

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

Fakulta elektrotechniky
a komunikačních technologií

BAKALÁŘSKÁ PRÁCE

Brno, 2016

Lukáš Černohous



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

METODA DYNAMICKÉHO BORCENÍ ČASU

DYNAMIC TIME WARPING

BAKALÁŘSKÁ PRÁCE

BACHELOR'S THESIS

AUTOR PRÁCE

AUTHOR

Lukáš Černošous

VEDOUCÍ PRÁCE

SUPERVISOR

Ing. Jan Klusáček

BRNO 2016



Bakalářská práce

bakalářský studijní obor **Automatizační a měřicí technika**

Ústav automatizace a měřicí techniky

Student: Lukáš Černohous

ID: 125394

Ročník: 3

Akademický rok: 2015/16

NÁZEV TÉMATU:

Metoda dynamického borcení času

POKyny PRO VYPRACOVÁNÍ:

1. Zpracujte rešerši různých variant algoritmu DTW a příkladů jeho praktického využití.
2. Vybrané metody naprogramujte.
3. Vyberte si jednu konkrétní oblast využití (rozpoznání hlasových povelů nebo zvuků, podpisu, ...) a získejte databázi záznamů, se kterými budete dále pracovat.
4. Vytvořte jednoduchý model, který bude klasifikovat na základě podobnosti stanovené pomocí DTW.
5. Vyhodnoťte dosažené výsledky.

DOPORUČENÁ LITERATURA:

Senin P.: Dynamic Time Warping Algorithm Review [online 2013-5]. Dostupné na <
http://www.researchgate.net/publication/228785661_Dynamic_Time_Warping_Algorithm_Review/file/32bfe5100f11a7929f.pdf >.

Termín zadání: 8.2.2016

Termín odevzdání: 23.5.2016

Vedoucí práce: Ing. Jan Klusáček

Konzultant bakalářské práce:

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

UPOZORNĚNÍ:

Autor bakalářské práce nesmí při vytváření bakalářské 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

Tato práce se zabývá popisem metody dynamického borcení času (DTW), jejím principem, variantami a způsobem výpočtu. Dále vyjmenovává některé typické případy její implementace. V praktické části se zaměřuje na vytvoření jednoduchého modelu verifikace podpisu na bázi DTW, jehož výsledky následně zhodnocuje. Model byl naprogramován v MATLABu.

Klíčová slova

Dynamické borcení času, DTW, varianty DTW, využití DTW, dynamické programování, MATLAB, verifikace podpisu, FAR FRR, EER, databáze SVC2004

Abstract

This thesis describes the method of dynamic Time Warping (DTW), its principles and method of calculation options. Additionally, it lists some of the typical examples of its implementation. In the practical part, it focuses on the creation of a simple signature verification model using DTW and it evaluates the results. The model was programmed in MATLAB.

Keywords

Dynamic Time Warping, DTW, variants of DTW, the use of DTW, dynamic programming, MATLAB, signature verification, FAR, FRR, EER, SVC2004 database

Bibliografická citace mé práce:

ČERNOHOUS, L. *Metoda dynamického borcení času*. Brno: Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií, 2016. 39 s. Vedoucí bakalářské práce Ing. Jan Klusáček.

PROHLÁŠENÍ

Prohlašuji, že svou bakalářskou práci na téma „Metoda dynamického borcení času“ jsem vypracoval samostatně pod vedením vedoucího bakalářské 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 bakalářské práce dále prohlašuji, že v souvislosti s vytvořením této bakalářské práce jsem neporušil autorská práva třetích osob, zejména jsem nezasáhl nedovoleným způsobem do cizích autorských práv osobnostních a/nebo majetkových a jsem si plně vědom následků porušení ustanovení S 11 a následujících autorského zákona č. 121/2000 Sb., o právu autorském, o právech souvisejících s právem autorským a o změně některých zákonů (autorský zákon), ve znění pozdějších předpisů, včetně možných trestněprávních důsledků vyplývajících z ustanovení části druhé, hlavy VI. Díl 4 Trestního zákoníku č. 40/2009 Sb.

V Brně

.....
(podpis autora)

Obsah

1 Úvod.....	6
2 Metoda dynamického borcení času.....	7
2.1 Princip algoritmu DTW.....	7
2.2 Algoritmus DTW [1].....	8
2.3 Omezení.....	10
2.3.1 Omezení na lokální souvislost a lokální strmost.....	10
2.3.2 Omezení na hraniční body.....	12
2.3.3 Globální vymezení pohybu bortivé funkce [1], [4].....	13
2.3.4 Omezení strmosti [3], [6].....	15
3 Výpočet vzdálenosti [1], [3].....	17
3.1 Váhová funkce.....	17
3.2 Normalizační faktor.....	18
4 Typy DTW.....	19
5 Využití DTW.....	21
5.1 Zpracování mluvené řeči.....	21
5.2 Aplikace v oblastech počítačového vidění.....	21
5.3 Zpracování biosignálů.....	21
5.4 Gelová elektroforéza.....	21
5.5 Klasifikace těles na základě jejich příčného profilu.....	22
5.6 Genová exprese.....	22
5.7 Data mining.....	22
6 Postup DTW.....	23
7 Závěr.....	24
Seznam použité literatury.....	26

1 Úvod

V praxi se setkáváme s případy, kdy je nutné porovnávat podobnost sekvencí nějakých údajů. Souhrnně lze říci, že tyto případy spojuje potřeba detekce konkrétního jevu, který alespoň přibližně předem známe. Máme tedy nějaký referenční vzorek, se kterým porovnááme zkoumané údaje a měříme míru shody mezi nimi.

Na tomto principu pracuje právě metoda dynamického borcení času (DTW), poměrně stará metoda, která je ale pro svou jednoduchost hojně využívána dodnes. V následujících kapitolách bude popsán základní princip této metody a její varianty vzniklé kombinací různých nastavení jejích parametrů. Dále budou vyjmenovány některé příklady jejích využití v různých oblastech lidské činnosti.

V praktická část této práce nejprve popisuje specifika vybrané oblasti využití – verifikace podpisu. Je zde popsáno sestavení konkrétního modelu, způsob jeho fungování a vyhodnocení jeho výsledků na vybrané databázi.

V závěru jsou shrnuty poznatky získané sestavením a otestováním modelu a vysvětleno jeho chování.

2 Metoda dynamického borcení času

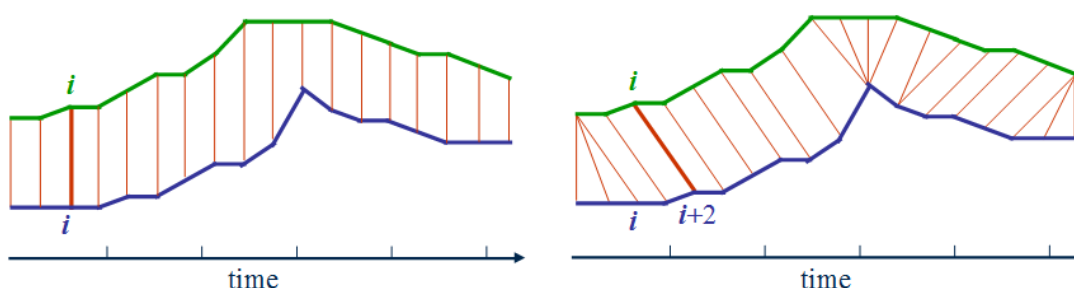
Metoda dynamického borcení času (DTW) je algoritmus využívající principu dynamického programování. Je známá přibližně od padesátých let minulého století, kdy ji objevil a začal popularizovat americký matematik Richard Ernest Bellman [1] [2]. Následně se tato metoda ujala především v oblasti automatické klasifikace slov. Její využití v této oblasti bylo důkladně zkoumáno již od sedmdesátých let [3]. Schopnost této metody efektivně měřit podobnost dvou časových průběhů při současně minimalizaci vlivu vzájemného posunu v čase nebo rychlosti jí zajistila popularitu i v řadě dalších oborů, z nichž některé budou popsány níže.

2.1 Princip algoritmu DTW

Jak již bylo řečeno, tato metoda funguje na principu porovnávání podobnosti dvou časových průběhů. Porovnávané průběhy zpracovávají ve většině úloh v praxi zpravidla nejsou. Mohou být vůči sobě posunuté či roztažené – nejsou zarovnané na časové ose. Vzhledem k tomu, že tyto odchylky nebývají lineární, nelze na ně ani aplikovat lineární porovnávání (např. měřením vzdálenosti odpovídajících si vzorků).

Při lineárním porovnávání například měřením Euklidovské vzdálenosti odpovídajících si bodů (resp. vektorů) můžeme vlivem nestejného zarovnání na časové ose chybně označit dva identické průběhy, které jsou vůči sobě jen posunuté, za nepodobné.

DTW umožňuje nelineárně přizpůsobit tyto průběhy zavedením časově nelineární tzv. *bortivé funkce*. Přizpůsobí buď testovaný časový průběh referenčnímu nebo oba navzájem tak, aby bylo dosaženo co největší shody. Rozdíl mezi lineárním a dynamickým porovnáváním ilustruje Obr. 1. níže.



Obr. 2.1: Rozdíl mezi lineárním a nelineárním (dynamickým) porovnáváním [6]

V rámci měření vzdálenosti u DTW může být též použito Euklidovské vzdálenosti. Rozdíl je ovšem v tom, že nejsou porovnávány (jen) body s odpovídajícím indexem, ale v závislosti na konkrétním nastavení podmínek algoritmu mohou být indexy i rozdílné.

2.2 Algoritmus DTW [1]

Při aplikaci metody DTW porovnáváme jednotlivé prvky (vzorky, příznaky) testovaného průběhu s daným prvkem referenčního průběhu a jeho sousedy. Mechanismus následného výběru nejbližšího prvku ovlivňují konkrétní aplikovaná omezení. Po dokončení výběru se algoritmus přesouvá o krok dál.

Porovnávané časové průběhy zpravidla tvoří posloupnost hodnot, případně posloupnost vektorů. Označme testovaný průběh (zde posloupnost vektorů)

$$\mathbf{A} = \{\mathbf{a}(1), \mathbf{a}(2), \dots, \mathbf{a}(n), \dots, \mathbf{a}(I)\}$$

a referenční průběh

$$\mathbf{B} = \{\mathbf{b}(1), \mathbf{b}(2), \dots, \mathbf{b}(m), \dots, \mathbf{b}(J)\} .$$

Pro úplnost $\mathbf{a}(n)$ reprezentuje n -tý vektor hodnot testovaného průběhu a $\mathbf{b}(m)$ je m -tý vektor hodnot referenčního průběhu. Algoritmus pak lze popsat jako hledání funkce DTW $m = \Psi(n)$, která reprezentuje optimální cestu. Ta minimalizuje funkci D celkové vzdálenosti mezi průběhy \mathbf{A} a \mathbf{B}

$$D(\mathbf{A}, \mathbf{B}) = \sum_{n=1}^I d[\mathbf{a}(n), \mathbf{b}(\Psi(n))] .$$

Vztah $d[\mathbf{a}(n), \mathbf{b}(\Psi(n))]$ je lokální vzdálenost mezi n -tým vektorem testovaného průběhu a m -tým ($m = \Psi(n)$) vektorem referenčního průběhu. Konkrétní vzorec pro výpočet vzdálenosti záleží na konkrétní aplikaci a formě dat. Lze například aplikovat výpočet Euklidovské vzdálenosti, případně Manhattanské a podobně.

Optimální cestu lze tedy vyjádřit vztahem mezi m a n . Je ovšem ještě nutné zavést několik podmínek zabraňujících algoritmu ve zkreslení a následném znehodnocení výsledků. Požadavky jsou následující:

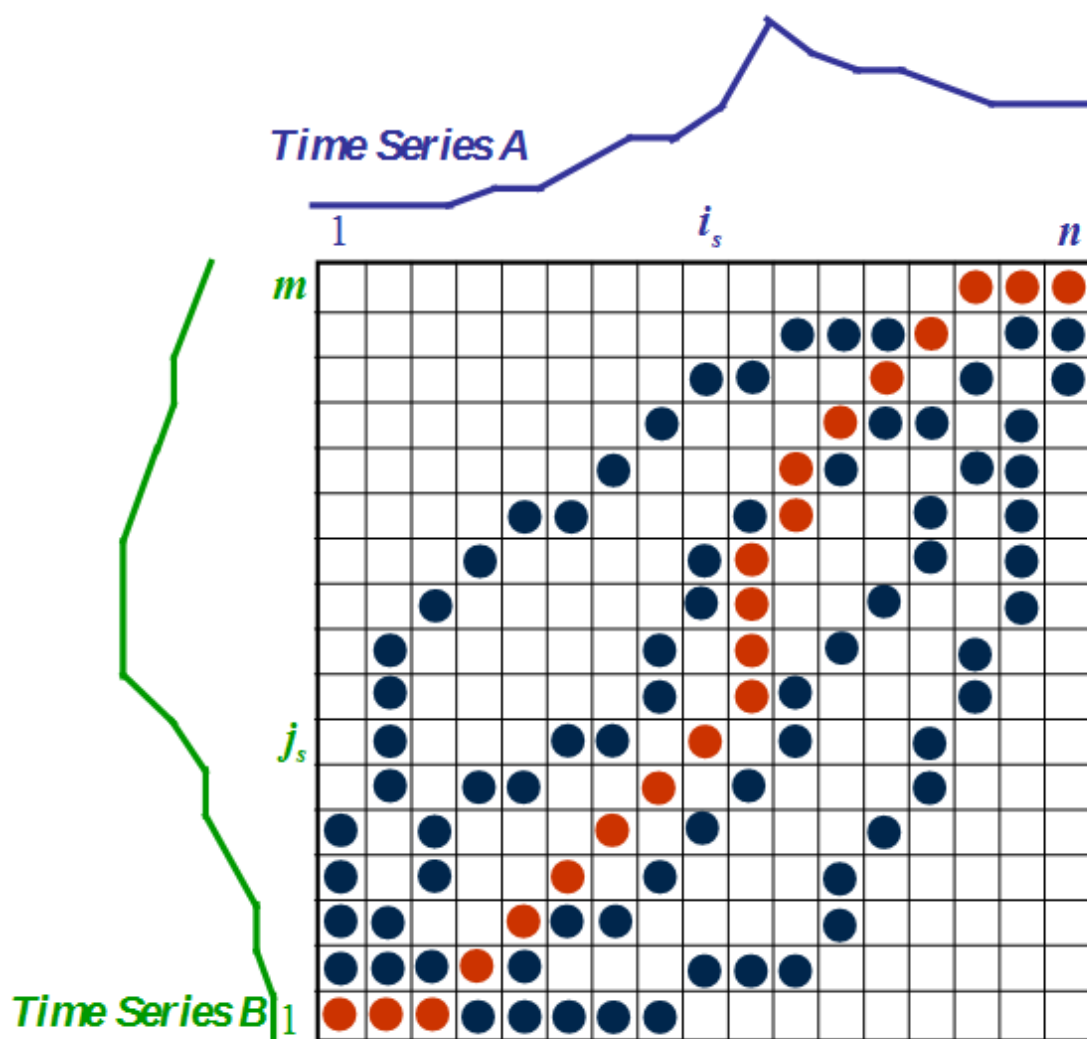
1. je třeba zajistit, aby algoritmus při porovnávání zpětně neopakoval důležité rysy průběhů: cesta se tedy nesmí vracet zpět v čase
2. je nežádoucí, aby algoritmus některé důležité rysy vynechával: cesta nesmí přeskakovat směrem dopředu v čase (tzn. nesmí se přerušit)
3. algoritmus musí brát v potaz oba průběhy v celé délce, ne jen jejich část
4. algoritmus nesmí přeskakovat nepodobné rysy a pozastavovat s u podobných
5. je třeba zabránit mapování velmi krátké části prvního průběhu do velmi dlouhé části průběhu druhého

Aby bylo možné definovat omezení vyplývající z následujících požadavků, je třeba nejdříve zavést obecnou časovou proměnnou k a obě časové proměnné m a n vyjádřit jako funkce k

$$\begin{aligned} n &= i(k), & k &= 1, \dots, K, \\ m &= j(k), & k &= 1, \dots, K, \end{aligned}$$

kde K vyjadřuje délku obecné časové osy.

Následující ilustrace jasně ukazuje, proč je důležité aplikovat na algoritmus určitá omezení: možných cest přes matici je mnoho, takže je možné, že jako optimální se bude jevit cesta, která ve skutečnosti výsledek zkresluje a znehodnocuje tak dosažené výsledky.



Obr. 2.2: Ilustrace možných variací optimální cesty [6]

2.3 Omezení

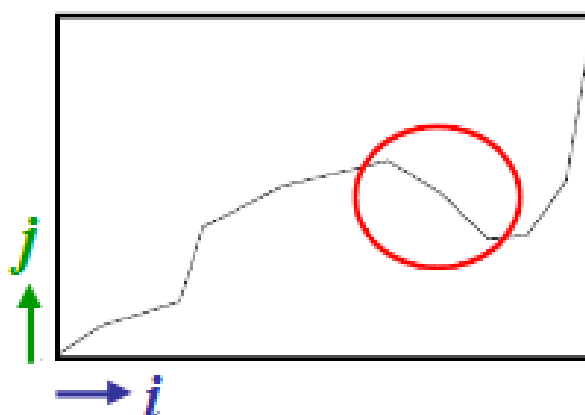
Omezující kritéria je třeba v algoritmu DTW aplikovat především z důvodu zajištění správné funkce porovnávání časových průběhů. Tím, že omezují možnost pohybu optimální cesty, snižují i nároky na výpočetní výkon.

Následující podkapitoly popisují omezení v pořadí daném v předchozí kapitole. Následuje tabulka jednotlivých variací algoritmu DTW v závislosti na aplikovaných omezeních.

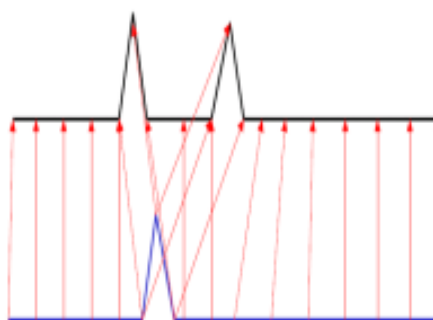
2.3.1 Omezení na lokální souvislost a lokální strmost

První omezení zabráňuje zpětnému vracení v čase a opakování porovnání části průběhu. To se projevuje jako pokles bortící funkce. Omezení tedy musí zaručit, aby funkce byla neklesající. Stanovuje požadavek na *monotónnost*

$$\begin{aligned}i(k-1) &\leq i(k), \\j(k-1) &\leq j(k).\end{aligned}$$



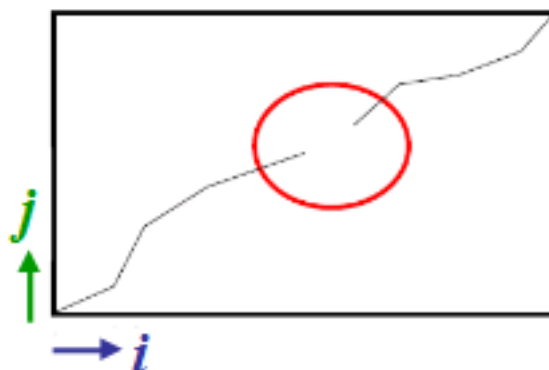
Obr. 2.3: Porušení požadavku na monotónnost [6]



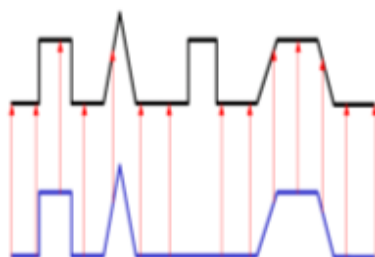
Obr. 2.4: Přeskočení zpět v čase a opětovné vyhodnocení důležitého rysu průběhu [6]

Přeskakování v čase dopředu a následné riziko vynechání důležitých rysů průběhu se projeví přerušáním bortící funkce. Zabránit tomuto jevu má požadavek na *spojitost*

$$\begin{aligned} i(k) - i(k-1) &\leq 1, \\ j(k) - j(k-1) &\leq 1. \end{aligned}$$



Obr. 2.5: Porušení požadavku na spojitost [6]



Obr. 2.6: Přeskočení důležitého rysu průběhu [6]

Kombinací požadavků na monotónnost a spojitost vzniká *omezení na lokální souvislost a lokální strmost*

$$\begin{aligned} 0 &\leq i(k) - i(k-1) \leq I^*, \\ 0 &\leq j(k) - j(k-1) \leq J^*, \end{aligned}$$

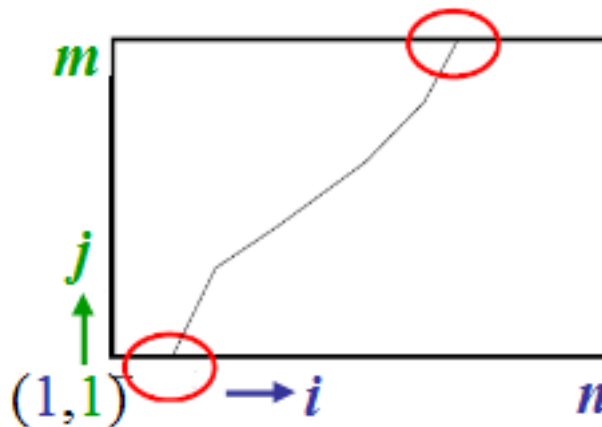
kde obvykle $I^*, J^* = 1, 2, 3$. Pokud $I^*, J^* = 1$, každý vektor se musí minimálně jednou opakovat (protože $i(k) = i(k-1)$ znamená, že se opakuje [5]). Je-li $I^*, J^* > 1$, znamená to, že algoritmus může při porovnání některé vektory vynechat.

2.3.2 Omezení na hraniční body

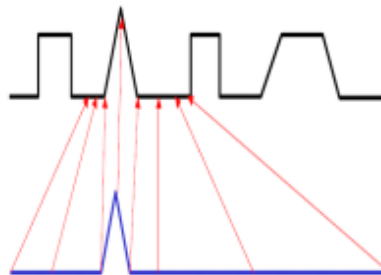
Aby algoritmus bral v potaz celé průběhy a ne jen jejich části, což by mohlo mít za následek umělé zlepšení výsledné vzdálenosti průběhů, je stanoveno *omezení algoritmu na hraniční body*

$$\begin{aligned} i(1) &= 1, & j(1) &= 1, \\ i(k) &= I, & j(k) &= J, \end{aligned}$$

kde K je délka zvolené obecné časové osy pro porovnání průběhů.



Obr. 2.7: Cesta nezačíná ani nekončí v hraničních bodech [6]



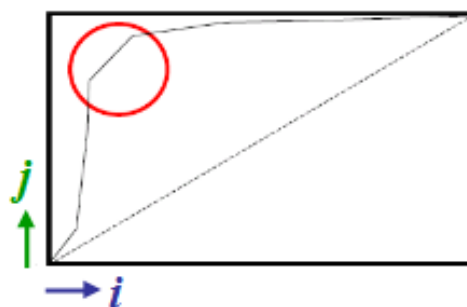
Obr. 2.8: Situace z obrázku 2.7 při porovnávání [6]

Jak je vidět na Obr. 2.7 a 2.8, pokud cesta nezačíná a/nebo nekončí v hraničních bodech $[1,1]$, $[I,J]$, je porovnávána jen část průběhu, zbytek je ignorován. To může opět vést ke zkreslení a následnému znehodnocení výsledku porovnávání.

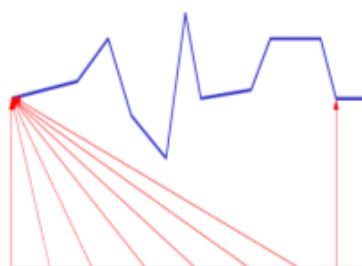
Omezení na hraniční body lze ale aplikovat pouze v případech, kdy porovnáváme průběhy známé délky (např. zpracování izolovaných slov [1]), tedy pokud jsou počáteční i koncové body přesně určeny.

2.3.3 Globální vymezení pohybu bortivé funkce [1], [4]

Pokud algoritmus přeskakuje nepodobné rysy a zbytečně se zdržuje u podobných, projeví se to výrazným vychýlením bortivé funkce od diagonály, která představuje ideální shodu. K zamezení takového chování je třeba globálně vymežit pohyb bortící funkce. K tomu jsou používány především tři hlavní způsoby ohraničení přípustné oblasti pohybu bortivé funkce.



Obr. 2.9: Výrazné vychýlení bortivé funkce od diagonály [6]



Obr. 2.10: Přeskakování nepodobných rysů a zdržování se podobným rysem [6]

Prvním způsobem je vymezení povoleného pásma [3]. Nazývá se též *Sakoe-Chiba band*, podle autorů, kteří jej začali aplikovat. Poprvé byl aplikován na porovnávání slov a vychází z předpokladu, že kolísání tempa řeči se v běžných případech nevyznačuje velkými výchylkami a bortící funkce se tudíž bude pohybovat v rámci určitého pásma podél diagonály. To je definováno vztahem

$$|i(k) - j(k)| \leq w.$$

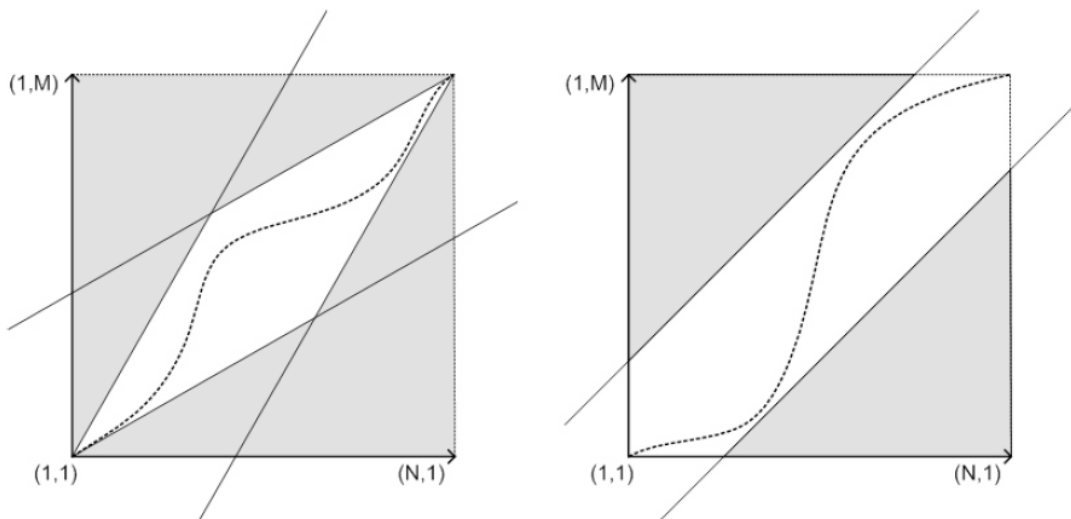
Šířka pásma w je vhodné celé číslo, které ale musí být větší než rozdíl $|I - J|$, aby bylo možné do pásma zahrnout i bod (I, J) . Tuto podmínku lze odstranit vztahem (zavedení směrnice dané poměrem stran):

$$\left| \frac{J}{I} i(k) - j(k) \right| \leq w.$$

Druhá varianta vychází ze *splnění omezení na hraniční body* a vymezuje přípustnou oblast následovně:

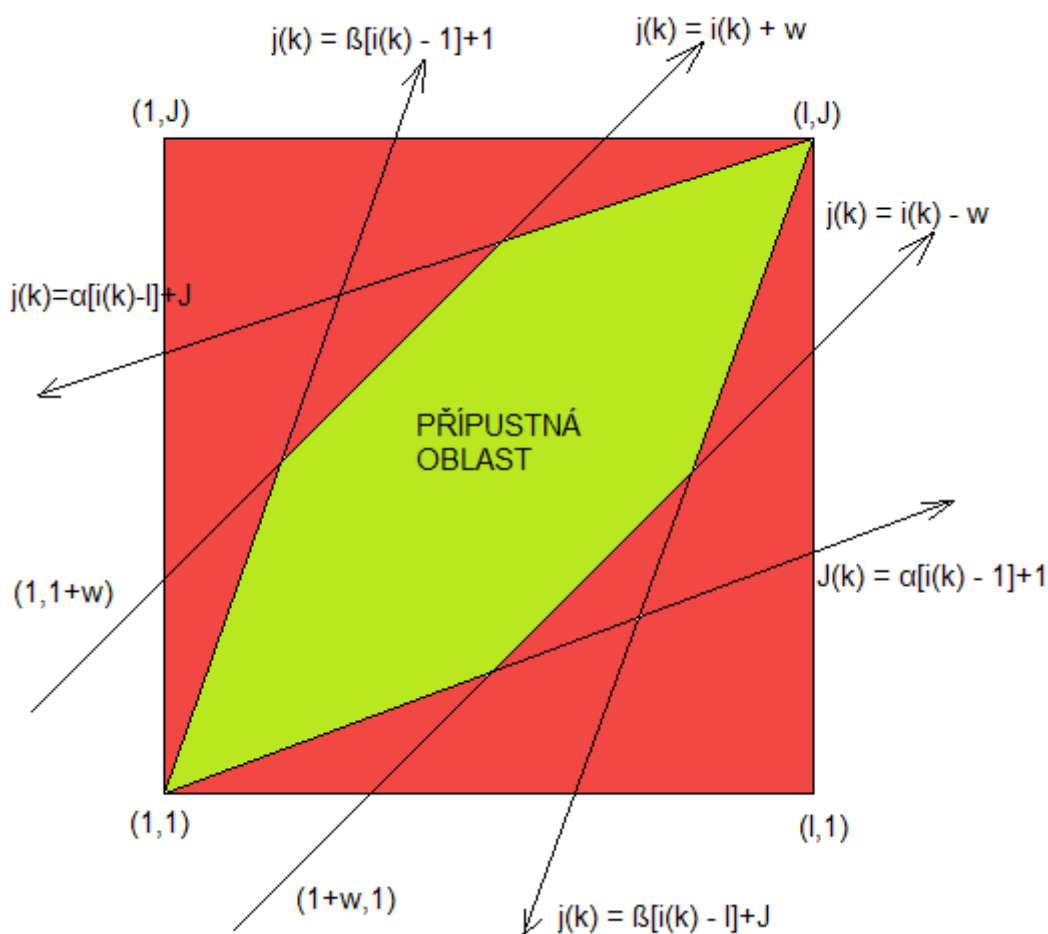
$$\begin{aligned} 1 + \alpha[i(k) - 1] &\leq j(k) \leq 1 + \beta[i(k) - 1], \\ J + \beta[i(k) - I] &\leq j(k) \leq J + \alpha[i(k) - I], \end{aligned}$$

kde α a β představují minimální, resp. maximální směrnice přímky vymezující přípustnou oblast. Někdy je toto vymezení nazýváno *Itakura paralelogram*, rovněž podle autora. Vymezení přípustné oblasti těmito dvěma způsoby ilustruje Obr. 2.11 níže, z Obr. 2.12 pak lze vyčíst příslušnost jednotlivých parametrů a vymezujících přímek.



Obr. 2.11: Základní způsoby globálního vymezení přípustné oblasti [4]

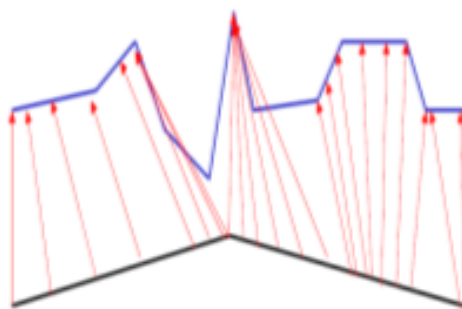
Poslední ze tří způsobů vymezení přípustné oblasti kombinuje předchozí dva. Vymezení tedy určují všechny tři vztahy z předchozích dvou způsobů. Tento způsob vymezení ilustruje Obr. 2.12.



Obr. 2.12: Ilustrace kombinovaného globálního vymezení přípustné oblasti

2.3.4 Omezení strmosti [3], [6]

Pohybuje-li se bortivá funkce příliš dlouho svisle, znamená to, že dochází k nežádoucímu zkreslení mapování velmi krátkého úseku referenčního průběhu do velmi dlouhého úseku testovaného průběhu. Analogicky to platí i při příliš dlouhém horizontálním pohybu. V takových případech je třeba zavést omezení strmosti. Pokud se bod $c(k)$ náležící bortící funkci pohybuje dopředu ve směru jedné z os po m kroků bez přerušení, dosáhne stanoveného maxima kroků a není mu v tomto směru povolen pohyb, dokud neurazí alespoň n kroků v diagonálním směru. Na základě těchto limitů lze potom stanovit míru omezení $P=n/m$. Čím větší tato hodnota je, tím přísnější je omezení. Pro $P=0$ neplatí žádné omezení strmosti, pro $P=\infty$ (při $m=0$) je bortivá funkce omezená pouze na diagonální pohyb. Pokud je tedy omezení příliš přísné, normalizace časové osy nebude efektivní. Na druhou stranu pokud bude omezení příliš uvolněné, zhorší se porovnávací schopnost algoritmu.



Obr. 2.13: Negativní efekt příliš velké strmosti části bortivé funkce [6]

3 Výpočet vzdálenosti [1], [3]

Obecný vztah pro určení celkové minimální vzdálenosti mezi referenčním průběhem A a testovaným průběhem B lze vyjádřit následovně:

$$D(A, B) = \min_{\{i(k), j(k), K\}} \left[\frac{\sum_{k=1}^K d[i(k), j(k)] \hat{W}(k)}{N(\hat{W})} \right],$$

$d[i(k), j(k)]$ je lokální vzdálenost mezi $n=i(k)$ prvkem časového průběhu A a $m=j(k)$ prvkem časového průběhu B , $\hat{W}(k)$ je hodnota váhové funkce pro k -tý úsek bortivé funkce a $N(\hat{W})$ je normalizační faktor, který je funkcí váhové funkce.

3.1 Váhová funkce

Váhová funkce slouží k upřednostnění/penalizaci konkrétních lokálních cest v matici vzdáleností. Závisí pouze na lokální cestě.

Používají se následující typy váhových funkcí:

typ a) symetrická váhová funkce

$$\hat{W}(k) = [i(k) - i(k-1)] + [j(k) - j(k-1)],$$

typ b) asymetrická váhová funkce

$$\begin{array}{ll} b1) & \hat{W}(k) = i(k) - i(k-1), \\ b2) & \hat{W}(k) = j(k) - j(k-1), \end{array}$$

typ c)

$$\hat{W}(k) = \min[i(k) - i(k-1), j(k) - j(k-1)],$$

typ d)

$$\hat{W}(k) = \max[i(k) - i(k-1), j(k) - j(k-1)],$$

přičemž $i(0) = j(0) = 0$.

3.2 Normalizační faktor

Normalizační faktor $N(\hat{W})$ je zaveden kvůli kompenzaci délky nebo počtu kroků bortivé funkce. Obecná definice:

$$N(\hat{W}) = \sum_{k=1}^K \hat{W}(k)$$

Pro jednotlivé typy váhových funkcí potom:

$$\begin{aligned} N(\hat{W}_a) &= \sum_{k=1}^K [i(k) - i(k-1) + j(k) - j(k-1)] = I + J \\ N(\hat{W}_{b1}) &= \sum_{k=1}^K [i(k) - i(k-1)] = I \\ N(\hat{W}_{b2}) &= \sum_{k=1}^K [j(k) - j(k-1)] = J \end{aligned}$$

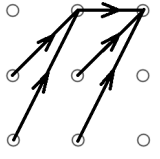
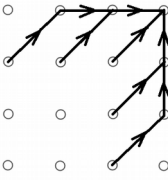
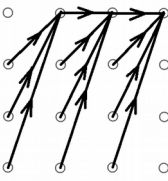
Pro váhové funkce typu c) a d) je normalizační faktor silně závislý na konkrétním průběhu bortivé funkce. Jeho hodnotu lze určit, ale ne rekurzivním algoritmem. Nejlépe se osvědčila volba faktoru nezávisle na průběhu bortivé funkce:

$$N(\hat{W}_c) = N(\hat{W}_d) = I.$$

4 Typy DTW

Na základě kombinací parametrů DTW popsaných výše lze rozlišit sedm základních typů funkce DTW. Souhrnně je popisuje následující tabulka [1].

Typ DTW	Typy lokálních omezení cesty	α	β	Typ $\hat{W}(k)$	Funkce částečné kumulované vzdálenosti $g(n, m)$
I		0	∞	a	$\min \begin{bmatrix} g(n, m-1) + d(n, m) \\ g(n-1, m-1) + 2d(n, m) \\ g(n-1, m) + d(n, m) \end{bmatrix}$
				b	$\min \begin{bmatrix} g(n, m-1) + d(n, m) \\ g(n-1, m-1) + d(n, m) \\ g(n-1, m) + d(n, m) \end{bmatrix}$
II		1/2	2	a	$\min \begin{bmatrix} g(n-1, m-2) + 3d(n, m) \\ g(n-1, m-1) + 2d(n, m) \\ g(n-2, m-1) + 3d(n, m) \end{bmatrix}$
				c	$\min \begin{bmatrix} g(n-1, m-2) + d(n, m) \\ g(n-1, m-1) + d(n, m) \\ g(n-2, m-1) + d(n, m) \end{bmatrix}$
III		1/2	2	a	$\min \begin{bmatrix} g(n-1, m-2) + 2d(n, m-1) + d(n, m) \\ g(n-1, m-1) + 2d(n, m) \\ g(n-2, m-1) + 2d(n-1, m) + d(n, m) \end{bmatrix}$
IV		1/2	2	b1	$\min \begin{bmatrix} g(n-1, m) + \kappa d(n, m) \\ g(n-1, m-1) + d(n, m) \\ g(n-1, m-2) + d(n, m) \end{bmatrix}$ $\kappa=1$ pro $j(k-1) \neq j(k-2)$ $\kappa=\infty$ pro $j(k-1) = j(k-2)$

V		1/2	2	d	$\min \begin{bmatrix} g(n-1, m-2) + 2d(n, m) \\ g(n-1, m-1) + d(n, m) \\ g(n-2, m-2) + 2d(n, m) + d(n, m) \\ g(n-2, m-1) + d(n-1, m) + d(n, m) \end{bmatrix}$
VI		1/3	3	a	$\min \begin{bmatrix} g(n-1, m-3) + 2d(n, m) + d(n, m-1) + d(n, m) \\ g(n-1, m-2) + 2d(n, m-1) + d(n, m) \\ g(n-1, m-1) + 2d(n, m) \\ g(n-2, m-1) + 2d(n-1, m) + d(n, m) \\ g(n-3, m-1) + 2d(n-2, m) + d(n-1, m) + d(n, m) \end{bmatrix}$
VII		1/3	3	a	$\min \begin{bmatrix} g(n-1, m-3) + 4d(n, m) \\ g(n-1, m-2) + 3d(n, m) \\ g(n-1, m-1) + 2d(n-2, m) \\ g(n-2, m-2) + 4d(n-1, m) + d(n, m) \\ g(n-2, m-2) + 3d(n-1, m) + d(n, m) \\ g(n-2, m-1) + 2d(n-1, m) + d(n, m) \\ g(n-3, m-3) + 4d(n-2, m) + d(n-1, m) + d(n, m) \\ g(n-3, m-3) + 3d(n-2, m) + d(n-1, m) + d(n, m) \\ g(n-3, m-1) + 2d(n-2, m) + d(n-1, m) + d(n, m) \end{bmatrix}$

5 Využití DTW

Metoda dynamického borcení času (nebo časové osy) se nabízí k využití ve všech případech, kdy je potřeba porovnat dvě sekvence signálu. Využití je pak dvojí: může jít přímo o finální porovnání s výběrem (knihovnou) známých, resp. žádaných sekvencí nebo je využita schopnost metody DTW synchronizovat sekvence v rámci předzpracování signálu pro jiné formy detekce/zpracování.

5.1 Zpracování mluvené řeči

Automatická klasifikace slov v mluveném projevu je vůbec prvním případem praktické implementace Metody dynamického borcení času. Od sedmdesátých let minulého století se touto problematikou zabývali například Hiroaki Sakoe a Seibi Chiba [3], kteří na toto téma vydali několik studií zabývajících se především vlivem aplikace různých kombinací omezujících kritérií algoritmu DTW.

5.2 Aplikace v oblastech počítačového vidění

V této oblasti metoda DTW našla uplatnění například v rozpoznávání gest ruky. Jednou z konkrétních aplikací tohoto zaměření je pak klasifikace znakové řeči.

Další aplikací metody DTW je klasifikace psaného slova [11]. Speciálním případem je v této oblasti pak rozeznávání podpisů.

5.3 Zpracování biosignálů

V tomto oboru je zpravidla nutné identifikovat nějaký vzor ve zpracovaném signálu. Ať už jde o vyhodnocování EKG či EEG signálu, vždy je možné aplikovat buď přímo metodu DTW jako samotný klasifikátor nebo ji využít „pouze“ jako nástroj pro synchronizaci sekvencí před jejich samotnou klasifikací například shlukovou analýzou. [13], [14]

5.4 Gelová elektroforéza

Elektroforéza obecně označuje množinu separačních metod využívajících ke klasifikaci chemických látek jejich odlišnou pohyblivost ve stejnosměrném elektrickém Poli. Gelová elektroforéza je pak metoda pro separaci a analýzu makromolekul (DNA, RNA, proteinů...) v závislosti na jejich velikosti, uspořádání a náboji. Využívá gelu o různé velikosti pórů, skrze které se fragmenty makromolekul pohybují.

Metoda dynamického borcení času je zde využita jako nástroj pro synchronizaci porovnávaných sekvencí v závěrečné fázi předzpracování. Pro konečné zpracování je použita shluková analýza. [15]

5.5 Klasifikace těles na základě jejich příčného profilu

Zde je metoda DTW využívána například ke klasifikaci projíždějících vozidel na základě jejich shora snímaného profilu. Snímání může být provedeno různými metodami (laserem, infračerveně, ...). Výstupem měření je zpravidla údaj o výšce profilu v časové sekvenci tak, jak vozidlo projíždí snímaným průřezem. Metoda DTW je zde použita jako finální klasifikátor [12].

5.6 Genová exprese

Cílem je vyhodnotit sekvenci hodnot reprezentujících sled nukleotidů Adeninu, Guaninu, Cytosinu a Thyaminu v různých kombinacích. Metoda DTW zde může být použita jako přímý klasifikátor i jako nástroj pro synchronizaci sekvencí. Jednotlivé hodnoty mohou být reprezentovány několika způsoby, zvolený postup je však stejný [7].

5.7 Data mining

Dolování sekvenčních dat neboli data mining je označení pro dobývání znalosti z databází potenciálně užitečných dat. Metoda DTW je v této problematice aplikována a diskutována jako možné zrychlení [8], [9].

5.8 Verifikace podpisu

Specifika této oblasti využití přiblíží praktická část. Zpravidla je v této oblasti používána nejjednodušší varianta DTW [16], [17], tedy varianta I z tabulky v kap. 4. Existují modifikace základního algoritmu, např. [18] pro zvýšení efektivity off-line verifikace, také [19] pro práci s podpisy načtenými s rozdílnou (známou) Vzorkovací frekvencí:

$$D(i, j) = \min \begin{cases} g(i-1, j-(F-1)) + d(r_i, q_i) \\ g(i-1, j-F) + d(r_i, q_i) \\ g(i-1, j-(F+1)) + d(r_i, q_i) \end{cases}$$

Základní dělení verifikace podpisu je na on-line a off-line [18]. Off-line verifikace spočívá v posouzení hotového podpisu (např. naskenovaného z dokumentu). On-line verifikace pracuje s daty získanými přímo za účelem zpracování, má tedy možnost pracovat s podstatně širší škálou hodnot (parametrů).

6 Postup DTW

Za předpokladu, že normalizační faktor je nezávislý na průběhu bortivé funkce, lze vztah pro výpočet celkové minimální vzdálenosti zjednodušit:

$$D(\mathbf{A}, \mathbf{B}) = [N(\hat{W})]^{-1} \left\{ \min_{\{(k), j(k), K\}} \sum_{k=1}^K d[i(k), j(k)] \hat{W}(k) \right\}$$

Takto definovaná vzdálenost je již vhodná k rekurzivnímu vyčíslení pomocí dynamického programování. Celý postup lze shrnout následovně:

1. *Matici lokálních vzdáleností* o velikosti $n \times m$ naplníme odpovídajícími hodnotami lokálních vzdáleností.
2. Vytvoříme si *matici kumulovaných vzdáleností*, která má oproti předchozí matici navíc nultý řádek a nultý sloupec. Tento řádek a sloupec kromě bodu (0,0) naplníme hodnotami ∞ , v bodě (0,0) bude nula.
3. Matici kumulovaných vzdáleností inicializujeme a začneme plnit hodnotami rekurzivně:

$$\text{Inicializace: } g[i(1), j(1)] = d[i(1), j(1)] \hat{W}(1)$$

$$\text{Rekurze: } g[i(k), j(k)] = \min_{\{i(k), j(k)\}} \{g[i(k-1), j(k-1)] + d[i(k), j(k)] \hat{W}(k)\}.$$

4. Konečná normalizovaná vzdálenosti

$$D(\mathbf{A}, \mathbf{B}) = [N(\hat{W})]^{-1} g[i(K), j(K)] = [N(\hat{W})]^{-1} g[I, J].$$

Dosažením jedné z váhových funkcí do vzorce pro rekurzi lze odvodit i potřebné rekurzivní vztahy k vyčíslování funkce DTW (poslední sloupec tabulky v předchozí kapitole).

7 Praktická část

Pro sestavení klasifikačního modelu pracujícího na principu metody Dynamického borcení času byla vybrána verifikace podpisu. Jde o vcelku specifické využití, jelikož v tomto modelu nejde přímo o klasifikaci (zařazení) testované sekvence do několika skupin, nýbrž o stanovení míry podobnosti testovaného podpisu s referenční skupinou podpisů (případně i jediným podpisem). Tato míra podobnosti je následně porovnávána s prahovou hodnotou, která rozhoduje o pravosti testovaného podpisu.

7.1 Databáze

Pro sestavení a zhodnocení modelu verifikace podpisu byla vybrána databáze SVC2004 dostupná bez nutnosti požádání na webu [21]. Sestává se ze dvou složek *Task1* a *Task2*, v nichž se nachází jednotlivé TXT soubory s daty.

Obě složky obsahují podpisy 40 uživatelů, od každého 40 podpisů. Celkem tedy 1600 souborů. Pro rozlišení jednotlivých podpisů jsou soubory pojmenovány následující formou: U1..40S1..40 (tedy např. *U2S39.txt*). Číslo za ‚U‘ identifikuje uživatele, číslo za ‚S‘ identifikuje konkrétní podpis. Jednotlivé vzorky sekvencí v každém textovém souboru jsou v řádcích, jednotlivé hodnoty oddělené mezerou. V prvním řádku každého souboru je údaj o počtu vzorků (řádků) dané sekvence, od druhého řádku dál již následují samotné hodnoty. Databáze byla sestavena tak, že prvních 20 podpisů jsou originály, dalších dvacet trénované podvrhy.

Ve sloupcích každého souboru jsou tedy jednotlivé zaznamenané veličiny každého podpisu. Jejich počtem se databáze složky *Task1* a *Task2* liší. Soubory v *Task1* obsahují hodnoty polohy hrotu souřadnice x a y , dále pak údaj o čase *time stamp* a údaj o kontaktu pera se záznamovým povrchem *button status*. Soubor těchto hodnot je vhodný pro off-line verifikaci.

Složka *Task2* obsahuje databázi souborů podpisů reprezentovaných sedmi hodnotami: souřadnice x a y , údaj o čase *time stamp* a údaj o kontaktu pera se záznamovým povrchem *button status*, *azimuth* (natáčení pera okolo osy z , orientace stupnice ve směru hodinových ručiček), *altitude* (náklon pera vůči ose z) a *pressure* neboli tlak (na podložku).

7.2 Verifikační model

Sestavení modelu pracujícího na principu dynamického borcení času lze rozdělit na několik fází. Základní je výběr varianty DTW. Jak již bylo zmíněno v kap. 5.8, nezřídka se využívají nebo přímo i vytváří další možné mutace algoritmu.

Dále je nutné vybrat databázi, na které bude model postaven. Databáze musí obsahovat jak množinu originálních podpisů (angl. *Genuine*) tak i množinu trénovaných padělků (angl. *Skilled forgery*).

Sada veličin poskytnutých zvolenou databází lze zpravidla ještě rozšířit o veličiny od těchto odvozené. Více veličin k porovnání může znamenat větší přesnost, ale zároveň jistě znamená i větší výpočetní zátěž a při výběru veličin nepodstatných či zkreslujících charakteristiku podpisu může dojít i k nežádoucímu chování modelu.

Rovněž lze předpokládat, že hodnoty v databázi nejsou nijak předzpracovány. To může způsobit nežádoucí zkreslení míry podobnosti podpisu (např. posun originálního podpisu v souřadném systému) a jeho následné chybné odmítnutí. Zároveň takový rozptyl hodnot zkreslí vzájemné podobnosti v referenční množině a tím může umožnit chybné přijetí padělku za originál.

Výběru algoritmu DTW a databáze vede k dalšímu bodu tvorby verifikačního modelu: stanovení strategie pro hodnocení. Spočívá v rozdělení množiny originálů na referenční a testovací podpisy a ve výběru (resp. způsobu výběru) množiny trénovaných padělků pro vyhodnocení úspěšnosti modelu.

Je-li stanoveno vše výše uvedené, zbývá ještě zvolit metodiku výpočtu podobnosti testovaného podpisu k referenční množině a definovat kritérium, které rozhodne, zda je testovaný podpis pravý či nikoliv. Na závěr se v rámci hodnocení chování modelu stanoví jeho úspěšnost.

Samotný algoritmus DTW, vzorec výpočtu lokální vzdálenosti a předzpracování dat včetně výběru vhodných veličin je inspirováno [16], výběr referenční množiny a metodika pro stanovení podobnosti pochází z [17] a rozdělení zbylých podpisů pro testování bylo stanoveno tak, aby byl využit plný potenciál databáze.

7.3 Zvolený algoritmus DTW

K výpočtu vzdálenosti dvou sekvencí je využit základní algoritmus DTW, bez vah (resp. s váhami rovnými 1) a bez omezení cesty, jako lokální vzdálenost slouží Euklidovská vzdálenost [16] (jako samostatná funkce). Algoritmus výpočtu DTW je naprogramován tak, že není třeba plnit celou matici lokálních vzdáleností, potřebná lokální vzdálenost se počítá zvlášť. Celý postup výpočtu DTW je zahrnut do funkce *DTW_bak()*, aby bylo možné jej opakovat ve smyčkách pro hromadné výpočty. Správnost fungování algoritmu vyla před jeho začleněním do modelu overěna pomocí vlastní matlabovské funkce *dtw()*.

```

function [ dtw_dist ] = DTW_bak(r, t, band, w)
%DTW_bak    Vzdálenost pomocí algoritmu DTW (Dynamic Time Warping)
mezi referenční sekvencí ref_v a testovanou
%test_v. Výsledná hodnota reprezentuje nepodobnost těchto sekvencí,
%tedy
%čím menší hodnota, tím větší podobnost.

%kontrola vstupních argumentů
switch nargin
    case 2
        w = [1,1,1];
        band = max(length(r), length(t));

    case 3
        if band<=0
            error('DTW_bak(): šířka pásma musí být větší než 0!');
        else
            band = max(length(r), length(t));
        end

    case 4
        if band<=0
            error('DTW_bak(): šířka pásma musí být větší než 0!');
        else
            if length(w)~=3
                error('DTW_bak(): vektor w musí obsahovat 3
hodnoty!');
            end
        end

    otherwise
        if nargin<2
            error('DTW_bak(): minimální počet vstupních hodnot je
2!');
        else
            error('DTW_bak(): příliš mnoho vstupních hodnot!');
        end
end

%kontrola rozměrů porovnávaných sekvencí
[n, vr] = size(r);
[m, vt] = size(t);

if vr ~= vt
    error('DTW_bak(): Délka vektorů porovnávaných sekvencí musí
souhlasit!');
end

```

```

%vytvoření a inicializace matice kumulovaných vzdáleností
DTW = ones(n+1, m+1) + Inf;
DTW(1,1) = 0;

%plnění oblasti matice DTW vymezené šířkou pásma
%vzdálenost vektorů je euklidovská vzdálenost
%sqrt( sum( (r(j-1,:) - t(i-1,:)).^2 ) )
for i = 2:(n+1)
    for j = max( 2,ceil((m/n)*(i-band)) ) :
min( m+1,ceil((m/n)*(i+band)) )
        DTW(i,j) = sqrt(sum((r(i-1,:) - t(j-1,:)).^2)) +
                    min((w(1)*DTW(i-1,j)),
                        min( (w(2)*DTW(i,j-1)),
                            (w(1)*DTW(i-1,j-1)) ) ) );
    end
end

% výsledná hodnota DTW

dtw_dist = DTW(end, end);

end %fcn DTW_bak

```

Z výše zobrazeného zdrojového kódu funkce *DTW_bak()* je zřejmé, že algoritmus dokáže pracovat i s vymezením pásma pohybu bortivé funkce a případně i s různým nastavením vah. Tyto vlastnosti byly do algoritmu zavedeny pro názornost v rámci bodu 2 zadání této práce.

7.4 Databáze a načítání dat

Na základě poznatků z [16] byla zvolena databáze *Task1* a z ní následně vybrány hodnoty (x, y, dx, dy, p) pro verifikační model. O načtení databáze do pracovního prostoru MATLABu se stará skript `main_load_norm.m`, který data načítá a rovněž se stará o jejich předzpracování, výpočet odvozených veličin a výběr vhodné sady veličin pro model. Níže je ukázka automatického načtení ve smyčce. Všechny tři následující ukázky patří do stejného skriptu (uveden výše).

```

cd Task2;
%seznam TXT souborů ve složce
soub = ls('*.txt');

%načtení TXT souborů do struktury 'a'
for i=1:length(soub)
    [pathstr,name,ext] = fileparts(soub(i,:));
    db.(name) = dlmread(soub(i,:), ' ');
end
%návrat a úklid
cd ..
clear soub pathstr name ext

```

7.4.1 Předzpracování [16]

Na vstupní veličiny (x, y, p) je aplikována geometrická translace (geometrický střed je pak v počátku) a statistická normalizace. Obojí je zřetelně vidět v následující části kódu:

```
% zajištění nulového průměru a jednotkové variance
-----

%geometrická translace a statistická normalizace
-----
%všechny hodnoty X a Y se posunou, aby střed podpisu ležel v počátku,
%normalizuje se i vektor přítlaku (obojí průměrnou hodnotou)

for i=1:40          %číslo uživatele
  for j=1:40        %číslo podpisu
    px = cat(2, 'U', num2str(i), 'S', num2str(j));
    xx = db.(px);

    %geom. trans.
    mean_x = sum(xx(2:end,1)) / (length(xx)-1);
    mean_y = sum(xx(2:end,2)) / (length(xx)-1);
    mean_p = sum(xx(2:end,7)) / (length(xx)-1);
    for k=2:length(xx)
      xx(k,1) = xx(k,1) - mean_x;
      xx(k,2) = xx(k,2) - mean_y;
      xx(k,7) = xx(k,7) - mean_p;
    end

    %stat. norm.
    sigma_x = sqrt(
      sum( (xx(2:end,1)-mean_x).^2) / (length(xx)-2)
    );
    sigma_y = sqrt(
      sum( (xx(2:end,2)-mean_y).^2) / (length(xx)-2)
    );
    sigma_p = sqrt(
      sum( (xx(2:end,7)-mean_p).^2) / (length(xx)-2)
    );
    for k=2:length(xx)
      xx(k,1) = (xx(k,1) - mean_x)/sigma_x;
      xx(k,2) = (xx(k,2) - mean_y)/sigma_y;
      xx(k,7) = (xx(k,7) - mean_p)/sigma_p;
    end

    db.(px) = xx;
  end
end

% konec normalizace
-----
```

7.4.2 Získání užitečných databáze

V této smyčce skriptu je ze struktury obsahující načtená a upravená data vytvářena nová struktury a částí původních dat a novými odvozenými daty. Během tvorby a ladění modelu se vyskytl problém s hodnotou dx nebo dy rovnou nekonečnu (kladnému nebo zápornému). Mohly za to dvě stejné po sobě jdoucí hodnoty veličiny *timestamp*, což vyvolalo dělení nulou. Tento problém obchází náhrada nuly ve jmenovateli dostatečně malým číslem.

```
% získání užitečných dat
-----

for i=1:40          %číslo uživatele
  for j=1:40        %číslo podpisu
    px = cat(2, 'U', num2str(i), 'S', num2str(j));
    xx = db.(px);
    yy = zeros(length(xx)-1,5);
    yy(1:end,1:2) = xx(2:end,1:2); %kopírování sloupců x a y
    yy(1:end,5) = xx(2:end,7);     %kopírování sloupce p
    for k=2:length(yy)

      %výpočet dx/dt
      dt = (xx(k+1,3) - xx(k,3));
      %dt=1;          %vyloučení hodnoty "timestamp"
      if dt == 0      %korekce proti Inf
        dt = 0.001;
      end
      yy(k,3) = ( yy(k,1) - yy(k-1,1) ) /dt;

      %výpočet dy/dt
      yy(k,4) = ( yy(k,2) - yy(k-1,2) ) /dt;
    end

    %vložení do struktury zpracovaných dat d
    d.(px) = yy;

  end
end
```

7.5 Tvorba modelu, pravidla pro verifikaci

Dle [17] je pro každého uživatele vytvořen model. Referenční množina podpisů je tvořena prvními pěti podpisy jmenovatele a mezi všemi navzájem jsou stanoveny vzdálenosti pomocí DTW. Aritmetický průměr dvou nejmenších je $avrMin$, pro dvě největší vzdálenosti je to $avrMax$.

Pro testovaný podpis jsou vypočítány vzájemné vzdálenosti a největší je přiřazena $SMax$, nejmenší $SMin$. Následně jsou vypočítány hodnoty:

$$diffMin = \frac{SMin - avrMin}{avrMin}$$

$$diffMax = \frac{SMax - avrMax}{avrMax}$$

Podpis je považován za originál, splňuje-li:

$$diff_{total} = \Sigma (diffMin + diffMax) \leq Th$$

kde Th je práh.

Výše uvedený postup lze udělat pro každou veličinu podpisu zvlášť a pak je sečíst. Rovněž je možné posuzovat jeden podpis jako celou sekvenci. Vyzkoušeno bylo obojí, následuje ukázka prvního případu:

```
%verifikace na základě DTW všech parametrů (x, y, dx/dt, dy/dt, p)
(zvlášť, pak skombinovat)
```

```
total_v = zeros(40, 35);
```

```
for i=1:40                                %uživatel
```

```
%-----
-
% tvorba referenčního modelu uživatele
-----
```

```
avg_mat = zeros(2, 5); %matice avrMin a avrMax (1. resp. 2.
řádek) pro k (sloupec)
```

```
for k=1:5                                %veličina
    d_vect = [];
```

```
    for a=1:4                            %podpis a
        pom = zeros(1, 5-a);
```

```

ind = 0;

for b = (a+1):5      %podpis b
    ind = ind+1;

    pa = cat(2, 'U', num2str(i), 'S', num2str(a));
    xa = d.(pa);
    vect_A = xa(1:end,k);

    pb = cat(2, 'U', num2str(i), 'S', num2str(b));
    xb = d.(pb);
    vect_B = xb(1:end,k);

    pom(ind) = DTW_bak(vect_A, vect_B);

end

d_vect = cat(2, d_vect, pom);

end

d_vect = sort(d_vect);
avg_mat(1,k) = (d_vect(1) + d_vect(2)) /2;      %avrMin pro k
avg_mat(2,k) = (d_vect(9) + d_vect(10)) /2;    %avrMax pro k

end

% konec tvorby referenčního modelu uživatele
-----

%-----
% testování ostatních podpisů uživatele
-----

for b=6:40

    S_mat = zeros(2,5);
    diff_mat = zeros(2,5);

    for k=1:5      %veličina
        s_vect = zeros(1,5);

        for a=1:5

            %referenční podpis
            pa = cat(2, 'U', num2str(i), 'S', num2str(a));
            xa = d.(pa);
            vect_A = xa(1:end,k);

```



```

        %testovaný podpis
        pb = cat(2, 'U', num2str(i), 'S', num2str(b));
        xb = d.(pb);
        vect_B = xb(1:end,k);

        s_vect(a) = DTW_bak(vect_A, vect_B);
    end

    S_mat(1,k) = min(s_vect);      %SMin pro k
    S_mat(2,k) = max(s_vect);      %SMax pro k

end

%celkový rozdíl od reference
diff_tot = 0;

for k=1:5
    diff_mat(1,k) = ( S_mat(1,k) - avg_mat(1,k) ) /
avg_mat(1,k);
    diff_mat(2,k) = ( S_mat(2,k) - avg_mat(2,k) ) /
avg_mat(2,k);

    diff_tot = diff_tot + diff_mat(1,k) + diff_mat(2,k);
end

total_v(i,b-5) = diff_tot;
end

% konec testování ostatních podpisů uživatele
-----

%-----
-

end      %konec vyhodnocování

%úklid
clear a avg_mat b d_vect diff_mat diff_tot i ind k pa pb pom S_mat
s_vect;
clear vect_A vect_B xa xb;

```

8 Vyhodnocení

Na základě změny prahové hodnoty byly pro všechny uživatele a podpisy vyčísleny následující:

FAR... chybné přijetí (poměr chybně přijatých podvrhů za originál ku počtu všech testovaných podvrhů)

FRR... chybné odmítnutí (poměr chybně odmítnutých originálů jako podvrhy ku celkovému počtu vzorů)

EER coby hodnota, kdy FAR = FRR je stanovena následujícím skriptem:

```
prah = 0:0.01:1;
FRR = zeros(1, length(prah));
FAR = zeros(1, length(prah));
verif = zeros(40,35);

for ind=1:length(prah)
    for i=1:40      %uživatel
        %stanovení rozdílu
        lo = min(total_v(i, 1:end));
        hi = max(total_v(i, 1:end));
        dif = hi-lo;

        for j=1:35      %podpis
            %verifikace: 1 = originál, 0 = podvrh
            if total_v(i,j) <= ( (dif*prah(ind))+lo )
                verific(i,j) = 1;
            end
        end
    end
end

% rozdělení pravdivostní matice verifikace na originály a podvrhy
mat_orig = verific(1:end,1:15);
mat_forg = verific(1:end,16:end);

%výpočet FAR
nR = (40*15) - sum(sum(mat_orig));
FAR(ind) = nR / (40*15);

%výpočet FRR
nA = sum(sum(mat_forg));
FRR(ind) = nA / (40*20);
end

%výpočet EER (průsečík dvou úseček)
ind=1;
while FAR(ind)>FRR(ind)
    ind = ind+1;
```

```
end

%přibližně:
EER = ( FAR(ind-1) + FAR(ind) )/2
prah_EER = ( prah(ind-1) + prah(ind) )/2

%vykreslení křivek FAR a FRR
figure();
plot(prah, FRR, prah, FAR);

%úklid
clear ind i lo hi dif j mat_orig mat_forg nR nAmprah FRR FAR verif
```

9 Závěr

Metoda dynamického borcení času (DTW) je již poměrně starou metodou, ale svou jednoduchostí a efektivitou (při správném nastavení) je i dnes hojně využívána. Velké popularitě se těší především v oboru rozpoznávání slov a mluvené komunikace s počítačem coby samotný klasifikátor, široké využití nachází i coby nástroj předzpracování sekvencí signálu v případech, kdy je před jejich klasifikací danou metodou nutné tyto sekvence synchronizovat na základě společných znaků. Mimo konkrétních případů zmíněných v této práci existuje i řada dalších aplikací této metody. Většinou jde ale o jiné využití téhož algoritmu v téměř identické podobě (s ohledem na specifika v podobě jiného formátu reprezentace dat či například volnějším požadavkům na omezující kritéria algoritmu).

Naprogramování algoritmu DTW v rámci splnění úkolu 2 této práce pro názornost naprosto postačuje. V této formě je pravděpodobně nejvyužívanější a v případě potřeby jej lze snadno upravit na jinou variantu z tabulky v kapitole 4 prostým přepsáním části, kde je vybírána minimální cesta z předchozích prvků matice kumulovaných vzdáleností. V případě výběru některé ze složitějších variant je samozřejmě třeba ještě přizpůsobit inicializaci matice kumulovaných vzdáleností – přidat řádky a sloupce vyplněné hodnotou *Inf* tak, aby korespondovaly s rozšířením hledání minima na vzdálenější prvky matice.

Při verifikaci prvním způsobem zmíněným v kap. 7.9 vyšlo EER 32% druhým způsobem 24,42%.

Tento výsledek je způsobem velkou podobností originálů a podvrhů, což vysvětluje [20] jako chybu při tvorbě databáze.

Seznam ilustrací

<i>Obr. 2.1: Rozdíl mezi lineárním a nelineárním (dynamickým) porovnáváním [6].....</i>	<i>4</i>
<i>Obr. 2.2: Ilustrace možných variací optimální cesty [6].....</i>	<i>6</i>
<i>Obr. 2.3: Porušení požadavku na monotónnost [6].....</i>	<i>7</i>
<i>Obr. 2.4: Přeskočení zpět v čase a opětovné vyhodnocení důležitého rysu průběhu [6].</i>	<i>7</i>
<i>Obr. 2.5: Porušení požadavku na spojitost [6].....</i>	<i>8</i>
<i>Obr. 2.6: Přeskočení důležitého rysu průběhu [6].....</i>	<i>8</i>
<i>Obr. 2.7: Cesta nezačíná ani nekončí v hraničních bodech [6].....</i>	<i>9</i>
<i>Obr. 2.8: Situace z obrázku 2.7 při porovnávání [6].....</i>	<i>9</i>
<i>Obr. 2.9: Výrazné vychýlení bortivé funkce od diagonály [6].....</i>	<i>10</i>
<i>Obr. 2.10: Přeskakování nepodobných rysů a zdržování se podobným rysem [6].....</i>	<i>10</i>
<i>Obr. 2.11: Základní způsoby globálního vymezení přípustné oblasti [4].....</i>	<i>11</i>
<i>Obr. 2.12: Ilustrace kombinovaného globálního vymezení přípustné oblasti.....</i>	<i>12</i>
<i>Obr. 2.13: Negativní efekt příliš velké strmosti části bortivé funkce [6].....</i>	<i>13</i>

Seznam použité literatury

- [1] PSUTKA, Josef. *Komunikace s počítačem mluvenou řečí*. Vyd. 1. Praha: Academia, 1996, 287 s. ISBN 80-200-0203-0.
- [2] BELLMAN, Richard E. *Dynamic programming*. New Jersey: Priceton, 1957, 339 s.
- [3] SAKOE, H. a S. CHIBA. Dynamic programming algorithm optimization for spoken word recognition. *IEEE Transactions on Acoustics, Speech, and Signal Processing*. 1978, vol. 26, issue 1, s. 43-49. DOI: 10.1109/tassp.1978.1163055.
- [4] SENIN, P. Dynamic time warping algorithm review. *Dynamic Time Warping Algorithm Review* [online]. 2013 [cit. 2015-01-05]. Dostupné z: <http://www2.hawaii.edu/~senin/assets/papers/DTW-review2008draft.pdf>
- [5] ČERNOCKÝ, Jan. Zpracování řečových signálů — studijní opora. In: [online]. 2006, 2006-12-06 [cit. 2015-01-05]. Dostupné z: http://www.fit.vutbr.cz/study/courses/ZRE/public/opora/zre_opora.pdf
- [6] TSIPORKOVA, Elena. *Dynamic Time Warping Algorithm for Gene Expression Time Series* [online]. [cit. 2015-01-05]. Dostupné z: <http://www.psb.ugent.be/cbd/papers/gentxwarper/DTWAlgorithm.ppt>
- [7] Skutkova, H.; Vitek, M.; Babula, P.; aj.: Classification of genomic signals using dynamic time warping. *BMC Bioinformatics*, ročník 14, č. Suppl 10, 2013: str. S1, ISSN 1471-2105, doi:10.1186/1471-2105-14-S10-S1.
- [8] AL-NAYMAT, Ghazi H. *NEW METHODS FOR MINING SEQUENTIAL AND TIME SERIES DATA*. University of Sydney, 2009. Dostupné také z: <http://hdl.handle.net/2123/5295>. Disertační práce.
- [9] *Data Mining a Trillion Time Series Subsequences Under Dynamic Time Warping*. In: RAKTHANMANON, Thanawin, Bilson CAMPANA, Abdullah MUEEN, Gustavo BATISTA, Brandon WESTOVER, Qiang ZHU, Jesin ZAKARIA a Eamonn KEOGH. Twenty-Third International Joint Conference on Artificial Intelligence, 2013, s. 3047-3051. Dostupné také z: <http://ijcai.org/papers13/Papers/IJCAI13-454.pdf>

- [10] INTERNATIONAL, IEEE MTT-S. *Microwave and Optoelectronics Conference (IMOC), 2009 SBMO/IEEE MTT-S International date, 3-6 Nov. 2009* [online]. [Piscatawy, N.J: IEEE, 2009 [cit. 2015-05-25]. ISBN 978-142-4453-573.
- [11] RATH, T.M. a R. MANMATHA. Word image matching using dynamic time warping. *2003 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2003. Proceedings* [online]. 2003 [cit. 2015-05-25]. DOI: 10.1109/cvpr.2003.1211511.
- [12] FILHO, Antonio Carlos Buriti da Costa, Joao Pereira de Brito FILHO, Renato Evangelista DE ARAUJO a Clayton Augusto BENEVIDES. Infrared-based system for vehicle classification. *2009 SBMO/IEEE MTT-S International Microwave and Optoelectronics Conference (IMOC)* [online]. IEEE, 2009, : 537-540 [cit. 2015-05-20]. DOI: 10.1109/IMOC.2009.5427528. ISBN 978-1-4244-5356-6. Dostupné z: <http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=5427528>
- [13] LORENC, P. Klasifikace srdečních cyklů. Brno: Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií, 2013. 47 s. Vedoucí diplomové práce Ing. Martin Vitek, Ph.D.
- [14] AARABI, Ardalan, Kamran KAZEMI, Reinhard GREBE, Hamid Abrishami MOGHADDAM a Fabrice WALLOIS. Detection of EEG transients in neonates and older children using a system based on dynamic time-warping template matching and spatial dipole clustering. *NeuroImage* [online]. 2009, **48**(1): 50-62 [cit. 2015-05-21]. DOI: 10.1016/j.neuroimage.2009.06.057. ISSN 10538119. Dostupné z: <http://linkinghub.elsevier.com/retrieve/pii/S1053811909006934>
- [15] SKUTKOVA, Helena, Martin VITEK, Sona KRIZKOVA, Rene KIZEK a Ivo PROVAZNIK. *Preprocessing and Classification of Electrophoresis Gel Images Using Dynamic Time Warping*. In: . ESG, 2013, s. 1609 - 1622. Int. J. Electrochem. Sci., 8. Dostupné také z: <http://www.electrochemsci.org/papers/vol8/80201609.pdf>
- [16] PASCUAL-GASPAR, J. M., V. CARDEÑOSO-PAYO a C. E. VIVARACHO-PASCUAL. *Practical On-Line Signature Verification*. , 1180. DOI: 10.1007/978-3-642-01793-3_119. Dostupné také z: http://link.springer.com/10.1007/978-3-642-01793-3_119

- [17] KHALIL, Mostafa I., Mohamed MOUSTAFA a Hazem M. ABBAS. Enhanced DTW based on-line signature verification: revue littéraire mensuelle. *2009 16th IEEE International Conference on Image Processing (ICIP)*. IEEE, 2009, , 2713-2716. DOI: 10.1109/ICIP.2009.5414166. ISBN 978-1-4244-5653-6. Dostupné také z: <http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=5414166>
- [18] TIAN, Wei a Jingyuan LV. A Different Approach to Off-Line Signature Verification Using the Optimal DTW Algorithm. *2012 International Conference on Computer Science and Service System*. IEEE, 2012, , 18-21. DOI: 10.1109/CSSS.2012.13. ISBN 978-0-7695-4719-0. Dostupné také z: <http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=6394251>
- [19] MARTENS, R. a L. CLAESEN. Dynamic programming optimisation for on-line signature verification. *Proceedings of the Fourth International Conference on Document Analysis and Recognition*. IEEE Comput. Soc, 1997, , 653-656. DOI: 10.1109/ICDAR.1997.620587. ISBN 0-8186-7898-4. Dostupné také z: <http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=620587>
- [20] YEUNG, Dit-Yan, Hong CHANG, Yimin XIONG, Susan GEORGE, Ramanujan KASHI, Takashi MATSUMOTO a Gerhard RIGOLL. *SVC2004: First International Signature Verification Competition*. , 16. DOI: 10.1007/978-3-540-25948-0_3. Dostupné také z: http://link.springer.com/10.1007/978-3-540-25948-0_3
- [21] *SVC2004 Download* [online]. 2004 [cit. 2016-04-23]. Dostupné z: <http://www.cse.ust.hk/svc2004/download.html>