s předlohami.

Nejprve byly předlohy naskenovány na velkoformátovém skeneru nacházejícím se na KMA na FAV (ZČU). Každá z předloh byla naskenována s rozlišením 600 DPI šedotónově v nekomprimovaném formátu **.tif* – výsledkem jsou dosti velké soubory (každý má velikost téměř 500 MB). Již při tomto prvotním skenování se objevil problém – z důvodu typu a konstrukce skeneru vznikaly v naskenovaných datech oblasti nespojitosti (předloha se pohybuje při průchodu skenerem a hliníková fólie místy nedoléhala na skleněnou desku skeneru; nejvíce toto bylo patrné v místech, kde na sebe navazují obrazy dvou snímačů). Ukázka nespojitostí na předlohách viz obrázky 6.2.1 a 6.2.2.



Obrázek 6.2.1: Ukázka nespojitosti na předloze tvořené úsečkami



Obrázek 6.2.2: Ukázka nespojitostí na šachovnicových předlohách

Dále ještě byly předlohy naskenovány na Katastrálním úřadu pro Plzeňský kraj. Typ skeneru na KÚ pro Plzeňský kraj je totožný se skenerem na KMA (taktéž se pohybuje předloha). Pro získání většího množství dat byly předlohy skenovány s menší hodnotou prahu (tloušťka čáry u mřížky je místy tenčí než 3 pixely) 1x černobíle při rozlišení 600 DPI a 1x šedotónově při 300 DPI (ve formátu *.tif). Stejně jako na skeneru nacházejícím se na KMA nedošlo ke korektnímu naskenování – opět na sebe vzor na některých místech nenasazuje jak by měl.

V rámci získání dat i z jiného typu skeneru bylo provedeno skenování na skeneru stolového typu v Zeměměřickém úřadu v Praze. Předlohy byly skenovány při rozlišení 600 DPI černobíle a 300 DPI šedotónově a uloženy byly ve formátu *.tif. U tohoto skeneru sice nedocházelo k nespojitostem vzoru na naskenovaném obraze (toto je dáno principem stolového skeneru – nepohybuje se u něj předloha nýbrž dochází k pohybu snímače) ale objevila se celá řada jiných problémů. Snímač má mít v menším rozměru 600 mm, ovšem šíře záběru snímače je ve skutečnosti o něco menší – 594 mm, navíc tento rozměr závisí na zvoleném režimu skenování. Protože jsou na předlohách kvůli orientaci umístěny ke třem rohům požadovaných rozměrů (750 × 600 mm) značky (tečky) nedošlo bohužel vlivem menší šíře záběru snímače k jejich úplnému naskenování (buď byly na jedné straně naskenovány všechny čtyři tečky ale pak už se nenaskenovala jediná tečka v levém horním rohu anebo byly naskenovány tečky po podélné straně předlohy ale už se nenaskenovaly všechny 4 tečky na straně druhé). Další problém se objevil u šachovnicového vzoru, kdy docházelo na černých šachovnicových polích k efektu velkých bílých teček uprostřed (zřejmě kvůli odrazivosti předlohy) a podařilo se nám je odstranit až po vyzkoušení celé řady možných skenovacích režimů, přičemž v některém z režimů se naskenovaný obrázek uložil s invertovanými barvami (tzn. černá je bílá a opačně). Bohužel tyto problémy nejsou zdokumentovány obrazovými daty, neboť jsme takováto skenování považovali za „chybná“ a pokračovali jsme dalším skenovacím režimem.

Obrazová data z těchto tří skenování jsou na příloženém DVD v adresáři „*Skenovane_predlohy*“.

6.3 Zhodnocení použitelnosti předloh

Jak již bylo popsáno v předchozí kapitole, skenování předloh provázela celá řada problémů. V této kapitole budou shrnuty poznatky ze skenování, práce s předlohou a možnost případného automatického vyhodnocení obrazu předlohy (analýza předlohy aplikací vytvořenou v rámci této diplomové práce).

- Po výrobě předloh byla ze strany ZÚ vydána zamítavá reakce na proměření všech souřadnic průsečíků mřížek, maximálně po osvědčení předloh by nechali proměřit např. každý pátý průsečík. Z tohoto důvodu se zpětně jeví síť, ve které jsou středy průsečíků vzdáleny 5 mm jako příliš hustá, dochází ke zbytečnému skenování bodů, které by se pak musely při eventuálním automatickém zpracování na počítači filtrovat. Navíc by výsledný obraz zabíral více místa než když by hustota bodů byla nižší.
- Při skenování se ukázalo, že rozměr předlohy je příliš velký (byť jen o necelý centimetr) – na skeneru stolového typu v Praze v ZÚ byla šíře záběru snímače menší než 600 mm a nedošlo tak k naskenování rohových identifikátorů natočení (identifikátory jsou vzdáleny právě 600 mm).
- Hliníková fólie místy nedoléhala na skleněnou desku skeneru, nejvíce to bylo patrné v místech překrytu snímačů. Tento problém zapříčiňuje, že se v obraze naskenované předlohy vyskytnou místa, na kterých vzor nenavazuje.
- Dále jsme po pečlivějším promyšlení dospěli k závěru, že tečky jako identifikátory natočení předlohy nejsou nejlepším řešením (obzvláště v kontextu faktu nejasně ohraničené předlohy). Dále by bylo lepší, aby byl identifikátor u každého z rohových křížků (a ne jen u třech). Je potřeba jasně identifikovat síť křížků, ať již z počítačového, tak i z pohledového hlediska. Při pohledové kontrole se jeví lépe, když je okamžitě jasné, jestli byla předloha dobře identifikována. Tečky nejsou na první pohled dobře odlišitelné. Identifikátorem může být libovolný útvar, pro každý roh jiný. Jako příklad můžeme uvést číslice. Z počítačového hlediska by postačil identifikátor jen u jednoho rohu. Při použití metody srovnávání se vzorem bude pro co nejlepší výsledky hodnotící funkce

žádoucí, aby se identifikátory od sebe co nejvíce lišily (toto samozřejmě platí i pro jiné metody).

- Dále se při skenování ukázalo, že kontinuální vzor po celé ploše hliníkové fólie také není nejlepším řešením. Zbytečně narůstá velikost obrazu naskenované předlohy a celkově je předloha subjektivně nepřehledná – na první pohled není jasné, jaká oblast předlohy je předmětem analýzy – toto je i „vina“ hustého vzoru a teček sloužících jako identifikátorů, které se ve vzoru snadno „ztratí“. Protože je vzor kontinuální po téměř celé ploše fólie tak na jeho okrajích ulpívají otisky prstů.
- S velikostí předlohy souvisí i tento bod – předloha se převáží v kartonovém obalu o něco větším než jsou rozměry hliníkové fólie a celková manipulace s takovýmto „zavazdlem“ není příliš komfortní. Při přepravě autobusem dokonce došlo u jedné fólie k mírnému prohnutí v jednom místě. Předloha by se musela buď srolovat anebo převážet v automobilu a i v tomto případě je otázkou, zda by se fólie nějak nemohla nějak poškodit.
- Problémem u předloh (zejména u šachovnicového vzoru) je také to, že ačkoliv povrch se vzorem vypadá matně, ve skutečnosti je lesklý a může tak docházet k nežádoucím odrazům při skenování.

Přestože je materiál rozměrově stálý (což je základní požadavek) nebude vhodný z jiných důvodů – jak z hlediska přepravy, tak z hlediska manipulace s ním (hrozí nebezpečí říznutí se). Dále také nebude vhodný proto, že fólie není dostatečně poddajná aby při průchodu skenerem byla plně přitlačena a nedocházelo k nespojitostem ve vzoru v místech překrytu snímačů (toto platí u skenerů kde je více snímačů v jedné řadě).

Vzor na předlohách by postačil v menší hustotě – dle ZÚ by mohly být vzdálenosti středů průsečíků maximálně 25 mm (přesné proměření budoucích referenčních souřadnic předlohy je nákladné – provádí se ve VÚGTK; více o proměrování předlohy v kapitole 2.2). Z hlediska automatické vektorizace a vyhodnocení přesnosti skeneru by nebyl problém s vyhodnocením sítě úseček, aplikace analyzuje stávající předlohu tvořenou křížky a u sítě úseček by se nacházely

středů křížení, tzn. postačilo by v aplikaci nastavit analyzátor dle rozměrů vzoru (vzor by se zpracovával jako kdyby to byla síť křížků).

Šachovnicové vzory nejsou vhodné z toho důvodu, že samotná metoda zjišťování geometrické přesnosti skeneru není přesnější než úsečky. Zpracování šachovnicového vzoru by bylo náročnější, s křížky se pracuje „komfortněji“. Zároveň se musí najít takový práh, aby byla splněna podmínka, že vzor bude opravdu tvořit čtverce, tzn. že na rozích se budou dotýkat a ne překrývat či že mezi nimi bude mezera. Šachovnicový vzor je tedy silně závislý na prahování, chyba prahování je v intervalu 0-1, kdežto u úseček se chyba prahování vzájemně „vyruší“ – chyba prahování u náběžné hrany křížku může nabývat hodnoty 0-1, u sestupné hrany má pak negativní znaménko (náběžná hrana černá barva = 1, sestupná hrana bílá barva = 0). U prahování křížků postačí nastavit práh tak, aby byly splněny Pokyny ČÚZK č. 32 – tloušťka čáry křížku mezi 3 až 6 pixely.

V tuto chvíli byl materiál předloh shledán nevhodným z výše uvedených důvodů (nedoléhání hliníkové fólie při skenování z důvodu silnějšího materiálu, obtížné a nešikovné přepravování předlohy a malá mechanická odolnost vůči promáčklinám). Vzor by postačil s menší hustotou, s jinými identifikátory rohů a s jinými rozměry. Vhodná by mohla být síť úseček, kterou by byla aplikace schopna zpracovat.

6.4 Doporučení pro nové předlohy

Jelikož se neprokázala vhodnost nových předloh z výše uvedených důvodů, byly stanoveny obecné zásady pro případnou výrobu jiné předlohy. Doporučení vycházejí z nabytých zkušeností při skenování předloh, z požadavků ZÚ a také při jejich formulování bylo myšleno na to, že by bylo vhodné, aby předloha byla automaticky zpracovatelná ve vytvořené aplikaci (bez nutnosti vytvářet jiný kód schopný analyzovat nový vzor předlohy).

- S ohledem na možnosti stolového skeneru v ZÚ by maximální rozměry vzoru neměly přesáhnout 750×590 mm. Doporučuje se, aby mezi kresbou vzoru a okrajem materiálu, na kterém bude vzor nanesen, byl pás bez vzoru (abychom při manipulaci s předlohou zbytečně nesahali na vynesení vzor a nezanechávali případné otisky).
- Vzor by měl být hustější než jaký je nanesen na stávající testovací předloze, z hlediska případného zaměření souřadnic průsečíků se jeví jako optimální vzdálenost středů průsečíků 20–25 mm (požadavek ZÚ je maximálně 25 mm) – tzn. 4x tolik souřadnic průsečíků.
- Lepší popis chyb skeneru než síť úseček by mohla poskytnout síť úseček otočená o n stupňů, popřípadě 6 úhelníková mřížka – zpracování těchto vzorů by však již bylo mimo rozsah této diplomové práce.
- Identifikátory rohů by měly být od sebe co nejlépe odlišitelné a počítačově zpracovatelné. Toto dobře splňují například číslice 1, 2, 3 a 4. Tečky jsou také dobře odlišitelné, ale subjektivně se pohledem hůře rozpoznává který roh je který.
- Materiál by bylo potřeba aby kromě rozměrové stálosti poskytoval v ideálním případě i komfortní přepravu a skladování nenáročné na prostor. Bylo by zapotřebí provést rozsáhlé testování velké škály zřejmě PET fólií, zda by se našel takový materiál, který by odpovídal požadavkům na rozměrovou stálost.

S ohledem na výše uvedená doporučení byly vytvořeny dva vzory předlohy, které se zároveň co nejvíce blíží vzoru původní předlohy a aplikace by je měla umět analyzovat (odpadne tak nutnost upravovat či vymýšlet jiný algoritmus). Jeden vzor jsou opět křížky, jejichž středy jsou od sebe vzdáleny cca 20 mm. Druhým vzorem je síť úseček, jejichž vzdálenost od sebe je taktéž cca 20 mm. Oba vzory mají jako identifikátory v rozích číslice, které jsou orientovány stejným způsobem jako na původní testovací předloze (tzn. u levého dolního rohu se nachází číslice 1, u levého horního rohu číslice 2, u pravého dolního rohu je číslice 3 a u pravého horního rohu se

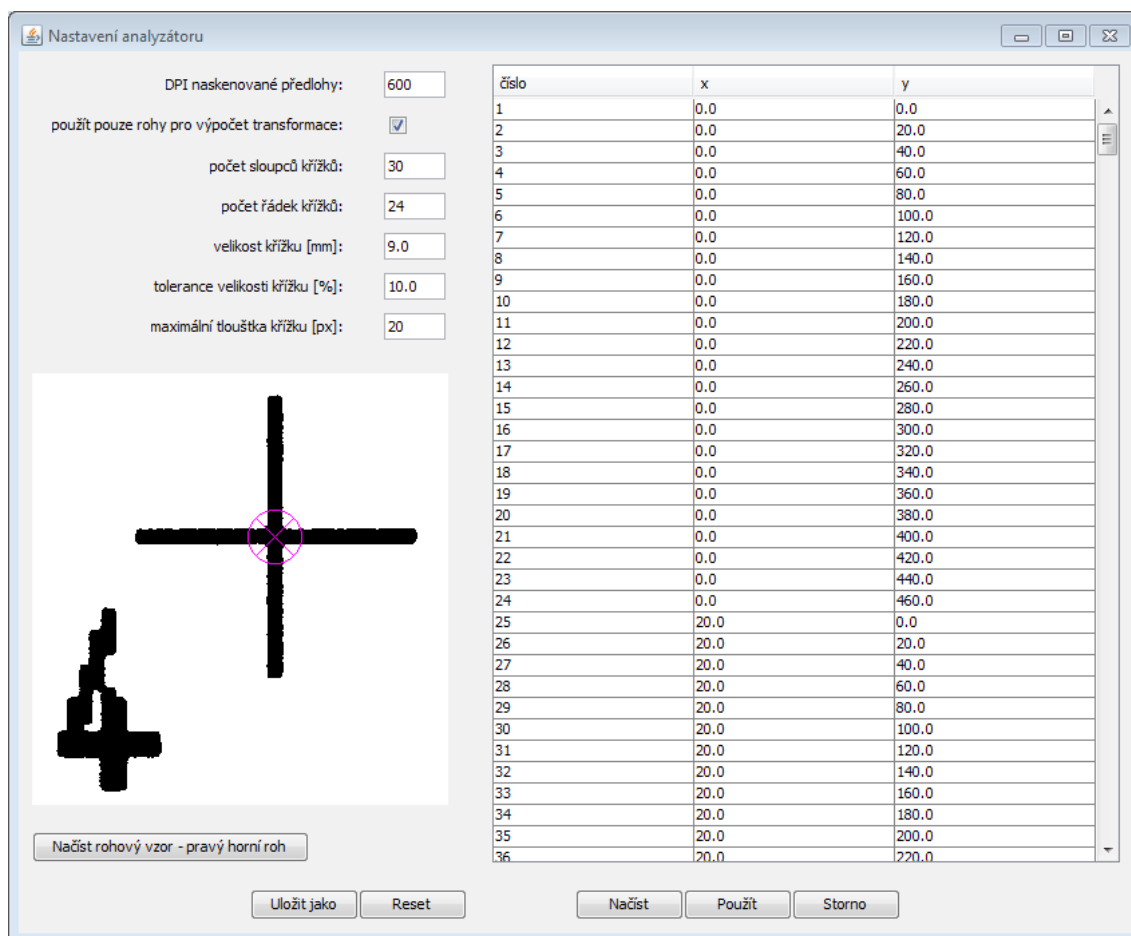
nachází číslice 4). Vzory byly vytištěny na papír a naskenovány černobíle při rozlišení 600 DPI ve formátu *.tif na skeneru nacházejícím se na KMA. Při skenování jsme bohužel nemohli dodržet požadavky uvedené v začátku 4. kapitoly – tloušťka úseček tvořících vzor je místy více než 10 pixelů. Je to dáno tím, že již při tisku byly úsečky vytištěny tlustěji než by bylo nutné. Pro účely získání naskenovaných předloh a jejich vyhodnocení aplikací však nesplnění požadavku tloušťky úseček nevádí. Obrazová data naskenovaných předloh se nachází na přiloženém DVD v adresáři „Skenovane_predlohy“.

6.4.1 Vyhodnocení jedné z navržených předloh aplikací

V následujícím textu si ukážeme postup nastavení a výstupy aplikace při vyhodnocení naskenovaného obrazu jedné z navržených papírových předloh. Pro demonstraci jsme vybrali vzor tvořený křížky na papírové předloze. Předloha byla skenována s rozlišením 600 DPI (výsledný obraz má $22\,400 \times 14\,400$ pixelů) a tloušťka úseček tvořících křížek je místy více než 10 pixelů (jak již bylo zmíněno výše).

Po spuštění aplikace a načtení předlohy je nutné změnit nastavení analyzátoru (viz obr. 6.4.1). Je potřeba změnit rozlišení na 600DPI, počet sloupců na 30 a počet řádek křížků na 24. Experimentováním s nastavením analyzátoru se přišlo na to, že pro tuto předlohu je nutno nastavit velikost křížku na 9 milimetrů a maximální tloušťka křížků byla zvolena 20 pixelů. Samozřejmě se musí načíst příslušný rohový vzor (pro tuto předlohu se nachází na přiloženém DVD v adresáři *Rohove_vzory*) a dále je nutné změnit referenční souřadnice v archivu *analyzer.ap* (postup výměny popsán v kapitole 5.2). Pro účely vyhodnocení této konkrétní předlohy byly referenční souřadnice vytvořeny jako kdyby středy křížků byly vzdáleny 20 milimetrů, tzn. souřadnice středů mají takovouto podobu: $[0,20][0,40][20,20]$ atd. Do výsledných charakteristik geometrické přesnosti se pak promítne chyba z tisku předlohy a chyby skeneru.

Konfigurace analyzátoru pro tuto předlohu byla uložena a lze jí nalézt na přiloženém DVD v adresáři *Aplikace_ukazky_nastaveni_a_dat*.



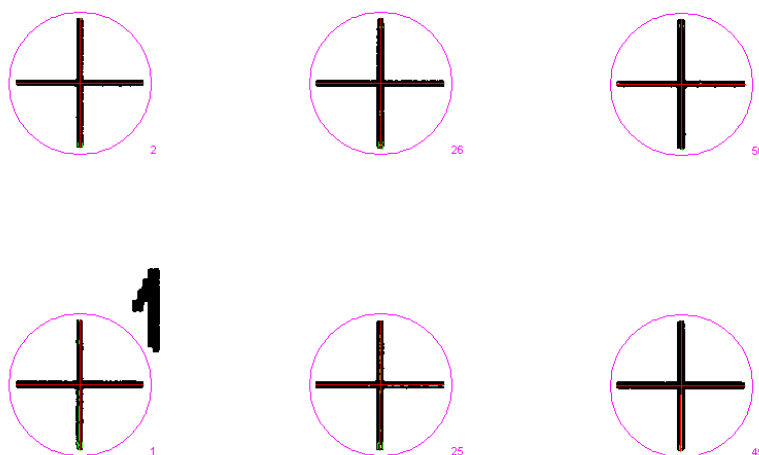
Obrázek 6.4.1: Nastavení analyzátoru pro vzor tvořený křížky na papírové předloze

Veškeré výstupy z této analýzy, které aplikace poskytuje (protokol, transformované souřadnice a vektorová mapa odchylek) jsou uloženy na DVD v adresáři uvedeném na předchozí straně. Ukázka části protokolu:

Chyby souřadnic [mm]:

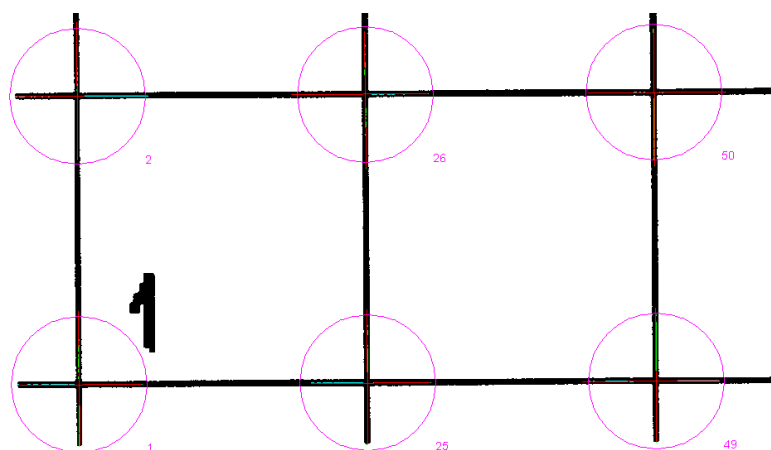
```
-----
Počet vzorků:                720
Střední souřadnicová chyba:  0.085
                             x      y      d
Střední chyba v poloze:     0.055  0.081  0.106
Maximální odchylka v poloze: 0.160  0.296  0.299
```

S vhodným nastavením analyzátoru byly nalezeny středy křížků. Snímek hlavního okna aplikace s několika vizualizovanými středy křížků s čísly si lze prohlédnout na obr. 6.4.2. V reálné situaci by se takto naskenovaná předloha ani nevyhodnocovala.



Obrázek 6.4.2: Snímek hlavního okna aplikace s vizualizovanými středy křížků

Se stejnou konfigurací analyzátoru lze vyhodnotit i druhou papírovou předlohu – mřížku. Mřížka byla naskenována s lehkým natočením, přesto došlo k úspěšnému nalezení průsečíků – viz obr. 6.4.3.



Obrázek 6.4.3: Snímek hlavního okna aplikace s vizualizovanými průsečíky mřížky

7 Závěr

Teoretická část diplomové práce popisuje některé metody zpracování digitalizovaného obrazu. V rámci diplomové práce byla shrnuta současná metodika testování geometrické přesnosti skenerů. Testování se provádí na základě Pokynů č. 32 ČÚZK, přičemž testování a vyhodnocení geometrické přesnosti skenerů provádí Zeměměřický úřad.

Jedním z cílů práce bylo navrhnout a ověřit novou testovací předlohu – předloha měla být na rozměrově stálém materiálu a proto byla jako materiál zvolena hliníková fólie, u které se ovšem vyskytlo několik problémů při skenování a celkově je hliníková fólie hodnocena jako ne příliš vhodný materiál (problematika s převozem, náchylnost k destruktivnímu ohybu, nedoléhavost při průchodu skenerem).

Dále byly navrženy algoritmy pro plně automatickou analýzu skenovaného obrazu testovací předlohy. Algoritmy byly navrženy s ohledem na robustnost a rychlost. Algoritmy jsou parametrizovatelné, díky čemuž je možné pomocí těchto algoritmů zpracovávat různé předlohy charakterově podobné vzoru současné testovací předlohy.

Hlavním cílem práce bylo navrhnout aplikaci, která s využitím navržených algoritmů nalezne na naskenovaném obraze testovací předlohy středy průsečíků mřížky a vypočte charakteristiky týkající se geometrické přesnosti skeneru, na kterém byla předloha naskenována. Aplikace pro analýzu obrazu testovací předlohy byla vytvořena v jazyce Java a pro zvýšení uživatelského komfortu byla vybavena grafickým uživatelským rozhraním (původně se předpokládalo ovládání pouze z příkazové řádky). Podrobný popis ovládání aplikace a nastavení parametrů analýzy obrazu je součástí diplomové práce.

Aplikace byla primárně vyvíjena pro zpracování naskenovaných obrazů současné testovací předlohy. Výsledky je možno porovnat s výsledky ruční vektorizace – kompletní atest provedený pracovníkem ZÚ se nachází na příloženém DVD v adresáři „*Atest*“ a obsahuje i obrazová data získaná při provádění testování.

Výstupem aplikace je protokol, ve kterém jsou uvedeny vypočtené statistiky týkající se geometrické přesnosti. Dále obsahuje matici afinní transformace či souřadnice nalezených středů průsečíků křížků – tyto údaje jsou nad rámec protokolu

atestu dle Pokynů č. 32 ČÚZK. Dalším výstupem je vektorová mapa odchylek (grafické zobrazení rozdílů mezi vypočtenými a referenčními souřadnicemi středů křížků). Vektorová mapa se vykresluje jako obrázek ve formátu **.png* v souladu s tím, jak je vyobrazena v atestu a navíc obsahuje čísla u rohových křížků, souřadnicovou soustavu a měřítko odchylek.

Protože parametry navržených algoritmů lze nastavit v aplikaci, je možné jako předlohu použít libovolnou mřížku a tedy i jednu z navržených předloh na hliníkové fólii. V aplikaci je možné analyzovat síť křížků či úseček s libovolnou vzdáleností středů průsečíků mřížky – vždy je nutné mít na paměti požadavky na vstupní data (na naskenovaný obraz testovací předlohy) – tzn. maximální natočení mřížky menší než 5° , data nepostižena šumem nebo jen minimálně a lokálně a mřížka je zkosená minimálně nebo vůbec. V kapitole 6.4.1 bylo ukázáno, že aplikace vyhodnotí i ne zcela ideální data – demonstrační předlohy byly pouze papírové, vzor byl vytištěn tlustšími úsečkami než by bylo potřeba.

Ve srovnání s ručním vyhodnocením, které vyžaduje až 8 hodin práce je automatizované vyhodnocení výrazně rychlejší (nanejvýš několik minut). Většinu času si vyžádá práce uživatele se zadáním souboru s obrazem předlohy a případné uložení výsledků. Vlastní analýza obrazu předlohy trvá pouze několik sekund.

Cíle práce byly splněny, nicméně bude zapotřebí provést ověření funkce aplikace porovnáním ručního a automatického vyhodnocení přesnosti na větším množství obrazů předloh.

Seznam zdrojů

- [1] ČERVENKA, M. Metoda nejmensich ctvercu [online]. ČVUT, server Herodes. [cit. 2011-03-30]. Dostupné z:
<http://herodes.feld.cvut.cz/mereni/mnc/mnc.php>.
- [2] HLAVÁČ, V. - SEDLÁČEK, M. Zpracování signálů a obrazů. 2. vydání. Praha: Vydavatelství ČVUT, 2007. ISBN 978-01-03110-0.
- [3] KOTEK, Z., et al. Metody rozpoznávání a jejich aplikace. 1. vydání. Praha: Academia, 1993. ISBN 80-200-0297-9.
- [4] MIČKA, P. Metoda nejmenších čtverců - Algoritmy.net [online]. [cit. 2011-03-30]. Dostupné z:
<http://www.algoritmy.net/article/1592/Nejmensi-ctverce>.
- [5] SHAPIRO, Linda G. - STOCKMAN, George C. Computer Vision. Upper Saddle River (New Jersey): Prentice Hall, 2001. ISBN 0-13-030796-3.
- [6] ŠONKA, M. - HLAVÁČ, V. - BOYLE, R. Image Processing, Analysis, and Machine Vision. Third edition. Toronto (Ontario): Thomson Learning, 2008. ISBN 978-0-495-08252-1.
- [7] UMBAUGH, SCOTT E. Computer Imaging: Digital Image Analysis and Processing. Boca Raton (Florida): CRC Press, 2005. ISBN 0-8493-2919-1.
- [8] VRANÁ, P. Digitalizace starých map zařízením Contex CRYSTAL G600. Bakalářská práce. Plzeň: ZČU. Fakulta aplikovaných věd, 2009. [cit. 2011-03-30]. Dostupné z:
http://www.gis.zcu.cz/studium/dp/2009/Vrana_Digitalizace_starych_map_zarizeni_m_Contex_CRYSTAL_G600_BP.pdf.
- [9] ŽÁRA, J., et al. Moderní počítačová grafika. 2. vydání. Brno: Computer Press, 2004. ISBN 80-251-0454-0.
- [10] ŽELEZNÝ, M. Zpracování digitalizovaných obrazů [online]. Přednášky k předmětu KKY/ZDO na FAV, ZČU. 2013. [cit. 2013-03-03]. Dostupné z:
http://www.kky.zcu.cz/uploads/courses/zdo/ZDO_aktual_130215.pdf.
- [11] Astralon - Wikipedia. [cit. 2013-03-21]. Dostupné z:
<http://pl.wikipedia.org/wiki/Astralon>.

- [12] Eclipse - The Eclipse Foundation open source community website [online]. The Eclipse Foundation, 2013. [cit. 2013-05-10]. Dostupné z:
<http://www.eclipse.org>.
- [13] JAMA: Java Matrix Package [online]. Poslední změna [2005-07-13]. [cit. 2011-01-10]. Dostupné z:
<http://math.nist.gov/javanumerics/jama>.
- [14] Oracle Technology Network for Java Developers [online]. Oracle Corporation, 2013. [cit. 2013-05-22]. Dostupné z:
<http://www.oracle.com/technetwork/java/index.html>.
- [15] Pokyny č. 32 pro skenování katastrálních map a grafických operátů dřívějších pozemkových evidencí [online]. Praha: Český úřad zeměměřický a katastrální, 2004. [cit. 2011-04-02]. Dostupné z:
<http://cuzk.cz/GenerujSoubor.ashx?NAZEV=10-POKYN32>.
- [16] Remanufactured KODAK TRENDSETTER VLF Platesetter [online]. Kodak Graphic Communications Group. [cit. 2011-05-08]. Dostupné z:
http://graphics.kodak.com/US/en/Special_Products/Remanufactured_CTP/Very_Large_Format/TRENDSETTER_VLF/default.htm.
- [17] TYPOS [online]. Plzeň: tiskařské závody, s.r.o. [cit. 2011-05-08]. Dostupné z:
www.typos.cz.
- [18] Vydavatelství VŠCHT Praha. VŠCHT v Praze. [cit. 2013-03-21]. Dostupné z:
http://vydavatelstvi.vscht.cz/knihy/uid_es-003/hesla/astralon.obchodni-nazvy.html.

Příloha A – Výstup aplikace – protokol

Jedná se o protokol z provedené analýzy, kde zpracováváný obraz *zup1_1.tif* byl analyzován s výchozími hodnotami analyzátoru. Obraz se nachází na DVD v adresáři *Atest*, protokol k této analýze se nachází na přiloženém DVD v adresáři „*Aplikace_ukazky_dat*“.

----== Protokol ==----

Začátek zpracování: 2013-05-27 10:20:15
Zpracování souboru: zup1_1.tif

Transformační klíč (matice transformace) (px -> mm):

```
-----  
0.063, 0.001, -51.082  
0.000, -0.063, 604.918  
0.000, 0.000, 1.000
```

Chyby transformačního klíče [mm]:

```
-----  
Střední chyba transf. klíče: 0.116  
Počet vzorků: 4  
Střední souřadnicová chyba: 0.082  
                  x          y          d  
Střední chyba v poloze: 0.018 0.114 0.116  
Maximální odchylka v poloze: 0.018 0.114 0.116
```

Chyby souřadnic [mm]:

```
-----  
Počet vzorků: 180  
Střední souřadnicová chyba: 0.093  
                  x          y          d  
Střední chyba v poloze: 0.070 0.086 0.119  
Maximální odchylka v poloze: 0.228 0.204 0.293
```

Souřadnice křížků:

	na předloze [mm]	v obrazu [px]	transformované [mm]	chyby [mm] (x, y, d)
1:	0.000 0.000	728.607 9543.092	0.018 0.114	0.018 0.114 0.116
2:	-0.053 49.990	735.118 8756.521	0.031 49.995	0.084 0.005 0.084
3:	-0.048 99.921	741.433 7968.486	0.031 99.969	0.079 0.048 0.093
4:	-0.080 149.887	747.339 7180.335	0.006 149.950	0.086 0.063 0.106
5:	-0.088 199.878	753.559 6392.093	0.000 199.937	0.088 0.059 0.106
6:	-0.123 249.858	759.432 5604.737	-0.028 249.868	0.095 0.010 0.096
7:	-0.099 299.812	765.712 4816.842	-0.030 299.832	0.069 0.020 0.072
8:	-0.121 349.760	771.445 4029.397	-0.066 349.769	0.055 0.009 0.056
9:	-0.160 399.740	777.381 3242.031	-0.089 399.700	0.071 0.040 0.081
10:	-0.154 449.702	783.033 2453.779	-0.131 449.687	0.023 0.015 0.027
11:	-0.181 499.651	788.894 1666.372	-0.160 499.621	0.021 0.030 0.037
12:	-0.191 549.609	794.417 879.905	-0.209 549.495	0.018 0.114 0.116
13:	49.962 -0.034	1515.082 9549.365	49.943 0.086	0.019 0.120 0.122
14:	49.908 49.947	1521.813 8762.215	49.970 50.004	0.062 0.057 0.085
15:	49.934 99.902	1527.829 7974.177	49.951 99.978	0.017 0.076 0.078
16:	49.891 149.896	1534.092 7186.084	49.948 149.955	0.057 0.059 0.083
17:	49.876 199.866	1539.919 6397.804	49.918 199.945	0.042 0.079 0.089
18:	49.837 249.846	1546.136 5610.498	49.912 249.872	0.075 0.026 0.079
19:	49.866 299.812	1552.295 4822.257	49.902 299.859	0.036 0.047 0.059
20:	49.834 349.758	1558.114 4035.048	49.871 349.780	0.037 0.022 0.043
21:	49.813 399.728	1563.963 3247.298	49.842 399.736	0.029 0.008 0.030
22:	49.799 449.694	1569.917 2459.054	49.819 449.723	0.020 0.029 0.035
23:	49.788 499.647	1575.549 1671.710	49.777 499.652	0.011 0.005 0.012
24:	49.767 549.597	1581.258 885.028	49.739 549.540	0.028 0.057 0.064
25:	99.917 -0.017	2301.770 9554.730	99.882 0.116	0.035 0.133 0.138

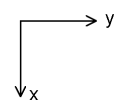
Příloha B – Výstup aplikace – vektorová mapa odchylek

Jedná se o vektory odchylek po provedení analýzy, kde zpracovávaný obraz: `zup1_1.tif` byl analyzován s výchozími hodnotami analyzátoru. Protokol k této analýze se nachází na přiloženém DVD v adresáři „*Aplikace_ukazky_dat*“.

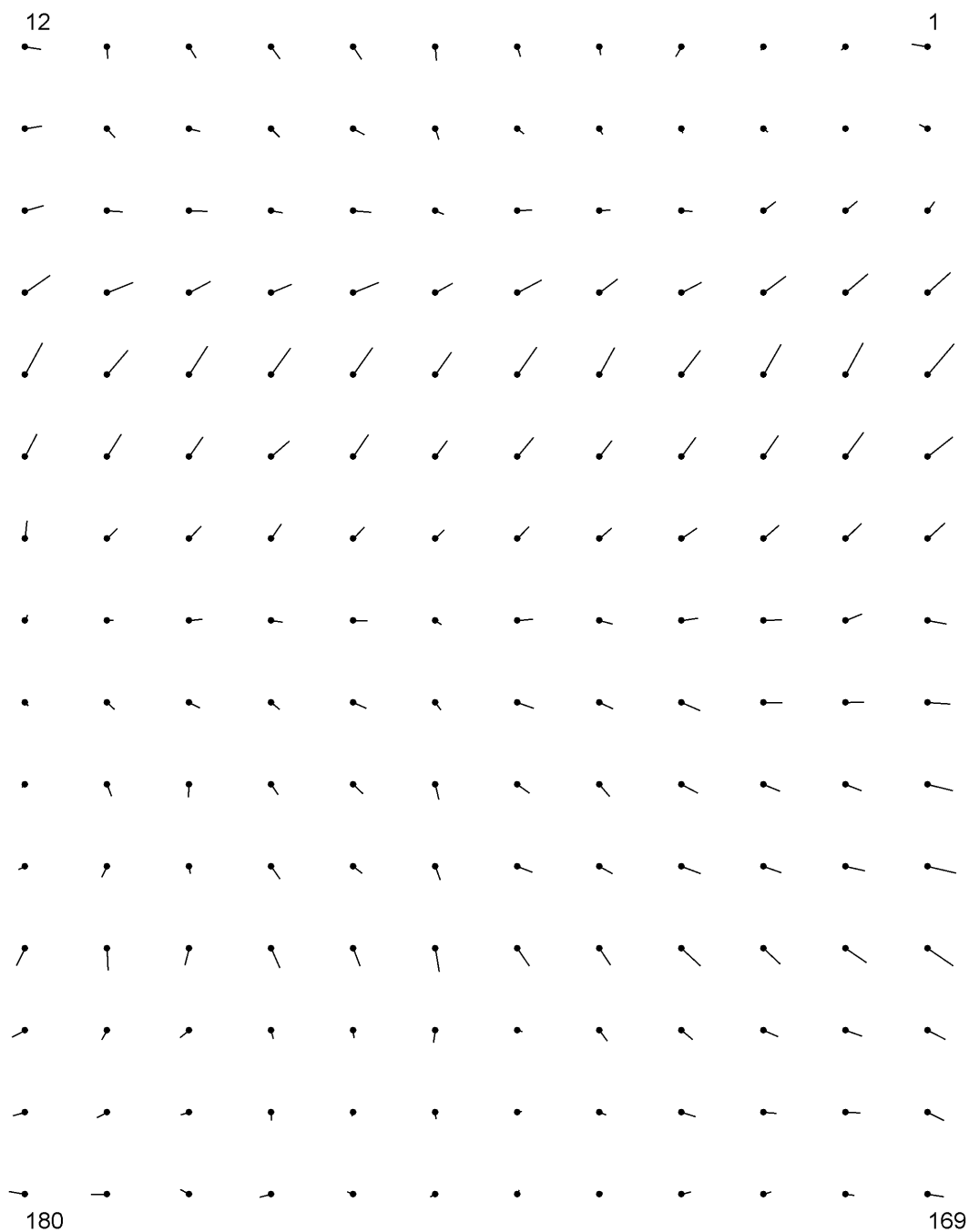
Testovaný soubor: zup1_1.tif

Chyba transformace: 0.116

[mm]	střední chyba	maximální chyba
v poloze	0.119	0.293
ve směru x	0.070	0.228
ve směru y	0.086	0.204



měřítka odchylky:
0.30 mm



Příloha C – Část protokolu ZÚ

Jedná se o část protokolu s hodnotami chyb z atestu skeneru Contex CRYSTAL G600 na KMA, který byl proveden pracovníkem ZÚ dne 22. 4. 2008 (více v [8]). Testovaným obrazem byl *zupl_1.tif*. Obraz se nachází na přiloženém DVD v adresáři *Atest*, kompletní protokol ve formátu **.pdf* (*Atest-2008-05.pdf*) tamtéž.

Výsledky testu:

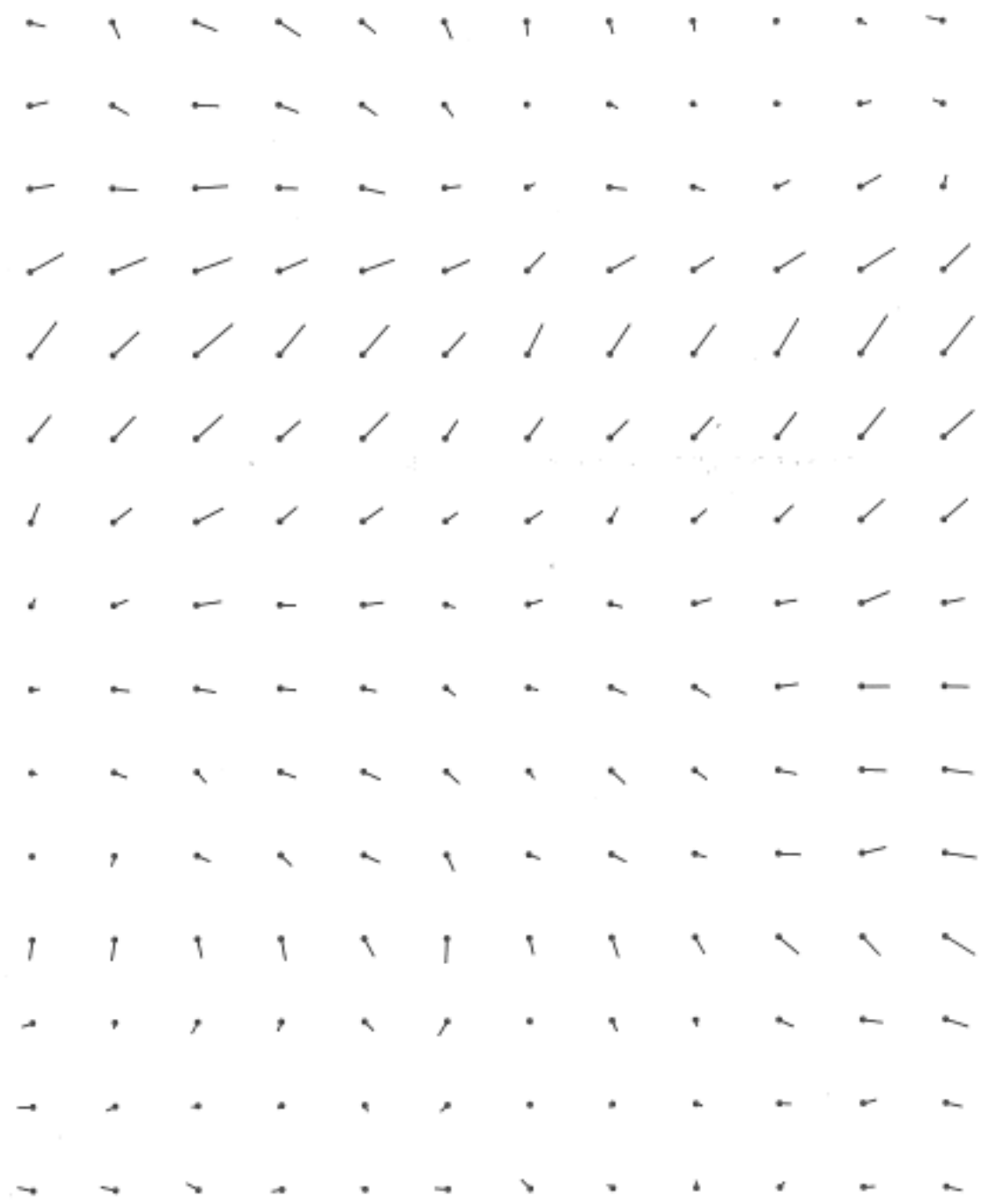
transformace:	afinní	
identické body:	4 rohové body mřížky	
střední chyba transformačního klíče:	0,11 mm	
kontrolní body:	č. 1 až 180	
střední chyba		maximální odchylka
v poloze	0,16	0,33
ve směru skenování	0,09	0,21
kolmo na směr skenování	0,12	0,29
souřadnicová m_{xy}	0,11	

Příloha D – Vektorová mapa odchylek ZÚ

Jedná se o vektorovou mapu odchylek z atestu skeneru Contex CRYSTAL G600 na KMA, který byl proveden pracovníkem ZÚ dne 22. 4. 2008 (více v [8]). Testovaným obrazem byl *zup1_1.tif*. Kompletní protokol ve formátu **.pdf* (*Atest-2008-05.pdf*) obsahující vektorovou mapu odchylek se nachází na přiloženém DVD v adresáři *Atest*.

Testovaný file: zup1
chyba transformace: 0.11 mm

[mm]	str. chyba	max. chyba (složka)
v poloze	0.16	0.33
ve smeru skenovani	0.09	0.21
ve smeru valce	0.12	0.29



Příloha E – Struktura přiloženého CD

- Aplikace
 - obsahuje spustitelnou aplikaci spolu se všemi nutnými náležitostmi (před spuštěním aplikace je nutno tento adresář s veškerým jeho obsahem překopírovat na disk, na kterém má uživatel právo k zápisu)
- Aplikace_ukazky_nastaveni_a_dat
 - obsahuje nastavení analyzátoru pro papírovou předlohu a výstupy aplikace pro tři předlohy
- Atest
 - obsahuje atest skeneru Contex CRYSTAL G600 ve formátu *.pdf (atest udělen na základě výsledků testování, které bylo provedeno dne 22.4.2008)
- Eclipse_projekt
 - obsahuje zdrojové kódy aplikace
- Instalacni_soubory
 - obsahuje instalační soubory *JRE* (běhové prostředí Java) ve verzi 6 a 7 pro 32-bit i 64-bit operační systémy *Windows*
- Obrazky
 - obsahuje obrázky ve formátu *.jpg a *.png, které se nachází v textu DP
- Pokyny_c_32
 - obsahuje pokyny č. 32 ČÚZK spolu s dodatky (formát *.rtf a *.pdf)
- Rohove_vzory
 - obsahuje rohový vzor pro současnou předlohu a pro papírovou předlohu
- Skenovane_predlohy
 - obsahuje obrázky naskenovaných předloh ze tří skenerů
- Text_DP
 - obsahuje text diplomové práce ve formátu *.pdf