



VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA ELEKTROTECHNIKY A KOMUNIKAČNÍCH TECHNOLOGIÍ

FACULTY OF ELECTRICAL ENGINEERING AND COMMUNICATION

ÚSTAV AUTOMATIZACE A MĚŘICÍ TECHNIKY

DEPARTMENT OF CONTROL AND INSTRUMENTATION

LÍCOVÁNÍ SKICI OBJEKTU S JEHO SNÍMKEM

OBJECTS OUTLINE AND IMAGE MATCHING

DIPLOMOVÁ PRÁCE

MASTER'S THESIS

AUTOR PRÁCE

AUTHOR

Bc. Petr Kvasnička

VEDOUCÍ PRÁCE

SUPERVISOR

Ing. Ilona Janáková, Ph.D.

BRNO 2018

Diplomová práce

magisterský navazující studijní obor **Kybernetika, automatizace a měření**

Ústav automatizace a měřicí techniky

Student: Bc. Petr Kvasnička

ID: 158181

Ročník: 2

Akademický rok: 2017/18

NÁZEV TÉMATU:

Lícování skici objektu s jeho snímkem

POKYNY PRO VYPRACOVÁNÍ:

Cílem práce je navrhnout algoritmus pro vizuální porovnání zvoleného, spíše plošného, objektu reprezentovaného snímkem či videosekvencí s jeho skicou (schématem, náčrtem, obrysem).

1. Proveďte rešerši možností zpracování obrazu z hlediska realizace dané úlohy.
2. Navrhněte vhodné objekty pro testování a demonstraci výsledků a vytvořte testovací databáze.
3. Na základě rešerše zvolte vhodné metody předzpracování obrazu a metody porovnávání/slícování (matching) se vzorem. Uvažujte rozdílná měřítka, natočení, perspektivu i jen částečnou viditelnost objektu.
4. Alespoň tři vybrané postupy implementujte, důkladně otestujte a vzájemně porovnejte.
5. Zhodnoťte výsledky a stanovte omezující podmínky.

DOPORUČENÁ LITERATURA:

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

BELONGIE, S., J. MALIK a J. PUZICHA. Shape Matching and Object Recognition Using Shape Contexts. IEEE TRANSACTIONS ON PATTERN ANALYSIS AND MACHINE INTELLIGENCE [online]. 2002(VOL. 24, NO. 24), 509-522 [cit. 2016-09-08]. Dostupné z: <http://graphics.stanford.edu/courses/cs468-08-fall/pdf/belongie-pami02.pdf>

Termín zadání: 5.2.2018

Termín odevzdání: 14.5.2018

Vedoucí práce: Ing. Ilona Janáková, Ph.D.

Konzultant:

doc. Ing. Václav Jirsík, CSc.
předseda oborové rady

UPOZORNĚNÍ:

Autor diplomové práce nesmí při vytváření diplomové práce porušit autorská práva třetích osob, zejména nesmí zasahovat nedovoleným způsobem do cizích autorských práv osobnostních a musí si být plně vědom následků porušení ustanovení § 11 a následujících autorského zákona č. 121/2000 Sb., včetně možných trestněprávních důsledků vyplývajících z ustanovení části druhé, hlavy VI. díl 4 Trestního zákoníku č.40/2009 Sb.

Abstrakt

Cílem diplomové práce je seznámení s problematikou detekování objektu s jeho náčrtem, schématem, případně obrysem, který by měl reprezentovat porovnávaný objekt. Práce se zabývá tím, jak tuto problematiku vyřešit. Část diplomové práce obsahuje teoretické metody, které zvládají vyhodnocovat míru podobnosti mezi neznámým objektem s jeho méně přesným náčrtem. Každá z těchto metod obsahuje vlastní úpravy z pohledu předzpracování obrazu. Práce se dále zabývá podrobněji některými metodami. Vybrané metody jsou realizované a testované na vytvořené databázi. V posledním kroku práce jsou tyto realizované metody vzájemně porovnány.

Klíčová slova

Vzdálenost, obrazy, podobnost, objekty, náčrtek

Abstract

The aim of the diploma thesis is to get acquainted with problem of detecting the object with sketch, diagram or contour, which should represent the object to be compared. The thesis deals with how to solve this problem. Part of the thesis contains theoretical methods that manage to evaluate the degree of similarity between the unknown object and less accurate sketch. Each of these methods includes customized edits from the viewpoint of image pre-processing. The thesis deals with some methods in more detail. Selected methods are implemented and tested on the created database. In the last step of the thesis these realized methods are compared to each other.

Keywords

Distance, pictures, similarity, objects, sketch

Bibliografická citace:

KVASNIČKA, P. *Licování skici objektu s jeho snímkem*. Brno: Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií, 2018. 66 s. Vedoucí diplomové práce Ing. Ilona Janáková, Ph.D..

Prohlášení

*„Prohlašuji, že svou diplomovou práci na téma *Lícování skici objektu s jeho snímkem* jsem vypracoval samostatně pod vedením vedoucí/ho diplomové práce a s použitím odborné literatury a dalších informačních zdrojů, které jsou všechny citovány v práci a uvedeny v seznamu literatury na konci práce.*

Jako autor uvedené diplomové práce dále prohlašuji, že v souvislosti s vytvořením této diplomové práce jsem neporušil autorská práva třetích osob, zejména jsem nezasáhl nedovoleným způsobem do cizích autorských práv osobnostních a jsem si plně vědom následků porušení ustanovení § 11 a následujících autorského zákona č. 121/2000 Sb., včetně možných trestněprávních důsledků vyplývajících z ustanovení části druhé, hlavy VI. díl 4 Trestního zákoníku č. 40/2009 Sb.

V Brně dne: **14. května 2018**

.....
podpis autora

Obsah

1	ÚVOD	10
2	METODY DETEKCE.....	11
2.1	Rozdílová a korelační metoda.....	11
2.2	Hausdorffova vzdálenost.....	13
2.3	Shape context	16
2.4	Inner distance	19
2.5	Active Appearance Models	21
2.6	Boundary Fragment Model	24
2.7	Hierarchical Matching.....	27
3	IMPLEMENTACE VYBRANÝCH METOD	30
3.1	Rozdílová metoda	30
3.2	Korelační metoda	31
3.3	Hausdorffova metoda.....	32
3.4	Shape context	35
3.5	Inner distance	41
4	TESTOVÁNÍ METHOD	47
4.1	Testování rozlišovacích schopností.....	47
4.2	Testování změny velikosti rozměrů a úhlů	57
4.3	Testování na databázi Tool2D	60
4.4	Testování na reálných snímcích.....	62
5	ZÁVĚR.....	65
	Citovaná literatura.....	67

Seznam obrázků

Obr. 2.1	Ukázka distancí transformace [5]	14
Obr. 2.2	Ukázka podoby předzpracování obrysu a ukázka histogramu pro zvolené body [6]....	16
Obr. 2.3	Ukázka podoby distanční sítě [9].....	19
Obr. 2.4	Ukázka rozdělení objektu na celé části propojené pohyblivým kloubem [9]	20
Obr. 2.5	Ukázka vzhledu hraničních bodů v testovacím obrázku [11]	22
Obr. 2.6	Ukázka výběru hraničního kandidátu a jeho následné rozšiřování [14]	25
Obr. 2.7	Ukázka Shape-tree, kde je vidět postupné rozkládání křivky na menší části [17].....	28
Obr. 3.1	Vývojový diagram pro rozdílové metody SSD a SSDN.....	31

Obr. 3.2 Vývojový diagram pro korelační metody CC, NCC a MNCC	32
Obr. 3.3 Vývojový diagram pro Hausdorffovu vzdálenost.....	33
Obr. 3.4 Vývojový diagram pro shape context	35
Obr. 3.5 Vývojový diagram pro TPS transformaci	36
Obr. 3.6 Ukázka vzhledu využití TPS transformace na vzoru a porovnávanému obrazu.....	37
Obr. 3.7 Ukázka log-polar, který rozděluje prostor u shape context.....	38
Obr. 3.8 Ukázka histogramu pro metodu Shape Context.....	38
Obr. 3.9 Vývojový diagram pro metodu Hungarian	39
Obr. 3.10 Zobrazení nalezení vazeb mezi dvěma objekty	41
Obr. 3.11 Vývojový diagram pro inner distance	42
Obr. 3.12 Vzhled propojení vnitřních vazeb zkoumaného objektu.....	43
Obr. 3.13 Diagram k vypočítání inner-anglu pro metodu Inner distance.....	44
Obr. 3.14 Ukázka tečny (červená) a všech počátečních směrů nejkratší vzdálenosti (modrá) ...	45
Obr. 4.1 Testovací obrazce pro hledání podobnosti.....	47
Obr. 4.2 Surf graf pro metodu SSDN testu 4.1	48
Obr. 4.3 Surf graf pro metodu MNCC testu 4.1.....	49
Obr. 4.4 Surf graf pro metodu Hausdorff testu 4.1	50
Obr. 4.5 Výsledky podobností pro metodu shape context pro první variaci testu	53
Obr. 4.6 Výsledky podobností pro metodu shape context pro druhou variaci testu	54
Obr. 4.7 Výsledky podobností pro metodu inner distance	54
Obr. 4.8 Obrázek abecedy pro testování rozpoznatelnosti znaků	55
Obr. 4.9 Výsledky Hausdorffovy metody na obrázek abecedy.....	55
Obr. 4.10 Výsledky shape contextu na obrázek abecedy	56
Obr. 4.11 Výsledky inner distance na obrázek abecedy	56
Obr. 4.12 Vzhled testovacích znaků pro testování reakce na změnu velikosti	57
Obr. 4.13 Surf graf pro metodu Hausdorff testu 4.2	57
Obr. 4.14 Výsledky testů velikosti pro metodu shape context.....	58
Obr. 4.15 Výsledky testů velikosti pro metodu inner distance	58
Obr. 4.16 Obrázek pro testování natočení.....	59
Obr. 4.17 Výsledek shape context na rotaci	59
Obr. 4.18 Výsledek inner distance na rotaci.....	60
Obr. 4.19 Výsledky testů databáze Tool2D pro shape context	61
Obr. 4.20 Výsledky testů databáze Tool2D pro inner distance.....	61
Obr. 4.21 Obrázek reálných snímků ovoce a zeleniny.....	62
Obr. 4.22 Nalezené pozice pro metodu MCCN u obrázku s reálnými objekty	62

Obr. 4.23 Výsledný tvar hran po aplikování upraveného Cannyho detektoru	63
Obr. 4.24 Surf graf pro metodu Hausdorff u obrázku s reálnými objekty	64
Obr. 4.25 Vyhodnocení podobnosti u metody shape context pro reálný snímek.....	64
Obr. 4.26 Vyhodnocení podobnosti u metody inner distance pro reálný snímek	64

Seznam tabulek

Tabulka č. 1 Tabulka rychlosti algoritmu pro různé typy rozdílových a korelačních metod pro 10 opakování	49
--	----

Seznam grafů

Graf č. 4.1 Závislost zrychlení v závislosti na velikosti vynechané oblasti pro zrychlovací metodu u metody Hausdorff	51
Graf č. 4.2 Závislost míry podobnosti pro počet opakování TPS transformace	52

Seznam zkratek

AAM	Active Appearance Model
ASM	Active Shape Model
PCA	Principal Component Analysis
TPS	Thin Plate Spline
BFM	Boundary Fragment Model
TPS	Thin Plate Spline
SSD	Suma Čtverců Odchylek
SSDN	Suma Čtverců Odchylek Normalizovaná
CC	Korelace obrazu a šablony
NCC	Normalizovaná korelace
MNCC	Modifikovaná Normalizovaná korelace
DP	Dynamic programming

1 ÚVOD

V počítačovém vidění se s ohledem na vyhledávání a sledování objektů lze setkat s dvěma pojmy. První pojem je pattern matching (český název „přízpusobení vzoru“). V pattern matching je snaha najít přesnou podobu obrysu v neznám obraze. Druhý pojem je pattern recognition (český název „rozpoznání vzoru“). V pattern recognition je snahou nalezení objektu, který má podobné vzhledové vlastnosti jako vzor, např. neznámý tvar vypadá jako hrnek, to znamená, aby neznámý objekt spadal do stejné kategorie. Problematika detekce je relativně obtížná úloha, jelikož je ovlivněna sugescí. Jinými slovy - člověk dokáže rozpoznat, co se zobrazuje na obrázku, ale pro počítač schopnost rozpoznání není tak jednoduchá. Této problematice se bude věnovat tato diplomová práce. Cílem bude nalézt metody, které budou schopné rozeznávat objekty podle podobného objektu, který bude nakreslený jako šablona, skica, případně náčrtek. Diplomová práce je rozdělena na několik částí. V první části jsou vypsány teoretické poznatky k metodám, které dokáží rozeznávat objekty podle vzoru. Vzory mohou mít přesně vypadající tvar nebo mít zdánlivou podobu. Tato práce je více zaměřená na zdánlivě podobné vzory, jako jsou šablony, skicy, náčrtky a případně objekty, které mají jiný tvar, ale patří do stejné kategorie. Kromě stejné podobnosti mohou mít vzory i jiné vady. Příkladem vad mohou být deformace, šum při získávání obrazu, vynechání části objektu a překrytí. Samozřejmě stejné problémy se mohou vyskytovat i v prohledávaném obraze. Kromě vad mohou být porovnávány obrázky v jiných podobách, například natočení objektů, změny měřítka anebo se mohou vyskytovat na jiných pozicích. Všechny tyto problémy se snaží vyřešit metody v první části práce. V druhé části práce jsou následně vybrány některé metody z první části a tyto metody jsou následně vytvořeny v programovacím rozhraní MATLAB Ra2014. V poslední části jsou aplikované algoritmy otestované na testovací databázi. Databáze je tvořena z jedné části jednoduchými symboly a z druhé části reálnými objekty. Na této databázi se bude následně testovat, jaké problematické vlastnosti jsou schopné řešit. Vlastnosti, které se budou vyhodnocovat, jsou například deformace objektu, natočení nebo změny měřítka. Kromě těchto vlastností se bude dále vyhodnocovat kvalita detekce, časová náročnost vytvořených algoritmů a schopnost reagovat na podobný tvar reprezentovaný náčrtem nebo obrysem. Ze získaných poznatků z testování budou následně metody vyhodnoceny a vzájemně porovnány.

2 METODY DETEKCE

Diplomová práce se zabývá těmito metodami:

- Rozdílová a korelační metoda
- Hausdorffova vzdálenost
- Shape context
- Inner distance
- Active Appearance Models
- Boundary Fragment Model
- Hierarchical Matching

Pro každou z těchto metod je vypsán postup, jak se budou realizovat, co dokážou vyhledat a jejich nevýhody.

2.1 Rozdílová a korelační metoda

Rozdílová a korelační metoda patří k jednoduchým detekčním metodám. Základní princip těchto metod spočívá v posunování hledaného vzoru, značeného písmenem t , po hlavním snímku obrazu, značeného písmenem f . Při každém posunu dojde k výpočtům podle daných vzorců a výsledek je zapsaný do matice. Při projití všech možných pozic se z výsledné matice určí minimální, případně maximální hodnota, která reprezentuje shodu v obraze. První zabývanou metodou je rozdílová metoda (2.1.1).

$$SSD(x', y') = \sum_{x, y \in T} (f(x + x', y + y') - t(x, y))^2 \quad (2.1.1)$$

Rozdílová metoda odečítá všechny prvky pro daný posun. Pokud je část hlavního obrazu stejná jako hledaná šablona, je výsledek pro daný posun rovný nule. Z toho je zřejmé, že z výsledné matice SSD se hledá minimum vyjádřeno vzorcem (2.1.2).

$$(x', y') = \arg \min_{(i, j)} \|SSD(i, j)\| \quad (2.1.2)$$

Nevýhodou této metody je nemožnost rozpoznat rotace šablony a také je nutné, aby se tvar hledaného vzoru přesně vyskytoval v hledaném obraze. Poslední nevýhodou je neschopnost si poradit se změnou jasů. Problém s jasnem lze z části odstranit upravením základního vztahu (2.1.1) do upravené verze (2.1.3). K základnímu vztahu jsou přidány funkce \bar{f} a \bar{t} . Funkce \bar{f} je brána jako průměr jasových hodnot pro daný posun, který se počítá vždy při každém posunu. Funkce \bar{t} je průměrná jasová hodnota

pro daný vzor a je počítána pouze jednou. Schopnost kompenzovat lokální změny jasových hodnot je zaručena jenom pro oblast velikosti šablony.

$$SSD_N(x', y') = \sum_{x, y \in T} \left((f(x + x', y + y') - \bar{f}(x', y')) - (t(x, y) - \bar{t}) \right)^2 \quad (2.1.3)$$

Další metoda je korelační, jejíž definiční vztah je uveden ve vzorci (2.1.4). Rozdíl ve výpočtu oproti rozdílové je ten, že místo vzájemného odečítání prvků se využívá prvků v pozicích. Po výpočtu všech korelací pro všechny posuny vzoru po obraze se za pomoci vztahu (2.1.5) vybere maximální hodnota, která reprezentuje shodu vzoru s objektem. Korelační metoda má stejné slabé stránky jako metoda rozdílová - přesněji detekce rotovaného objektu a neschopnost detekovat jasové výkyvy.

$$CC(x', y') = \sum_{x, y \in T} (f(x + x', y + y') \cdot t(x, y)) \quad (2.1.4)$$

$$(x', y') = \arg \max_{(i, j)} \|CC(i, j)\| \quad (2.1.5)$$

Pro odstranění jasových hodnot je základní vztah obohacen průměry \bar{f} a \bar{t} , jejichž význam je stejný jako u rozdílové metody a přidáním proměnné N, která má pro normalizovanou korelaci vztah uveden ve vzorci (2.1.7), vzniká normalizovaná korelace uvedena vztahem (2.1.6), která je schopná si poradit s lineární podobou jasu.

$$NCC(x', y') = \frac{1}{N} \sum_{x, y \in T} \left((f(x + x', y + y') - \bar{f}(x', y')) \cdot (t(x, y) - \bar{t}(x, y)) \right) \quad (2.1.6)$$

$$N = \sqrt{D_f(x', y') \cdot D_t} \quad (2.1.7)$$

Kde $D_f(x', y')$ je převzata část ze vzorce (3.1.3) pro normalizovanou rozdílovou metodou, která vyjadřuje rozdíl průměru jasových hodnot vůči aktuálnímu posunu obrazu vyjádřena ve vztahu (2.1.8) a D_t je převzata část ze vzorce (2.1.3) pro normalizovanou rozdílovou metodou, která vyjadřuje rozdíl průměru jasu vůči šabloně vyjádřena ve vztahu (2.1.9).

$$D_f(x', y') = \sum_{x, y \in T} \left(f(x + x', y + y') - \bar{f}(x', y') \right)^2 \quad (2.1.8)$$

$$D_t = \sum_{x, y \in T} (t(x, y) - \bar{t})^2 \quad (2.1.9)$$

Nevýhodou normalizované korelace je, že pokud v hledaném obrazu nalezneme plochou část, tak se bude D_f ze vztahu (2.1.7) rovnat nule a tím dojde ve vztahu (2.1.6) k dělení nulou. Pro odstranění se využívá modifikovaná normalizovaná korelace, která vzniká dosazením za N ve vztahu (2.1.6) vztahem (2.1.10). [1] [2]

$$N = D_f(x', y') + D_t \quad (2.1.10)$$

2.2 Hausdorffova vzdálenost

Hausdorffova vzdálenost je definovaná ve vzorci (2.2.1), kde A a B jsou množiny bodů. Funkce h je definovaná v (2.2.2).

$$H(A, B) = \max(h(A, B), h(B, A)) \quad (2.2.1)$$

$$\vec{h}(A, B) = \max_{a \in A} \min_{b \in B} \|a - b\| \quad (2.2.2)$$

Základní princip Hausdorffovy vzdálenosti spočívá v zjištění, zdali jsou si obě množiny A a B podobné. V první řadě se bere jeden prvek z množiny A a zjistí se vzdálenosti k prvkům z druhé množiny B . Při zjištění všech vzdáleností pro zvolený bod z množiny A se vybere minimální vzdálenost a cyklus se opakuje pro další prvek A . Po projetí všech prvků z množiny A a nalezení příslušných minimálních vzdáleností, je pak vybrána největší minimální vzdálenost, která reprezentuje míru shody prvků množiny A a B . Pokud je výsledné číslo malé, tak jsou si množiny podobné, naopak pokud je výsledné číslo velké, tak jsou množiny rozdílné.

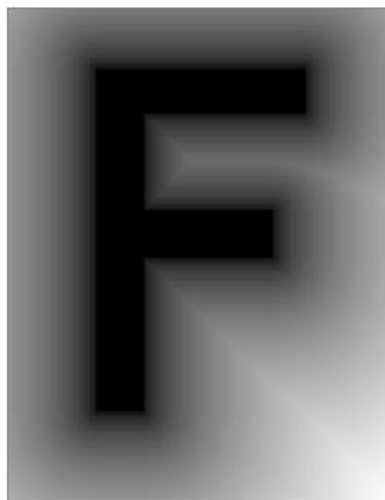
Pro možnost použití Hausdorffovy vzdálenosti k detekci vzoru v obraze je nutné doplnit základní vztah tak, aby bral v potaz pohyb vzoru po obraze. Upravený vztah je definovaný ve vzorci (2.2.3), kde symbol \oplus reprezentuje Minkowskovu sumu. Upravený vztah nyní projíždí všechny možné pozice v obraze a pro každou pozici vypočítává shodu za pomoci Hausdorffovy vzdálenosti. Při projetí všech pozic pak z nalezených vzdáleností vybere tu nejmenší.

$$M_T(A, B) = \min_t H(A, B \oplus t) \quad (2.2.3)$$

Pro vylepšení výpočtu Hausdorffovy vzdálenosti se využívají distanční transformace. Základní vztah pro výpočet distanční transformace je uveden ve vzorci (2.2.4). Princip distanční transformace spočívá v nalezení nejbližší vzdálenosti pixelu v obraze k nejbližšímu nenulovému pixelu obrazu. Metoda využívá binární obraz. Ukázka vzhledu distanční transformace je ukázána na obrázku Obr. 2.1. [5]

$$D_p(p) = \min_{q \in P} \|p - q\| \quad (2.2.4)$$

Kde D_p vyjadřuje distanční transformaci, p vybraný bod v obraze a q nejbližší nenulový bod.



Obr. 2.1 Ukázka distancí transformace [5]

Pro určení Hausdorffovy vzdálenosti z digitálních dat je optimální aplikovat na obraz a vzor hranový detektor. Pomocí hranového detektoru se získá binární obraz, který reprezentuje nalezené hrany v obraze. Na binární obrazy se aplikují distanční transformace a vzniknou transformace D pro vzor a D' pro obraz. Následně se vezme vztah (2.2.2) a upraví se pro digitální obrazy. Vztah F_B , jež má definiční vztah ve vzorci (2.2.5), je určen k vyhledávání bodu ve vzoru s distanční mapou vytvořenou z obrazu. Vztah F_A , jež má definiční vztah ve vzorci (2.2.6), slouží k vyhledávání části obrazu s distanční mapou vzoru. Oba dva vztahy se dosadí do vztahu (2.2.1) a vznikne nám Hausdorffova vzdálenost pro digitální obraz, která je uvedena ve vztahu (2.2.7).

$$F_B[x, y] = \max_{k,l} B[k, l] D'[k + x, l + y] \quad (2.2.5)$$

Kde $B[k, l]$ je brána jako binární matice hledaného vzoru, D' distanční mapa získaná z obrazu A .

$$F_A[x, y] = \max_{k,l} A[k, l] D[k - x, l - y] \quad (2.2.6)$$

Kde $A[k, l]$ je brána jako binární matice obrazu, D distanční mapa získaná z obrazu B .

$$F[x, y] = \max(F_A[x, y], F_B[x, y]) \quad (2.2.7)$$

Hausdorffova vzdálenost umožňuje i detekci neúplných objektů, například pokud je hledaný objekt zakrytý jiným objektem. Pro docílení možnosti detekovat neúplné objekty je nutné upravit definiční vztah (2.2.2), tak aby místo vyhledávání maximální hodnoty vyhledával nejvyšší v pořadí, například desátou největší hodnotu. Upravená definice je ukázána ve vzorci (2.2.8). Přidáním možnosti nalezení nejvyšší

hodnoty v pořadí přidáme možnost ignorovat některé body, které jsou mimo požadovanou oblast, což umožňuje najít obraz, který je částečně zakrytý.

$$h_K[x, y] = K_{b \in B}^{th} \min_{a \in A} \|a - b\| \quad (2.2.8)$$

Kde K^{th} je požadované pořadí vzdálenosti a je definováno vztahem (2.2.9).

$$K^{th} = \lfloor f_1 q \rfloor \quad (2.2.9)$$

Kde f_1 umožňuje výběr, kolik bodů vzorů se bude odpovídat shodě mezi vzorem a obrazem. Hodnota f_1 je požadována v rozsahu $0 \leq f_1 \leq 1$ a proměnná q uvádí počet bodů ve vzoru. V případě, že $f_1 = 1$, tak vztah odpovídá základní definici (2.2.2).

Hausdorffova vzdálenost dále umožňuje vyhledávat objekty, které jsou otočené pod různými úhly. Pro tyto účely se vztah (2.2.3) upraví na vztah (2.2.10).

$$M_E[A, B] = \min_{t, \theta} H(A, (R_\theta B) \oplus t) \quad (2.2.10)$$

Kde H je Hausdorffova vzdálenost a R_θ je rotační matice. [4]

Cílem je, aby na každou rotaci vzoru B byla vypočítána Hausdorffova vzdálenost. Je důležité, aby rotace objektu byla pozorovatelná, to znamená, že výpočtový interval θ , který udává úhel otočení objektů, ve kterém se bude počítat Hausdorffova vzdálenost, by měl být tak velký, aby změna rotace byla poznatelná od nejmíň jednoho pixelu rastru. Rotace vzoru B je uvedena ve vzorci (2.2.11).

$$r_j = \|b_j - c_r\| \quad (2.2.11)$$

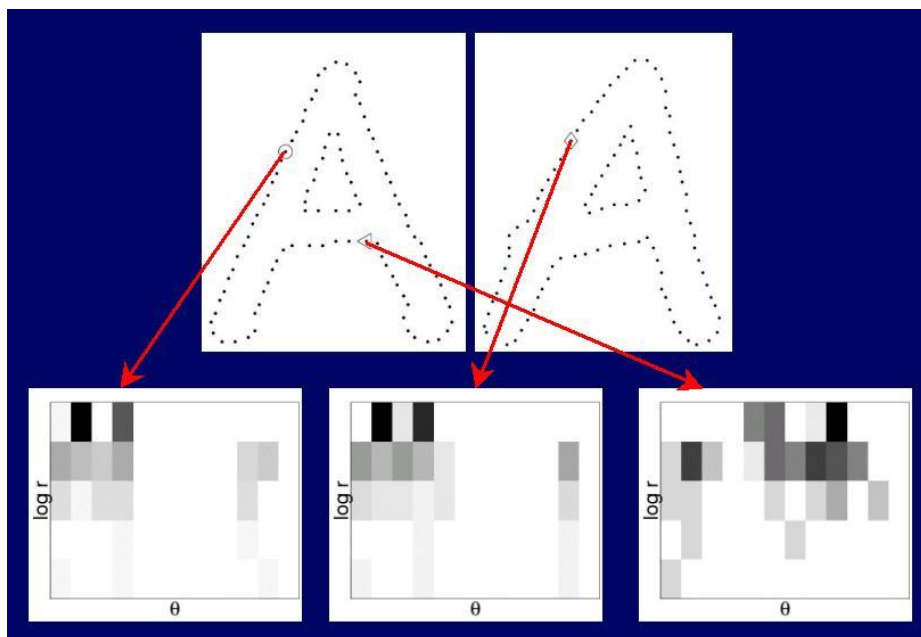
Kde b_j udává bod vzoru množiny B , c_r vyjadřuje střed rotace a r_j je nová pozice způsobena rotací množiny B .

Pro výpočet rotací podle požadovaného intervalu rotace θ , je vytvořena pomocná matice Q , do které se postupně zapisují vypočítané Hausdorffovy vzdálenosti pro úhly natočení v závislosti na intervalu θ . Matematický popis pomocné matice Q je uveden ve vztahu (2.2.12).

$$Q[i] = h((R_{i \Delta \theta} B) \oplus t, A) \quad (2.2.12)$$

2.3 Shape context

Metoda Shape context je metoda, která využívá pro porovnání obrazu obrysy objektu. Obrysy objektů získává za použití hranového detektoru a následného rozložení těchto hran na body, které zdánlivě vytváří obrys objektu na snímku. Podoba rozložení bodů je ukázána na obrázku Obr. 2.2.



Obr. 2.2 Ukázka podoby předzpracování obrysu a ukázka histogramu pro zvolené body [6]

Shape context je brán jako deskriptor, který udává vzdálenosti nejbližších bodů pro zvolený bod hrany. Výpočet deskriptoru je uveden ve vztahu (2.3.1).

$$h_i(k) = \#\{p_j: j \neq i, x_j - x_i \in \text{bin}(k)\} \quad (2.3.1)$$

Kde funkce bin reprezentuje rovnoměrně rozloženou logaritmicko-polární oblast a $h_i(k)$ je histogram shape contextu. Dále je ve vzorci využita Euklidova vzdálenost mezi body, vyjádřena $x_j - x_i$.

Pokud je nějaký bod blízko k testovanému bodu, tak je v histogramu zvýrazněn tmavší barvou a matematicky je vyjádřen velkým číslem. Pokud se bude testovaný bod vzdalovat, tak jeho matematická hodnota se bude snižovat a v histogramu se bude zesvětlovat. Ukázka vzhledu histogramu pro bod je ukázaná v dolní části obrázku Obr. 2.2. Pro každý bod v obrázku je vytvořen histogram popisující vzájemné vzdálenosti mezi body. Po projetí všech bodů v obrysu vzniká histogram h_i pro jeden obraz. Pak se algoritmus aplikuje pro druhý porovnávaný objekt. Tím vzniká druhý histogram h_j . Pro vyhodnocení podobnosti obou histogramů se využívá statistika X^2 , která je definovaná ve vzorci (2.3.2).

$$C_{ij} = \frac{1}{2} \sum_{k=1}^K \frac{[h_i(k) - h_j(k)]^2}{h_i(k) + h_j(k)} \quad (2.3.2)$$

Kde C_{ij} je cena podobnosti tvaru dvou objektů a $h_j(k)$ histogramů.

Po aplikování a projítí všemi dostupnými body je dalším krokem provést minimalizaci. Minimalizace se provádí za pomoci vzorce (2.3.3). V případě, že nedošlo k nalezení dvojic bodů p_i a q_i , čili jinými slovy vyjádřeno, že jeden bod v prvním porovnávaném objektu nemá ekvivalentní bod v druhém obraze, je k ceně vybraného bodu přidána penalizace.

$$H(\pi) = \sum_i C(p_i, q_{\pi(i)}) \quad (2.3.3)$$

Kde $H(\pi)$ je minimalizace ceny podobnosti a C je cena podobnosti dvou porovnávaných tvarů.

Metodu porovnávání za pomoci shape contextu je možné upravit tak, aby byla odolná proti změně měřítka, pohybům po obraze a rotacím. Nezávislost na měřítku a posunům po obraze je možné docílit vypočítáním průměrné vzdálenosti mezi páry v hledaném objektu. Odolnost proti rotacím objektu se dají pokrýt použitím tečny ke každému bodu vůči ose x . Díky tomu se získají úhly tečen, které zůstávají zachované i přes rotace objektu po obraze. Touto úpravou je možné docílit kompletní odolnosti vůči rotacím objektu. Výhodnou vlastností metody shape context je to, že dokáže ignorovat lehké tvarové vady v obraze. Odolnost shape contextu vůči poruchám je možné zvýšit, když lze rozdělit části prohledávaného obrazu na části, které patří do hledaného vzoru a části které jsou mimo obraz. Části, které nepatří do hledaného obrazu, nevyužíváme k sestavení histogramu. [7]

Pro určení podobnosti mezi dvěma objekty je možné využít rovinnou transformaci. Nejvíce využívaná transformace je thin plate spline (TPS). Jedná se o transformaci, která se snaží spojit všechny body do jedné velké oblasti. TPS se při své tvorbě snaží, aby výsledný tvar roviny byl co možná nejmíň deformovaný. Principiálně je na prvním objektu vytvořena síť, která se deformuje tak, aby se podobala druhému objektu. Z výsledné deformované sítě je pak možné určit rozdíly mezi dvěma tvary. Definiční vztah pro TPS je definován ve vztahu (2.3.4).

$$f(x, y) = a_1 + a_x x + a_y y + \sum_{i=1}^n w_i U((x_i, y_i - (x, y))) \quad (2.3.4)$$

Kde a_1, a_x, a_y jsou afinní složky transformace, w_i neafinní složky, funkce U ve tvaru $U(r_i) = r_{ij}^2 \log r_i$, kde r_{ij} je Euklidova vzdálenost mezi mezníky i a j na porovnávacím objektu. [10]

Hlavní vlastností TPS transformace je najít plochu, která je nejmíň zdeformovaná. Tato vlastnost se nazývá ohybová energie a je uvedena ve vztahu (2.3.5).

$$I_f = \iint \left(\frac{\partial^2 f}{\partial x^2} \right)^2 + 2 \left(\frac{\partial^2 f}{\partial x \partial y} \right)^2 + \left(\frac{\partial^2 f}{\partial y^2} \right)^2 dx dy \quad (2.3.5)$$

Pro podmínku, aby se dosáhlo integrace druhé derivace ve vzorci (2.3.5) je nutné, aby platily následující podmínky (2.3.6), získané ze vztahu (2.3.4).

$$\sum_{i=1}^n w_i = \sum_{i=1}^n w_i x_i = \sum_{i=1}^n w_i y_i = 0 \quad (2.3.6)$$

Pro nalezení všech afinních parametrů a neafinních parametrů se využívá vztahu (2.3.7), který se nazývá matice deformačních energií, značená písmenem L .

$$L = \begin{pmatrix} K & P \\ P^T & 0 \end{pmatrix} \begin{pmatrix} w \\ a \end{pmatrix} = \begin{pmatrix} v \\ 0 \end{pmatrix} \quad (2.3.7)$$

Kde $K = U(\|(x_i, y_i) - (x_j, y_j)\|)$ vycházející ze vzorce (2.3.4), parametr P je matice, která má tři sloupce ve tvaru $(1, x_i, y_i)$ a má tolik řádků, kolik je bodů v porovnávaném objektu. Parametr a je vektor s afinními složkami. Parametr w je vektor s neafinními složkami.

V případě, že ve vypočítaném parametru v se může vyskytovat šum, je proto vhodné provést normalizaci. Normalizace se provádí pomocí vztahu (2.3.8)

$$H[f] = \sum_{n=1}^n (v_i - f(x_i, y_i))^2 + \lambda I_f \quad (2.3.8)$$

Kde parametr λ se nazývá normalizační koeficient. Jedná se o vektor, který určuje hladkost interpolace.

Po aplikování výše uvedeného postupu je dále potřeba určit shape distance. Pro určení shape distance se vychází ze vzorce (2.3.9)

$$D = aD_{ac} + D_{sc} + bD_{be} \quad (2.3.9)$$

Kde D_{ac} vyjadřuje cenu vzhledu objektu, D_{sc} vzdálenost shape contextu a D_{be} cenu transformace. Výpočet prvního parametru D_{ac} je uveden ve vztahu (2.3.10) a vyjadřuje jasové rozdíly při použití okénkových funkcí, nejčastěji se používá Gaussové okno.

$$D_{ac}(P, Q) = \frac{1}{n} \sum_{i=1}^n \sum_{\Delta \in Z^2} G(\Delta) [I_P(p_i + \Delta) - I_Q(T(q_{\pi(i)}) + \Delta)]^2 \quad (2.3.10)$$

Kde I_P a I_Q jsou porovnávané obrazy v šedotónovém tvaru, symbol Δ značí rozdílový vektor offsetu a G je funkce okna. Výpočet tohoto parametru se provádí po nalezení nejlepší shody TPS transformace. Druhý parametr D_{sc} je uveden ve vztahu (2.3.11)

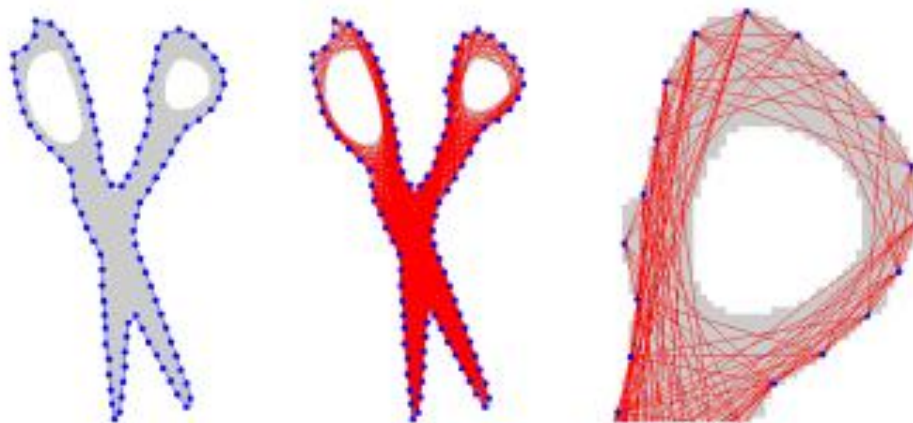
$$D_{sc}(P, Q) = \frac{1}{n} \sum_{p \in P} \arg \min_{q \in Q} C(p, T(q)) + \frac{1}{m} \sum_{q \in Q} \arg \min_{p \in P} C(p, T(q)) \quad (2.3.11)$$

Kde parametr $T(q)$ je odhad TPS transformace.

Pro výpočet posledního parametru D_{be} je využita ohybová energie, která je uvedena ve vzorci (2.3.5).

2.4 Inner distance

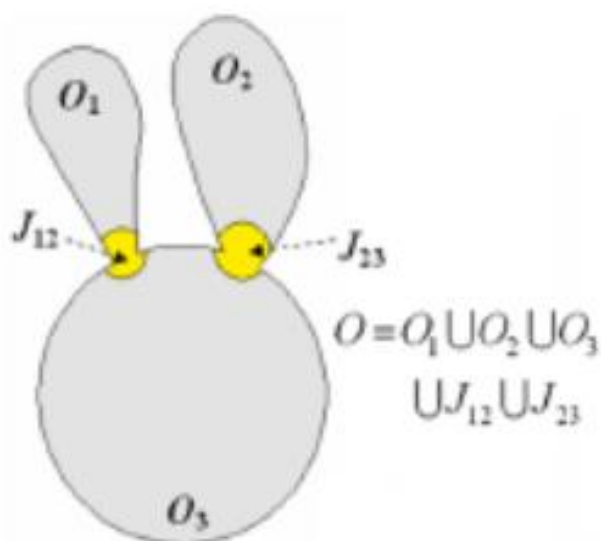
Inner-distance je metoda, která bere v ohled i vnitřní strukturu objektu oproti Shape context, kde není možné určit vnitřní rozložení objektu. Inner distance využívá stejnou počáteční ideologii jako shape context. Obě metody vycházejí ze stejných počátečních bodů, které jsou znázorněny na obrázku Obr. 2.2. Další postup zpracování je jiný. Prvním krokem je vytvoření mapy, kde je snaha propojit všechny body reprezentující obraz přímkou a následné vypočítání vzdálenosti těchto přímek. Následně jsou všechny přímky testovány, jestli jsou celé obsaženy uvnitř vybraného obrysu. Jinými slovy pokud se spoj mezi dvěma body protne a prorazí hranu objektu, tím se dostane mimo vnitřní strukturu a výsledná přímka se nezapočítává. S touto podmínkou nám vznikne síť, která reprezentuje vnitřní tvar objektu. Ukázka výsledné sítě objektu je znázorněna na obrázku Obr. 2.3. Na prostředním objektu v obrázku Obr. 2.3 je zřetelně vidět, kde se vyskytují prázdné oblasti neboli díry v objektu.[8]



Obr. 2.3 Ukázka podoby distanční sítě [9]

Po vytvoření sítě další část algoritmu spočívá v zjištění nejkratší možné vzdálenosti mezi dvěma body. Celkový výpočet by měl trvat $O(n^3)$ času, kdy $O(n)$ je čas, kdy se testuje, jestli přímka mezi dvěma body je součástí vnitřní struktury. Hlavní výhoda využití Inner distance spočívá ve schopnosti přesně odhadnout vnitřní strukturu, jelikož se síť snaží propojit pouze přímo vnitřně viditelné body.

Další výhodou Inner distance je odolnost vůči změně tvaru způsobené pohybem v kloubech objektu. Objekt lze rozdělit na několik nezávislých částí, které se budou značit písmenem O a ohybný kloub, který vzájemně propojuje části O . Ohybný kloub je značen písmenem J . Myšlené rozdělení objektu je ukázáno na obrázku Obr. 2.4. Při ohýbání kloubu J nedochází ke změně vnitřní vzdálenosti hlavních částí O . Při ohybu se mění pouze vnitřní vzdálenost oblasti kloubu J . Tyto změny jsou způsobeny deformací při ohýbání kloubu. V každém případě je změna vnitřní vzdálenosti nepatrná vůči hlavním částem, které zůstávají tvarově stejné i při ohybu v kloubech.



Obr. 2.4 Ukázka rozdělení objektu na celé části propojené pohyblivým kloubem [9]

Metoda Inner distance se dá využít v metodě Shape context, kde lze nahradit Euklidovu vzdálenost, která je použita ve vzorci (2.3.1). Další postup vylepšeného Shape contextu o inner distance spočívá v nalezení úhlu obrysové tečny mezi dvěma body. Nevýhoda této tečny je ta, že je ovlivněna změnou tvaru způsobeného ohybem kloubů. Tento problém se v každém případě dá vyřešit, když se vezme obrysová tečna počátečního bodu a dále se vytvoří tečna vůči směru nejkratší cesty, pak tyto dvě tečny spolu vytváří úhel, který zůstává stejný nezávisle na ohybu v kloubech.[8] Daný úhel se nazývá inner-angle a jeho hlavní vlastností je schopnost zachovat si vlastnosti vůči ohybům v kloubech u zvoleného objektu. V shape contextu je tento úhel využit pro orientaci. Po vytvoření Shape context histogramu je postup algoritmu stejný. Výpočet statistiky X^2 pomocí vzorce (2.3.2) a nalezení minimalizace ceny shody $H(\pi)$ je ukázán ve vzorci (2.3.3).

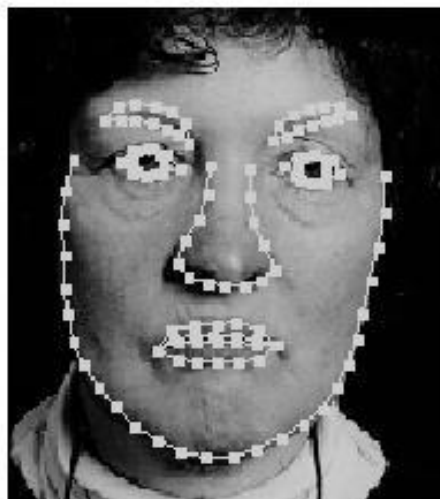
Z důvodu, že obrys se skládá z bodů, které jsou postupně skládány v chronologickém pořadí, je možné využít pro řešení podobnosti objektů dynamické programování. Hlavní výhodou využití dynamického programování je zvýšení rychlosti výpočtu algoritmu. V každém případě je zde podmínka, že oba porovnávané objekty mají shodné startovní a poslední pozice. Pokud tato podmínka není splněna, je nutné najít jiný shodný startovní bod a tím roste časová náročnost porovnávacího algoritmu. Tuto vadu lze částečně kompenzovat, pokud se bude ke shodě používat několik fixně nastavených bodů, značených písmenem k . Princip realizace dynamického programování je ve vztahu (2.4.1) [22].

$$H(i, j) = \min \begin{cases} H(i-1, j-1) + c(i, j) \\ c(i-1, j) + \tau \\ c(i, j-1) + \tau \end{cases} \quad (2.4.1)$$

Hlavní výhodou využití dynamického programování a inner distance oproti shape contextu je v lepším získání porovnání dvou objektů. Další výhodou je ta, že algoritmus se ovlivňuje pouze změnou dvou parametrů. První parametr je penalizace za neshodu dvou pozic bodů v objektu, zmíněných v předešlé kapitole 2.3 a parametrem k , který určuje, kolik bodů se bude snažit najít počáteční shodu mezi dvěma porovnávanými objekty. Poslední velkou výhodou je to, že není nutné vytvářet žádné transformační převody objektu, které se musí využívat u shape contextu. Nevýhodou metody IDSC je to, že rozdílné objekty mohou mít podobný tvar. Další nevýhoda nastane, pokud se pozorovaný objekt překryje s jiným objektem. V těchto případech účinnost metody IDSC začne slábnout.

2.5 Active Appearance Models

Metoda active appearance model (AAM) je založena na vytvoření modelu z poskládání referenčních dat a následném detekování vytvořeného modelu v neznámém prostoru. Metoda AAM vychází jako vylepšení metody active shape models (ASM). Prvotní metoda ASM principiálně pracuje tak, že se vytvoří testovací databáze, ve které jsou obrysy obrázku vyznačené sérií hraničních bodů, které popisují pozorovaný tvar. Ukázka těchto hraničních bodů je ukázána na obrázku Obr. 2.5.



Obr. 2.5 Ukázka vzhledu hraničních bodů v testovacím obrázku [11]

Po vytvoření hraničních bodů ve všech testovacích snímcích je dalším krokem vytvoření vektoru x , který obsahuje všechny hraniční body ve všech testovacích snímcích. Tento postup se nazývá Point Distribution Model (PDM). Po vytvoření vektoru x je následně využita metoda principal component analysis (PCA). Vzorec využití metody PCA pro vypočítání vektoru x je ukázán ve vzorci (2.5.1).

$$x = \bar{x} + P_s b_s \quad (2.5.1)$$

Kde \bar{x} je průměr všech testovacích tvarů, P_s ortogonální režimy variace, b_s tvarové parametry.

Kromě vytvoření vektoru x , který reprezentuje tvarové variace testovacích dat, je nutné vytvořit model, který bude vyjadřovat jasové hodnoty v bodech, které tvoří hraniční body. Získání jasových hodnot z okolí hraničních bodů je možné docílit prokládáním testovacích obrazů na sebe použitím triangulační metody z testovacích snímků nebo využití TPS transformace uvedené v kapitole 2.3. Z výsledného snímku je následně získána jasová hodnota. Z důvodu nepřesnosti jasové hodnoty způsobené osvětlením je vhodné provést normalizaci použitím offsetu a měřítka. Použití normalizace je uvedeno ve vzorci (2.5.2).

$$g = \frac{g_{im} - \beta 1}{\alpha} \quad (2.5.2)$$

Kde g je výsledná jasová hodnota po normalizaci, g_{im} nenormalizovaná jasová hodnota, β offset a α měřítko.

Z normalizovaných hodnot jasových složek pro všechny testovací snímky je stanoven průměr a následně je znovu použita metoda PCA, tentokrát pro jasové hodnoty. Vztah je uveden ve (2.5.3).

$$g = \bar{g} + P_g b_g \quad (2.5.3)$$

Kde g je výsledný vektor jasových hodnot kolem hraničních bodů, \bar{g} je průměr normalizované jasových hodnot, P_g jsou ortogonální režimy variace, b_g jasové parametry.

Z důvodu, že modely pro obrys a modely pro jas jsou rozdílné, je nutné využít znovu statistiky PCA. Prvním krokem je vytvořit společný vektor b , který vyjadřuje vlastnosti tvaru a vzhledu. Vektor b se poskládá z tvarových parametrů b_s a jasových parametrů b_g . Vztah pro výpočet vektoru b je uveden ve vztahu (2.5.4). [11]

$$b = \begin{pmatrix} w_s b_s \\ b_g \end{pmatrix} = \begin{pmatrix} w_s P_s^T (x - \bar{x}) \\ P_g^T (g - \bar{g}) \end{pmatrix} \quad (2.5.4)$$

Kde w_s je váhová matice, která kompenzuje rozdíly mezi modelem obrysu a jasového modelu.

Po vytvoření společného vektoru b , který vyjadřuje parametry pro tvar obrysu a jasů je na tento vektor použita statistika PCA a vznikne vztah (2.5.5)

$$b = Qc \quad (2.5.5)$$

Kde c je vektor parametrů, které ovlivňují modely jasů a obrysu a Q eigenvector. Za použití lineární algebry je možné společný vektor c dosadit přímo do základních tvarů (2.5.1) a (2.5.3) vzniknou vztahy (2.5.6) a (2.5.7).

$$x = \bar{x} + P_s W_s Q_s c \quad (2.5.6)$$

$$g = \bar{g} + P_g Q_g c \quad (2.5.7)$$

Z výše uvedeného postupu je vytvořený model a nyní se přistoupí k problému rozpoznání modelu v neznámém snímku. Cílem je, aby rozdíl mezi vytvořeným modelem a neznámým objektem byl minimální a proto se zavádí rozdílový vektor δI , který vyjadřuje jasové hodnoty mezi modelem a neznámým objektem ve snímku. Vztah pro rozdílový vektor je uveden ve vztahu (2.5.8).

$$\delta I = I_i - I_m \quad (2.5.8)$$

Kde I_i je vektor jasových hodnot neznámého snímku a I_m vektor jasových hodnot modelu. Cílem je, aby velikost rozdílového vektoru δI byla co nejmenší. Této vlastnosti se docílí změnou parametru modelu, který je obsažený ve vektoru c uvedeném ve výše uvedeném postupu při získávání modelu z testovacích snímků. Pro úspěšné rozpoznání objektu ve snímku je nutné řešit vazbu mezi rozdílovým vektorem

δI a chybami parametru ve vektoru parametrů modelu c . Propojení této vazby určuje vztah (2.5.9).

$$\delta c = A\delta I \quad (2.5.9)$$

Kde δc jsou nové korigované parametry, δI rozdílový vektor a A hledaná aproximace. Pro nalezení nových parametrů se využívá úpravou již stávajících parametrů modelu. Je důležité, aby stávající parametry modelu nebyly ovlivněny šumem. Parametry modelu se záměrně přemístí na neznámý model a vypočítá se rozdíl mezi parametry. Při přemístění dochází ke změně pozice, měřítka a orientace. Je vhodné tyto změny zaznamenat a přidat do všech nových parametrů obsažených ve vektoru δc . Při získání nových parametrů δc se vytvoří nový model c , na který je znovu vytvořen model tvaru x a normalizovaný jasový model g_m . Následně je získaný vzorek g_s , který se získá z modelu tvarů x . Z normalizovaného vzoru g_s je následně odečtena normalizovaným jasovým modelem g_m a je získána chyba vzoru δg , která je dosazena do vzorce (2.5.9) místo rozdílového vektoru δI . [11]

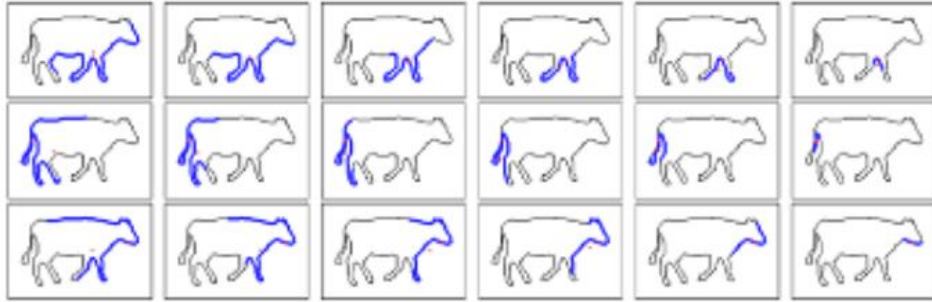
Hlavní výhodou Active appearance models je ta, že umožňuje rychle prohledat a následně najít shodu. Další výhodná vlastnost AAM je ta, že je možné snadno přizpůsobit model na nový hledaný obraz. V každém případě je zde riziko, že model nebude schopen porovnat nový obraz. Chyba může být způsobena počátečním odhadem polohy objektu, který se musí dělat ručně. Chyba porovnání může být dále způsobena nedostatečným množstvím testovacích snímků, protože AAM vyžaduje velkou databázi testovacích vzorů objektů v různých podobách.

2.6 Boundary Fragment Model

Boundary Fragment je metoda, která pro detekci objektů využívá části obrysů, které tvoří celkový obrys objektu. Jedná se o metodu, která využívá učení na sadě testovacích snímků, které obsahují hledaný vzor. Hlavní ideologie je rozdělení hranice objektů na menší části, které tvoří slabší detektor a následně slabší detektory poskládat a vytvořit silnější detektor, který je tvořen za pomoci učícího algoritmu Adaboost.

Prvním krokem pro použití metody BFM je vytvoření testovací databáze, které obsahují požadované objekty. Prvek v testovací databázi, na který se bude algoritmus učit, by měl být ideálně celý pozorovatelný a hledaný tvar objektu by měl být označený v hraničním obdélníku, který vymezuje prostor. Po sestavení všech testovacích snímků s požadovaným objektem se pro postup učení musí snímek upravit tak, aby se daly rozpoznat jeho hrany. Na obrázek se tedy použije jeden z možných hranových detektorů. V citované práci [14] je doporučeno pro vytvoření hranového obrázku využití Cannyho detektoru. Po získání snímku, na kterém jsou pouze hrany, se náhodně vyberou některé hrany, které tvoří hranici objektu. Zvolené hrany se budou brát jako hraniční kandidáti. Následně se každý kandidát postupně rozšiřuje, tak jak je ukázáno na obrázku Obr. 2.6. Při každém rozšíření dochází k vypočítání ceny shody

s validačními snímky. V posledním kroku je ke kandidátní hranici vytvořen vektor, který směřuje ke středu objektu.



Obr. 2.6 Ukázka výběru hraničního kandidátu a jeho následné rozšiřování [14]

Pro výpočet celkové shody objektu je využit vzorec (2.7.1).

$$C_{total}(\gamma_i) = C_{match}(\gamma_i) \cdot C_{loc}(\gamma_i) \quad (2.7.1)$$

Kde $C_{match}(\gamma_i)$ udává vzdálenost kandidáta vůči validačním snímkům a $C_{loc}(\gamma_i)$ udává vzdálenost středu objektu mezi validačním snímkem a odhadnutým středem získaným z testovacích snímků.

Pro výpočet $C_{match}(\gamma_i)$ se využívá Chamfer vzdálenost, která se využívá pro výpočet vzdálenosti uvedené ve vzorci (2.7.2).

$$distance(\gamma_i, I_{u_i}) = \frac{1}{|\gamma_i|} \min_{\gamma_i I_{u_i}} \sum_{t \in \gamma_i} DT_{I_{u_i}}(t) \quad (2.7.2)$$

Kde $DT_{I_{u_i}}(t)$ udává výsledek použití Chamfer vzdálenosti. Následně po získání vzdálenosti je výsledná $C_{match}(\gamma_i)$ získaná ze vzorce (2.7.3).

$$C_{match}(\gamma_i) = \frac{\sum_{i=1}^{L^+} distance(\gamma_i, P_{u_i})/L^+}{\sum_{i=1}^{L^-} distance(\gamma_i, P_{u_i})/L^-} \quad (2.7.3)$$

Kde L^+ udává počet správných validačních snímků, které odpovídají testovacímu objektu a naopak L^- udává počet nesprávných validačních snímků, na kterých je rozdílný tvar.

Při získání celkové ceny pro kandidáta $C_{total}(\gamma_i)$ je nutné se rozhodnout, jestli kandidát bude brán nebo bude zamítnut. Pro rozhodnutí se celková cena porovná s nastavitelnou hraniční hodnotou, která rozhoduje, jestli kandidát bude brán v potaz nebo zanikne. V případě, že kandidát bude mít celkovou velikost menší než je požadovaná hranice th_{hi} , tak je přijatý. Přijetí kandidáti následně vytvářejí strukturu,

kteřá nese geometrickou podobu tvaru a informaci o středu objektu. Při vytvoření struktury se dále přistupuje k užití učicích detektorů.

Učící detektory jsou využity dva. První slabý detektor h_i , který se snaží poskládat k hraničních kandidátů a z nich následně určí požadované parametry. První požadovaný parametr je, aby hranice byly obsaženy ve validačním snímku. Druhá podmínka je ta, aby vektor směřující ke středu ukazoval na stejnou oblast pro všechny kandidáty k . Poslední podmínka slabého detektoru je ta, že všechny vektory středu jsou v oblasti, kde se vyskytuje opravdový střed objektu. Pokud jsou všechny výše vypsané podmínky splněny, tak na výstupu detektoru bude jednička a v případě, že jedna z podmínek není splněna, tak na výstupu detektoru bude nula. Matematický popis slabšího detektoru h_i je ukázán ve vztahu (2.7.4).

$$h_i(I) = \begin{cases} 1 & \text{když } D(h_i, I) < th_{hi} \\ 0 & \text{jinak} \end{cases} \quad (2.7.4)$$

Kde th_{hi} je požadovaná maximální přípustná vzdálenost a $D(h_i, I)$ vzdálenost pro určité hraniční kandidáty k , jejíž výpočet je uveden ve vztahu (2.7.5).

$$D(h_i, I) = \frac{1}{m_s^2} \cdot \sum_{j=1}^k distance(\gamma_j, I) \quad (3.7.5)$$

Kde $distance(\gamma_j, I)$ Chamfer distance ze vztahu (3.7.2) a m_s je váha odhadu středu objektu.

Nyní po vytvoření slabšího detektoru h_i se přistoupí k vytvoření silnějšího detektoru, který bude vybírat ze slabšího detektoru nejlepší shodu. Pro sestavení silnějšího detektoru je využito principu AdaBoost. Pro sestavení detektoru se nejprve získají vzdálenosti pro všechny validační snímky vůči všem kandidátům. Vztah silnějšího detektoru využívající AdaBoost je uveden ve vztahu (3.7.6).

$$H(I) = sign\left(\sum_{i=1}^T h_i(I) \cdot w_{h_i}\right) \quad (3.7.6)$$

Kde T je počet iterací, $h_i(I)$ slabší detektor zkoumaného snímku a w_{h_i} je váha slabšího detektoru.

Algoritmus pracuje tak, že v prvním kroku nastaví všechny váhy w_{h_i} na stejnou hodnotu. Následně se v cyklu s počtem opakování T snaží vypočítat chybu detekce pro každý h_i . Následně je vybrán h_i , který dosahoval nejmenší chyby a zvýší se mu váha.

Nyní jsou naučené slabší a silnější detektory a přistoupí se k detekování naučeného tvaru na neznámých snímcích. Pro vybraný neznámý snímek se podobně

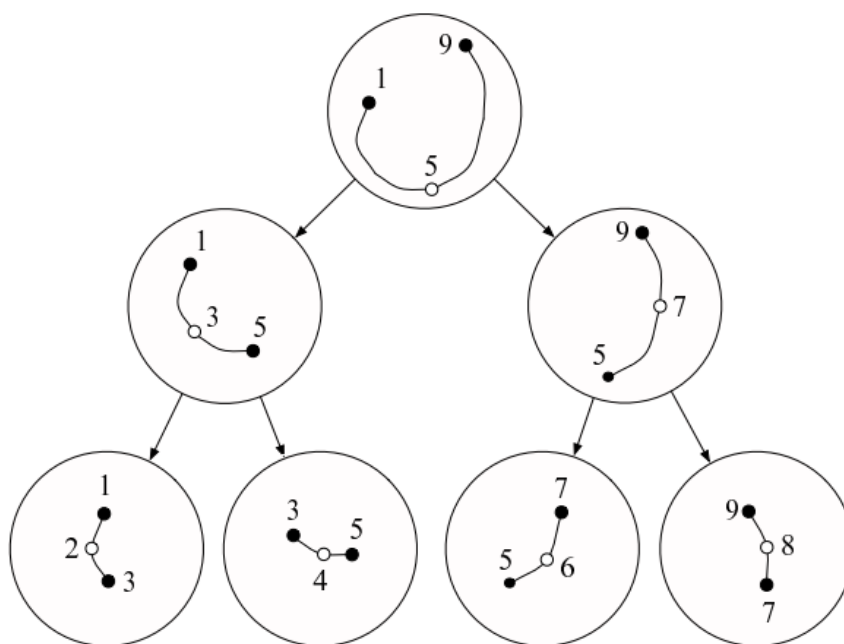
jako u testovacích snímků vytvoří obrázek, který bude ukazovat hrany všech objektů na neznámém snímku. Následně po sestavení snímku hran neznámého obrázku je hranový snímek porovnán s hraničními kandidáty, kteří jsou obsaženi ve slabších detektorech h_i . Z důvodu, že za normálních okolností je obrázek prohledáván jako celek, je nutné, aby pro nalezení více variací stejného obrázku byla upravena váha w_{h_i} pro všechny slabší detektory h_i . Změna váhy se provádí za pomoci Hough voting prostoru. V tomto prostoru každý slabší detektor hlasuje za pomoci své váhy, která určuje střed objektu. Hlasy se následně sbírají při použití sledovacího okna $W(x_n)$, které má kruhový tvar se specificky nastavitelnou vzdáleností. Následně se ve sledovacím okně spočítají všechny hlasy a pokud součet hlasů v okně je větší než detekční práh, tak je okno bráno jako kandidát, který odpovídá naučenému modelu. Pokud je okno přijato, tak slabší detektory, jejichž hlasy jsou obsaženy ve vybrané oblasti, jsou zpětně segmentovány. Následně jsou okolní oblasti, které nejsou uvnitř hran odstraněny za pomoci morfologických operací. V poslední fázi je odpovídající objekt zarámován v hraničním rámci. [15]

Metoda BFM může být odolná v určité míře vůči rotaci, změně úhlu pohledu a měřítku. Metoda BFM je odolná vůči malým rotacím objektu. Tato odolnost je způsobena využitím použití Chamefer vzdálenosti, která dokáže kompenzovat odchylky způsobené rotací. Pro kompenzování změny měřítko je možné použít normalizaci. Metodu lze upravit tak, že se normalizuje výpočet středu objektu. Metoda BFM umožňuje úpravu i vůči změně pohledu na daný objekt. Pro docílení odolnosti je vhodné některé testovací snímky natočit a použít je pro nové naučení.

2.7 Hierarchical Matching

Metoda Hierarchical matching je metoda, která využívá obrysy objektů. Obrys objektů se tato metoda snaží rozložit na menší celky, které následně vloží do vývojového stromu, který pak slouží k porovnávání s jiným objektem.

Principiálně je vytvořena křivka A , která reprezentuje obrys objektu. Křivka se skládá z bodů (a_1, \dots, a_n) . Nyní se rozdělí křivka na dvě části za pomoci rozdělovacího bodu a_i . Rozdělovací bod a_i může být umístěn kdekoliv v křivce. Po umístění rozdělovacího bodu z jedné křivky A , se obrazně vytvoří dvě křivky A_1 a A_2 . Následně je stejný postup aplikován na křivky A_1 a A_2 a tím vznikají další rozdělení křivky A na drobnější části. Tímto postupem vzniká binární strom, který se v Hierarchical Matching nazývá shape-tree. Ukázka tohoto postupu rozkládání je znázorněna na obrázku Obr. 2.7. Na daném obrázku je patrné, že na začátku pyramidy je zachycena celá křivka, která obsahuje informace o počátečním a koncovém bodu křivky. Naopak na spodku pyramidy je znázorněna vzdálenost mezi jednotlivými body. Kromě toho je možné pozorovat, pod jakým úhlem se jednotlivé body spojují.



Obr. 2.7 Ukázka Shape-tree, kde je vidět postupné rozkládání křivky na menší části [17]

Pokud je vytvořen shape-tree z nějaké křivky, je možné pomocí informací, které jsou obsaženy v shape-tree, provést rekonstrukci dané křivky. Je možné vytvořit identickou kopii postupným poskládáním jednotlivých vrstev stromu. Kromě toho je možné upravit všechny body tak, aby obrysový tvar zůstal stejný, ale současně byl objekt posunutý, otočený nebo zvětšený případně zmenšený. Další výhodou je ta, že lze výslednou křivku deformovat přidáním šumu do bodů, které jsou obsaženy ve vrstvách stromu. Pokud se následně objekt zrekonstruuje, bude viditelně deformovaný, ale opticky bude jeho tvar vypadat jako výchozí křivka nebo objekt.

Pokud je potřeba využít metodu Hierarchical matching pro rozpoznávání vzoru A s neznámým objektem B , jsou nutné sestavy shape-tree pro oba prvky A a B . Cílem správného vyhodnocení je to, aby si oba stromy byly co nejvíce podobné, tak aby deformace způsobená mapováním objektů vzorem byla co nejmenší. Tato skutečnost se vyhodnocuje sčítáním deformací v každé vrstvě tak, že je bráno v potaz o jakou vrstvu se jedná. Například na vrchní straně je velmi malá tolerance vůči deformaci, jelikož se ve výsledku může jednat o jiný objekt a naopak v nižších vrstvách je tolerance větší, protože ve spodní části se vyskytuje nepatrná část křivky. Hlavním cílem detekce je nalezení rozdělovacího bodu a_i v prvku B , kde stejnou roli má rozdělovací bod b_j . V případě, že jsou si oba body podobné, lze říct, že rozdělené části jsou si rovněž podobné. Podobnost obou křivek lze matematicky popsat rovnicí (2.7.1).

$$\psi(A, B) = \min_{b_j \in B} (\psi(A_1, B_1) + \psi(A_2, B_2) + \lambda_A \cdot dif(L_A, L_B)) \quad (2.7.1)$$

Kde λ_A je váha ohledu deformace, L_A a L_B část segmentu stromu pro obě křivky A a B , který obsahují počáteční, rozdělovací a konečný bod. Poslední funkce *dif* určuje rozdíl vzdáleností mezi body v segmentu pro obě křivky.

Pro využití této metody v obrazech, které obsahují více objektů, je nutné provést předzpracování dat. Z obrazu pro pozorování je vytvořen za pomoci hranového detektoru obraz obsahující pouze hrany objektu. Následně se vyberou všechny obrisy a porovnají se s hledaným vzorem s využitím dynamického programování.

3 IMPLEMENTACE VYBRANÝCH METOD

V diplomové práci byly prakticky realizovány tyto metody:

- Rozdílové metody
- Korelační metody
- Hausdorffova vzdálenost
- Shape context
- Inner distance

Metody jsou testovány na pokusném snímku, na kterém se nacházejí objekty, které se budou vybrané metody snažit nalézt. Každá metoda bude vyhodnocena z pohledu správného nalezení požadovaného objektu a časové náročnosti nalezení objektu. Všechny výše vypsane metody byly programovány v MATLABU verze R2014a.

3.1 Rozdílová metoda

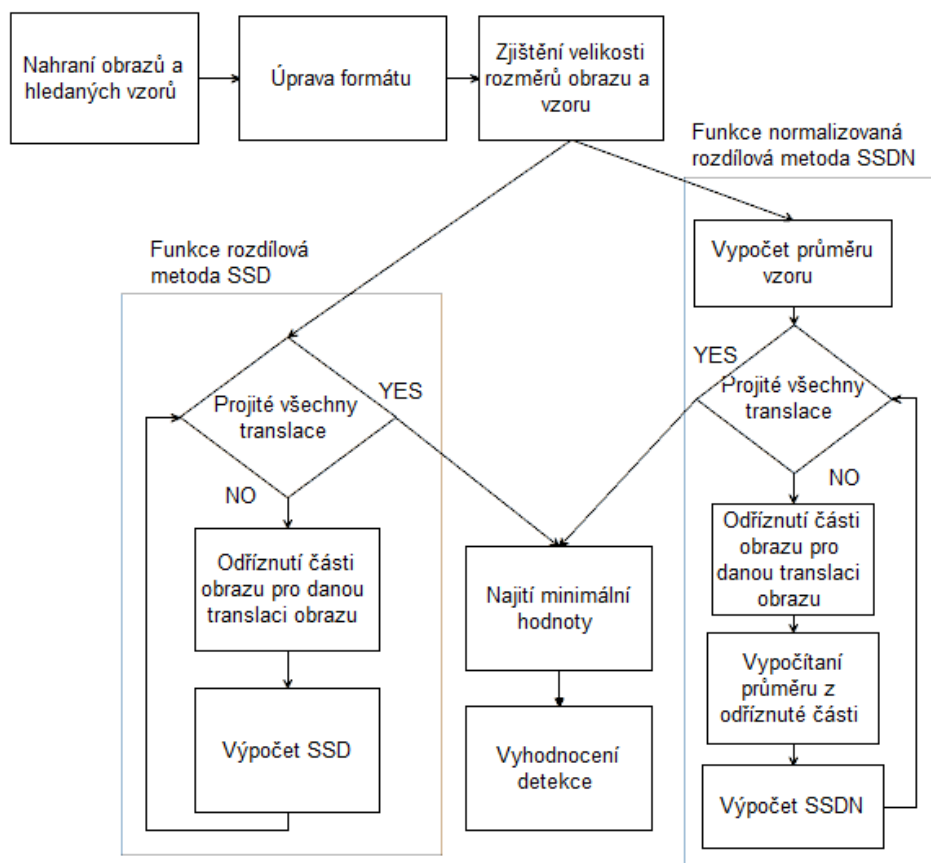
První prakticky aplikovaná metoda je rozdílová. Teorie k této metodě je popsána v kapitole 2.1. Program algoritmu programovaný v Matlabu je uveden na přiloženém CD ve složce *Programy/Roz_Kor_Hausd*. Zkrácený výpis operací pro metody rozdílové je uveden v blokovém diagramu na obrázku Obr. 3.1.

Na začátku programu je nahrán pozorovaný obraz. Následně je nahráný hledaný vzor, který se bude vyhledávat v pozorovaném obraze. Po nahrání obrazových dat je potřeba udělat úpravu formátu. První úpravou je převedení z barevného formátu na šedotónový. Druhou úpravou je převedení číselného formátu z `uint8` na `double`, abychom mohli obrazová data vzájemně sčítat. Po úpravách formátu jsou následně zjištěny velikostní rozměry pozorovaného obrazu a hledaného vzoru. Nyní, jelikož v teorii jsou uvedeny dvě metody, je nutné vytvořit dvě funkce. První funkce řeší základní rozdílovou metodu a druhá funkce normalizovanou rozdílovou metodu.

Pro první funkci se projíždí vzorem po obraze, tak že se vyřízne z obrazu aktuální oblast výskytu vzoru. Vyříznutá oblast má rozměry jako hledaný vzor. Vyříznutá oblast obrazu je následně proložena vzorem a dochází k výpočtu SSD podle vzorce (2.1.1).

Postup pro druhou funkci je obdobný, jen s tím rozdílem, že před cyklem je vypočítána průměrná hodnota vzoru a před výpočtem SSDN podle vzorce (2.1.3) je vypočítána průměrná hodnota z odříznuté části obrazu pro aktuální posun vzoru.

Po projití všech možných pozic obrazu oběma funkcemi je následně z výstupních matic zjištěna pozice, která má nejmenší matematickou hodnotu. Takto získané pozice vyjadřují nejlepší shodu v pozorovaném obraze.



Obr. 3.1 Vývojový diagram pro rozdílové metody SSD a SSDN

3.2 Korelační metoda

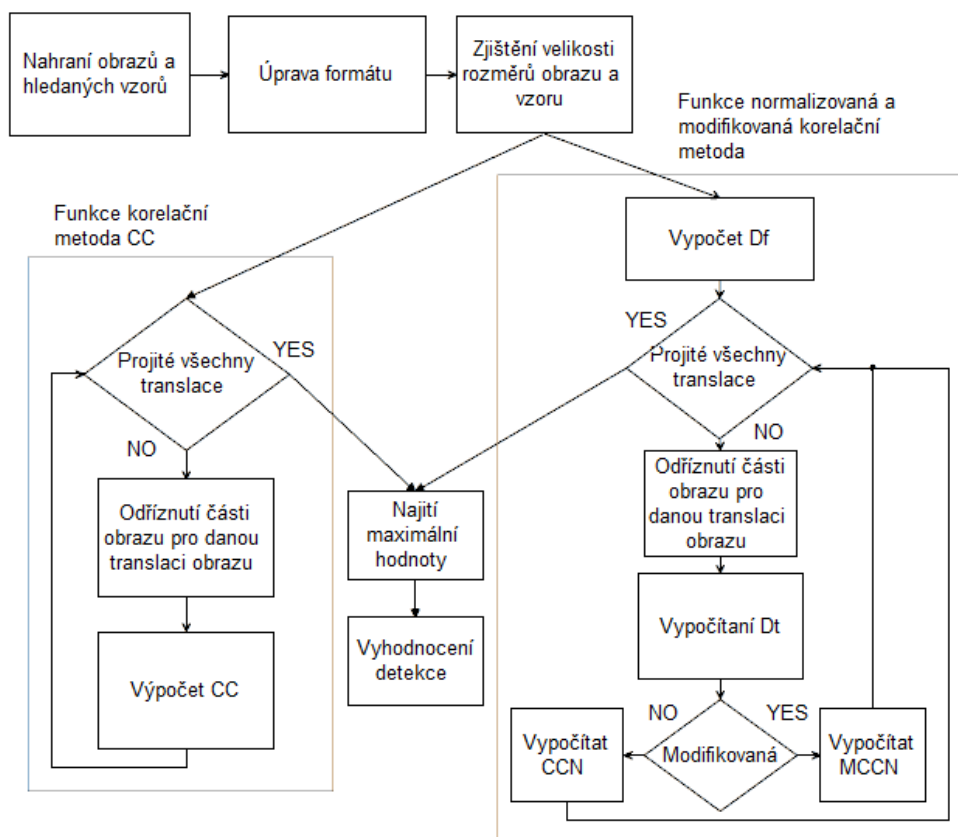
Druhá prakticky aplikovaná metoda je korelační. Teorie k této metodě je vypsána v kapitole 2.1. Program algoritmu programovaný v Matlabu je uveden na příloženém CD ve složce *Programy/Roz_Kor_Hausd*. Zkrácený výpis udělaných operací pro metody korelační jsou uvedeny v blokovém diagramu na obrázku č. 3.2.

Obdobně jako v metodě rozdílové je prvotní postup stejný, tj. nahrání obrazu a vzoru. Úprava formátu obrazových dat a zjištění rozměru obrazu a vzoru. Nyní, jelikož v teorii jsou uvedeny tři metody, je nutné vytvořit tři funkce. První funkce řeší základní korelační metodu, druhá funkce normalizovanou korelační metodu a poslední funkce řeší modifikovanou normalizovanou korelační metodu.

Pro první funkci se projíždí vzorem po obraze tak, že se vyřízne z obrazu aktuální oblast výskytu vzoru. Vyříznutá oblast má rozměry jako hledaný vzor. Vyříznutá oblast obrazu je následně proložena vzorem a dochází k výpočtu CC podle vzorce (2.1.4).

Další dvě funkce mají společný postup. Před začátkem cyklu je vypočítán parametr D_f , který je uveden ve vzorci (2.1.8). Následně dochází k odříznutí hlavního obrazu podle aktuální pozice vzoru a vypočítání parametru D_t podle vzorce (2.1.9).

Nyní dochází v obou funkcích k rozdílu výpočtu N uvedené ve vzorci (2.1.6). Pro normalizovanou korelační metodu je využit vzorec (2.1.7) a pro modifikovanou normalizovanou korelační metodu je využit vzorec (2.1.10).



Obr. 3.2 Vývojový diagram pro korelační metody CC, NCC a MNCC

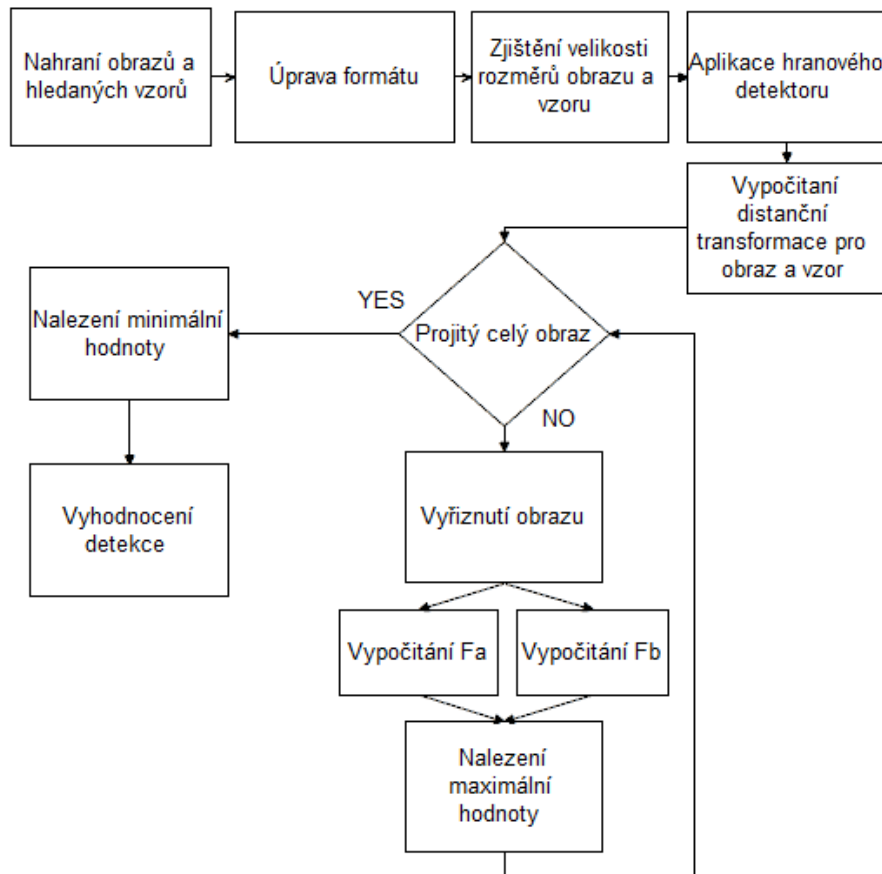
3.3 Hausdorffova metoda

Třetí prakticky realizovaná metoda je metoda využívající Hausdorffovu vzdálenost. Teoretické poznatky k této metodě je možné najít v kapitole 2.2. Výpis principu algoritmu pro metodu Hausdorff je uvedena v blokovém diagramu na obrázku Obr. 3.3. Zdrojový kód programu pro Hausdorffovu metodu je uveden na příloženém CD ve složce *Programy/Roz_Kor_Hausd*

V prvním kroku Hausdorffovy metody je nahrání obrazu a vzoru, úprava formátu obrazových dat a zjištění rozměru obrazu a vzoru. Na obraz a vzor je následně aplikován hranový detektor. V práci bylo použito detekce za použití *canny* metody s využitím příkazu *edge* v *matlabu*. Následně je vytvořena funkce, která sestaví distanční mapu z hranových obrazů. Distanční mapa je v binárním tvaru a je sestavena na principu vzorce (2.2.4). Jsou vytvořeny dvě distanční mapy D pro pozorovaný obraz a D' pro vzor. Program následně projíždí postupně celý obraz a vždy vyřizne část pozorovaného obrazu a pokryje ji se vzorem. Následně se snaží najít F_A a F_B . F_A se

najde znásobením obrazu a distanční mapy vzoru D . Po znásobení se najde maximální hodnota a ta se zapíše do F_A pro aktuální pozici vzoru. Určení F_B je principiálně stejné jen s tím rozdílem, že se násobí vzor s distanční mapou obrazu D' .

Po projití všech možných variací pozic je následně ze všech hodnot F_A a F_B zjištěna minimální hodnota, která následně odpovídá pozici nejlepší podoby vzoru v obraze.



Obr. 3.3 Vývojový diagram pro Hausdorffovu vzdálenost

K Hausdorffově metodě byly následně provedeny dvě modifikace. První modifikací bylo aplikování vzorce (2.2.8), který umožňuje zvýšit citlivost na případné překrytí hledaného objektu jiným objektem. Toto vylepšení je vytvořeno ve funkci $K_HAS(BWHled, Da, f1)$ a $L_HAS(Db, BW, f2)$, kde pomocí parametru $f1$ a $f2$ se nastavuje potřebná tolerance, ostatní parametry slouží pro tvary obrazu.

Druhou aplikovanou modifikací bylo využití rotace vzoru B s následným vypočítáním hodnoty Hausdorffovy vzdálenosti pro každou rotaci v jednom posunu obrazu. Ze všech vzdáleností získaných z rotací se následně vybere ta největší, která bude reprezentovat Hausdorffovu vzdálenost pro daný posun. Rotační modifikace je vytvořena ve funkci $ROT_HAUSR(B, DA, DB, M, U, P, pra, w)$, kde parametry B, DA a DB obsahují požadované obrazy v příslušných tvarech. Další parametr M umožňuje volit, jestli se rotace bude počítat důkladně pro nejmenší možný posun, který umožňuje

rozpoznat rotaci objektu nebo rotace budou voleny operátorem. V případě manuálního nastavení je možné rotaci upravovat parametry U a P . Parametr U nastavuje úhel jednoho kroku rotace a parametr P obsahuje počáteční a konečný úhel. Poslední parametry pra a w umožňují nastavit prahové hodnoty, která umožňují vyskočit z rotace, pokud bude nalezená vzdálenost velká než nastavený prah.

První věc, která se v této funkci provádí, je výpočet nejdelší vzdálenosti aktivního pixelu od středu vzoru. Výpočet středu vzoru vychází ze vzorce (3.3.1), získaná z [20].

$$(\bar{x}, \bar{y}) = \left(\frac{1}{n} \sum_{j=1}^n x_j, \frac{1}{n} \sum_{j=1}^n y_j \right) \quad (3.3.1)$$

Nejdelší vzdálenost aktivního pixelu od středu je následně využita k výpočtu úhlu, který zaručí, že každý zkoumaný úhel otočení způsobí, že se vzor B posune alespoň o jeden pixel. Při zjištění úhlu následně začne prohledávání obrazu. Každý posun je porovnán se všemi rotacemi vzoru, a to od 0° až do 360° . Kromě této možnosti je možné využít manuální mód, který se aktivuje příkazem $M\sim=I$.

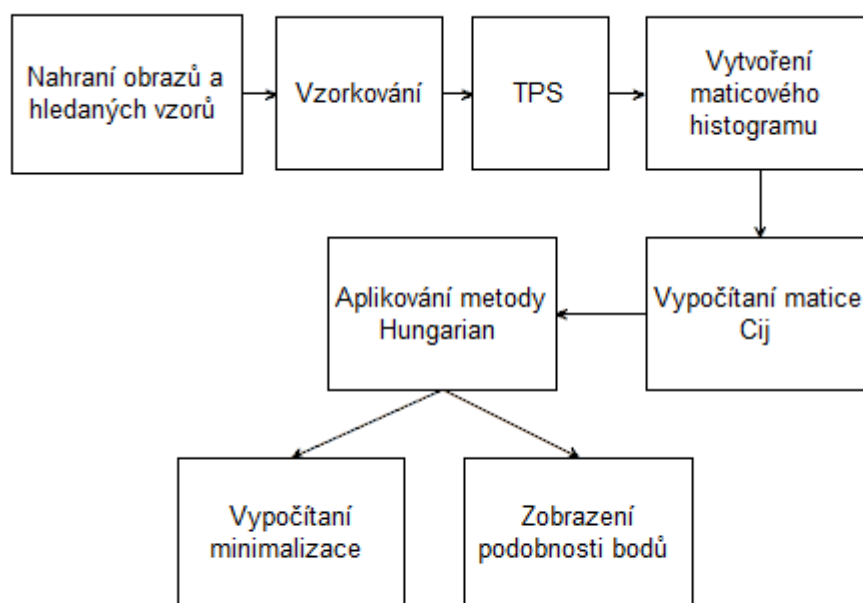
Pro rotování vzoru byla využita matlabovská funkce *imrotate*, která umožňuje rotovat obraz podle zvoleného úhlu. Z důvodu, že při rotaci dochází ke zvětšování rozměrů okrajů, je operace *imrotate* nastavena v módě *crop*, který zachovává původní velikost vzoru. Při projití všech rotací se vybere největší vzdálenost a zapíše se úhel nejlepší shody pro daný posun obrazu. Cyklus se následně opakuje pro další posun až do vyčerpání všech možných posunů.

Z důvodu časové náročnosti algoritmu, hlavně pro modifikaci využívající automatickou rotaci, byly aplikovány zrychlovací úpravy. První zrychlení, které se týká základní verze, je naprogramováno ve funkcích $K_HAS_OMEZ(B,DA,f1,prah,ob)$ a $L_HAS_OMEZ(B,DA,f2,prah,ob)$, kde parametr *prah* uvádí maximální přípustnou velikost Hausdorffovy vzdálenosti a *ob* udává oblast vynechané oblasti. Princip zrychlení spočívá v přeskočení okolní oblasti, pokud bude nalezena Hausdorffova vzdálenost větší než zvolený prah. Tato vlastnost je způsobena použitím distanční transformace, kde každý pixel má uvedenou vzdálenost k nejbližšímu pixelu. Tato vzdálenost se mění plynule. Z tohoto důvodu při posunu nedochází k velkým změnám vzdálenosti, a proto lze říct, že okolní oblasti budou mít přibližně stejnou hodnotu. Oblasti takto zvolené se následně přeskakují.

Další zrychlovací úpravy se týkají výhradně rotační úpravy, protože je tato modifikace časově náročná. První spočívá v nalezení středu vzoru v distanční mapě. Ze středu lze následně získat prahovou hodnotu, která bude následně využita pro porovnání středové hodnoty získané při ořezání posunované oblasti. Pokud je středová hodnota jiná než prahová hodnota, lze říct, že se jedná o jiný objekt. Tato vlastnost je zaručena tím, že střed rotovaného vzoru se s rotací nemění.

3.4 Shape context

Čtvrtá prakticky realizovaná metoda je shape context. Teoretické poznatky k této metodě jsou popsány v kapitole 2.3. Zkrácený výpis udělaných operací pro metodu shape context je uvedena v blokovém diagramu na obrázku Obr. 3.4. Všechny uvedené funkce je možné najít na přiloženém CD ve složce *Programy/Shape*.

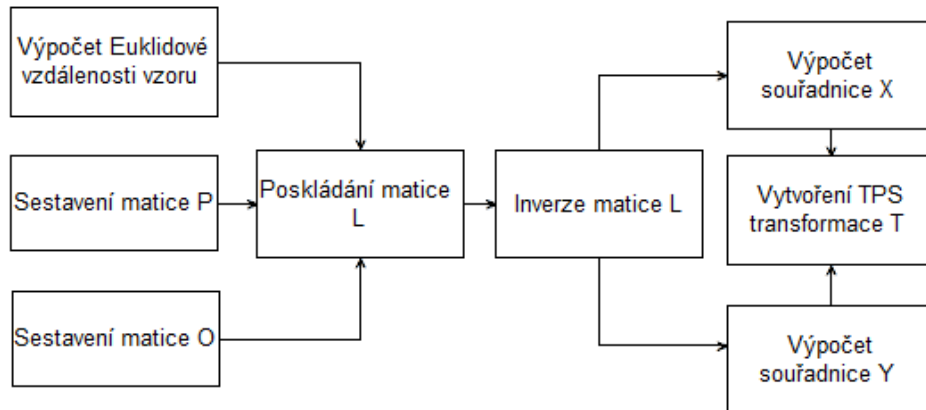


Obr. 3.4 Vývojový diagram pro shape context

V prvním kroku shape contextu jsou nahrány vzory a obrazy, úprava formátu obrazových dat a zjištění rozměrů. Následně dochází ke vzorkování obrazu a vzoru za pomoci vlastní vytvořené funkce $Body4_0(obraz, pp)$, kde proměnná pp udává celkový počet bodů. Ve funkci $Body4_0$ je přídavná funkce $Body3_2(obraz, vzd, Stara)$, kde proměnná vzd udává velikost rozptylu vzorkování a proměnná $Stara$ obsahuje vzor předcházející. Ve funkci $Body3_2$ dochází k vytvoření hranového obrazu za pomoci matlabovské funkce $edge$ nastavené jako Cannyho detektor hran. Následně se testuje, zda v proměnné $Stara$ se vyskytuje nějaká křivka, pokud ano, tak se její pozice smaže z aktuálního obrazu. Tímto je zaručeno, že se v dalším kroku vybere jiná hrana v obraze. Po přejetí této podmínky se za pomoci matlabovské operace $find$ nalezne první aktivní pixel. Souřadnice tohoto pixelu jsou následně použity v matlabovské operaci $bwtraceboundary$. Operace $bwtraceboundary$ umožňuje vyhledat všechny body, které jsou vzájemně propojeny, a to v chronologickém pořadí od počátečního bodu. Po získání všech bodů dochází k vytvoření rozptylu bodu pomocí proměnné vzd . Podoba upraveného tvaru pak vypadá podobně jako na obrázku Obr. 2.2 uvedeného

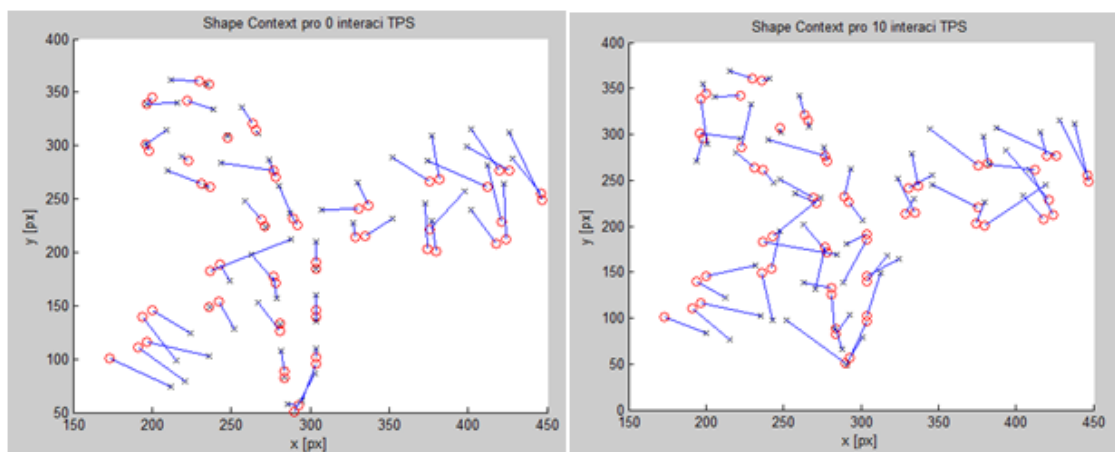
v kapitole 2.3. Ze všech vzorků, které byly získány z funkce *Body3_2* se následně vybere tolik vzorků, kolik je v proměnné *pp* a tímto je ukončen proces vzorkování.

Nyní při sestavení požadovaného vzorkování se přistoupí k vytvoření funkce řešící TPS transformaci, která umožňuje zlepšit porovnávací schopnosti a současně zvyšuje stabilitu nalezení dvojic při využití metody Hungarian. Pro účely transformace byla vytvořena funkce $TPSI(S1,S2)$, kde parametry *S1* a *S2* obsahují souřadnice obrazu a vzoru. Blokový diagram prováděných operací pro funkci TPS je uveden na obrázku Obr. 3.5.



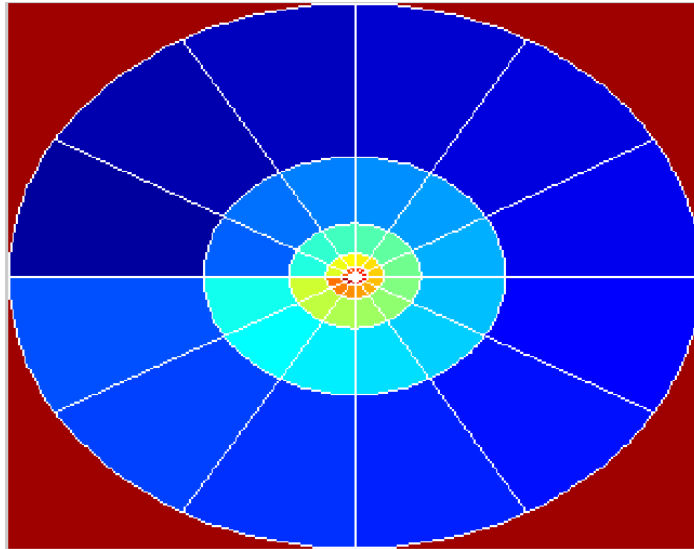
Obr. 3.5 Vývojový diagram pro TPS transformaci

V první části jsou vypočítány Euklidovské vzdálenosti všech souřadnic, které obsahují vzor. Výpočet se provádí za pomoci matlabovské operace *pdist* a výstup z této operace se převádí na čtvercovou matici za pomoci operace *squareform*. Následně se vytvoří matice *P*, která má rozměry *počet souřadnic* \times 3. Matice *P* v prvním sloupci obsahuje jedničky a v dalších sloupcích souřadnice vzoru. Dále je sestavena matice *O* s rozměry 3×3 , která obsahuje nuly. Všechny matice jsou následně sloučeny do jedné matice *L* ve tvaru (2.3.7). Z matice *L* je následně vytvořena inverze. Inverzní matice je následně vynásobena s vektorem *vx* a *vy*. Vektory *vx* a *vy* obsahují souřadnice neznámého obrazce. Pro každou souřadnici *x* a *y* je vypočítaná transformace *Tx* a *Ty*. Tyto transformace jsou následně odečteny od souřadnic vzoru. Výsledný rozdíl aplikování TPS transformace je uveden na obrázku Obr. 3.6.



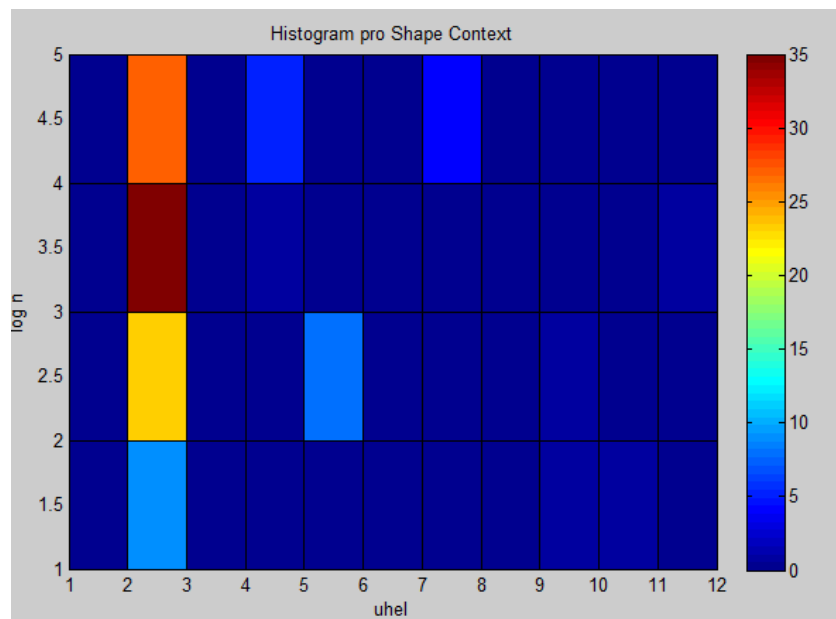
Obr. 3.6 Ukázka vzhledu využití TPS transformace na vzoru a porovnávanému obrazu

V dalším kroku algoritmu je nutné vytvořit Shape histogram. K jeho vytvoření bylo postupováno podle práce [23]. K vytvoření histogramu slouží funkce $Shape_Cij_MATRIX(S1,S2,Scal,L,nb,konst)$, kde proměnná $Scal$ udává počet oblastí, do kterých se bude přiřazovat vzdálenost bodů. Vzdálenost rozdělování oblastí je uvedena parametrem L . Pro rozdělení úhlů slouží parametr nb a poslední parametr $konst$ obsahuje penalizační hodnotu pro výpočet C_{ij} . V této funkci se vypočítává histogram pro oba obrazy. V prvním kroku funkce je volána funkce $S_vzdálenost(S1,Scal,L)$, která v prvním kroku vypočítá Euklidovu vzdálenost mezi všemi body. Výpočet Euklidovy vzdálenosti se provádí za pomoci matlabovské operace $pdist$. Výstup z této operace je následně převeden do čtvercové matice K za pomoci operace $squareform$. Ze všech hodnot matice je vypočítán průměr a tento průměr se následně vydělí ze všech hodnot. V dalším kroku dochází k přiřazování do oblastí, do kterých okolní body spadají. Počet přiřazených oblastí závisí na velikosti $Scal$ a rozdělovací vzdálenost je udána parametrem L . V případě metody maticového zápisu je využita vzdálenost L , která se v každém rozdělení velikostí zdvojnásobí. Každý bod matice vzdáleností je následně rozdělen do přiřazených vzdáleností. Výsledkem je matice obsahující čísla v rozsahu 0 až $Scal$, s tím, že čím je hodnota větší, tím je bod blíže k výchozímu bodu. Pro grafickou názornost byl vytvořen obrázek Obr. 3.7, kde rozdělení prostoru funkcí $S_vzdálenost$ je možné pozorovat jako počet kružnic od středu kružnice.



Obr. 3.7 Ukázka log-polar, který rozděluje prostor u shape context

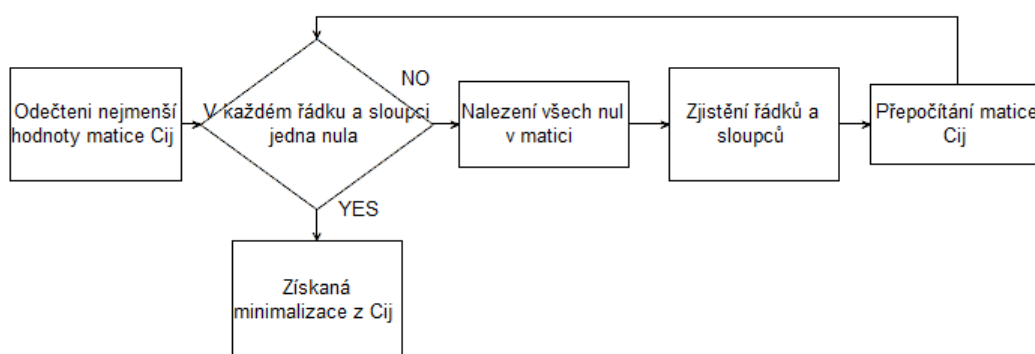
Po vytvoření histogramu vzdálenosti je následně volána funkce *Shap_uhel(S)*, která obdobným způsobem vytvoří histogram pro úhly. Stejným způsobem je vytvořena matice *K*. Následně je vypočítán úhel mezi body za pomoci matlabovské operace *atan2*, který vypočítá úhel mezi body. V případě, že úhel je záporný, tak se k tomuto úhlu přičte 2π . Tím je zaručena lepší rozeznatelnost bodu v histogramu. Získaný úhel je následně rozdělen do oblastí podle parametru *nb*. Tento krok se provádí ve funkci *S_histog(K_log, K_uhl, nb, na)*, kde parametr *K_log* obsahuje matici rozdělených bodů podle vzdáleností a parametr *K_uhl* obsahuje úhly mezi body. V této funkci dochází k vytvoření histogramu. Histogram se skládá tak, že se prohledávají obě rozdělovací matice a sčítají se stejné kombinace rozdělení. Výsledný tvar histogramu je pak ve tvaru (*nb.scal x počet_bodů*). Tvar možného výsledného histogramu je možné vidět na obrázku Obr. 3.8.



Obr. 3.8 Ukázka histogramu pro metodu Shape Context

Po vytvoření histogramu obrazu a vzoru je další krok aplikování statistiky X^2 , který vypočítává rozdílovou matici C_{ij} . Výpočet C_{ij} je prováděn ve funkci *Shap_Cij2* (*Zak, Hle, konst*), kde proměnné *Zak* a *Hle* obsahují souřadnice obrazů. V této funkci dochází k výpočtu statistiky X^2 za pomoci vzorce (2.3.2). Výsledek je následně podroben matlabovské operaci *chi2pdf*, která ze statistiky vytváří p-score, které je v rozsahu 0 až 1 a udává pravděpodobnost shody dvou porovnávaných histogramů.

Ze získané matice je dále nutné nalézt její minimalizaci. Pro možnosti vypočítání minimalizace ceny je nutné aplikovat metodu Hungarian. Aplikovaný algoritmus vychází z [19]. Blokovaný diagram pro metodu Hungarian je uveden na obrázku Obr. 3.9.



Obr. 3.9 Vývojový diagram pro metodu Hungarian

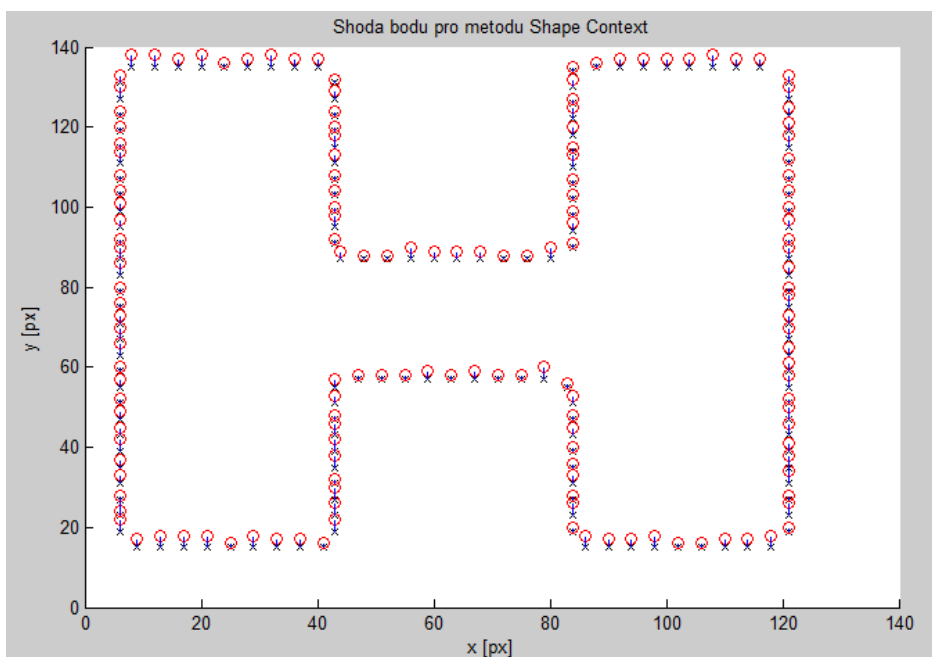
Algoritmus metodu Hungarian je vytvořen ve vlastní funkci *Hungarian* (*H, S1, S2*), kde proměnná *H* je využita pro matici C_{ij} dále *S1* a *S2*, obsahují souřadnice obrazu a vzoru. V prvním kroku funkce dochází k nalezení nejmenší hodnoty v řádcích proměnné *H*. Následně je nejmenší hodnota odečtena z vybraného řádku. Postup je následně opakován pro sloupce. Výsledkem je, že v každém řádku je alespoň jedna nula. Následně je použita funkce *Shape_Hled_NUL_MARKER2(H)*, která slouží k nalezení nejmenšího možného propojení všech nul v matici *H* s tím, že se berou celé řádky případně sloupce. V prvním kroku funkce *Shape_Hled_NUL_MARKER2* jsou nalezeny řádky, kde se vyskytuje pouze jedna 0. Řádky, které splňují podmínku, jsou evidovány v pomocné proměnné *M*. V dalším kroku dochází ke stejnému procesu pro sloupce. Tento postup se následně opakuje, dokud všechny nuly nebudou označeny. V případě, že nastane situace, kdy v řádcích je více než jedna nula, tak se vybere první hodnota v řádku od levého kraje s tím, že zbývající nuly v řádku a sloupci nebudou dále vedeny.

Tímto postupem se může stát, že některý již evidovaný řádek bude obsahovat 0, která byla přeškrtnuta sloupcem. Tento problém řeší další část programu, která je obsažena ve funkci *Shape_Hled_NUL_RADSLOUP4(M, H)*. Zmíněná funkce řeší výše uvedený problém a současně vytváří evidence řádků a sloupců. V prvním kroku funkce dochází k přeskládání pořadí proměnné *M*, která obsahuje souřadnice *X* a *Y* nul, které splňovaly podmínku. Pořadí se přeskládá od nejmenšího po největší pro hodnotu řádků

X. Následně se postupně testuje, jestli jsou v proměnné M všechny řádky. Pokud nějaký řádek chybí, je následně zapsán do pomocné proměnné $Neni$. Proměnná $Neni$ obsahuje řádky, které nesplnily podmínku a na těchto řádcích jsou nalezeny 0. Při nalezení nuly dochází k evidování sloupce, na kterém byla nalezena 0. V případě, že v evidovaném sloupci je nula, která byla přijata a je její souřadnice obsažena v proměnné M , je nutné tuto nulu odstranit a pustit celý proces ve funkci $Shape_Hled_NUL_RADSLOUP4(M,H)$ znovu, tak dlouho, dokud velikost pomocné proměnné $Neni$ je nulová. V posledním kroku jsou následně zapsány zbylé řádky a sloupce, které prošly všemi podmínkami.

Evidované řádky a sloupce jsou následně použity v další funkci $Shape_Hled_NUL_OPRAVA_H2(H,RADKY,SLOPC)$, která slouží k matematické opravě matice H v závislosti na nalezených řádcích a sloupcích. Funkce využívá pomocnou proměnou SIT , která prezentuje pozici každého prvku matice H . V prvním kroku SIT obsahuje pouze nuly. Následně jsou v řádcích vytvořeny jedničky v závislosti na proměnné $RADKY$. Obdobně je pak přičtena jednička pro sloupce. Tímto postupem matice SIT obsahuje nuly, jedničky a dvojky. Cílem korekce matice H je to, aby se z pozic matice H , která je v matici SIT pod nulou našla nejmenší hodnota. Tato nejmenší hodnota je následně odečtena ze všech pozic, kde je SIT nulová. Dále je nejmenší hodnota přičtena k pozicím SIT , kde se vyskytuje dvojka. U oblastí, které jsou v SIT označeny dvojkou, došlo k protnutí řádků a sloupců. Pro aplikování požadovaného matematického postupu jsou vytvořeny pomocné proměnné $SIT0$ a $SIT2$, které obsahují jedničky na pozicích, kde se vyskytují nuly, případně dvojky. Tyto pomocné proměnné jsou následně pro násobeny s maticí H a sečteny případně odečteny. Tím vzniká nová matice H , která se následně použije v celém procesu znovu.

Proces se opakuje, až dokud velikost evidovaných souřadnic M , které prošly všemi úpravami, nebude stejně velká jako velikost čtvercové matice H . Pokud podmínka bude splněna, proměnná M obsahuje nuly, které jsou poskládány tak, že v každém řádku a sloupci je aspoň jedna nula. Pro účely Shape contextu to znamená, že matice M udává shodu pozice z jednoho obrazu s druhým obrazem. Pro názornost je ukázka této shody uvedena na obrázku Obr. 3.10.

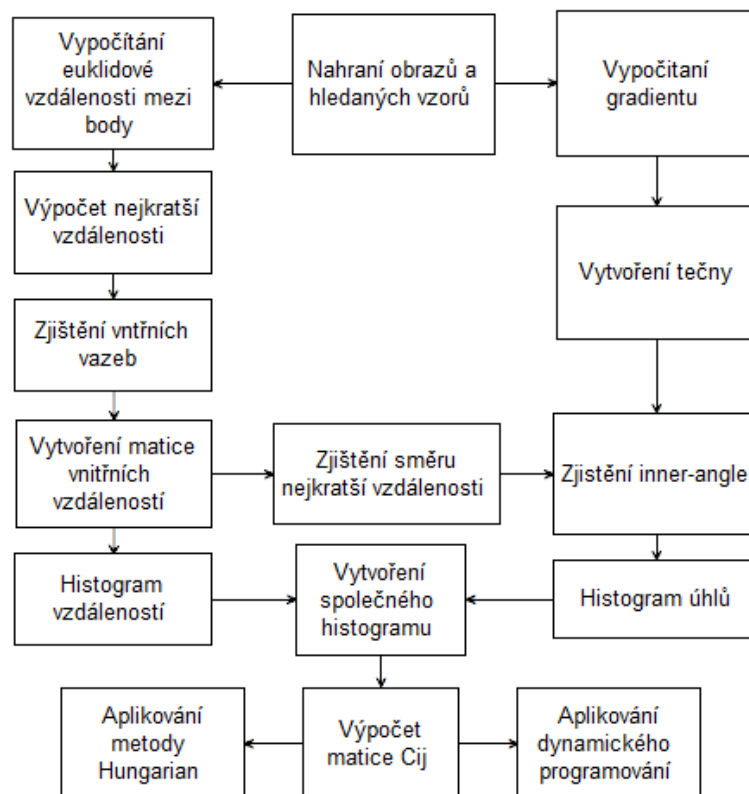


Obr. 3.10 Zobrazení nalezení vazeb mezi dvěma objekty

3.5 Inner distance

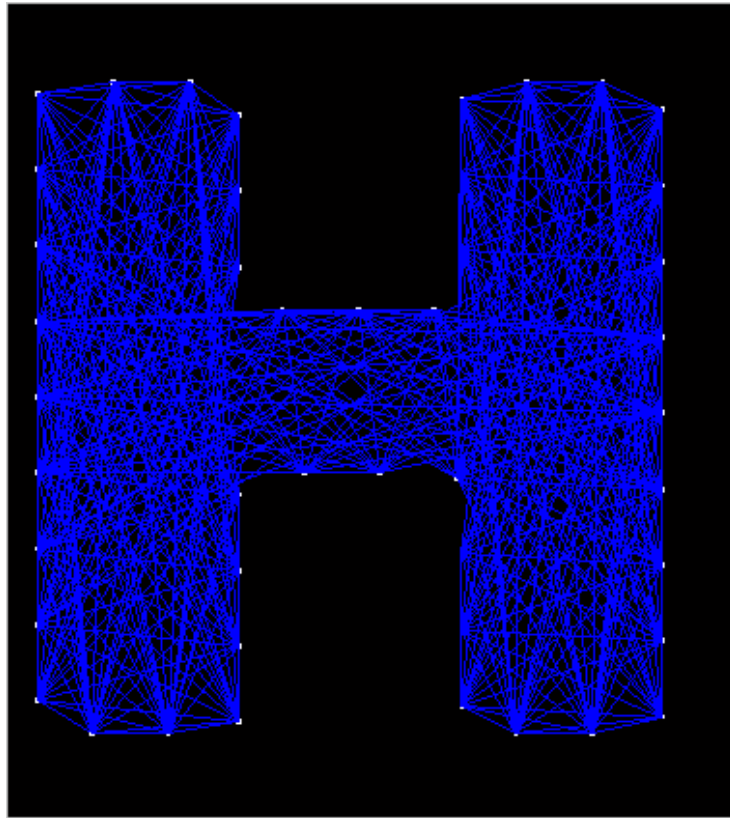
Pátá prakticky realizovaná metoda je inner distance. Teorie k této metodě je popsána v kapitole 2.4. Zkrácený výpis udělaných operací pro metodu inner distance je uveden v blokovém diagramu na obrázku Obr. 3.11. Programy k inner distance jsou uvedeny na příloženém CD ve složce *Programy/Inner*.

Prvním krokem Inner distance je nahrání vzoru a obrazu, úprava formátu obrazových dat a zjištění rozměrů. Následně je obraz navzorkován za pomoci funkce *Body4_I*, která je stejná jako v metodě shape context uvedená v kapitole 3.4. Ze získaných bodů je následně spočítána Euklidova vzdálenost za pomoci matlabovské operace *pdist*. Následně je použitý další příkaz *squareform*, který vzdálenosti z předešlé funkce převede z vektoru do čtvercové matice. V dalším kroku je nutné z matice odstranit vzdálenosti, které nejsou uvnitř objektu. Pro zajištění této vlastnosti se využívá matlabovské operace *polyxpoly*, která umožňuje zjistit, jestli došlo k propojení hranice polynomu. K vytvoření polynomu slouží hraniční body, které byly získány z funkce *Body*. K propojování nebo případnému vytváření děr polynomu je vytvořena vlastní funkce *Poly(T,Pk,O)*, kde proměnná *T* udává nový polynom, *Pk* starý polynom a *O* vybírá, jestli se nový polynom *T* bude připojovat nebo bude tvořit nový tvar. Funkce *Poly* slučuje souřadnice dvou objektů. Poslední parametr funkce určuje, jestli má vzniknout nový polynom přidáním hodnoty NaN jako oddělovač.



Obr. 3.11 Vývojový diagram pro inner distance

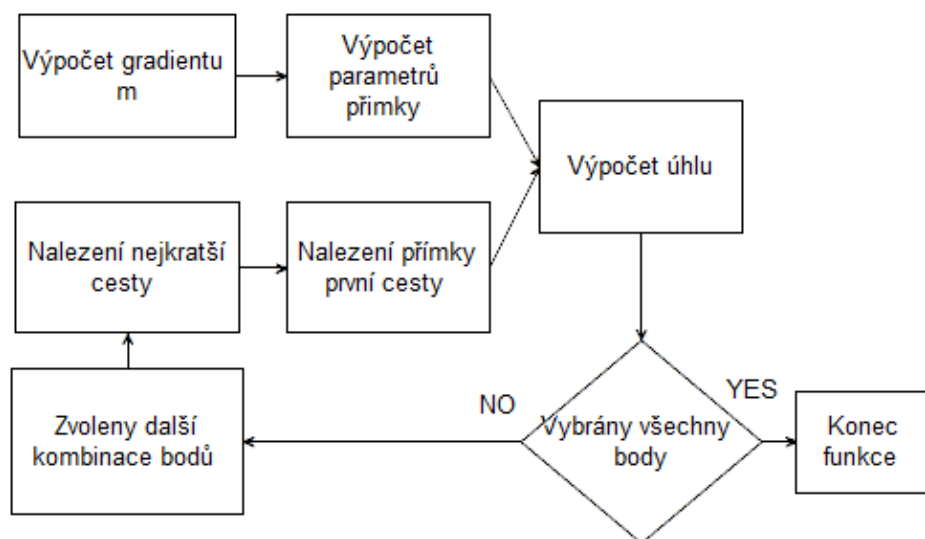
Použitím příkazu *polyxpoly* je možné odstranit velkou část nežádáných vzdáleností, ale některé vzdálenosti zůstávají, a to především ty, které jsou proti sobě odděleny volným prostorem. Z tohoto důvodu byla vytvořena vlastní funkce *Hrany(NOV, l, okraj, z)*, která zvětší okraje o zvolenou velikost pixelu parametrem *z*. Tímto krokem se vytvoří přídavná vrstva, která odstraní zbylé nežádané oblasti. Výsledné propojení je ukázáno na obrázku Obr. 3.12, kde je možné vidět vzájemné vazby všech vnitřních bodů.



Obr. 3.12 Vzhled propojení vnitřních vazeb zkoumaného objektu

Při získání vzdáleností matice je pro další účely nutné najít inner-angle. Pro nalezení tohoto úhlu slouží funkce $Inner_uhel(DG, sam, samCEL)$, kde proměnná DG tvoří stromovou strukturu vytvořenou příkazem $sparse$, sam obsahuje body okraje objektu s vybranou vzdáleností a $samCEL$ obsahuje všechny body okraje objektu. Princip vyhledávání inner-anglu je zobrazen v diagramu na obrázku Obr. 3.13. Námět k tomuto postupu vychází z [21].

V dalším kroku je cílem nalézt nejkratší vzdálenosti mezi všemi body. Pro tyto účely jsou použity matlabovské operace $sparse$ a $graphallshortestpaths$. Operace $sparse$ převádí vytvořenou matici do podoby grafu a umožňuje využití operace $graphallshortestpaths$. Operace $graphallshortestpaths$ umožňuje nalézt nejkratší vzdálenosti pro všechny body z obrazu a nabízí několik metod, jak vypočítat nejkratší vzdálenost. Všechny vzájemné vzdálenosti se následně použijí do vlastní funkce $I_vzdalenost(dist, Scale, L)$, kde proměnná $dist$ představuje vzdálenosti bodů získané příkazem $graphallshortestpaths$, proměnná $Scale$ představuje počet rozdělení vzdálenosti a proměnná L vyjadřuje výchozí vzdálenost. Princip funkce $I_vzdalenost$ je obdobný jako funkce $S_vzdalenost(S, Scal, L)$, která byla využita v metodě Shape context, jež je uvedena v kapitole 3.4. Jediný rozdíl je ten, že místo vzájemné vzdálenosti všech bodů, jsou použity jejich vnitřní vzdálenosti obsažené v proměnné $dist$.

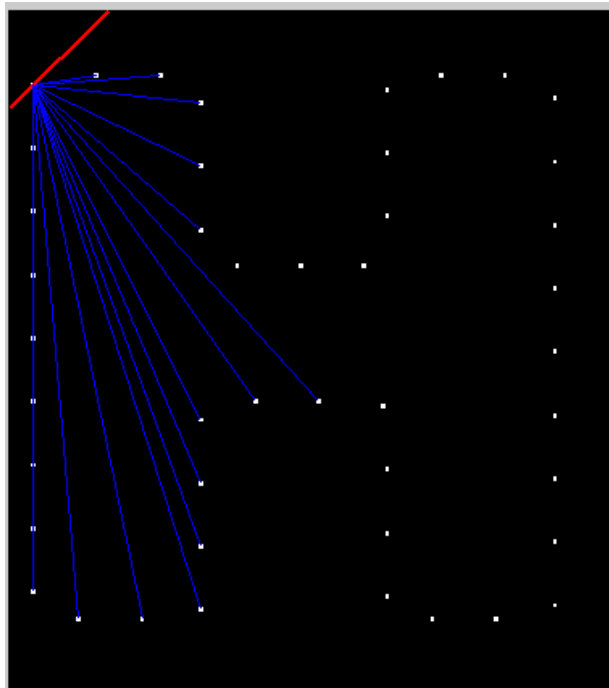


Obr. 3.13 Diagram k vypočítání inner-anglu pro metodu Inner distance

Hlavním cílem pro nalezení inner-anglu je získání tečny kontury pro testovaný bod. Z tečny lze následně získat úhel, který svírá s nejkratší cestou při vyhledávání nejkratší vzdálenosti. Pro získání tečny je nutné v prvním kroku získat gradient m . Gradient se dá získat za pomoci vzorce (3.5.1) tak, že se gradient m vypočítá pro všechny souřadnice, které jsou obsažené v proměnné $samCEL$. Pro výpočet gradientu je využita matlabovská operace $gradient$. Gradient se počítá pro souřadnice x a y zvlášť. Následně jsou tyto gradienty vzájemně vyděleny a vzniká tím výsledný gradient m . Matematický popis získání gradientu je uveden ve vztahu (3.5.1).

$$m = \frac{y_2 - y_1}{x_2 - x_1} \quad (3.5.1)$$

Z výsledného gradientu m je následně vypočítán tangent tečny kontury za pomoci matlabovské operace $atan2$. Nyní jsou vytvořeny všechny úhly tečny a je dále nutné z těchto bodů vybrat ty, které jsou obsaženy v proměnné sam . V proměnné sam jsou body, na které je vytvořena matice nejkratších vzdáleností. Úspěšným nalezením shody je následně vytvořena tečna pro daný bod. Vzhled nalezení a vykreslení tečny je zobrazen červeně na obrázku Obr. 3.14.



Obr. 3.14 Ukázka tečny (červená) a všech počátečních směrů nejkratší vzdálenosti (modrá)

Při nalezení tečny je dále nutné najít směr nejkratší vzdálenosti mezi počátečním a prvním směrovaným bodem. Pro tyto účely je využita stromová struktura DG , která obsahuje všechny cesty vzdáleností mezi body. Pro nalezení cesty je využita matlabovská operace `graphshortestpath`, kde jedna z výstupních proměnných obsahuje celou nejkratší cestu. Pro účely vytvoření přímky je nutné znát alespoň dva body. První bod je zvolený počáteční bod a druhý bod je vybrán jako druhý v pořadí v proměnné `path`, která reprezentuje cestu nejkratší vzdálenosti mezi dvěma body. Přímky nejkratších vzdáleností pro všechny body vůči jednomu bodu jsou vykresleny na obrázku Obr. 3.14 modře.

Při získání obou přímek je v posledním kroku nutné nalézt úhel mezi těmito přímkami. Úhel se počítá za pomoci matlabovského příkazu `atan` tak, že se vypočítá tangent ze vzorce (3.5.1) pro každou přímku zvlášť a následně výsledky odečtou. Tento krok je následně opakován, dokud není vytvořena matice K_{uhl} , která má čtvercový tvar a velikost stejnou jako počet bodů obsažených v proměnné `sam`.

Po získání matice obsahující vnitřní úhly je dalším krokem spojení této matice s maticí vzdáleností. K tomuto účelu slouží funkce `S_histog(K_log, K_uhl, nb, scale)`. Funkce `S_histog` projíždí všechny variace oblastí, které vznikají rozdělováním prostoru. Celkový počet rozdělených oblastí je násobek mezi hodnotou v proměnné `Scale` a proměnné `nb`. Rozdělený prostor vypadá obdobně jako použitý kruh na obrázku Obr. 3.5, použitý v shape contextu.

Tím vzniká histogram pro inner distance. Proces se následně opakuje pro druhý obraz a z obou histogramů je sestavena rozdílová matice C_{ij} za pomoci funkce.

Při vypočítání matice C_{ij} je následně použita funkce $Hungarian1(H,S1,S2)$, která je stejná jako v minulé kapitole 3.4. Výsledný vzhled je obdobný jako na obrázku Obr. 3.8 z minulé kapitoly.

Kromě výpočtu matice C_{ij} byl realizovaný jiný postup výpočtu minimalizace. Jedná se o postup, který je popsán v teorii uvedené v kapitole 2.3. Jedná se o využití dynamického programování. Pro účely dynamického programování byla vytvořena funkce $DP_prog(C,t)$, kde parametr C obsahuje matici C_{ij} a parametr t obsahuje hodnotu penalizace, pokud není shoda bodu. Pro metodu Inner distance je optimální použití dynamického programování, jelikož metoda Inner splňuje podmínky chronologického navázání bodů. Realizovaný je vztahem (2.4.1).

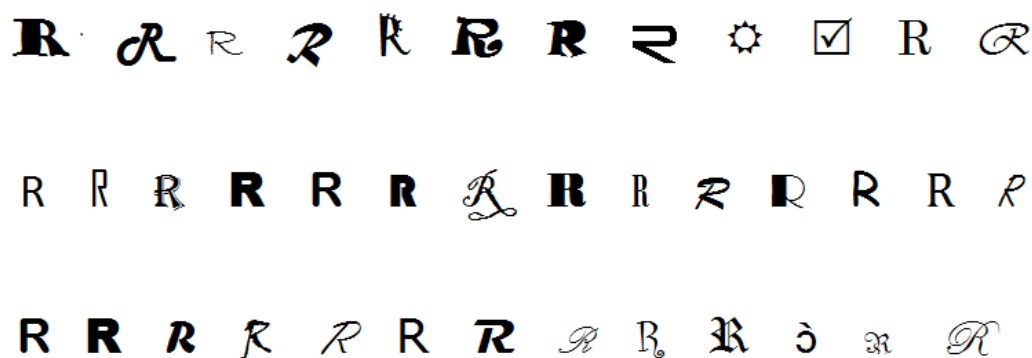
Pro zajištění podmínky, že pozice startovního a konečného bodu je pro oba obrazy stejná, byla vytvořena funkce $DP_fix(H,k)$, kde proměnná H obsahuje C_{ij} a k udává požadovaný rozptyl. Princip funkce spočívá v přeskládání řádků matice C_{ij} , tím je vytvořeno to, že dochází k výběru jiné dvojice bodů vůči oběma porovnávaným obrazům. Tímto je zaručené, že dochází k nalezení shodného počátečního bodu. Parametrem k je možné nastavovat přeskokování řádku. Takže při nastavení $k=1$ jsou přeskládány všechny řádky matice. Při nastavení $k=4$ se se posunou vždy 4 řádky. Upravená matice je následně použita ve funkci $DP_prog(C,t)$ a ze všech výsledků je nalezena nejmenší hodnota, která reprezentuje minimalizaci ceny matice C_{ij} .

4 TESTOVÁNÍ METHOD

Pro účely diplomové práce byly vytvořeny testovací obrázky. Každý testovací obrázek je uveden v příslušných kapitolách. Všechny obrázky byly vytvořeny za pomoci programu Malování v operačním systému Windows 7 a některé byly upraveny programovým rozhraním Matlab. Testovány budou rozlišovací schopnosti detekce při různých deformacích znaku, změnou velikosti, změnou natočení a rozpoznáním podobnosti vůči jiným tvarům. Všechny testovací snímky je možné najít na přiloženém CD v adresáři *Galerie*.

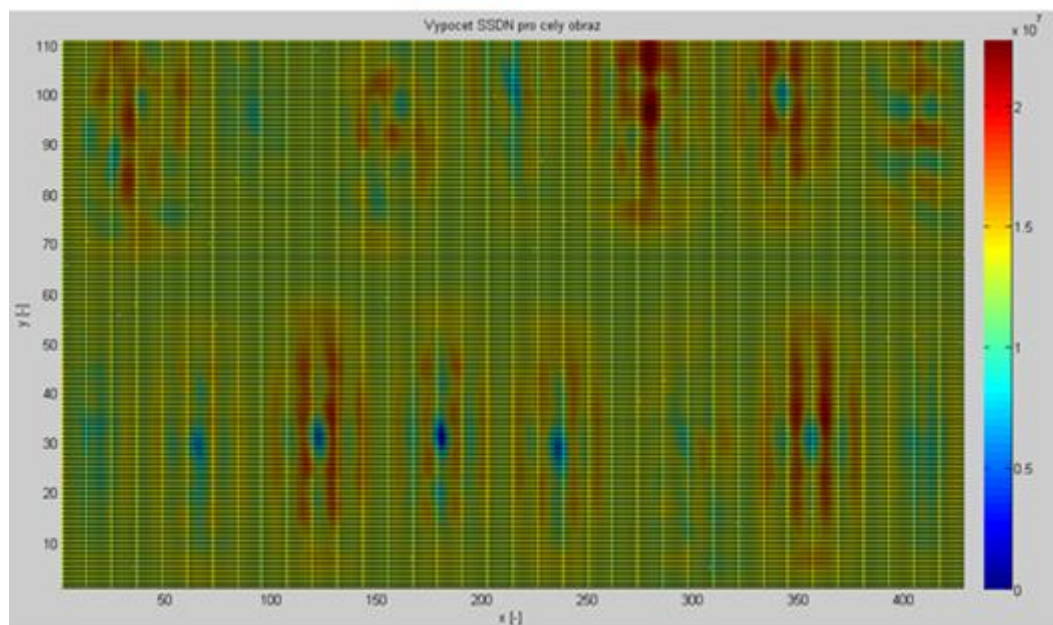
4.1 Testování rozlišovacích schopností

Účelem první kapitoly je seznámení a otestování funkčnosti základních verzí testovaných metod. K testování rozlišovacích schopností byl vytvořen obrázek Obr. 4.1. Na obrázku jsou zobrazeny různé fonty pro písmeno R. Z těchto písmen se vybere jedno písmeno a toto písmeno bude následně porovnáváno s vybranými metodami. Celková velikost obrazce je 504x294 px.



Obr. 4.1 Testovací obrazce pro hledání podobnosti

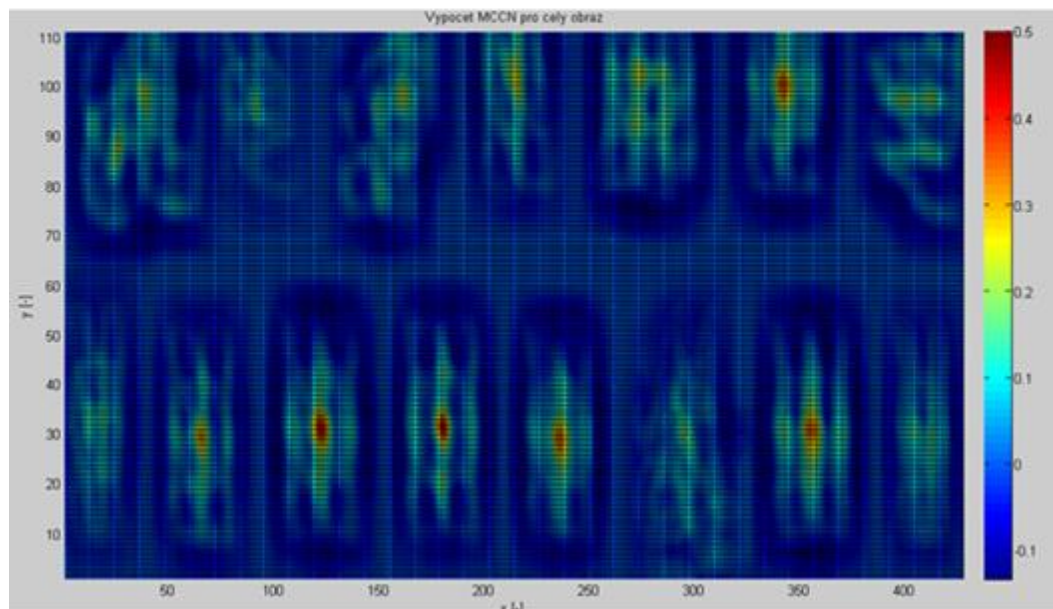
První metody, které se budou testovat, jsou metody rozdílové a korelační. Vyhledávání probíhá tak, že se hledaný vzor, který je ukázán na obrázku Obr. 4.2 v červeném rámečku, přikládá k obrazu a vypočítává podobnost pro daný posun. Tím se vytváří matice, která má velikost (*velikost obrazu - velikost vzoru*). Každá pozice v matici reprezentuje posun vzoru po obraze. Vzhled této matice je ukázán na obrázku Obr. 4.2 a 4.3 v surf grafech. Pro přehlednost je vybráno jenom několik symbolů z hlavního obrazu. Zkrácená verze obrázku má rozměry 462x149 px.



Obr. 4.2 Surf graf pro metodu SSDN testu 4.1

Na obrázku Obr. 4.2 je zobrazená výsledná podoba matice pro metodu SSDN. Shoda podobnosti pro metodu SSDN je závislá na nejmenší velikosti pozice v matici. V případě tohoto surfu je shoda vyznačena modrou barvou. Ze surfu je možné vyhodnotit, že byl nalezen požadovaný vzor. Dále je ze surfu patrné, že byla nalezena podobnost s protějšími znaky, jelikož mají velice podobnou strukturu. Co se týče jiných znaků, je možné pozorovat, že některé variace nebyly vůbec nalezeny, případně je jejich velikost nerozeznatelná. Tento výsledek se dá očekávat, jelikož se jedná o jednoduchou metodu, která jenom porovnává vzájemnou podobnost pixelů.

Obdobný princip je ukázán na obrázku Obr. 4.3, který obsahuje surf pro metodu MNCC. V surfu pro metodu MNCC je snaha najít oblasti v matici, které mají největší hodnotu. I zde je vidět, že došlo ke shodě vzoru s obrazem. V každém případě je patrné, že došlo k více detekcím podobnosti oproti metodě SSDN. Nově detekované znaky jsou dva znaky *R* na pravé straně.



Obr. 4.3 Surf graf pro metodu MNCC testu 4.1

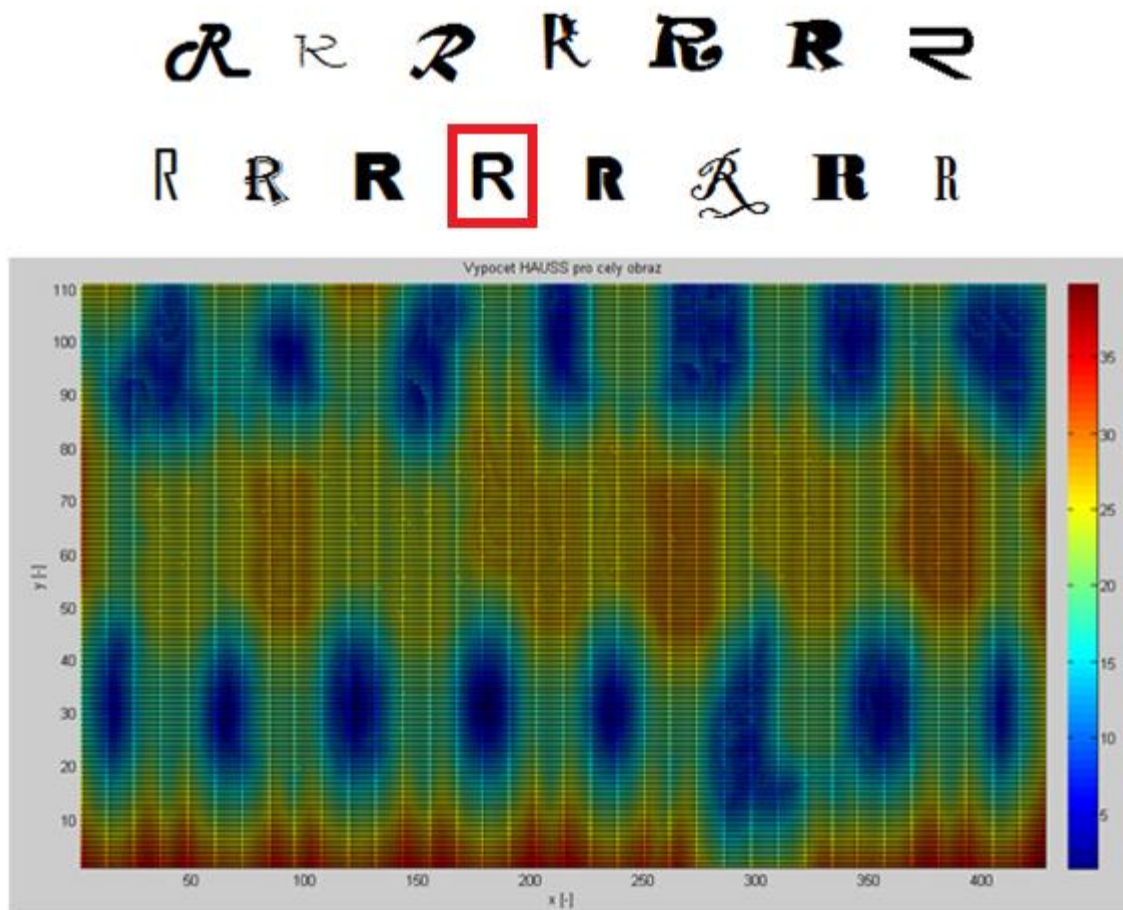
V diplomové práci byly realizovány všechny variace rozdílových a korelačních metod. Výsledky pro pokus s rozlišováním podobnosti jsou ukázány v tabulce č. 1. Kromě úspěšnosti je v tabulce uveden průměrný čas výpočtu.

	METODY				
	SSD	SSDN	CC	NCC	MNCC
Nalezená pozice [-]	ANO	ANO	NE	ANO	ANO
Čas výpočtu [s]	22,5	65,7	20,1	50,3	50,2

Tabulka č. 1 Tabulka rychlosti algoritmu pro různé typy rozdílových a korelačních metod pro 10 opakování

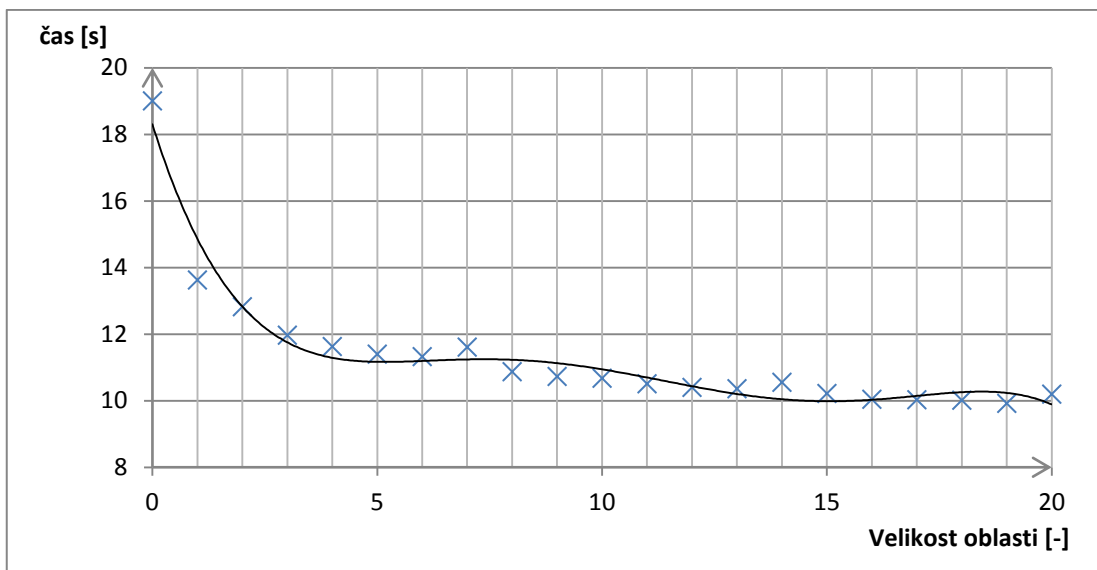
Z tabulky je možné vyčíst, že základní neupravené verze jsou pro obě metody přibližně stejně časově náročné. Naopak vylepšené metody trvají dvakrát déle. Tato vlastnost je logická, jelikož vylepšené verze vypočítávají dodatečně průměry z oblastí. Metody nevykazovaly žádné abnormality, takže časy byly přibližně stejné. Z pohledu přesností nalezení všechny metody kromě CC našly požadovaný vzor v obraze. Metoda CC nebyla schopná nalézt podobnost. Tato vlastnost je očekávaná, jelikož metoda CC má problémy nalézt hledaný vzor, pokud je v obraze více podobných objektů jako vzor. Z obrázků Obr. 4.2 a 4.3 je možné vyvodit, že metody rozdílové a korelační jsou vhodné k nalezení přesného tvaru vzoru.

Další zabývanou metodou je metoda využívající Hausdorffovy vzdálenosti. Na metodu byl aplikovaný stejný porovnávání obraz jako v předešlých metodách. Výsledný výsledek je ukázán na obrázku Obr. 4.4. Parametry citlivosti byly zvoleny $f_1=1$ a $f_2=1$. Jedná se tedy o základní verzi, která vychází ze vzorce (2.2.7).



Obr. 4.4 Surf graf pro metodu Hausdorff testu 4.1

Z obrázku Obr. 4.4 je vidět, že detektor reaguje na všechny typy znaku R. Je navíc zjevné, že intenzita modré barvy je nejvíc patrná u hledaného vzoru. Ze surf grafu je vidět, že dochází ke změnám vzdálenosti plynule, proto je možné otestovat zrychlovací úpravu, která byla uvedena v kapitole 3.3. Princip zrychlení je omezení vyhledávání okolí, pokud je pro danou lokaci Hausdorffova vzdálenost větší než práh. Pro vyhodnocení efektivity uvedeného principu zrychlení byl upravený algoritmus použitý k prohledání částečného testovacího obrázku, který je uvedený na obrázku Obr. 4.4. Pro lepší orientaci jsou výsledky zobrazeny v grafu č. 4.1.



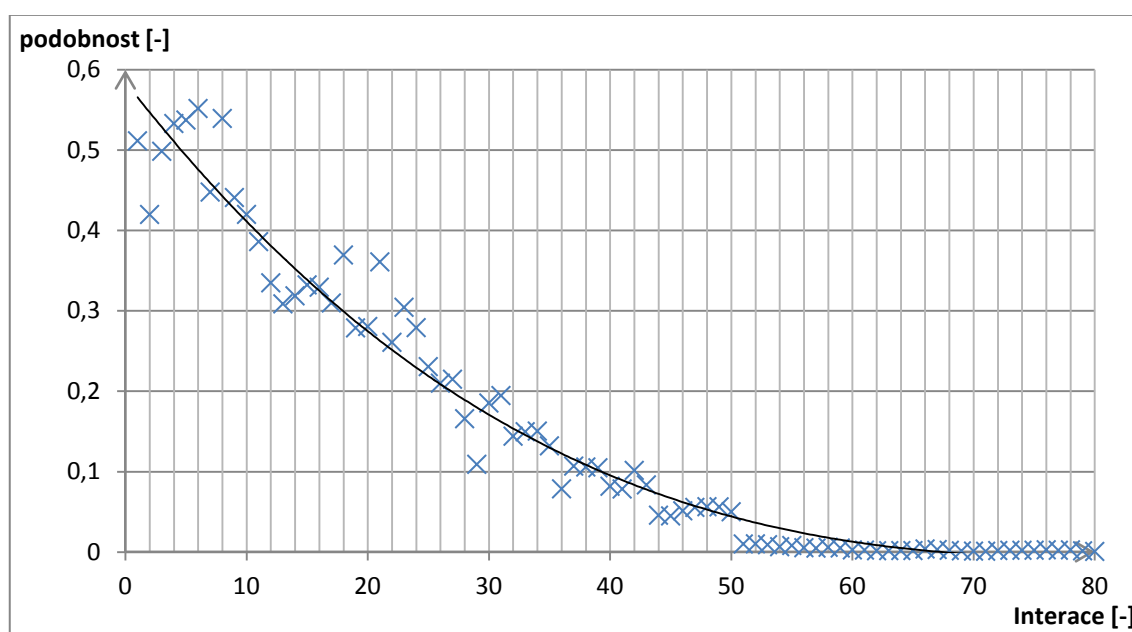
Graf č. 4.1 Závislost zrychlení v závislosti na velikosti vynechané oblasti pro zrychlovací metodu u metody Hausdorff

Další metoda, která se bude testovat je shape context. Princip detekce bude odlišný oproti předešlé metodě. Každý znak bude vyextrahovaný z obrázků Obr. 4.1 a následně porovnán s požadovaným vzorem. Pro zautomatizování extrakce byla využita metoda amplitudové projekce. Více informací o této metodě je možné zjistit v [13]. Jedná se o jednoduchou metodu, která sčítá všechny pixely v každém sloupci a řádku. Z těchto součtů je následně možné rozdělit obraz na oddělené oblasti, kde v každé oblasti je pouze jeden znak. Funkce, která provádí amplitudovou projekci se nazývá *AmpProj(ob)* a její zdrojový kód je uveden na CD ve složce *Programy/Shape*.

Naprogramovaná verze shape contextu umožňuje zvolit si mnoho parametrů. Prvním možným parametrem je možnost výpočtu histogramu využitím grafického obrazce uvedeného na obrázku Obr. 4.5 nebo využití maticového principu využitého v inner distance. U histogramu je možné si vybírat počet oblastí pro vzdálenosti na a výběr počtu oblastí pro rozdělení úhlů nb . Další možnost, jež lze upravit, je počáteční násobek vzdálenosti L . Úpravou těchto parametrů dochází k následujícím změnám: v případě, že parametr na bude malý, dochází ke snižování počtu rozdělovacích oblastí, což má za následek snížení rozdílové odchylky s tím, že toto lze chápat jako zkreslený výsledek, jelikož některé body nebudou započítány do histogramu, neboť jejich vzdálenost je mimo pokrytou oblast. Naopak pokud na bude velký, dojde k pokrytí všech bodů do histogramu s tím, že rozdělovací oblasti, které jsou vzdálené a neobsahují body, zpomalují rychlost výpočtu algoritmu, jelikož se tyto oblasti též musí testovat na přítomnost bodů. Velmi obdobný princip je i při úpravě úhlů nb . Výběrem malého rozdělení úhlů dochází k zmenšování rozdílů. Při malém nb vzniká riziko, že naprogramovaná metoda Hungarian nedokáže zjistit minimalizaci rozdílové matice C_{ij} a zkolabuje. Při testování většího množství rozdělovacích oblastí nb dochází k zvětšení přesnosti podobnosti. Tato vlastnost se vyplatí pouze tehdy, pokud se bude vyhledávat přesný objekt v případě, že se ve vyhledávaném objektu vyskytují odchylky

od ideálního stavu. V tomto případě může mnoho rozdělovacích polí zhoršit vyhledávání objektu. Kromě této vlastnosti dochází ke zvětšování časové náročnosti výpočtu. Z těchto poznatků je doporučeno volit na tak, aby oblasti rozdělení vzdáleností pokryly pokud možno většinu vzorkovaných bodů a nb byl zvolen mezi 6 až 12 s tím, že menší nb snižuje výpočtovou stabilitu programu.

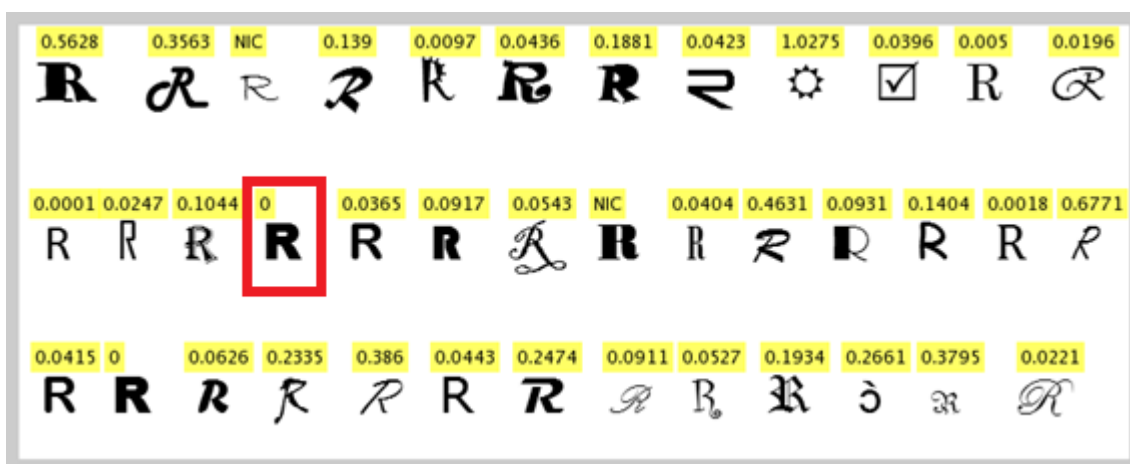
Dalším krokem, jež lze upravovat v shape kontextu, je možnost si zvolit počet iterací TPS transformace, která umožňuje zlepšovat kvalitu porovnávání. Vylepšení pozorování v závislosti na počtu opakování TPS funkce je zobrazeno v grafu č. 4.2. Graf byl sestaven s parametry $L=0,125$, $na=4$ a $nb=8$. Z grafu č. 4.2 je možné vyčíst, že TPS transformace drasticky zvyšuje detekční schopnosti, ale velké množství TPS iterací může způsobit to, že dojde ke ztrátě rozpoznávacích schopností.



Graf č. 4.2 Závislost míry podobnosti pro počet opakování TPS transformace

Z výše uvedených poznatků se nyní přistoupí k testování shape kontextu. Jako první variace testu byly použity tyto parametry: Počet vzorků: 60, $L=0,125$, $na=4$, $nb=8$ a $TPS=1$. Výsledný tvar je možné vidět na obrázku Obr. 4.5. Každý znak byl vzájemně testován s hledaným vzorem vyznačeným červeným rámečkem.

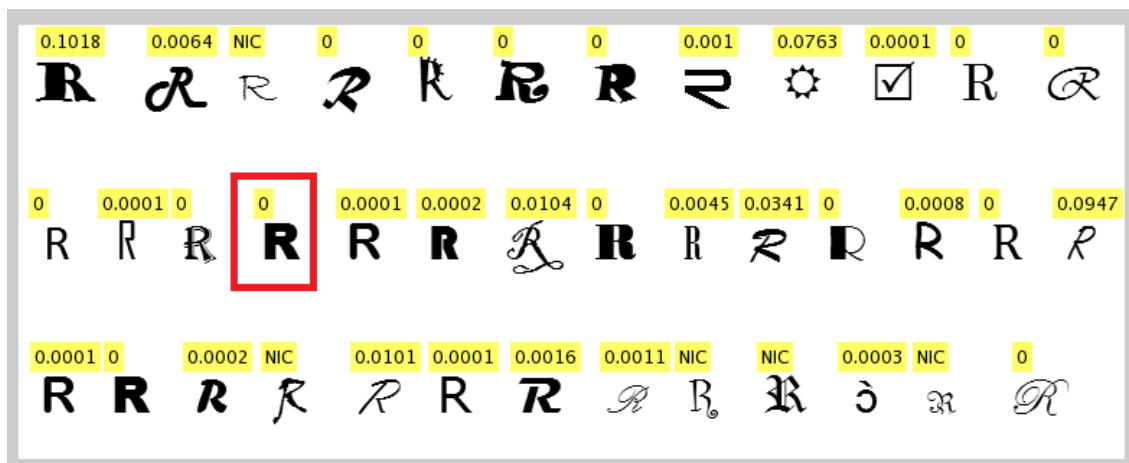
V obrázku Obr. 4.5 jsou v žlutých rámečcích uvedeny hodnoty minimalizované ceny podobnosti. Vysoká podobnost porovnávaných objektů je udána malou hodnotou naopak větší hodnota udává, že míra podobnosti je malá. V některých případech může hodnota v žlutém rámečku obsahovat hodnotu NIC , což znamená, že došlo k chybě výpočtu. Chyba výpočtu může nastat ve dvou případech. V prvním případě dojde k chybě výpočtu TPS, což způsobuje špatné rozložení vzorků. Tuto chybu lze spravit snížením počtu vzorků. Druhý chybový stav může nastat při výpočtu minimalizace u naprogramované metody Hungarian.



Obr. 4.5 Výsledky podobností pro metodu shape context pro první variaci testu

Na obrázku Obr. 4.5 byla aplikovaná metoda shape context. Vzor je označený červeným rámečkem. Z obrázku je patrné, že některé znaky udávají vysokou míru podobnosti. Nejblíže podobný znak hledanému je na levé straně s hodnotou 0,0001. Je vidět, že tvarově odpovídá a vypadá, že je zmenšený, na to shape context dokáže reagovat. Zajímavé je to, že malé hodnoty podobnosti byly nalezeny i pro tvary s různými deformacemi. Naopak je zde dost symbolů, které nevykazují velkou hodnotu a tím jsou méně podobné. Například první znak s hodnotou 0,5628. Nejmenší podobnost dosahoval znak slunce s hodnotou 1,0275. Je vidět, že se jedná o úplně jiný znak než hledaný. Naopak znak naproti s hodnotou 0,0396 vykazuje velmi velkou podobnost, v každém případě je důležité si uvědomit, že tento znak má určité části podobné. Dále je vidět, že metoda má problém se zkosením, jelikož jsou znaky, které vykazují vysokou hodnotu a tím nejsou podobné hledanému vzoru. V poslední řadě je zde jeden znak, který vykazuje chybovou hodnotu NIC. Z pohledu výpočetní rychlosti k výpočtu jednoho znaku je potřeba 2 sekundy.

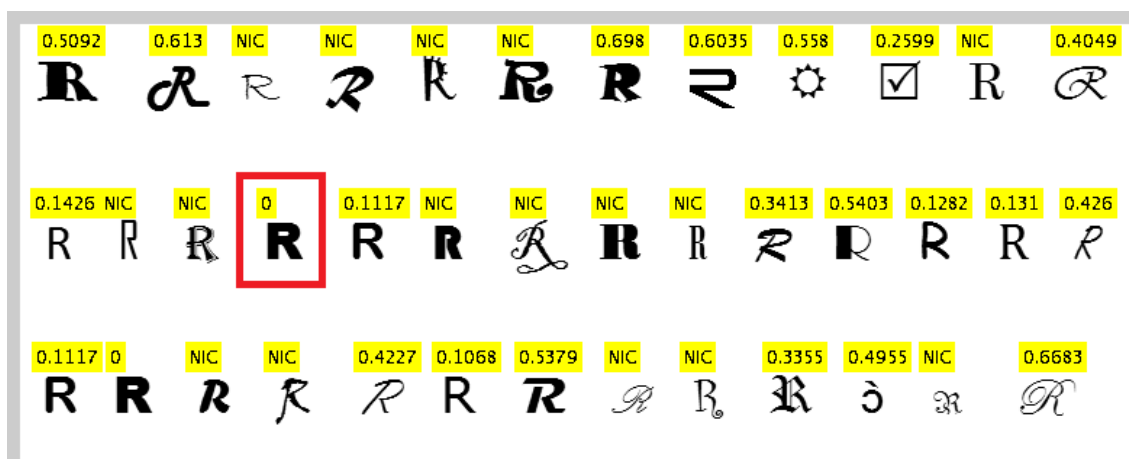
Pro možnosti otestování efektivity TPS transformace byl vytvořen druhý test, který má stejné parametry jako předchozí s tím rozdílem, že počet iterací TPS je nastavený na 30. Výsledky testu je možné pozorovat na obrázku Obr. 4.6



Obr. 4.6 Výsledky podobností pro metodu shape context pro druhou variaci testu

Na obrázku Obr. 4.6 lze sledovat, že úprava TPS způsobuje, že znaky jsou nyní více respektované a všechny znaky vykazují mnohem větší podobnost. Navíc je vidět, že znak, který v minulém testu obsahoval chybovou hodnotu, nyní vykazuje hodnotu 0. V každém případě je vidět, že kromě zvýšení citlivosti, došlo také ke zhoršení stability výpočetní metody Hungarian. Je tedy zřejmé, že využitím TPS dochází k zlepšení výsledku s rizikem, že může dojít k nestabilitě výpočetní metody.

Další testovanou metodou je inner distance. Vzor byl zvolen stejný a parametry byly zvoleny: počet vzorků 60, $L=0,125$, $na=4$ a $nb=8$. Přidavně byly prohledávány možnosti korespondence bodů využitých při výpočtu dynamického programování. Výsledky porovnání jsou ukázány na obrázku Obr. 4.7.

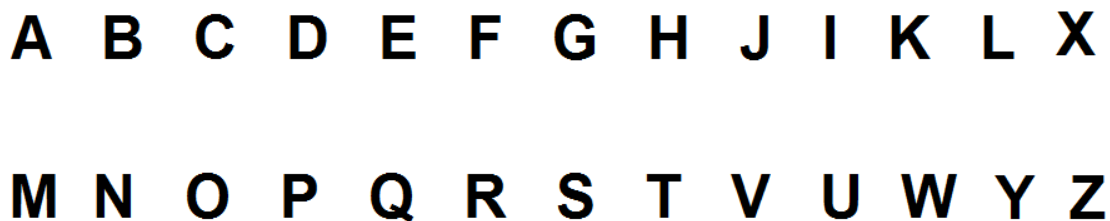


Obr. 4.7 Výsledky podobností pro metodu inner distance

Z testovacího snímku Obr. 4.7 je možné pozorovat, že došlo k několika detekcím znaku R. Je vidět, že všechny znaky, které mají hodnotu menší než 0,2 vykazují velmi podobné rysy jako hledaný znak. Dále je vidět, že spousta znaků nebyla detekována. Především jsou to znaky, které mají malou tloušťku. Chyba je způsobena omezením naprogramovaného algoritmu, který nebyl schopen vytvořit správně vnitřní vazby. Chyba je způsobena příliš malou tloušťkou testovaných objektů. Z pohledu časové

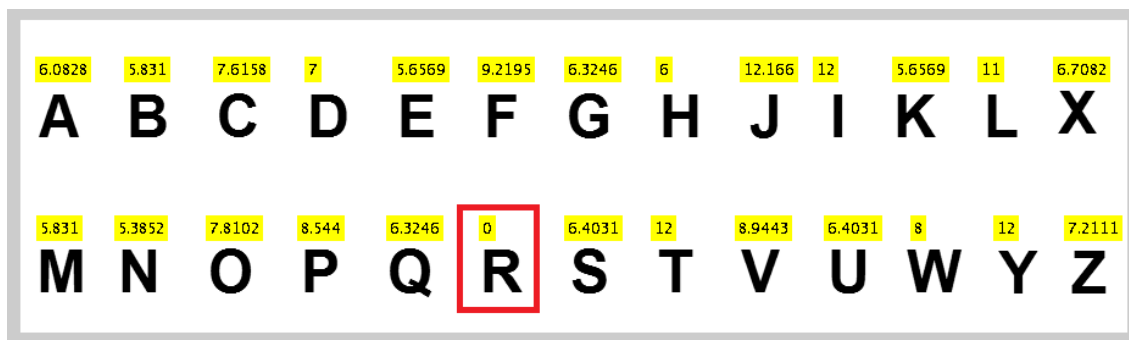
náročnosti výpočtu algoritmu byl jeden znak v průměru vypočítán za 12 sekund. Velký čas je způsoben projížděním všech možných kombinací matice C_{ij} .

Pro další porovnání byl vytvořen obrázek Obr. 4.8, který obsahuje abecedu se stejným fontem a velikostí. Velikost obrázku je 908x256 px.



Obr. 4.8 Obrázek abecedy pro testování rozpoznatelnosti znaků

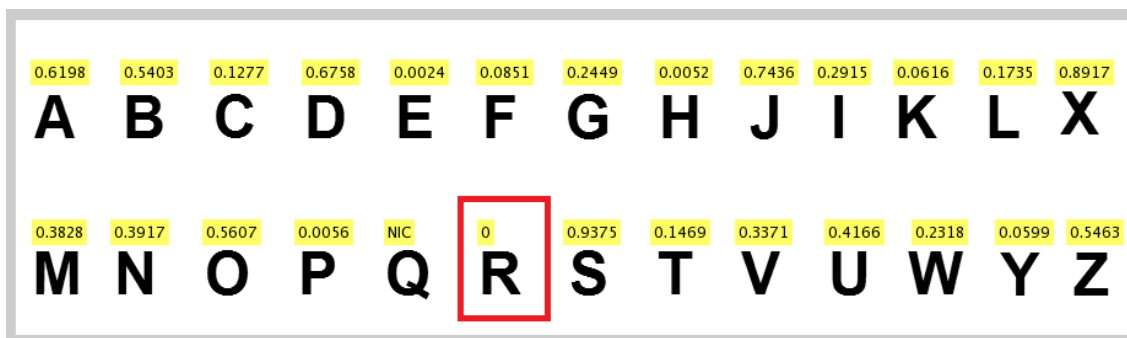
První testovaná metoda, na které je testován obrázek Obr. 4.8, je metoda využívající Hausdorffovu vzdálenost. Vyhodnocení pro všechny znaky jsou uvedeny na obrázku Obr. 4.9.



Obr. 4.9 Výsledky Hausdorffovy metody na obrázek abecedy

Na obrázku Obr. 4.9 jsou uvedeny výsledky pro metodu Hausdorff. Čísla uvedená v rámečcích udávají nejmenší nalezenou vzdálenost v okolí testovacího znaku. Z pohledu číselných hodnot platí, že čím menší, tím přesněji vypadá k hledanému tvaru. Z obrázku lze vypožorovat, že neblíže podobný tvar je symbol N s hodnotou vzdáleností 5,3852. Časová náročnost zpracování obrázku je 101 sekund.

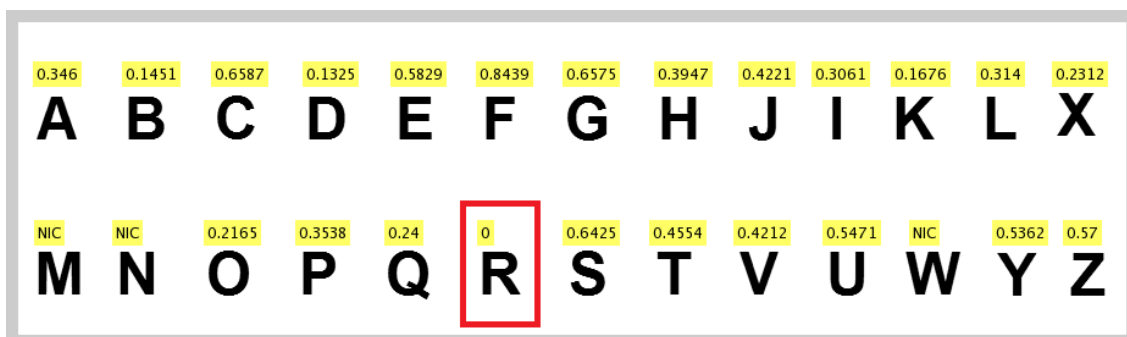
Druhá testovaná metoda, na které je testován obrázek Obr. 4.8, je shape context s parametry: počet vzorů 60, $L=0,125$, $na=4$, $nb=8$ a $TPS\ iterací=0$. Vyhodnocení pro všechny znaky jsou uvedeny na obrázku Obr. 4.10.



Obr. 4.10 Výsledky shape contextu na obrázek abecedy

Z obrázku Obr. 4.10 je možné pozorovat, že nejbližší podobné vzoru je písmeno P, které dosáhlo hodnoty 0,0056. Tato shoda je očekávaná, jelikož písmeno P je velmi podobné písmenu R. Podobnou logiku lze použít i pro písmeno F, případně i písmeno K a H. Všechna tato písmena obsahují části z testovacího vzoru. Naopak je zde několik chybových znaků, jako jsou písmena Y nebo E. Písmeno E dosáhlo nejlepší podobnosti. Naopak hůř dopadlo písmeno B, které by mělo mít menší hodnotu, jelikož obsahuje části z písmena R. Pouze jediné písmeno nebylo ohodnoceno podobností. Jedná se o písmeno Q, které obsahuje chybovou hodnotu NIC.

Třetí testovaná metoda, na které je testován obrázek Obr. 4.8, je inner distance s parametry: počet vzorů 60, $L=0,125$, $na=4$ a $nb=8$. Vyhodnocení pro všechny znaky jsou uvedeny na obrázku Obr. 4.11.

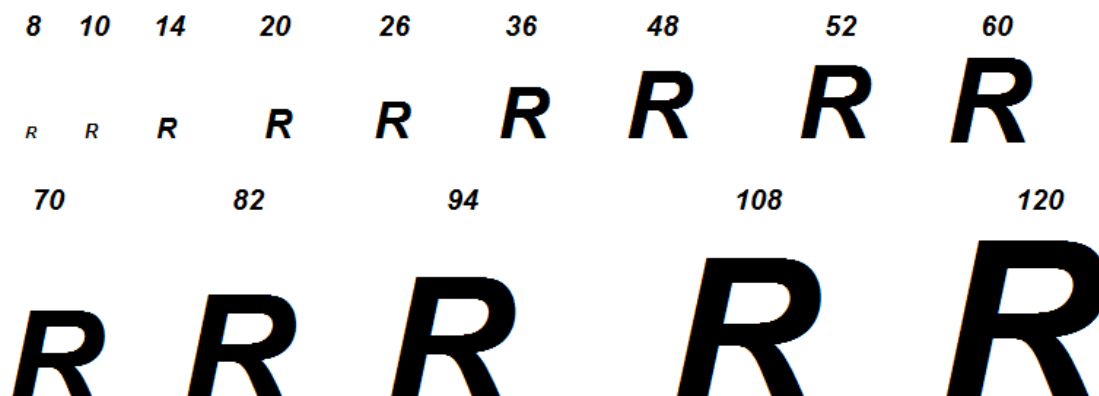


Obr. 4.11 Výsledky inner distance na obrázek abecedy

Druhý testovací obrázek Obr. 4.11 byl oproti předcházejícímu mnohem striktnější se svojí podobností vůči okolním znakům v abecedě. Znaky, které měly v předešlém testu malou hodnotu, jsou u metody inner distance s mnohem větší hodnotou. Naopak lépe se vyhodnotilo písmeno B, které získalo hodnotu 0,1451. Jako nejlépe v testu dopadlo písmeno D s hodnotou 0,1325. Tři písmena nebyla ohodnocena: W, M a N.

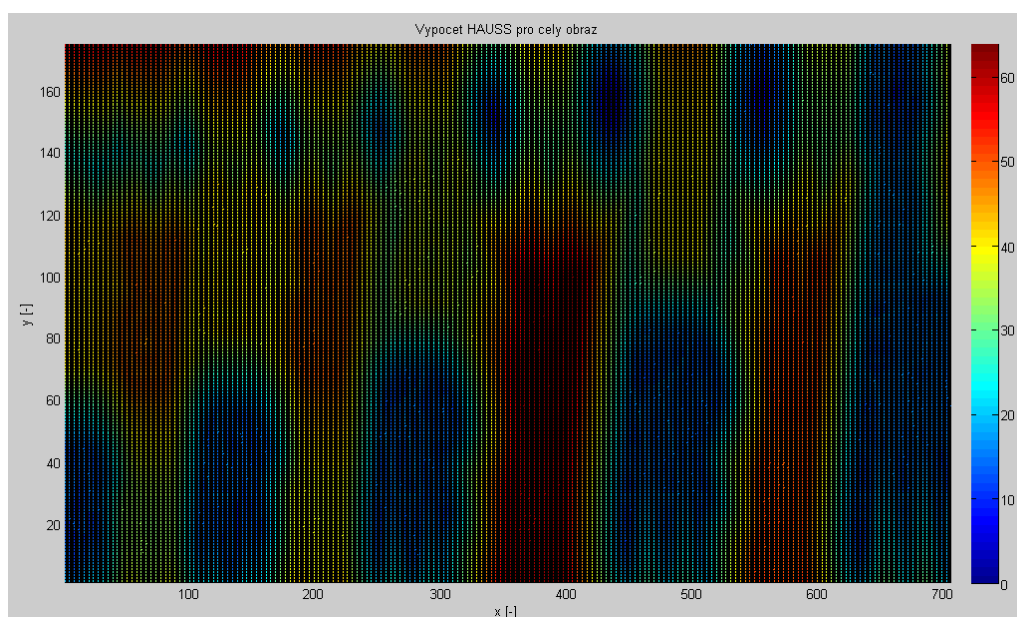
4.2 Testování změny velikosti rozměrů a úhlů

V této části testování budou metody podrobeny testu pomocí obrázku Obr. 4.12, který obsahuje stejný znak s různými velikostmi. Čísla uvedená nad znakem udávají velikost vybraného znaku. Znakem pro porovnání bylo zvoleno písmeno R s velikostí 36. Následně byl obrázek Obr. 4.12 podroben všem aplikovaným metodám. Velikost obrázku je 763x240 px.



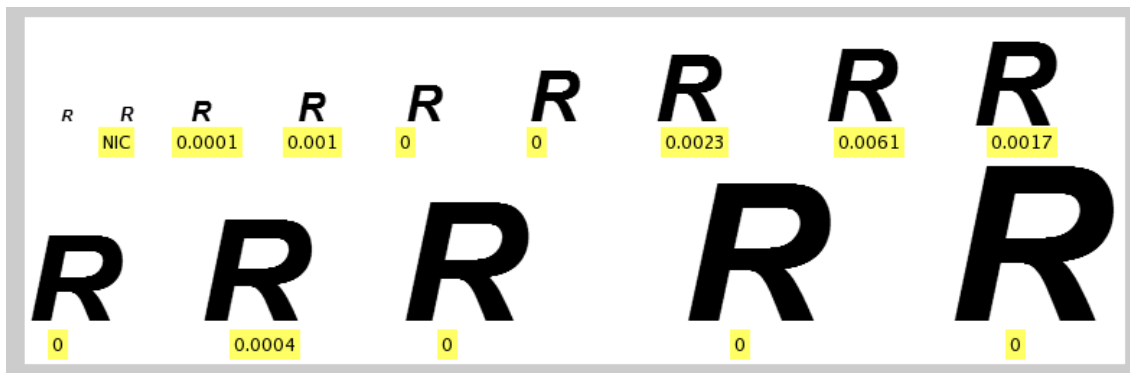
Obr. 4.12 Vzhled testovacích znaků pro testování reakce na změnu velikosti

Na obrázek Obr. 4.12 byla aplikovaná metoda Hausdorffova. Surf obsahující vzdálenosti je ukázán na obrázku Obr. 4.13. Z obrázku je patrné, že metoda nedokáže rozpoznat písmena, která jsou menší než vzor. Zlomová velikost je kolem 26. Naopak je metoda schopná lépe rozeznat zvětšené symboly s tím, že je možné pozorovat, že dochází ke slučování oblastí mezi velikostmi 120 a 60. Rychlost zpracování celého obrázku trvalo 86 sekund.



Obr. 4.13 Surf graf pro metodu Hausdorff testu 4.2

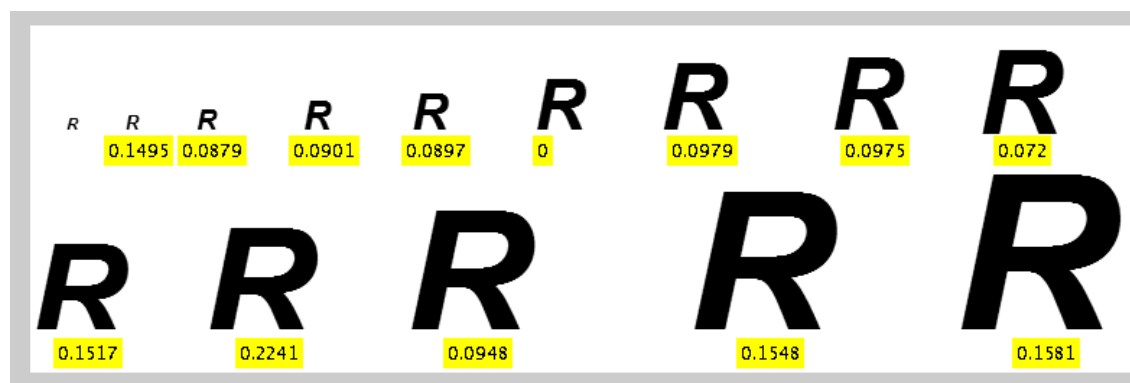
Pro metodu shape context byl obdobně použitý obrázek Obr. 4.12 se stejným vzorem písmene R s velikostí 36. Pro test byly použity tyto parametry: počet vzorů 60, $L=0,125$, $na=4$, $nb=8$ a $TPS=0$. Výsledky testovaných znaků jsou uvedeny na obrázku Obr. 4.14.



Obr. 4.14 Výsledky testů velikosti pro metodu shape context

Jako první je důležité upozornit, že naprogramovaná metoda řešící vzorkování není schopná vzorkovat příliš malé objekty. Vzorkovací funkce *Body* využívá operaci *bwtraceboundary*, která vykazuje chybu při testování vzoru R s velikostí 8. V každém případě je vidět, že ani velikost 10 není na tom o nic lépe, jelikož v jejím případě došlo k získání tvaru, ale nedošlo k vypočítání podobnosti přes funkci *Hungarian*. S pohledem na ostatní symboly je vidět, že metoda v základním nastavení si dokáže poradit se zvětšenými tvary. V případě, že jsou využity úpravy bodů využitím TPS, dochází ke zhoršení podobnosti, jelikož TPS způsobuje deformaci, která si nezachovává podobnost vzdáleností.

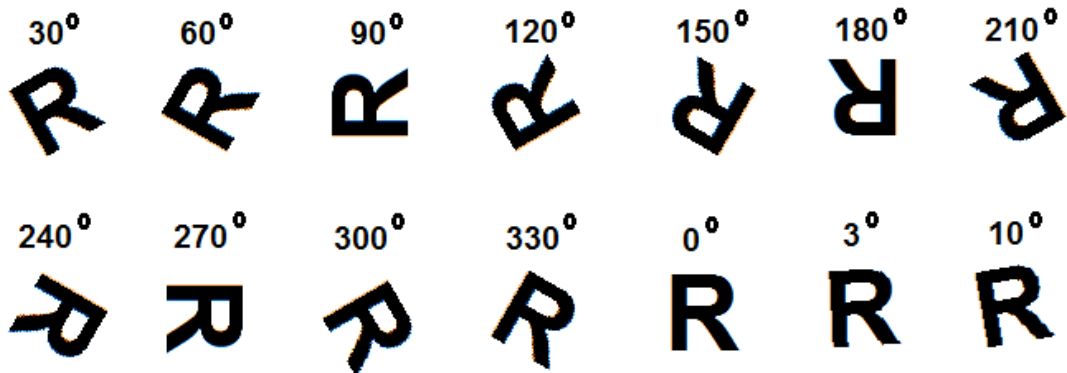
Další testovanou metodou je inner distance. Vzor byl zvolený stejný a parametry byly zvoleny: počet vzorů 60, $L=0,125$, $na=4$, $nb=8$. Výsledek je ukázán na obrázku Obr. 4.15.



Obr. 4.15 Výsledky testů velikosti pro metodu inner distance

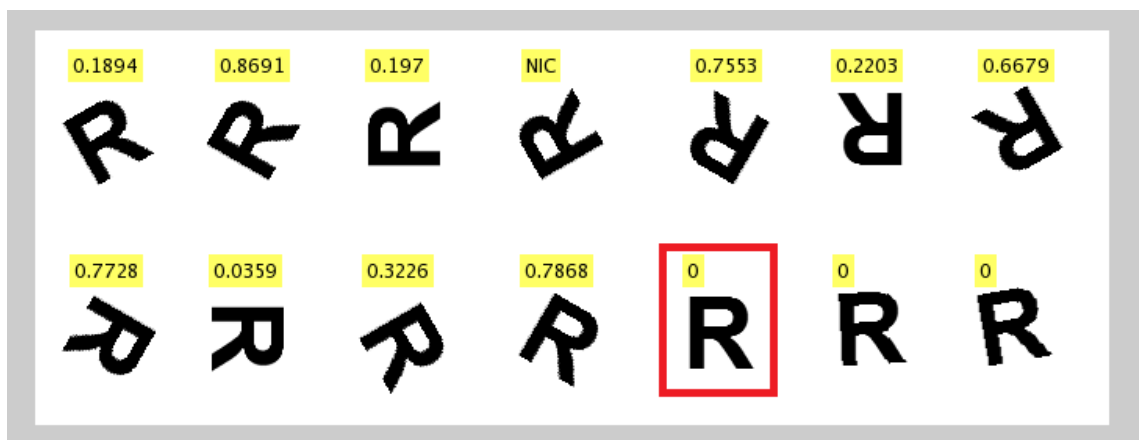
Z testu pro metodu inner distance lze rozhodnout, že metoda je schopná dobře rozeznat znaky v určité míře velikostí. Oproti shape context byla metoda schopna rozeznat druhý znak R s velikostí 10. Hůře dopadly znaky větší velikostí než 70.

Další testování bude aplikováno na schopnost rozlišovat rozdílnost natočení. Pro testování rozdílnosti byl vytvořen obrázek Obr. 4.16. Obrázek obsahuje písmeno R s různými směry natočení, které jsou uvedeny nad příslušným znakem. Rozměry testovacího obrázku je 687x252.



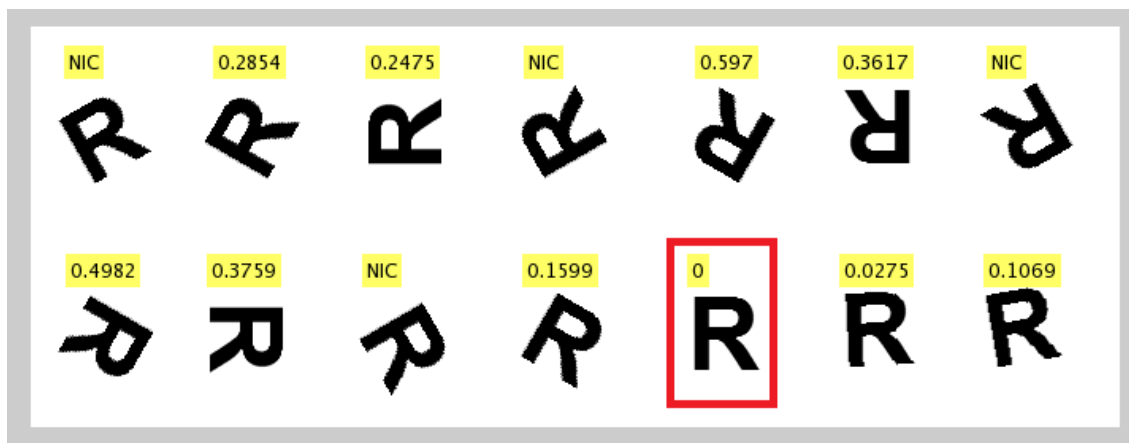
Obr. 4.16 Obrázek pro testování natočení

První testovaná metoda na obrázek pro rotace byla testována na metodu shape context. Výsledky je možné vidět na obrázku Obr. 4.17.



Obr. 4.17 Výsledek shape context na rotaci

Na obrázku Obr. 4.17 jsou výsledky pro metodu shape context. Je vidět, že metoda shape context je schopná přirozeně tolerovat malé odchylky v rotaci. Lze to pozorovat u posledních znaků, které obsahují nulovou hodnotu. Malých hodnot dále nabývají vzory, které jsou natočeny vždy po 90°, z nichž nejbližší podobnému byl znak při 270° s hodnotou 0,0359.



Obr. 4.18 Výsledek inner distance na rotace

Na obrázku Obr. 4.18 je možné pozorovat výsledky pro metodu inner distance. Metoda inner distance reaguje citlivěji na rotaci než shape context. Navíc jsou změny v rotacích mnohem lépe pozorovatelné. V každém případě stabilita výpočtu je menší než u shape context. Zpracování každého znaku trvalo přibližně 12 sekund.

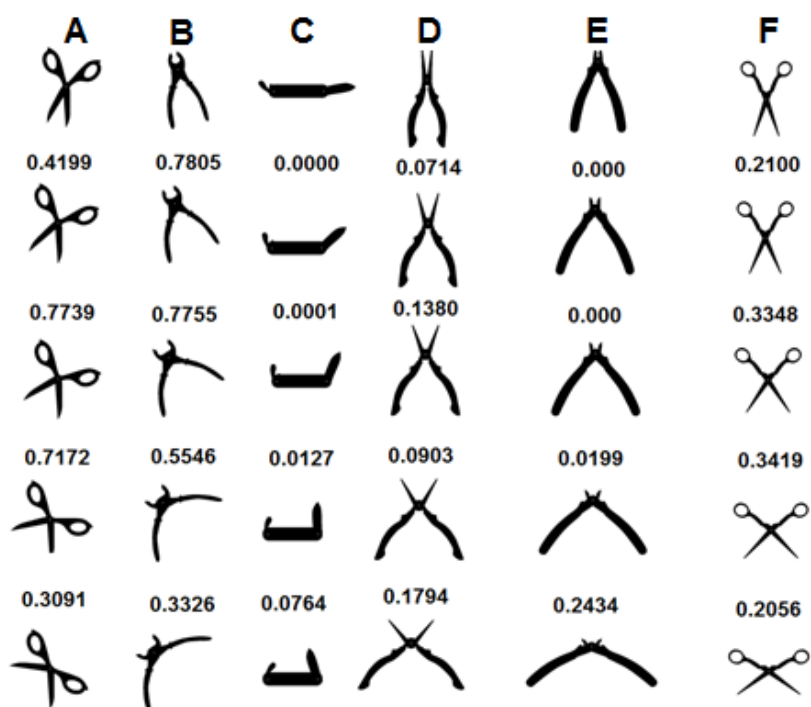
Při testování korelačních a rozdílových metod na test rotace se ukázalo, že metody byly schopné rozeznat znaky 3° a 10°. Zbývající natočení byla obtížně pozorovatelná. Jediný znak, který vykazoval zvýšené hodnoty, byl znak s rotací 180°, který alespoň z pohledu natočení odpovídá stejným rozměrům jako hledaný vzor.

Z pohledu testu na metodu Hausdorffa byla nalezena podobná závislost jenom s tím rozdílem, že byly nalezeny všechny rotace v násobcích 30°. Měly přibližně stejnou závislost. Rychlost výpočtu trvala 101 sekund. Dále byla otestována modifikace Hausdorffa, která umožňuje detekovat rotace. Při nastavení prohledávaného rozsahu od 0° po úhel 360° a při nastavených krocích 90° byla tato úprava schopna rozpoznat znaky v příslušných úhlech v násobcích 90°. Modifikace využívá zrychlovací úpravy při nastavení $w=10$ a $pra=40$, byla schopná prohledat obraz za 67 sekund, což je rychlejší než základní metoda.

4.3 Testování na databázi Tool2D

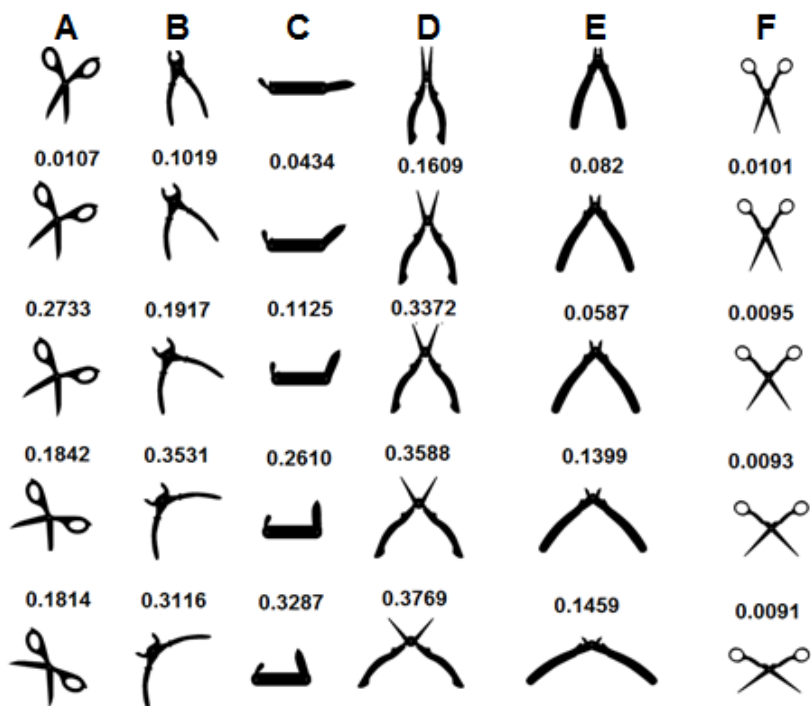
V dalším testu je využita veřejná databáze Tool2D, která vychází z těchto prací [27],[28] a je možné ji získat z těchto webových stránek [26]. Databáze Tool2D obsahuje 6 různých objektů a tyto objekty mají několik možných zobrazení. Například nůžky, které se postupně rozevírají. Každý testovací snímek má rozměry 635x476 px.

Cílem databáze je otestovat, jaký vliv má na míru podobnosti změna vnitřní struktury při ohýbání objektů. Tato vlastnost by teoreticky měla nejlépe rozeznávat inner distance. V první řadě je na tuto databázi aplikován shape context. Výsledky pro míry podobnosti mezi objekty je možné vyčíst na obrázku Obr. 4.19. Na obrázku Obr. 4.19 jsou v prvním řádku vzory, které se následně porovnávají s příslušným sloupcem. Výsledná hodnota porovnání je uvedena vždy nad porovnávaným vzorem.



Obr. 4.19 Výsledky testů databáze Tool2D pro shape context

Z testu na obrázku Obr. 4.19 je možné pozorovat, že vzor kapesní nůž C byl při všech vnitřních pohybech dobře rozpoznáný. Velmi podobných výsledků dokazovaly i vzory D a E. Naopak velmi špatně dopadly vzory A a B, kde vnitřní pohyby nebylo možné rozpoznat. Z pohledu rychlosti byl každý vzor zpracován za 3,33 sekund.

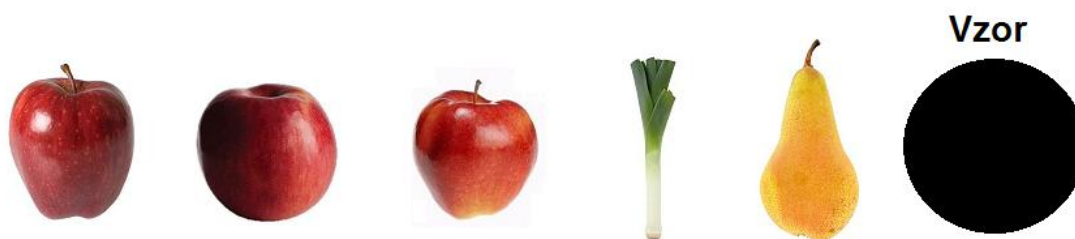


Obr. 4.20 Výsledky testů databáze Tool2D pro inner distance

Na obrázku Obr. 4.20 byla pro změnu aplikována metoda inner distance. Pokud se porovnájí výsledky s předcházejícím testem shape context, je možné vidět, že metoda inner distance si dokázala lépe poradit se vzory A, B a F , než metoda shape context. Naopak shape context lépe zvládal C, D a E . Rychlost výpočtu každého vzorku pro metodou Inner distance trval v průměru 16,83 sekund.

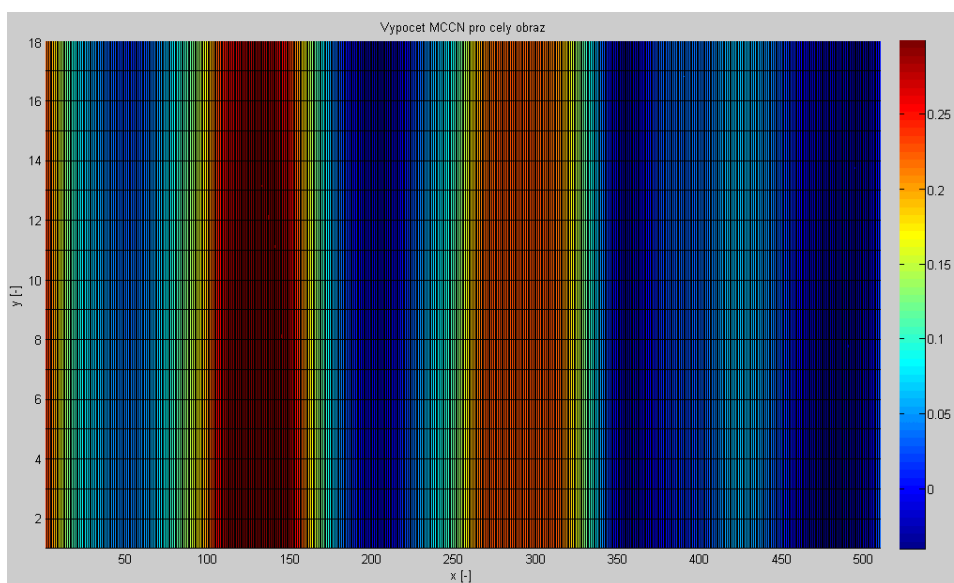
4.4 Testování na reálných snímcích

Pro testování na reálných snímcích byl vytvořen testovací obrázek Obr. 4.21. Testovací obrázek obsahuje několik variací ovoce a zeleniny. Snímky použitých předmětů byly získány z těchto webových stránek [30],[31] a [32]. Snímek obsahuje 3 variace jablek, dále jeden pórek a jednu hrušku. V obrázku je dodatečně zobrazen testovací vzor, který reprezentuje požadovaný tvar.



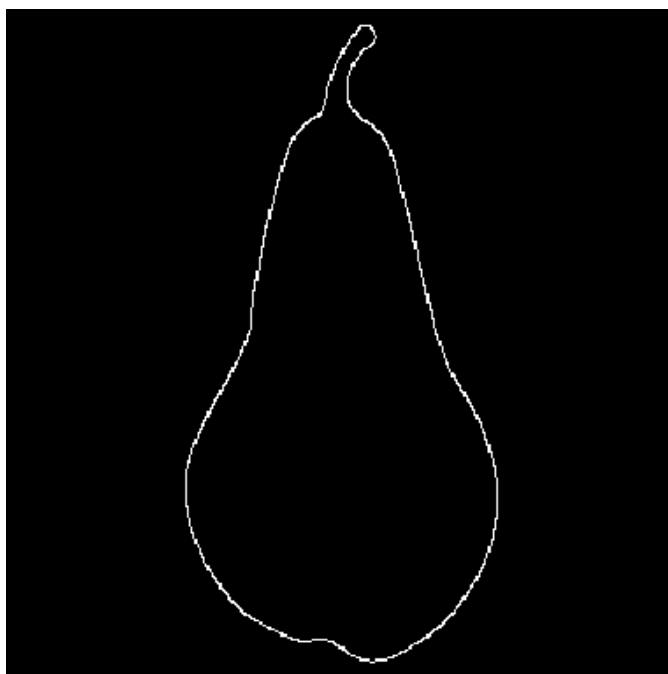
Obr. 4.21 Obrázek reálných snímků ovoce a zeleniny

Jako první jsou testovány metody korelační a rozdílové. Jelikož jsou si tyto metody podobné, byla testována pouze jedna z nich. Na obrázku Obr. 4.22. je možné pozorovat výsledek prohledávání v surf grafu. V surfu je vidět, že nejlepší podobnost dosáhlo druhé jablko. Ostatní jablka byly nalezeny také, ale jejich hodnota je menší. Ostatní obrázky nebyly touto metodou pozorovány.



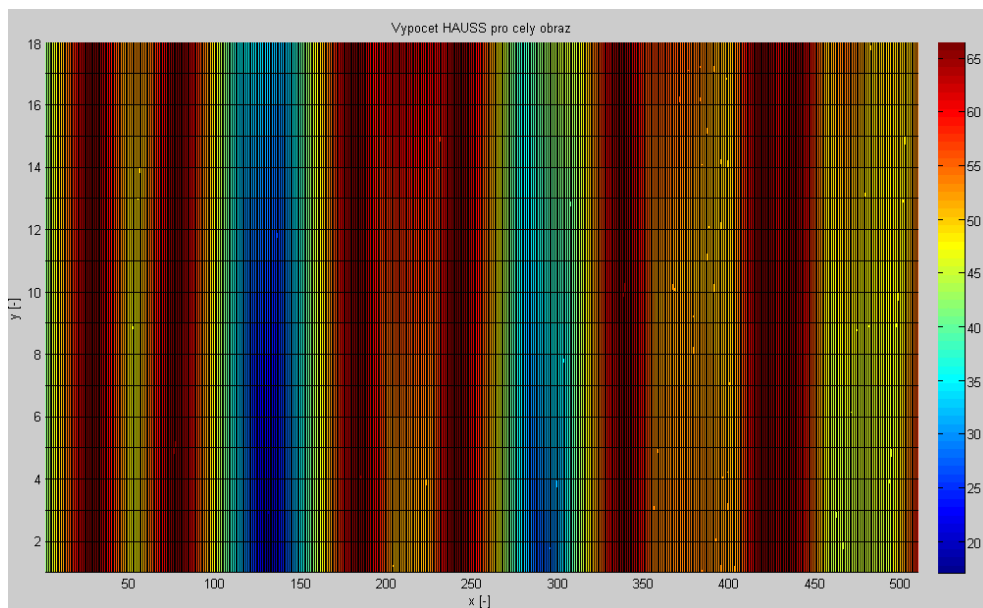
Obr. 4.22 Nalezené pozice pro metodu MCCN u obrázku s reálnými objekty

Pro další metody bylo nutné provést úpravy v algoritmu. Úpravou bylo to, že bylo nutné upravit vyhledávání hran, tak aby byly pokud možno, co nejméně ovlivněné hranami, které nerepresentují tvar objektu. Tyto nadbytečné hrany negativně ovlivňují detekci u naprogramovaných metod. Pro vyhlazení hran představující objekty byla nalezena inspirace z této práce [29]. V principu bylo aplikováno prahování obrazu na požadovanou hodnotu a z upravené verze byl vytvořen binární obraz. Dále se na tento binární obraz byl aplikován vylepšený Cannyho detektor hran, který je použitý v matlabovském příkazu *edge*. Vylepšení se týká nastavením parametru prahu a sigmy, které je možné upravovat Cannyho detektor. Všechny reálné snímky byly upraveny tímto detektorem. Výsledek této úpravy je uveden na obrázku Obr. 4.23. Funkce, které obsahují vylepšený detektor, byly označeny písmenem R např. *Body2_R*, která je použita u *inner distance*.



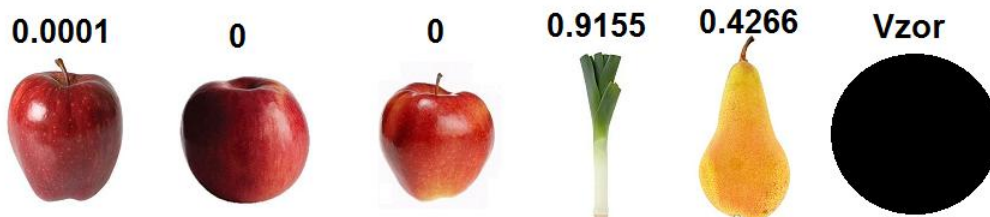
Obr. 4.23 Výsledný tvar hran po aplikování upraveného Cannyho detektoru

S touto úpravou byla testována další metoda Hausdorff, jejíž test je ukázán na obrázku Obr. 4.24. Z obrázku lze pozorovat, že metoda našla tři jablka. Test proběhl velmi obdobně jako v minulém testu, který využíval principu MCCN.



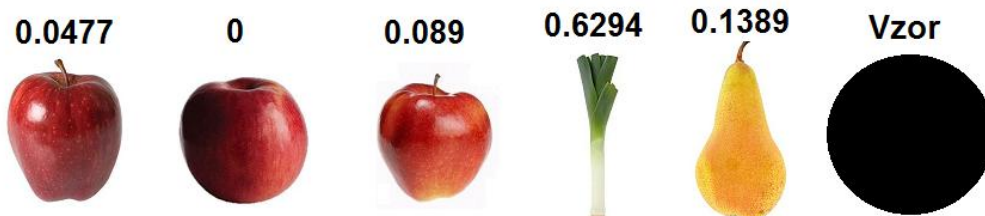
Obr. 4.24 Surf graf pro metodu Hausdorff u obrázku s reálnými objekty

Další aplikovanou metodou byla shape context. Výsledky je možné vidět na obrázku Obr. 4.25. V testu na obrázku Obr. 4.25 jsou uvedeny hodnoty podobnosti. Hledaný vzor dokázal vyhledat požadované tvary jablek s velmi velkou přesností. Jiné tvary byly vyhodnoceny jako jiné objekty.



Obr. 4.25 Vyhodnocení podobnosti u metody shape context pro reálný snímek

V posledním testu na obrázku Obr. 4.26 byl aplikován inner distance. S porovnáním se shape contextem byl schopný rozeznat požadované objekty. V každém případě byl lépe schopný rozpoznat rozdílnost objektů, která je způsobena deformacemi tvaru. Posledním rozdílem je větší podobnost hrušky než u shape context.



Obr. 4.26 Vyhodnocení podobnosti u metody inner distance pro reálný snímek

5 ZÁVĚR

Hlavním cílem diplomové práce bylo seznámení s problematikou detekce objektu v neznámém obraze za pomoci vzoru nebo šablony. V první části práce jsou popsány motivace a problémy, které mohou nastat při detekování šablon v obraze.

Stěžejní část práce je věnovaná teorii k metodám, které lze aplikovat pro detekci, ke každé metodě je popsán postup, jak metodu realizovat. Tato část také obsahuje teoretické poznatky k omezujícím vlastnostem, které jednotlivé metody nezvládají. Dále je v této části popsáno, pro jaké aplikace jsou uvedené metody použitelné. Z této teoretické části byly vybrány některé metody a v další části práce byly tyto metody prakticky zrealizované za pomoci rozhraní MATLAB Ra2014.

V druhé části práce byly provedeny algoritmy pro tyto metody: rozdílové, korelační, Hausdorff, shape context a inner distance. Výpisy principů algoritmů jsou popsány v příslušné kapitole 3. V poslední části práce byla vytvořena testovací databáze, která ověřovala funkčnost vybraných metod. Testovací databáze obsahovala objekty, které měly rozdílnou velikost, úhel natočení, deformace anebo obsahovaly znaky, které nepatří do stejné kategorie jako požadovaný vzor. Kromě toho byla také vytvořena databáze s reálnými objekty. Na tuto databázi byly otestovány naprogramované algoritmy a došlo k ověřování teoretických poznatků.

První prakticky realizované metody jsou rozdílové a korelační. Tyto metody odpovídaly teoretickým poznatkům. Z praktického hlediska jsou tyto metody použitelné pouze pro velmi přesné objekty s malými odchylkami v podobnosti. Metody měly jednu velkou výhodu, že se daly se použít k prohledávání obrazů pouze s převodem na šedotónový tvar. Z pohledu časové náročnosti je rychlost ovlivněna převážně velikostí prohledávaného obrazu, velikostí vzoru a typem verze metody.

Další řešenou metodou byla metoda využívající Hausdorffovu vzdálenost. Metoda využívá stejného principu vyhledávání jako rozdílové a korelační, a proto se dá použít k prohledávání celých obrazů. Velkou výhodou je možnost realizovat zrychlovací úpravy, které dokázaly zrychlit detekci v obraze bez rizika, že by nebyla nalezena nejlepší shoda podobnosti. Další velkou výhodou, která byla prokázána, je ta, že metoda je schopná detekovat natočení hledaného objektu. Nevýhodou této úpravy je zvýšení výpočetní náročnosti. Navíc pokud není nastavený vhodný úhel mezikroku, nemusí být požadovaný tvar nalezen. Z pohledu kvality vyhledávání je metoda schopná lépe rozeznávat rozdíly ve tvarech než metody rozdílové a korelační.

Další realizovanou metodou je shape kontext. Bylo ověřeno, že je tato metoda vhodná spíše k porovnávání dvou objektů než k celoplošnému vyhledávání obrazu. Na druhou stranu je metoda shape context schopná lépe rozeznat odchylky od ideálního tvaru. Bylo prakticky ověřeno, že je metoda imunní vůči translaci objektu. Dále je možné upravit metodu tak, že je schopná rozlišovat změny v měřítku. Nepovedlo se zprovoznit verzi, která je schopná řešit natočení objektu. V každém případě je metoda schopná rozlišit drobné změny v natočení. K metodě byla vytvořena TPS transformace,

kteřá vylepřovala rozpoznávací schopnosti. S ohledem na stabilitu realizovaného kódu může algoritmus zkolabovat při výpočtu TPS transformace anebo výpočtem minimalizace ceny podobnosti. Z pohledu časové náročnosti je rychlost ovlivněna počtem vzorů v obraze, velikostí rozdělovacího prostoru, počtem TPS iterací a výpočtem minimalizace ceny prováděno za pomoci metody Hungarian. I přesto všechno byla tato metoda nejrychlejší.

Poslední prakticky realizovanou metodou je inner distance, která má podobné praktické využití jako shape context. Její hlavní výhodou je schopnost rozlišovat díry, navíc nevyžaduje TPS transformace nebo jiné dodatečné modely. Metoda byla schopná lépe rozlišovat rozdíly mezi stejnými prvky než metoda shape context. Metoda je také odolná proti translaci a je schopná detekovat objekty s jiným měřítkem než hledaný vzor. Nebylo dosaženo kompletní imunity na změny měřítka. Realizovaný algoritmus Inner distance je vhodný na mohutnější objekty, jelikož pokud je porovnáván objekt tenký, algoritmus nemusí být schopný nalézt nejkratší cesty. Dalším limitním faktorem je nacházení inner-anglu, který nemusí být pro daný bod stabilní. Tato vlastnost je ovlivněna strukturou hrany objektu. Z pohledu výpočetní rychlosti realizovaného algoritmu je metoda inner distance pomalejší než shape context. Zvýšená rychlost je ovlivněna vyhledáváním nejkratších cest, hledáním inner-anglu a výpočtem minimalizace ceny využitím dynamického programování.

Další zmíněné metody jsou uvedeny v teoretické části a nebyly prakticky realizovány. První teoreticky zmíněná metoda je active apperance model. Metoda active apperance model využívá databázi snímků hledaných vzorů, na kterých jsou vytvořeny hraniční body. Deformace hraničních bodů potom slouží k nalezení shody vzoru v hledaném obraze. Nevýhodou této metody je riziko, že objekt nebude nalezen a to je způsobeno špatným určením počáteční pozice pro body. Použitelnost této metody je hlavně v detekci lidského obličeje nebo končetin.

Druhá teoreticky zmíněná metoda má název boundary fragment model. Hlavním cílem této metody je sestavení testovací databáze požadované třídy vzoru. Pomocí vytvořené databáze jsou následně naučeny slabší a silnější klasifikátory, které potom slouží k detekování vzoru v obraze. Metoda je využitelná pro rozpoznávání objektu do stejné třídy.

Poslední zmíněnou metodou je hierarchical matching. Metoda hierarchical matching vytváří shape-tree. Jedná se o postupné rozkládání křivky na dvě části za pomoci rozdělovacího bodu. Vytvořený strom se dá následně použít pro rekonstrukci nebo k nalezení podobnosti objektů v neznámém obraze. Velkou výhodou této metody je možnost detekovat části křivek.

CITOVANÁ LITERATURA

- [1] HORÁK, Karel. Dynamické obrazy [online]. In: . Brno, s. 41 [cit. 2017-11-17]. Dostupné z: http://midas.uamt.feec.vutbr.cz/ROZ/roz_cz.php
- [2] DANĚK, Ondřej. *Registrace obrazů buněk podle intenzit v obraze* [online]. Brno, 2006 [cit. 2017-11-19]. Dostupné z: https://is.muni.cz/th/60931/fi_m/xdanek2-dp.pdf. Diplomová práce. Masarykova univerzita. Vedoucí práce RNDr. Petr Matula, Ph.D.
- [3] BOUTS, Mark. *Comparing Images Using the Hausdorff Distance* [online]. , 4 [cit. 2017-11-21]. Dostupné z: <http://www.cs.uu.nl/docs/vakken/gmspr/Comparing%20Images%20Using%20the%20Hausdorff%20Distance2.doc>
- [4] HUTTENLOCHER, Daniel, Gregory KLANDERMAN a William RUCKLIDGE. Comparing Images Using the Hausdorff Distance. *IEEE TRANSACTIONS ON PATTERN ANALYSIS AND MACHINE INTELLIGENCE* [online]. 1992, **15**(9), 14 [cit. 2017-11-24]. Dostupné z: <https://web.stanford.edu/class/cs273/refs/hausdorff.pdf>
- [5] HORÁK, Karel. *Defektoskopie a klasifikace* [online]. In: . Brno, s. 30 [cit. 2017-11-24]. Dostupné z: http://midas.uamt.feec.vutbr.cz/ROZ/roz_cz.php
- [6] BELONGIE, Serge, Jitendra MALIK a Jan PUZICHA. *Matching with Shape Contexts* [online]. [cit. 2017-12-08]. Dostupné z: https://www2.eecs.berkeley.edu/Research/Projects/CS/vision/shape/sc_digits.html
- [7] BELONGIE, Serge, Jitendra MALIK a Jan PUZICHA. Shape Matching and Object Recognition Using Shape Contexts. *IEEE TRANSACTIONS ON PATTERN ANALYSIS AND MACHINE INTELLIGENCE* [online]. 2002, **24**(24), 14 [cit. 2017-12-08]. Dostupné z: <https://people.eecs.berkeley.edu/~malik/papers/BMP-shape.pdf>
- [8] LING, Haibin a David JACOBS. *Using the Inner-Distance for Classification of Articulated Shapes* [online]. [2005], , 8 [cit. 2017-12-08]. ISSN 1063-6919. Dostupné z:

<https://pdfs.semanticscholar.org/7f74/479974089970108c0ba3b2b5111115ab05fe.pdf>

- [9] LING, Haibin a David JACOBS. *Shape Classification Using the Inner-Distance: IEEE TRANSACTION ON PATTERN ANALYSIS AND MACHINE INTELLIGENCE* [online]. 2007, **29**(2), 44 [cit. 2017-12-08]. Dostupné z: https://www.google.cz/url?sa=t&rct=j&q=&esrc=s&source=web&cd=4&ved=0ahUKEwiyNHvjfvXAhWPZ1AKHfqBcn4QFghBMAM&url=http%3A%2F%2Fvc.cs.nthu.edu.tw%2Fhome%2Fpaper%2Fcodfiles%2Fcjhung%2F200801081550%2FShape_Classification_Using_the_Inner-Distance.ppt&usg=AOvVaw2hz2bDFZX61fS1fAadDYU-
- [10] KLÍMA, Ondřej. *FORENZNÍ IDENTIFIKACE DĚTSKÝCH OBLIČEJŮ* [online]. Brno, 2011 [cit. 2017-12-11]. Dostupné z: https://www.vutbr.cz/www_base/zav_prace_soubor_verejne.php?file_id=117559. BAKALÁŘSKÁ PRÁCE. VYSOKÉ UČENÍ TECHINCKÉ V BRNĚ. Vedoucí práce Ing. Vitězslav Beran, Ph.D.
- [11] COOTES, T.F, G.J. EDWARDS a C.J TAYLOR. *Active Apperance Models* [online]. Manchester: Wolfson Image Analysis Unit, [2001], , 16 [cit. 2017-12-21]. ISSN 0162-8828. Dostupné z: <https://www.cs.cmu.edu/~efros/courses/LBMV07/Papers/cootes-eccv-98.pdf>
- [12] QI, Xiaojun. *Active Shape Model and Active Appearance Model* [online]. , 82 [cit. 2017-12-21]. Dostupné z: <http://digital.cs.usu.edu/~xqi/Teaching/REU05/Notes/ASM.pdf>
- [13] JANÁKOVÁ, Ilona. *Detekce geometrických primitiv* [online]. Brno: Computer Vision Group, s. 49 [cit. 2017-12-21]. Dostupné z: http://midas.uamt.feec.vutbr.cz/POV/pov_cz.php
- [14] OPELT, Andreas, Axel PINZ a Andrew ZISSERMAN. *A Boundary-Fragment-Model for Object Detection* [online]. Graz, , 14 [cit. 2017-12-28]. Dostupné z: <https://www.robots.ox.ac.uk/~vgg/publications/2006/Opelt06/opelt06.pdf>
- [15] FRISHBERG, Amit. *Boundary Fragment Model for Object Detection* [online]. , 38 [cit. 2017-12-28]. Dostupné z: http://moodle.cs.huji.ac.il/cs10/file.php/67777/Amit_Frishberg_Boundary-Fragment-Model_for_Object.pdf

- [16] FENG, Chen. *2D Shape Matching: Inner Distance + Shape Context* [online]. Michigan: Dept. of Civil & Environmental Engineering, 2010, s. 43 [cit. 2017-12-28]. Dostupné z:
http://web.eecs.umich.edu/~silvio/teaching/EECS598_2010/slides/10_14_Chen.pdf
- [17] FELZENSZWALB, Pedro a Joshua SCHWARTZ. *Hierarchical Matching of Deformable Shapes* [online]. Chicago, , 8 [cit. 2018-01-01]. ISSN 1063-6919. Dostupné z: <https://cs.brown.edu/~pff/papers/hier.pdf>
- [18] HORÁK, Karel. *Porozumění obsahu obrazu* [online]. Brno: Computer Vision Group, 2017, s. 54 [cit. 2018-01-01]. Dostupné z:
http://midas.uamt.feec.vutbr.cz/ROZ/Lectures/07_Porozumeni_obsahu_obrazu.pdf
- [19] PATEL, Mayank. *Notes on 'Hungarian Algorithm for assignment problem'* [online]., 2016, , 1 [cit. 2018-04-08]. Dostupné z:
http://home.iitk.ac.in/~mpatel/cs300/5B/cs300_a4.pdf
- [20] STEGMANN, Mikkel B. a David Delgado GOMEZ. *A Brief Introduction to Statistical Shape Analysis* [online]. Lyngby, 2002, , 15 [cit. 2018-04-16]. Dostupné z: https://graphics.stanford.edu/courses/cs164-09-spring/Handouts/paper_shape_spaces_imm403.pdf
- [21] BOURNE, Murray. Gradient (or slope) of a Line, and Inclination. *Interactive Mathematics* [online]. 2018 [cit. 2018-04-22]. Dostupné z:
<https://www.intmath.com/plane-analytic-geometry/1b-gradient-slope-line.php>
- [22] LING, Haibin a David W. JACOBS. *Shape Matching for Foliage Database Retrieval* [online]. Philadelphia, [2009], , 21 [cit. 2018-05-07]. Dostupné z:
<https://pdfs.semanticscholar.org/cd89/511fa2abf790ab186bcedf03c0732c97acf0.pdf>
- [23] PETERSEN, Newton. *Shape Contexts* [online]. In: . 2008, s. 26 [cit. 2018-05-08]. Dostupné z:
<http://www.cs.utexas.edu/~grauman/courses/spring2008/slides/ShapeContexts425.pdf>

- [24] HONZÍK, Petr. *Základy statistiky používané ve strojovém učení* [online]. In: . Brno: FEKT VUT, 2014 [cit. 2018-05-08]. Dostupné z: http://midas.uamt.feec.vutbr.cz/STU/Lectures/02_Zaklady_statistiky_pouzivane_ve_strojovem_uceni.pdf
- [25] Search Statistics How To: Statistics for the rest of us!. *Search Statistics How To* [online]. Statistics How To, 2018 [cit. 2018-05-08]. Dostupné z: <http://www.statisticshowto.com/probability-and-statistics/chi-square/>
- [26] BRONSTEIN, Alexander, Michael BRONSTEIN a Ron KIMMEL. Datasets. *Datasets* [online]. c2007-2010 [cit. 2018-05-11]. Dostupné z: http://tosca.cs.technion.ac.il/book/resources_data.html
- [27] BRONSTEIN, Alexander, Michael BRONSTEIN a Ron KIMMEL. *Numerical Geometry of Non-Rigid Shapes*. Springer, 2008. ISBN 978-0-387-73301-2.
- [28] BRONSTEIN, Alexander M., Michael M. BRONSTEIN, Alfred M. BRUCKSTEIN a Ron KIMMEL. Analysis of two-dimensional non-rigid shapes. *International Journal of Computer Vision* [online]. 2008, **78**(1), 39 [cit. 2018-05-11]. Dostupné z: <http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.77.1281&rep=rep1&type=pdf>
- [29] Boundary Extraction in MATLAB. *IMAGE PROCESSING* [online]. 2012 [cit. 2018-05-13]. Dostupné z: <https://angeljohnsy.blogspot.com/2012/01/boundary-extraction-in-matlab.html>
- [30] Atlas jablek: 16 oblíbených odrůd. *Vitalia* [online]. 2016 [cit. 2018-05-13]. Dostupné z: <https://www.vitalia.cz/clanky/atlas-jablek-16-oblíbených-odrůd/#p-3>
- [31] Hruška abate. *TESCO: Jablka a hrušky* [online]. Praha: Tesco, c2018 [cit. 2018-05-13]. Dostupné z: <https://nakup.itesco.cz/groceries/cs-CZ/products/2001120059629>
- [32] Pórek. *AkcniCeny.cz* [online]. Praha: MAFRA, 2018 [cit. 2018-05-13]. Dostupné z: <https://www.akcniceny.cz/akce/porek-ks/>
- [33] JEŽEK, František. *KMA/GPM - TPS - Thin-Plate Splines* [online]. In: . Plzeň, 2010, s. 38 [cit. 2018-05-14]. Dostupné z: <http://geometrie.kma.zcu.cz/index.php/www/content/download/1560/4400/file/P>

2-GPM-TPS_Thin-

Plate_Splines.pdf?PHPSESSID=00fbd24337e73b208f0ebf38aa17fb4b.

Seznam příloh – obsah CD

- **Diplomová práce v pdf** – Tato dokumentace
- **Adresář Programy** – Obsahují realizované programy
 - **Inner** – obsahuje algoritmus inner distance
 - **Roz_Kor_Hausd** – obsahuje algoritmy
 - **Shape** – obsahuje algoritmus shape kontext
- **Adresář Galerie** – Obsahují seznam použité databáze
 - **Vzory** – obsahuje testovací vzorky
 - **Tools** – obsahuje databázi Tools2D
 - **Databaze** – obsahuje reálné snímky
- **Adresář zdroje** – Obsahují zdroje, které byly použity