

ŠVOČ 2014
Ústí nad Labem

Rozpoznanie písaného textu matematickými metódami

Meno a Priezvisko študenta :	Monika Miklašová
Škola:	Slovenská Technická Univerzita
Fakulta:	Stavebná fakulta
Ročník a program/odbor štúdia:	2. roč., 2. stupeň, matematicko-počítačové modelovanie
Vedúci práce:	Ing. Michal Smíšek
Konzultant:	prof. RNDr. Karol Mikula, DrSc.
Katedra:	Katedra matematiky a deskriptívnej geometrie

Annotation

This work deals with mathematical methods of image processing in order to analyze a handwritten text. Conventionally used methods use blurring of 2D image of a letter and then compare it to an atlas of other, already blurred letters. Unlike conventional methods, we aim to represent a letter as a system of curves, to obtain 1D descriptors of these curves, and also create a concept of an atlas for classification of these. The reduction of dimensionality of this task results in possibility to formulate simpler and computationally more efficient methods to compute distance between two letters, yielding a metric.

A feasible way to represent a letter by a system of curves is considering outer boundary of this letter. The system of curves thus reduces into a simple closed curve. A feasible way to obtain a descriptor of this curve is to use signed curvature of the curve. In order to have a notion of blurring of the descriptor, to overcome imperfections in the written letters, we use curvature diffusion (often called also Lagrangian diffusion).

Since we are talking about 1D, there are multiple ways to formulate a metric to compute distances between descriptors of various letters. In this work, we consider computing absolute value of two descriptors subtracted as a first metric, computing distance function from the graph of one descriptor and evaluating it in points of the second descriptor as a second metric and computing distance function from the graph of one descriptor and evaluating it in points of graph of the second descriptor as a third metric. This way we demonstrate, it is easy to formulate an arbitrary metric, once we have reduced the problem to 1D.

Anotácia

Práca sa zaoberá matematickými metódami spracovania obrazu za účelom analýzy ručne písaného textu. Tradične používané metódy sú založené na zhladení písmen v 2D obrázku a ich následnom porovnaní s atlasom iných zhladených písmen. Na rozdiel od tradičných metód je našim cieľom reprezentovať písmeno pomocou systému kriviek na získanie 1D deskriptora týchto kriviek a taktiež vytvoriť návrh atlasu na ich klasifikáciu. Redukcia dimenzie nám umožňuje formulovať jednoduchšie a výpočtovo efektívnejšie metódy na výpočet vzdialenosti medzi dvomi krivkami.

Spôsob ako reprezentovať písmeno pomocou sústavy kriviek je nájdenie vonkajšej hranice daného písmena. Systém kriviek sa tým zjednodušuje na jednoduchú uzavretú krivku. Na získanie deskriptora krivky používame znamienkovú krivostnú funkciu. Na odstránenie nedokonalostí používame krivostnú difúziu (často nazývaná aj Lagrangeovska difúzia).

V 1D existuje niekoľko spôsobov, ako sformulovať metriku pre výpočet vzdialenosti medzi deskriptormi jednotlivých písmen. V práci sa zaoberáme výpočtom absolútnej hodnoty rozdielu dvoch deskriptorov (prvá metrika), výpočtom vzdialenostnej funkcie z grafu jedného deskriptora a následným vyhodnotením týchto hodnôt v bodoch grafu druhého deskriptora (druhá metrika) a výpočtom vzdialenostnej funkcie z grafu jedného deskriptora a

vyhodnotením týchto hodnôt v grafe druhého deskriptora (tretia metrika).
Týmto spôsobom demonštrujeme, že akonáhle zredukujeme problém na 1D,
vieme sformulovať rôzne metriky.

Obsah

Úvod	1
1. Úvod do problematiky	2
2. Definícia písmena pomocou deskriptora	3
2.1 Thresholding	3
2.2 Dilatácia	3
2.3 Získanie krivky z hranice písmena	4
2.4 Krivosť	5
3. Lagrangeovská difúzia	7
3.1 Numerická interpretácia lagrangeovskej difúzie	8
3.2 Thomasov algoritmus	9
3.3 Sherman-Morrisonová formula	9
3.3.1 Postup v programe	10
4. Invariantnosť voči počiatočnému bodu	11
5. Metriky	13
5.1 Rozdielová metrika	13
5.2 Bodovo-vzdialenostná metrika	14
5.2.1 Vzďialenostná funkcia	14
5.2.2 Funkcionalita Bodovo-vzdialenostnej metriky	16
5.3 Grafovo-vzdialenostná metrika	16
6. Popis softvéru	18
6.1 Opis grafického rozhrania	19
7. Experimenty	21
7.1 Porovnanie vplyvu lagrangeovských parametrov na deskriptor	21
7.1.1 Porovnanie deskriptora pri zmene veľkosti časového kroku τ	21
7.1.2 Porovnanie deskriptora pri zmene počtu časových krokov	22
7.1.3 Porovnanie deskriptora pri zmene veľkosti parametra ϵ	23
7.2 Porovnanie deskriptorov modifikovaného písmena	23
7.2.1 Porovnanie deskriptora pri zmene veľkosti písmena	23
7.2.2 Porovnanie deskriptora pri zmene rotácie písmena	24
7.2.3 Porovnanie deskriptorov rôznych podôb písmena a	26
7.3 Nevýhody krivostného deskriptora	27

8. Záver	28
Literatúra	29

Úvod

V dnešnej dobe sú aplikácie rozpoznávajúce text často používané. Všeobecný názov takéhoto typu aplikácii je OCR (optical character recognition). Používa sa na zjednodušenie a urýchlenie činností ako napríklad transformácia ručne písaného dokumentu do počítačom rozoznatelného textu. V automobilovom priemysle je použitý v dotykových navigáciách a v senzoroch rozpoznávajúcich cestné značky, čím prispieva k zvýšeniu bezpečnosti na cestách.

Cieľom tejto práce bolo rozpoznanie ručne písaného textu pomocou metód obrazového spracovania, pričom sme sa chceli vyhnúť tradičným metódam využívajúce napríklad neurálne siete či porovnávanie zhladených písmen z textu s databázou iných zhladených písmen [1]. V práci sme popísali metódy, pomocou ktorých sme sa chceli dopracovať k vhodnému deskriptoru dostatočne opisujúcemu písmeno. Ďalej sme vytvorili metódy, ktoré pracujú s deskriptormi písmen a odhadujú ich podobnosť. Na záver sme popísali grafické rozhranie softvéru a pomocou experimentov sme porovnali výstupné hodnoty metrík.

1. Úvod do problematiky

Ako som už spomínala, naším prvým cieľom bola definícia vstupného 2D písmena pomocou krivky. Naš počiatočný zámer bol vypočítať skeleton a pomocou neho definovať 2D písmeno zo vstupného obrázka. Skeleton môžeme vypočítať viacerými postupmi - morfológické operácie opísané v [2] alebo pomocou funkcie vzdialenosti [3]. Vzhľadom k jeho topológii nám skeleton neposkytoval informáciu o počiatočnom a koncovom bode, teda sme nevedeli definovať skeleton ako krivku pre každé písmeno. Vzhľadom na predchádzajúcu skúsenosť, sme si zvolili za najefektívnejšiu krivkovú reprezentáciu písmena jeho vlastnú hranicu.

Ďalším cieľom bolo zvoliť deskriptor, ktorý nám poskytne vhodnú informáciu o krivkách písmen. Deskriptor mal byť rezistentný voči zmene veľkosti krivky písmena, zmene jej rotácie či prípadnému šumu. Preto sme si ako deskriptor zvolili krivosť [4].

Výstupom krivosti je diskretná funkcia, ktorú sme potrebovali zhladiť. Preto sme na krivostnú krivku aplikovali rovnicu vedenia tepla. Táto metóda nebola vhodná vzhľadom na výpočtový čas. Preto sme si zvolili na vyhladenie lagrangeovskú difúziu, ktorá nám zároveň umožnila redistribúciu bodov krivky a tým presnejšie určenie krivosti.

Ďalej sme navrhli metriky, pričom sme postupovali od najjednoduchšej, ktorá vykonávala odčítanie až po zložitejšiu využívajúcu funkciu vzdialenosti.

Celý postup sme implementovali v softvéri.

2. Definícia písmena pomocou deskriptora

Ako najefektívnejšiu metódu na získanie hranice písmena sme si zvolili metódu využívajúcu dilatáciu. Vzhľadom k tomu, že dilatácia pracuje najlepšie s binarizovaným obrázkom, museli sme najprv na vstupný obrázok aplikovať metódu zvanú thresholding.

2.1 Thresholding

Thresholding patrí medzi najjednoduchšie a základné metódy obrazového spracovania [2]. Pomocou thresholdingu binarizujeme šedotónové obrázky a to tak, že sa zvolí hodnota, na základe ktorej thresholding pracuje. Všetky hodnoty pixelov väčšie alebo rovné ako daná hodnota sa nastaví na 255 a všetky menšie na 0. Keďže nevieme dopredu, či bude vstupný obrázok šedotónový, pred tým ako naňho použijeme thresholding ho automaticky upravíme na šedotónový.

2.2 Dilatácia

Dilatácia je ďalšia základná operácia v obrazovom spracovaní [2]. Pracuje pomocou ľubovoľne zvoleného štruktúrného elementu. Je nemenný voči posunutiu a je založený na Minkowského sume : $(a, b) + (c, d) = (a + c, b + d)$. Dilatácia je definovaná nasledovne:

$$A \oplus B = \bigcup_{b \in B} A_b, \quad (2.1)$$

kde A je binárny obrázok a B je štruktúrny element.

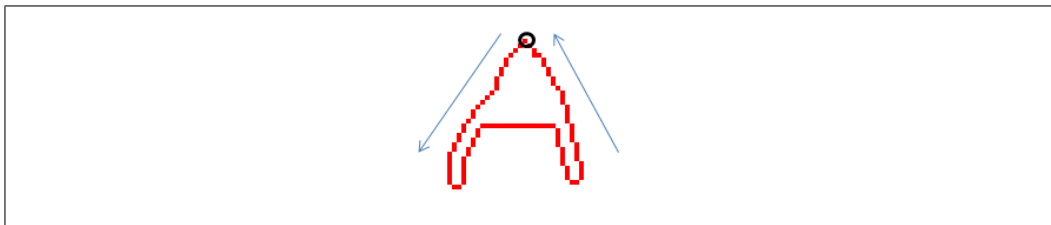
My sme si zvolili 4-susedový štruktúrny element veľkosti 3×3 . Ako vidieť na obrázku (2.1), hranicu písmena sme dostali odrátaním pôvodného písmena od písmena s dilatáciou.



Obr. 2.1: Štruktúrálny element, pôvodné písmeno, dilatácia, hranica písmena

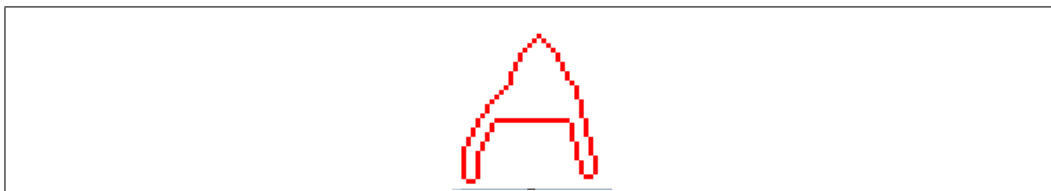
2.3 Získanie krivky z hranice písmena

Počiatočný bod krivky sme zadefinovali ako najvyššie položený pixel patriaci hranici písmena. Od tohto pixela sme postupovali v protismere hodinových ručičiek s tým, že sme vybrali len najbližší pixel v tomto smere a postupne sme ukladali body v danom poradí až kým sme sa nedostali späť k počiatočnému pixelu.



Obr. 2.2: Smer výberu pixelov, počiatočný bod je označený krúžkom

Ako vidieť na obrázku (2.3), pomocou tohto postupu sa taktiež odstránili nepotrebné pixely.



Obr. 2.3: Finálna hranica písmena

Vzhľadom na to, že sme spočiatku nevedeli, aký typ metrík budeme používať a čo všetko budú vyžadovať, dĺžka kriviek bola pre každé písmeno rôzna. Avšak vzhľadom na neskoršie poznatky sme sa rozhodli upraviť krivky získané z hraníc písmen na rovnaký počet bodov N , ktorý bol definovaný podľa krivky s najväčším počtom pixelov. Ďalej sme potrebovali správne určiť dĺžku kroku h . Preto sme si vypočítali celkovú dĺžku L upravovanej krivky, vzhľadom na euklidovslú vzdialenosť bodov, v tomto prípade stredov pixelov, a vydělili sme ju s dĺžkou N . S daným krokom sme sa pohybovali po krivke s ohľadom na dĺžku L a postupne sme ukladali nové hodnoty.

2.4 Krivosť

Ďalším krokom našej práce bol výpočet krivosti, ktorú sme si zvolili za deskriptor kriviek opisujúcich písmeno. Krivosť je definovaná nasledovne:

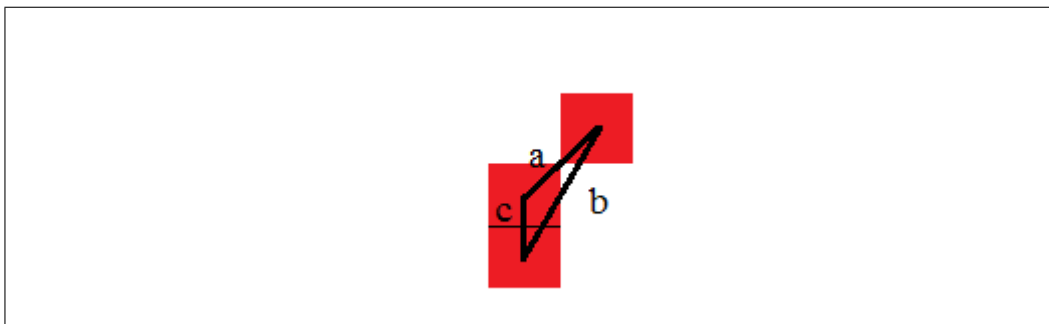
$$\kappa = \frac{1}{R} , \quad (2.2)$$

kde R je polomer príslúchajúcej kružnice. Veľkosť polomeru opísanej kružnice vyjadruje nasledovný vzťah :

$$R = \frac{a}{\sin\alpha} = \frac{b}{\sin\beta} = \frac{c}{\sin\gamma} , \quad (2.3)$$

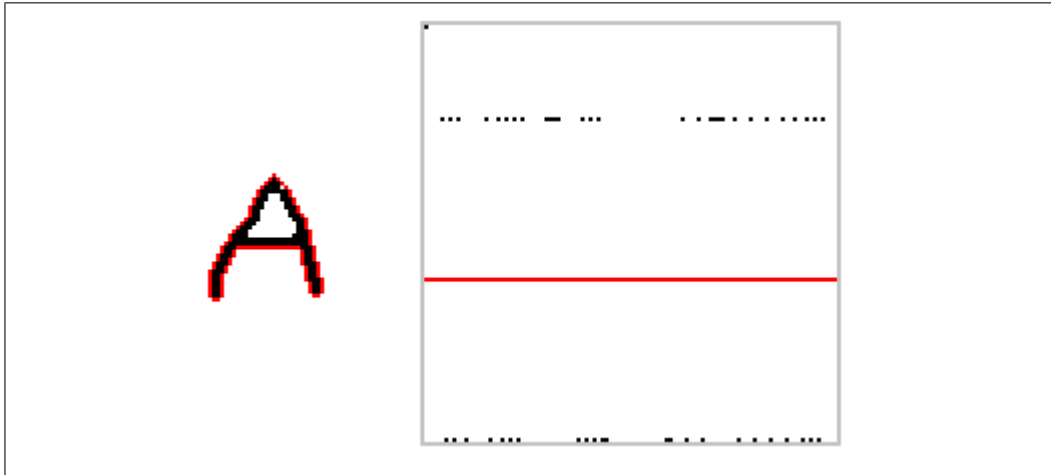
kde a , b a c sú dĺžky strán a α , β a γ sú uhly v príslušnom trojuholníku. Dĺžky strán trojuholníka sme vypočítali s použitím Pytagorovej vety a s použitím arcus cosinus sme získali príslušné uhly.

$$\alpha = \arccos\left(\frac{a^2 - b^2 - c^2}{2bc}\right) , \quad (2.4)$$



Obr. 2.4: Zobrazenie trojuholníka (časť z hranice písmena), ktorému opíšeme kružnicu na nájdenie krivosti pre stredný pixel.

Krivosť sme vzhľadom na ďalšiu prácu ukladali do pola aj v preškálovanom tvare vzhľadom na dĺžku danej krivky, teda najvyššiu hodnotu krivosti sme nastavili na veľkosť dĺžky krivky (počet pixelov krivky), najnižšiu na nulu a ostané odvíjajú sa od týchto hodnôt.

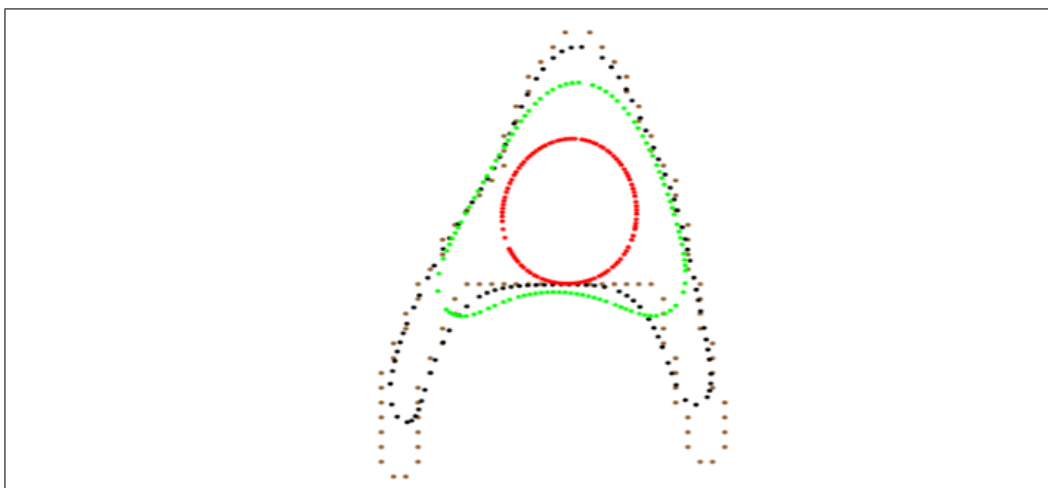


Obr. 2.5: Originálne písmeno s vypočítanou hranicou, graf krivosti (červená čiara znázorňuje x-ovú os v bode $y=0$, najvyššia hodnota krivosti je rovná 1 a najnižšia je rovná -0.63 . Vidíme, že graf funkcie nadobúda konečný počet hodnôt - v tomto prípade 4. Dôvodom je diskrétna povaha postupnosti pixelov reprezentujúca hranicu písmena.)

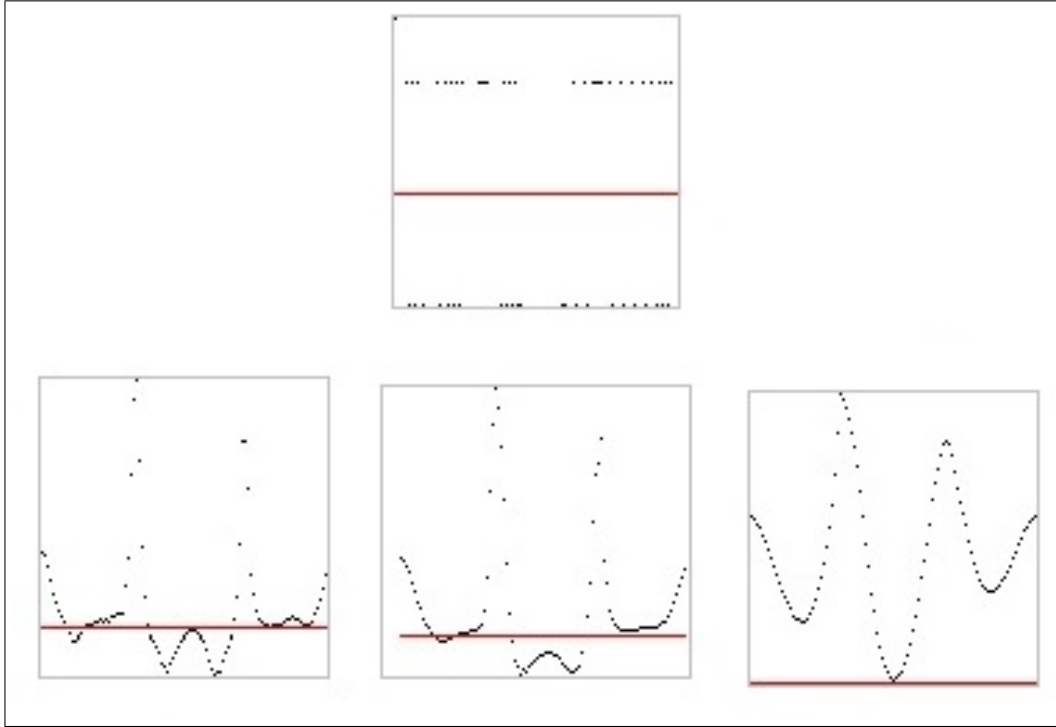
Keďže vidíme, že graf krivosti krivky, zobrazený na obrázku (2.5), nadobúda iba 4 rôzne hodnoty, na prvý pohľad chaoticky usporiadané, nejaví sa v takomto stave ako vhodná voľba deskriptora. Riešením by bolo a) zvoliť presnejší (viacbodový) systém pre odhad krivosti v danom bode, alebo b) realizovať evolúciu krivky, ktorá spôsobí jej postupné zhladzovanie. Keďže druhý postup je lepšie moderovateľný, zvolili sme ten.

3. Lagrangeovská difúzia

Ako vidieť na obrázku (2.5), krivku bolo potrebné zhladiť. Ako som už spomínala, zvolili sme si na to lagrangeovskú difúziu. Lagrangeovská difúzia je veľmi podobná vedeniu tepla a pracuje s krivosťou krivky. S narastajúcou krivosťou narastá aj rýchlosť difúzie, teda časti s najväčšou krivosťou sa vyhladzujú rýchlejšie ako ostatné. Ako vidieť na obrázku (3.1) a (3.2), negatívne časti krivky sa postupne posúvajú smerom von a pozitívne smerom dovnútra krivky. Krivka sa s narastajúcim počtom krokov Lagrangeovskej difúzie zaobluje a zmršťuje.



Obr. 3.1: Hranica písmena (hnedá farba), hranica po 5 (modrá), po 10 (zelená), po 25 krokoch lagrangeovej difúzie (červená).



Obr. 3.2: Krivostná krivka určená z hranice písmena, krivosť po 5, po 10, po 25 krokoch lagrangeovej difúzie

3.1 Numerická interpretácia lagrangeovskej difúzie

Hlavná schéma, od ktorej sa odvíjajú ďalšie úpravy je lagrangeovská difúzia s redistribúciou, bližšie opísaná v [4]:

$$x_t = \epsilon x_{ss} + \alpha x_s \quad (3.1)$$

Nasledovný výraz definuje diskretnú semi-implicitnú schému konečných objemov :

$$\frac{h_i^m + h_{i+1}^m x_i^{m+1} - x_i^m}{2 \tau} = \epsilon \left(\frac{x_{i+1}^{m+1} - x_i^{m+1}}{h_{i+1}^m} - \frac{x_i^{m+1} - x_{i-1}^{m+1}}{h_i^m} \right) + \alpha \left(\frac{x_{i+1}^{m+1} - x_{i-1}^{m+1}}{2} \right), \quad (3.2)$$

Rovnicu (3.2) vieme zapísať v skrátrenom tvare:

$$-A_i^m x_{i-1}^{m+1} + B_i^m x_i^{m+1} - C_i^m x_{i+1}^{m+1} = D_i^m, \quad (3.3)$$

kde D_i^m je pravá strana,

$$A_i^m = -\frac{\alpha_i^m}{2} + \frac{\epsilon_i^m}{h_i^m}, \quad C_i^m = \frac{\alpha_i^m}{2} + \frac{\epsilon_i^m}{h_{i+1}^m} \text{ a } B_i^m = (H_i^m + A_i^m + C_i^m), \quad (3.4)$$

pričom

$$H_i^m = \frac{h_{i+1}^m + h_i^m}{2\tau} \quad (3.5)$$

Systém (3.2) sme riešili pomocou Thomasovho algoritmu a Sherman-Morrisonovej formuly vzhľadom k tomu, že je cyklický a trojdiagonálny.

3.2 Thomasov algoritmus

Thomasov algoritmus je zjednodušená forma Gaussovej eliminácie používaný na riešenie trojdiagonálnych systémov rovníc. Ako prvé počíta nové hodnoty nasledovným spôsobom:

$$c'_i = \begin{cases} \frac{C_i}{B_i} & ; i = 1 \\ \frac{C_i}{B_i - A_i c'_{i-1}} & ; i = 2, 3, \dots, n-1 \end{cases} \quad (3.6)$$

$$d'_i = \begin{cases} \frac{D_i}{B_i} & ; i = 1 \\ \frac{D_i - A_i d'_{i-1}}{B_i - A_i c'_{i-1}} & ; i = 2, 3, \dots, n, \end{cases} \quad (3.7)$$

kde A_i, B_i a C_i sme vypočítali z rovníc (3.4),

a pomocou spätnej substitúcie počíta výsledné hodnoty:

$$x_n = d'_n, \quad x_i = d'_i - c'_i x_{i+1} \quad ; i = n-1, n-2, \dots, 1. \quad (3.8)$$

3.3 Sherman-Morrisonová formula

Sherman-Morrisonová formula [5] sa v lineárnej algebre využíva na riešenie cyklických trojdiagonálnych systémov.

$$\begin{bmatrix} b_1 & c_1 & & & \alpha \\ a_2 & b_2 & c_2 & & \\ & a_3 & b_3 & \ddots & \\ & & \ddots & \ddots & c_{n-1} \\ \beta & & & a_n & b_n \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ \vdots \\ x_n \end{bmatrix} = \begin{bmatrix} d_1 \\ d_2 \\ d_3 \\ \vdots \\ d_n \end{bmatrix}, \quad (3.9)$$

kde je $\alpha = a_n, \beta = c_1$ a n je počet riadkov systému.

Sherman-Morrison zjednodušuje výpočet takéhoto typu matíc vzhľadom k tomu, že pôvodnú maticu A upraví do tvaru $(A + u \otimes v)$. Teda lineárny

system (3.9) bude vyzerat' nasledovne:

$$(A + u \otimes v).x = b \quad (3.10)$$

S použitím rovníc (3.11) odvodených z (3.10) získame vektory y a z ,

$$A.y = b \quad A.z = u \quad (3.11)$$

pomocou ktorých vieme vypočítat' hľadane hodnoty :

$$x = y - \left[\frac{v.y}{1 + (v.z)} \right].z \quad (3.12)$$

3.3.1 Postup v programe

V programe sme si u a v zadefinovali nasledovne:

$$u = \begin{bmatrix} \gamma \\ 0 \\ 0 \\ \vdots \\ \beta \end{bmatrix} \quad v = \begin{bmatrix} 1 \\ 0 \\ 0 \\ \vdots \\ \frac{\alpha}{\gamma} \end{bmatrix}, \quad (3.13)$$

pričom γ je ľubovoľný parameter, ktorý sme si vzhľadom na zachovanie presnosti zadefinovali ako $\gamma = -b_1$ a vektory u a v majú veľkosť n .

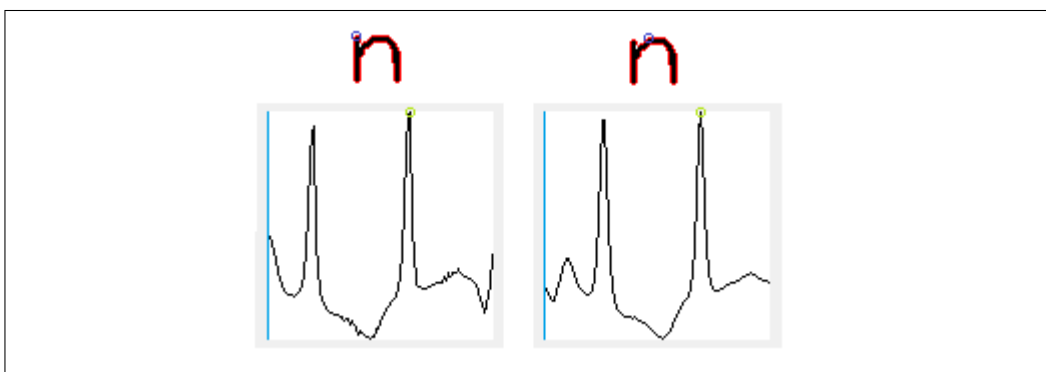
Teda matica A je trojdiagonálna časť systému (3.9) s dvomi modifikovanými členmi:

$$b'_1 = b_1 - \gamma, \quad b'_n = b_n - \frac{\alpha\beta}{\gamma} \quad (3.14)$$

Potom sme vyriešili rovnice (3.11) pomocou už spomínaného Thomasovho algoritmu. A nakoniec sme dosadením hodnôt vypočítaných pomocou Thomasovho algoritmu do z rovnice (3.12) získali hľadaný vektor.

4. Invariantnosť voči počiatočnému bodu

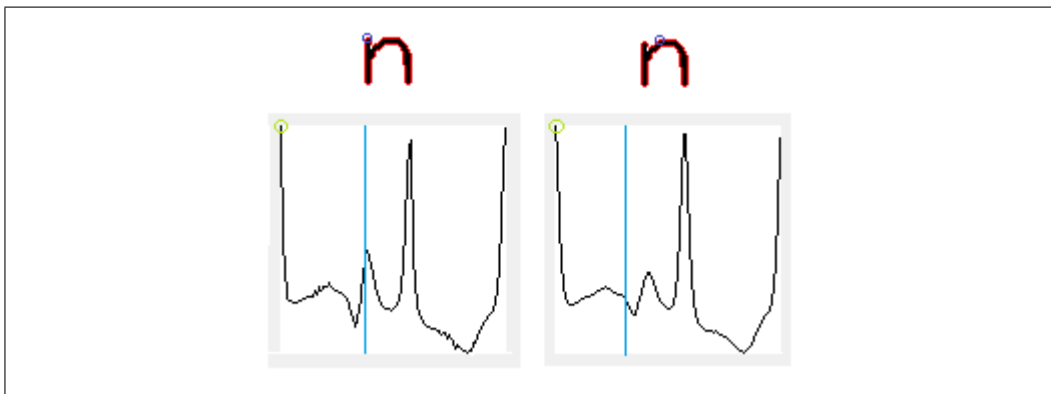
Po preskúmaní správania deskriptorov sme si všimli, že počiatočný bod, t.j. prvý najvrchnejší pixel patriaci hranici písmena, je citlivý voči rotácii a menším zmenám písmena, čo môže, pri niektorých písmenách, znamenať problém. Na obrázku (4.1) je bližšie znázornená situácia, pri ktorej nastáva spomínaný problém. Môžeme si všimnúť, že porovnávame podobné písmená, pričom majú rôzny počiatočný pixel označený kružnicou. Ako vidieť, deskriptory sú síce podobné, ale posunuté, čo môže spôsobiť menej presnú informáciu.



Obr. 4.1: Porovnanie deskriptorov podobných písmen s rôznymi polohami počiatočného pixela.

Na základe daného poznatku sme sa rozhodli zvoliť si jeden pixel, vzhľadom na ktorý posunieme daný deskriptor. Ako najvhodnejší sme si určili pixel, ktorý reprezentuje globálne maximum, teda najvyššiu hodnotu krivosti (na obrázku (4.1) je vyznačený zelenou kružnicou). Obrázok (4.2) znázorňuje výsledok spomínaného posunu. Je vidieť, že napriek rôznym počiatočným pixelom sa 2D grafy deskriptorov dvoch podobných písmen na seba omnoho viac podobajú.

Na obrázkoch (4.1) a (4.2) môžeme vidieť modrú čiaru prechádzajúcu grafom. Pomocou nej zvýrazňujeme pôvodnú polohu počiatočného pixela (označený kružnicou na hranici písmena) v grafe. Toto značenie sa bude objavovať aj v nasledujúcom texte.



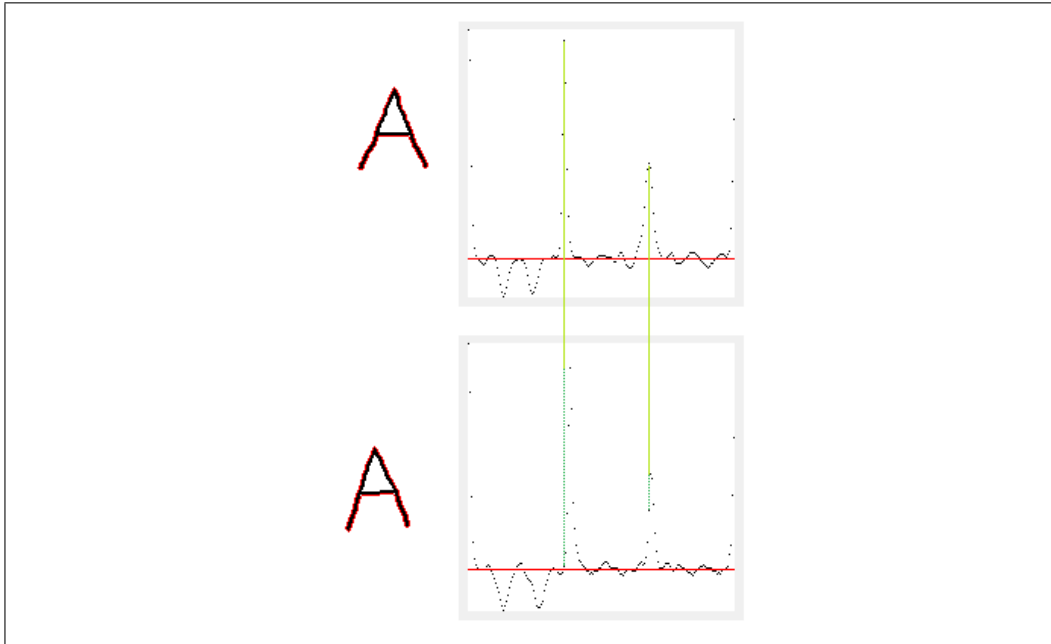
Obr. 4.2: Porovnanie posunutých deskriptorov podobných písmen

5. Metriky

Po získaní vhodnej krivky, respektíve deskriptora definujúceho vstupné písmeno, sme mohli uvažovať nad vhodnými metrikami, ktoré nám budú viesť definovať podobnosť kriviek reprezentujúcich písmená na základe vzdialenosti. Cieľom je určiť, o aké strojom čitateľné písmeno sa jedná.

5.1 Rozdielová metrika

Za prvú metódu sme si zvolili jednoduché odčítavanie dvoch kriviek. Teda porovnáваме konkrétny (skúmaný) deskriptor s deskriptorom uloženým v atlase a to tak, že prechádzame naraz po oboch krivkách a robíme abstrakciu ich hodnôt, ktoré následne sčítujeme a ukladáme. Vzhľadom k tomu, že majú všetky deskriptory definovanú jednotnú dĺžku, sa naša práca značne zjednodušila. Za najvhodnejší deskriptor z atlasu vyberieme ten, ktorému sme napočítali najnižší rozdiel od skúmaného deskriptora v absolútnej hodnote. Ako vidieť na obrázku (5.1), môže nastať situácia, pri ktorej deskriptory vyzerajú takmer rovnako ale jeden z nich je posunutý o pár pixelov a Rozdielová metrika tým pádom nemusí vhodne určiť skúmaný deskriptor.



Obr. 5.1: Prvý stĺpec : skúmané písmená. Druhý stĺpec : 2D grafy deskriptorov príslušných písmen.

Na obrázku (5.1) je pomocou zelených čiar vyznačené, o koľko je prvý deskriptor posunutý vzhľadom na druhý deskriptor. Slabozelená čiara smeruje od pixela prvého deskriptora k pixelu druhého deskriptora, s ktorým by sa podľa správnosti mal pixel prvého deskriptora porovnávať. Tmavozelená prerušovaná čiara smeruje k pixelu, s ktorým sa bude pixel prvého deskriptora vskutočnosti porovnávať. Môžeme si všimnúť, že pri odpočítaní daných pixelov (hodnôt) nám vznikne veľký rozdiel.

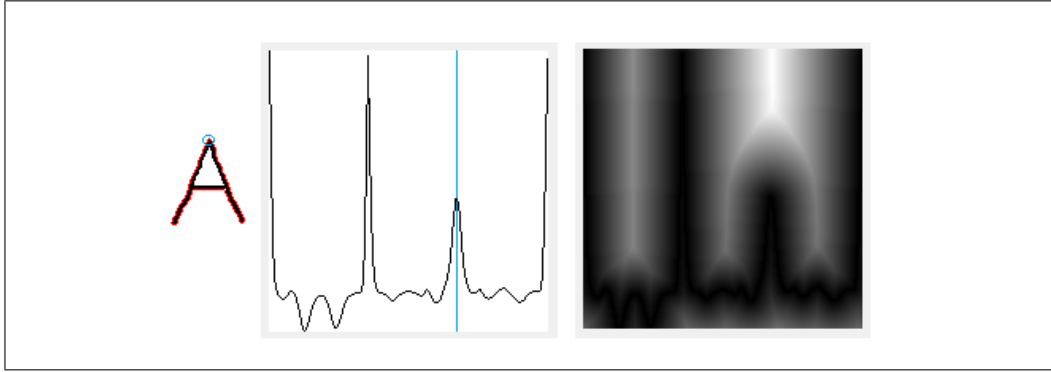
5.2 Bodovo-vzdialenostná metrika

Vzhľadom na skúsenosti, ktoré sme nadobudli skúmaním výsledkov z Rozdielovej metriky, sme vytvorili Bodovo-vzdialenostnú metriku, ktorá využíva vzdialenostnú funkciu. Pomocou nej ošetríme nedostatky rozdielovej metriky.

5.2.1 Vzdialenostná funkcia

Vzdialenostnú funkciu sme zostrojili pomocou Rouy-Tourin schémy, bližšie definovanú v [3]. Rovnica, od ktorej sa celý postup v [3] odvíja je Eikonalova rovnica s časovou relaxáciou:

$$d_t + |\nabla d| = 1 \quad (5.1)$$



Obr. 5.2: Pôvodné písmeno, 2D graf deskriptor, preškálovaná vzdialenostná funkcia 2D grafu deskriptora.

počítaná na oblasti $\Omega \times [0, T_D]$, kde Ω je doména obrázka s Dirichletovou podmienkou:

$$d(x, t) = 0, \quad x \in \Omega_0 \subset \Omega \quad (5.2)$$

Na numerický výpočet rovnice (5.1) s podmienkou (5.2) sme použili explicitnú časovú diskretizáciu s časovým krokom τ_D a pomocou Rouy-Tourin schémy sme diskretizovali (5.1) v priestore s krokom h_D . Rouy-Tourin schéma vyzerá pre (5.1) nasledovne:

$$d_{ij}^{n+1} = d_{ij}^n + \tau_D - \frac{\tau_D}{h_D} \sqrt{\max(M_{ij}^{-1,0}, M_{ij}^{1,0}) + \max(M_{ij}^{0,-1}, M_{ij}^{0,1})} \quad (5.3)$$

pričom

$$M_{ij}^{pq} = (\min(d_{i+p, j+q}^n - d_{ij}^n, 0))^2. \quad (5.4)$$

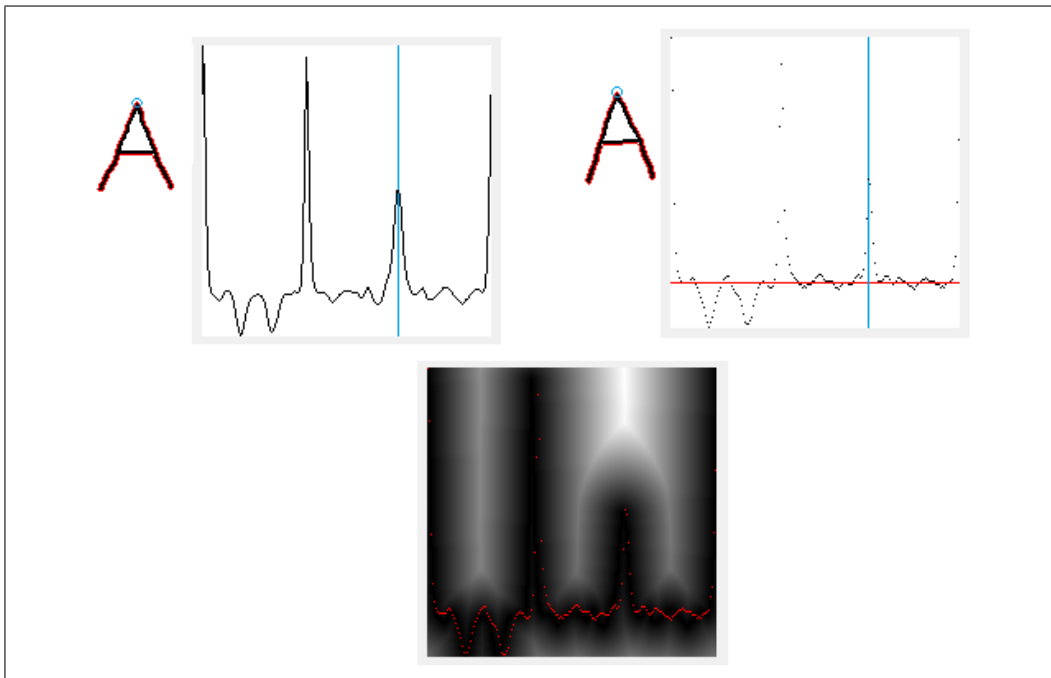
Na obrázku (5.2) je pre lepšiu viditeľnosť zobrazená preškálovaná vzdialenostná funkcia, pričom čierna má hodnotu 0 a biela 255. Teda je vidieť, ako sa postupne "zvyšuje hodnota" pixelov s narastajúcou vzdialenosťou od pôvodnej krivky deskriptora.

Vzhľadom na to, že deskriptor je graf krivosti jednoduchej uzavretej krivky, v neskorších etapách práce sme usúdili, že výsledky budú presnejšie, ak budeme počítat vzdialenostnú funkciu ku grafu funkcie tak, že na hornom a dolnom okraji budeme uvažovať nulovú Neumannovú podmienku a na pravom a ľavom okraji budeme uvažovať cyklickú okrajovú podmienku.

5.2.2 Funkcionalita Bodovo-vzdialenostnej metriky

Ako prvý krok sme potrebovali diskretný tvar krivky transformovať na 2D graf funkcie jej krivosti vzhľadom na požiadavky vzdialenostnej funkcie. Následne sme vypočítali vzdialenostnú mapu, ktorú sme si uložili. Tento proces sa vykonáva automaticky pre daný deskriptor po jeho pridaní do atlasu.

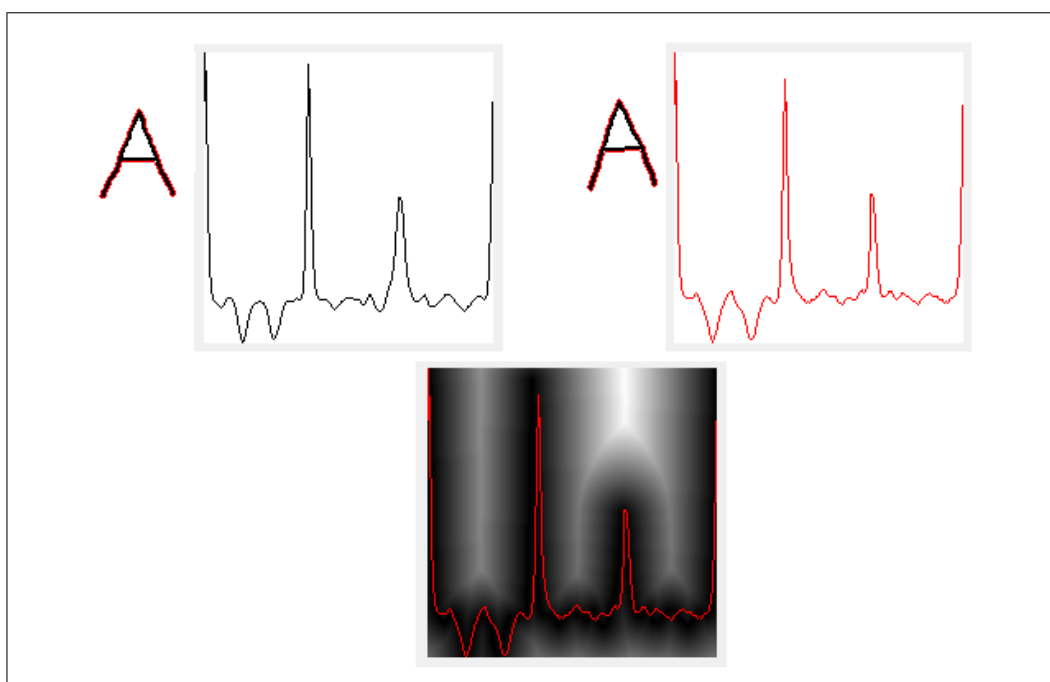
Ďalej sme mohli porovnávať skúmaný deskriptor s deskriptormi v atlase a to tak, že sme sa pohybovali po skúmanom deskriptore a zároveň sme sledovali, akú hodnotu má aktuálny bod vo vzdialenostnej mape deskriptora v atlase, následne sme hodnoty od seba odčítali. Bodovo-vzdialenostná metrika vyberá ako najvhodnejší deskriptor z atlasu pre skúmaný deskriptor ten, pri ktorom je absolútna hodnota sumy odčítaných hodnôt najnižšia.



Obr. 5.3: Prvý riadok : Pôvodné písmená s deskriptormi. Druhý riadok : preškálovaná vzdialenostná funkcia deskriptora prvého písmena s deskriptorom druhého písmena (červená farba).

5.3 Grafovo-vzdialenostná metrika

Pracuje rovnako ako Bodovo-vzdialenostná metrika s rozdielom, že graf vzdialenostnej funkcie deskriptora v atlase porovnáваме s 2D grafom skúmaného deskriptora. Týmto spôsobom vieme presnejšie určiť podobnosť ako pomocou Bodovo-vzdialenostnej metriky.

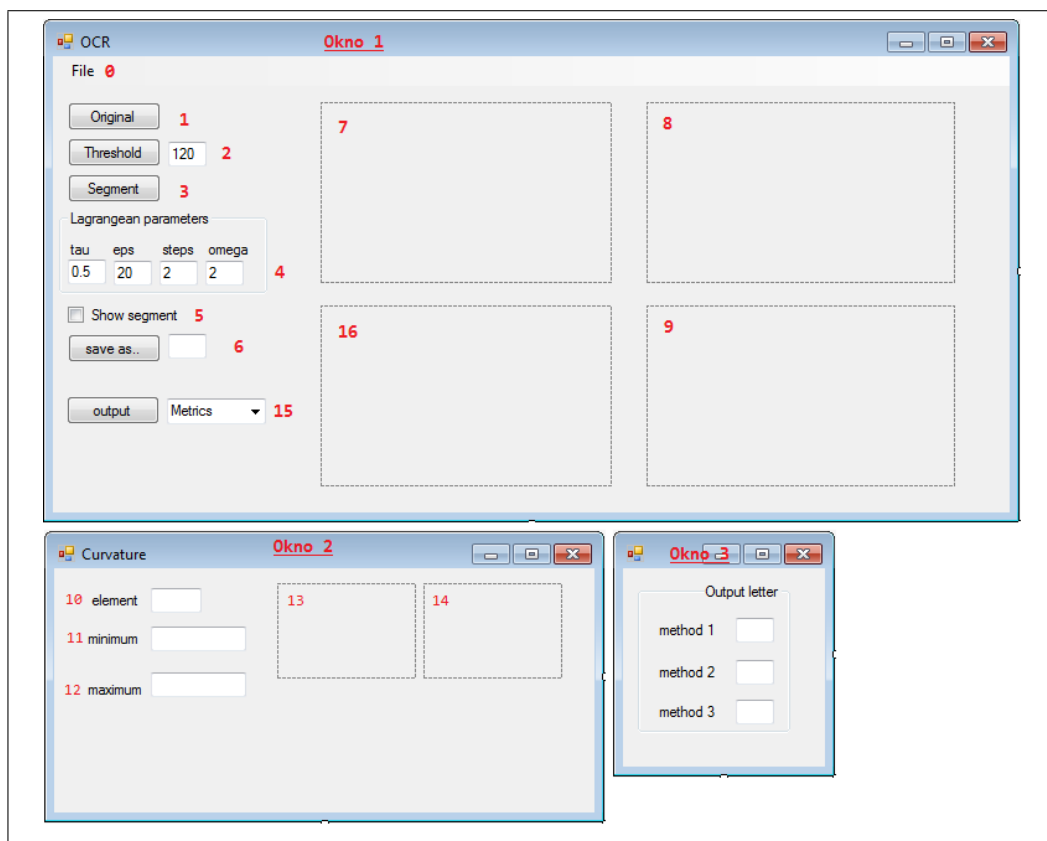


Obr. 5.4: Prvý riadok : Pôvodné písmená s 2D grafmi deskriptorov. Druhý riadok : preškálovaná vzdialenostná funkcia deskriptora prvého písmena s 2D grafom deskriptora druhého písmena (červená farba).

6. Popis softvéru

Všetky metódy sme implementovali v softvéri, ktorý sme nazvali všeobecnou skratkou OCR. Softvér bol vytvorený v Microsoft Visual Studio 2013 v jazyku C# 5.0 .

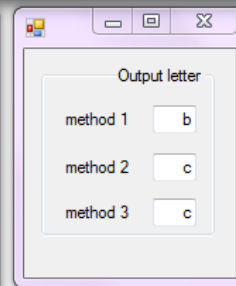
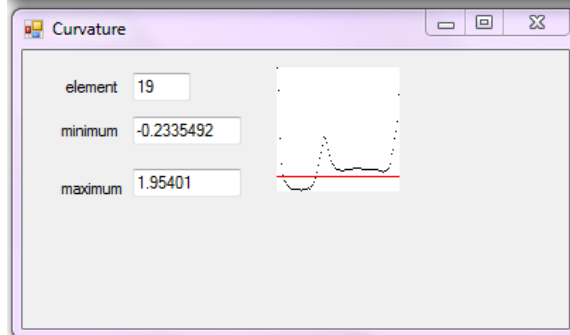
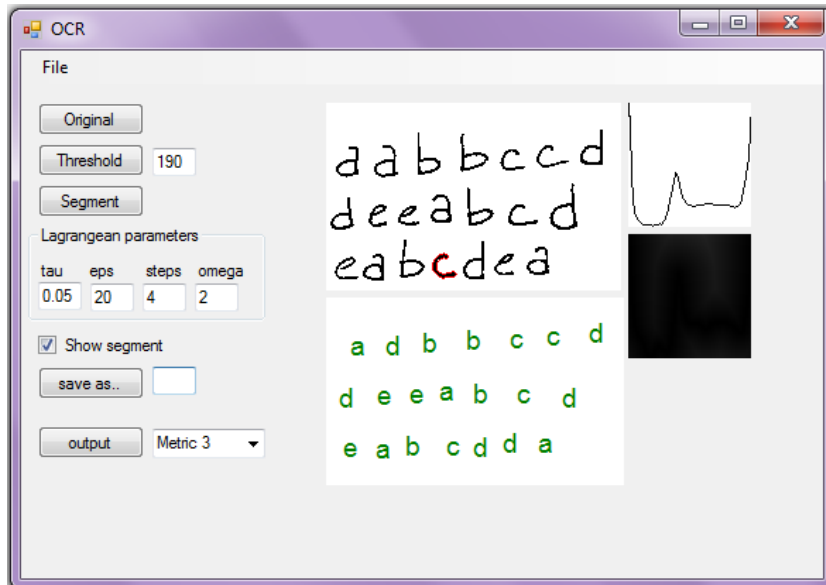
Jeho základná myšlienka je, že užívateľ nastaví parametre na získanie najvhodnejšieho deskriptora. Akonáhle program vytvorí a uloží, okrem iného, deskriptory pre každé písmeno zo vstupného obrázka, užívateľ vyznačí písmeno (kliknutím naňho), ktoré chce uložiť, napíše do textboxu jeho názov (počítačom rozoznateľný znak - písmeno/číslo z klávesnice) a potvrdí ho, čím sa dané písmeno uloží do atlasu. Do atlasu sa ukladá deskriptor, vzdialenostná mapa a užívateľom zvolený názov. Akonáhle klikneme na ďalšie písmeno, zobrazí sa nám výstup z jednotlivých metrík.



Obr. 6.1: Grafické rozhranie

6.1 Opis grafického rozhrania

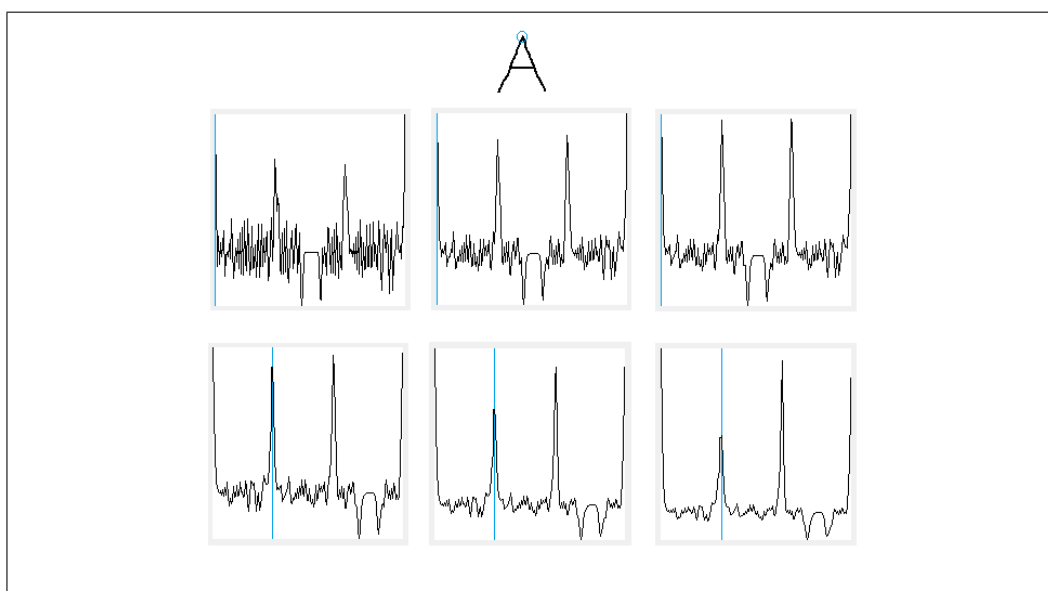
1. Pomocou File → Open (0) sa nám objaví dialógové okno, pomocou ktorého si môžeme zvoliť ľubovoľný vstup (obrázok) z počítača alebo pripojeného externého disku.
 2. Tlačítko Threshold (2) nám umožňuje aplikovať thresholding na základe hodnoty, ktorú zadáva užívateľ pomocou príslušného textBoxu.
 3. Pomocou tlačítka Segment (3) sa vykonajú všetky potrebné operácie na získanie a uloženie deskriptorov pre každé písmeno z aktuálneho vstupného obrázka. Pred použitím tlačítka Segment treba nastaviť parametre vstupujúce do Lagrangeovej difúzie (4).
 4. Do pictureBoxu (7) sa načíta užívateľom zvolený vstup (obrázok). PB (7) reaguje na klik s myšou.
 5. CheckBox zaktívňime, ak chceme, aby sa v pictureBoxe (7) zobrazila aj hranica daného písmena (aktívne písmeno), na ktorý sme klikli v (7).
 6. V pictureBoxe (8) sa po kliknutí na písmeno v (7) objaví 2D graf deskriptora aktívneho písmena a zároveň sa nám objaví Okno 2.
 7. TextBox (10) vypíše polohu aktívneho písmena v poli, textBox (11) vypíše minimálnu hodnotu krivosti (deskriptora) aktívneho písmena a textBox (12) vypíše maximálnu hodnotu krivosti (deskriptora) aktívneho písmena.
 8. V pictureBoxe (13) sa nám objaví graf diskkrétnej krivky deskriptora aktívneho písmena. PB (13) reaguje na dvoj-klik myšou.
 9. Po dvoj-kliku myšou na pB (13) sa nám graf z pB (13) objaví v pB (14) a bude tam dovtedy, kým opäť neklikneme do pB(13). Tento pictureBox slúžil hlavne pri sledovaní zmeny deskriptorov písmen zo vstupu.
 10. Ak chceme deskriptor aktívneho písmena uložiť do atlasu, napíšeme do textBoxu (6) jeho názov, teda počítačom rozoznateľný znak definujúci písmeno a stlačíme tlačítko save as..
 11. V pB (9) sa nám zobrazí graf vzdialenostnej funkcie práve uloženého písmena. Akonáhle máme v atlase aspoň jeden uložený deskriptor spolu s jeho názvom, po kliknutí do pictureBoxu (7) sa nám otvorí okno 3, ktoré nám prezentuje meno deskriptora (počítačom čitateľný znak) ktorí určili jednotlivé metriky.
 12. Ak chceme vidieť konečnú transformáciu textu do počítačom čitateľných znakov, pomocou scrollBaru (15) si zvolíme metriku, ktorej predikciu chceme vidieť a stlačením príslušného tlačítka output sa v pictureBoxe (16) objaví daný text.
- Tlačítkom Original (1) vrátíme upravovaný vstupný obrázok do pôvodného stavu.
- Nasledovný obrázok ilustruje použitie popísaného softvéru v praxi.



7. Experimenty

7.1 Porovnanie vplyvu lagrangeovských parametrov na deskriptor

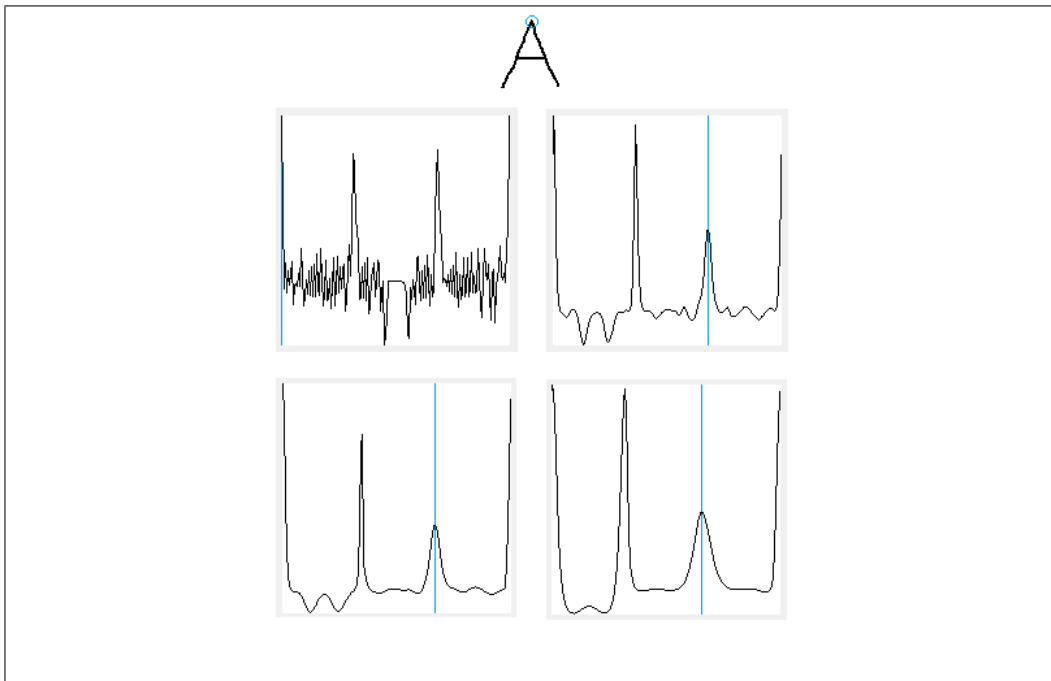
7.1.1 Porovnanie deskriptora pri zmene veľkosti časového kroku τ



Obr. 7.1: Deskriptory písmena A pri zmene τ : ($\tau = 0.01, \epsilon = 20$), ($\tau = 0.03, \epsilon = 20$), ($\tau = 0.04, \epsilon = 20$), ($\tau = 0.05, \epsilon = 20$), ($\tau = 0.07, \epsilon = 20$), ($\tau = 0.09, \epsilon = 20$).

Na obrázku (7.1) je vidieť, ako sa deskriptor s narastajúcim τ vyhladzuje.

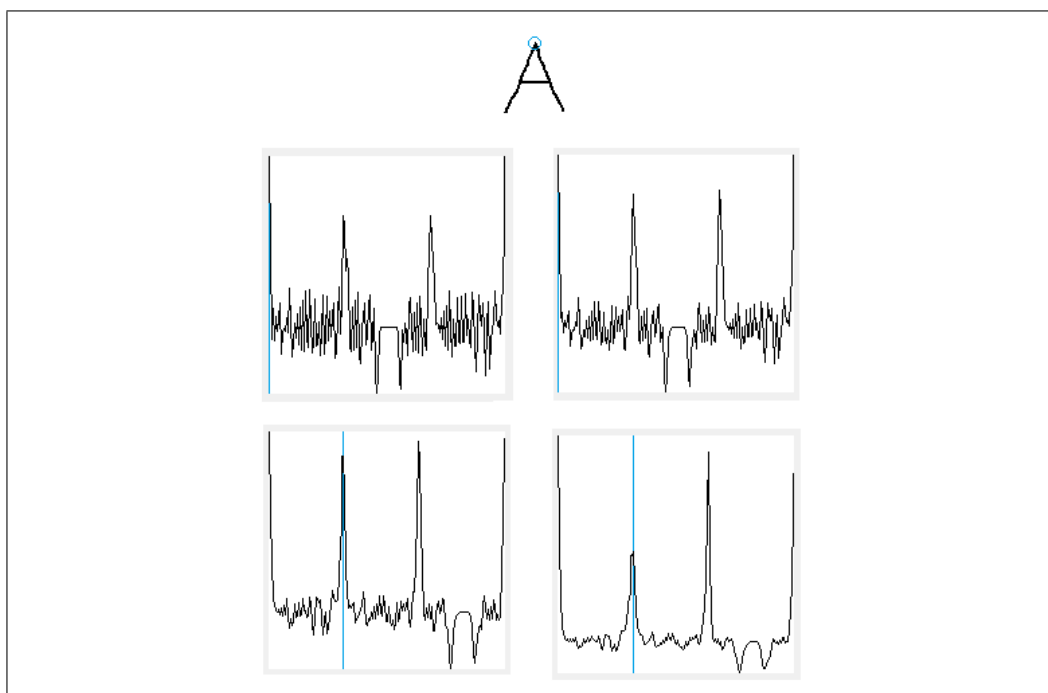
7.1.2 Porovnanie deskriptora pri zmene počtu časových krokov



Obr. 7.2: Deskriptory písmena A pri zmene počtu časových krokov : $t = 1, 5, 10, 20$.

Na obrázku (7.2) si môžeme všimnúť, že s narastajúcim počtom časových krokov sa deskriptor vyhladzuje.

7.1.3 Porovnanie deskriptora pri zmene veľkosti parametra ϵ

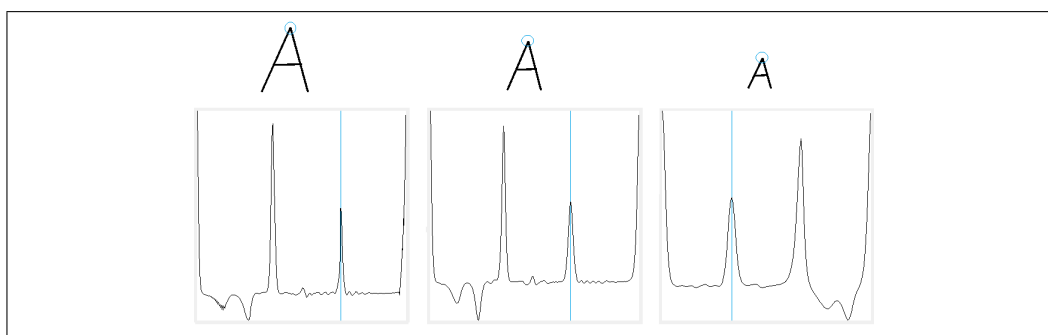


Obr. 7.3: Deskriptor písmena A pri zmene ϵ :5, 10, 20, 40

7.2 Porovnanie deskriptorov modifikovaného písmena

V tomto experimente skúmame vplyv zmeny rotácie a zmeny veľkosti písmen na deskriptory. Na obrázkoch (7.4) a (7.5) môžeme vidieť, že pri spomínaných vplyvoch sa deskriptory relatívne dosť podobajú.

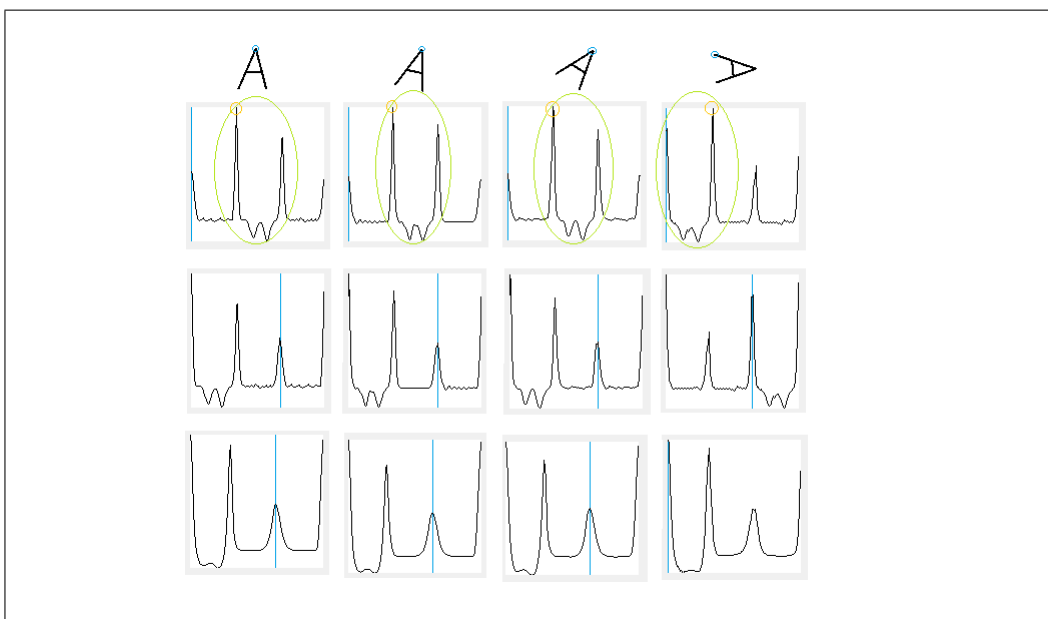
7.2.1 Porovnanie deskriptora pri zmene veľkosti písmena



Obr. 7.4: Deskriptor písmena A pri zmene jeho veľkosti.

Na obrázku (7.4) si môžeme všimnúť, že 2D graf deskriptora najmenšieho písmena *A* (posledný), je na rozdiel od predošlých dvoch značne posunutý. Je to spôsobené tým, že pri zhladzovaní deskriptorov vzniklo globálne maximum v inom bode. Nastáva to pri písmenách, ktorých deskriptory obsahujú dve a viac viditeľných maxím. Takúto situáciu vyriešime tak, že zdefinujeme v atlase písmeno *A* pod dvomi deskriptormi.

7.2.2 Porovnanie deskriptora pri zmene rotácie písmena

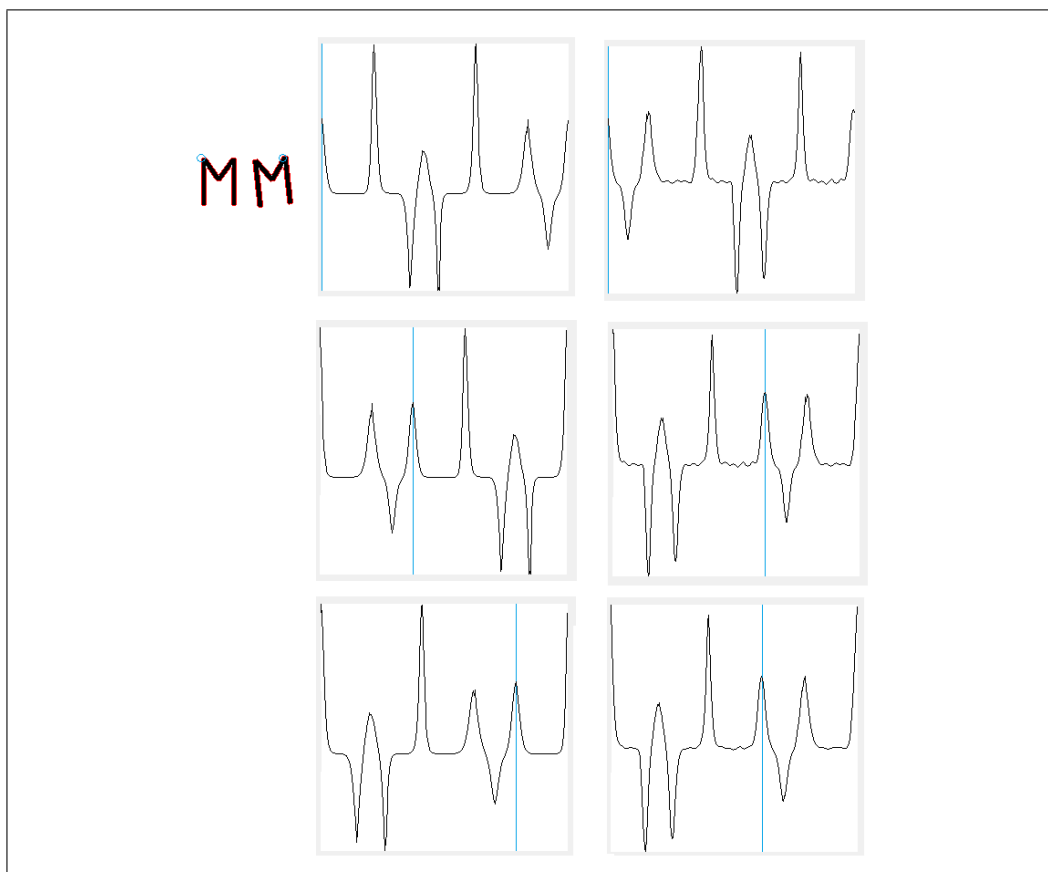


Obr. 7.5: Prvý riadok : Rotácia písmena *A*. Druhý riadok : Deskriptory príslušných písmen s neošetrenou invariabilitou počiatku ($\epsilon = 0.05$, $\tau = 20$, počet časových krokov = 5). Tretí riadok : Deskriptory príslušných písmen invariantných voči počiatku ($\epsilon = 0.05$, $\tau = 20$, počet časových krokov = 5). Štvrtý riadok : Deskriptory príslušných písmen invariantných voči počiatku ($\epsilon = 0.05$, $\tau = 20$, počet časových krokov = 7).

Na obrázku (7.5) je zobrazená rotácia písmena *A*. Pri rotácii sa mierne modifikujú písmená, preto si môžeme všimnúť drobný šum na deskriptoroch. Keď si všimneme časti vyznačené zelenou elipsou, môžeme vidieť, ako sa deskriptor pod vplyvom rotácie posúva. Prvé 3 deskriptory v druhom riadku sú značne podobné, lebo majú jednoznačne rovnaký počiatok. Posledný deskriptor je viditeľne posunutý vzhľadom na odlišný počiatok. Tento problém mal byť vyriešený pomocou posunu grafu deskriptora voči globálnemu maximu (invariantnosť voči počiatku). Avšak, keď sa zameriame na globálne maximum deskriptorov (vyznačených v druhom riadku oranžovým krúžkom) vidíme, že deskriptor posledného písmena má iné globálne maximum ako predošlé. Je to spôsobené šumom, ktorý vznikol

pri rotácii, a následným zhladením. Preto môžeme v predposlednom riadku vidieť prípad, ktorý sme spomínali v predošlej podsekcii. Písmeno s najväčšou rotáciou (posledné) má vzľadom na ostatné značne posunutý deskriptor. V štvrtom riadku vidíme, že pri zvýšení počtu časových krokov sa deskriptor posledného písmena zmodifikoval natoľko, že globálne maximum vzniklo na mieste rovnakom ako majú ostatné deskriptory.

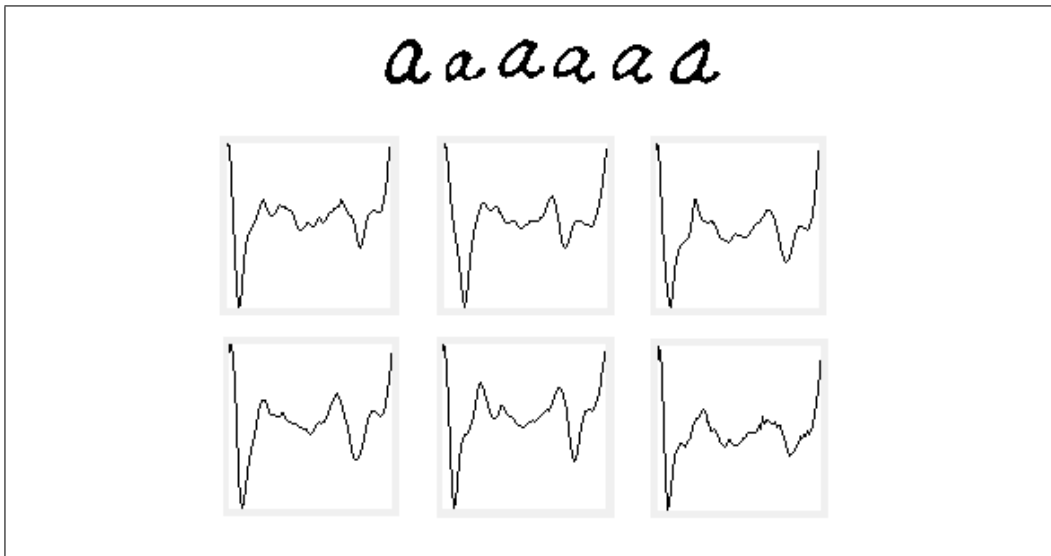
Podobný prípad môžeme pozorovať na nasledovnom obrázku (7.6).



Obr. 7.6: Prvý riadok : Deskriptory príslušných písmen neinvariantných voči počiatocnému bodu ($\epsilon = 0.05$, $\tau = 20$, počet časových krokov = 5) . Druhý riadok : Deskriptory príslušných písmen invariantných voči počiatocnému bodu ($\epsilon = 0.05$, $\tau = 20$, počet časových krokov = 5). Tretí riadok : Deskriptory príslušných písmen invariantných voči počiatocnému bodu ($\epsilon = 0.05$, $\tau = 20$, počet časových krokov = 6).

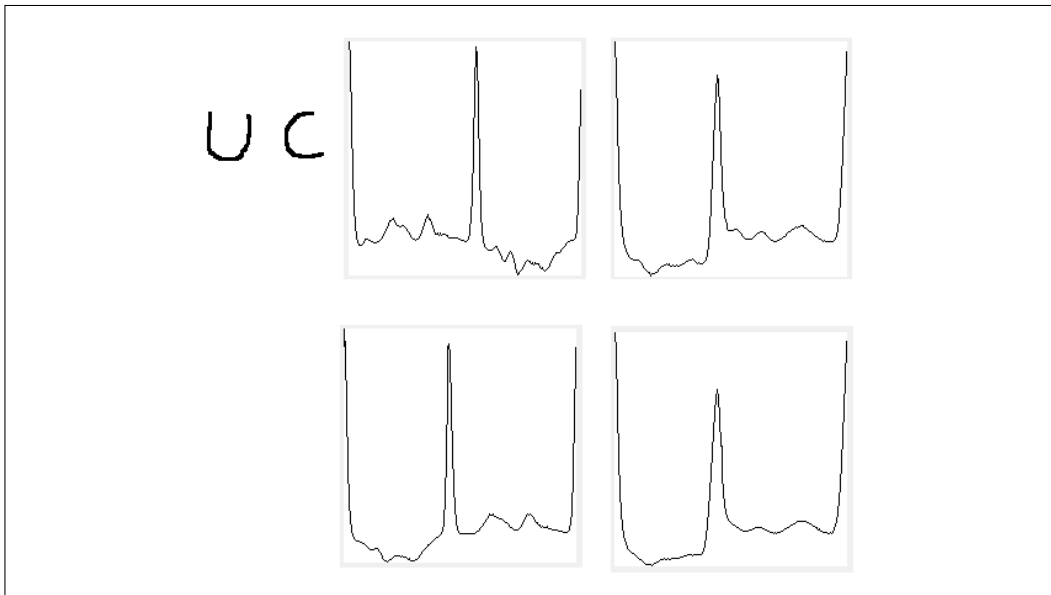
7.2.3 Porovnanie deskriptorov rôznych podôb písmena

a



Obr. 7.7: Deskriptory rôznych podôb písmena *a* pri lagrangeovských parametroch $\tau = 0.05$, $\epsilon = 20$, počet krokov lagrangeovskej difúzie = 4, $\omega = 2$.

7.3 Nevýhody krivostného deskriptora



Obr. 7.8: Prvý riadok : Deskriptory príslušných písmen ($\epsilon = 0.05$, $\tau = 20$, počet časových krokov = 5). Druhý riadok : Deskriptory príslušných písmen ($\epsilon = 0.05$, $\tau = 20$, počet časových krokov = 7).

Na obrázku (7.8) vidíme, že dve rôzne písmená môžu mať podobné deskriptory. Ďalším príkladom sú písmená t a x alebo M a W .

8. Záver

Podarilo sa nám na základe metód a postupov opísaných v práci zostrojiť aplikáciu, ktorá je schopná na základe vytvoreného atlasu rozpoznať ručne písaný text.

Literatúra

- [1] Shunji Mori, Ching Y. Suen, Kazuhiko Yamamoto: *Historical Review of OCR Research and Development* , IEEE
- [2] Rafael C. Gonzalez, Richard E. Woods: *Digital Image Processing, Third Edition*, University of Tennessee, MedData Interactive
- [3] Paul Bourguine, Peter Frokovič, Karol Mikula, Nadine Peyriéras, Mariana Remešíková: *Extraction of the intercellular skeleton from 2D images of embryogenesis using eikonal equation and advective subjective surface method*, CREA, Ecole Polytechnique-CNRS, 1 rue Descartes, 75005, Paris, France,
Department of Mathematics, Slovak University of Technology, Radlinského 11, Bratislava
- [4] Martin Balažovjeh, Karol Mikula, Mária Petrášová, Jozef Urbán: *Lagrangian method with topological changes for numerical modelling of forest fire propagation* , Department of Mathematics and Descriptive Geometry, Slovak University of Technology, Bratislava, Slovakia
- [5] William H. Press, Saul A. Teukolsky, William T. Vetterling, Brian P. Flannery: *Numerical Recipes, 3rd edition*, Cambridge University Press, The Edinburgh Building, Cambridge CB2 8RU, UK