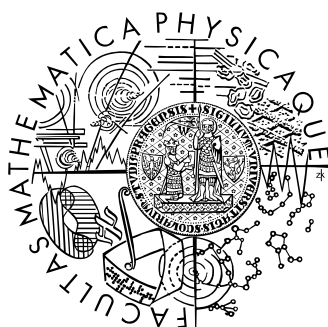


Univerzita Karlova v Praze
Matematicko-fyzikální fakulta

BAKALÁŘSKÁ PRÁCE



Tomáš Zámečník

Reprezentace křivek pro morfometrii

Kabinet software a výuky informatiky

Vedoucí bakalářské práce: RNDr. Josef Pelikán

Studijní program: Informatika, obor programování

2010

Poděkování

Děkuji svému vedoucímu, RNDr. Josefu Pelikánovi, za cenné rady, spolupráci během mé práce, poskytnutí literatury a koordinaci s týmem Morphome3cs. Dále děkuji za spolupráci všem členům týmu Morphome3cs, zvláště pak Bc. Jakubu Kratochvílovi za pomoc při statistickém zpracování dat a za úpravu PCA metody v prostředí Morphome3cs pro potřeby mé práce.

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

V Praze dne

Tomáš Zámečník

Obsah

1 Úvod	6
1.1 Křivky v antropologii	6
1.2 Určení pohlaví jedince na základě zkoumání tvaru kosti pánevní	6
1.3 Incisura ischiadica major	6
1.4 Vstupní vzorky antropologického výzkumu	6
1.5 Ruční analýza tvaru křivky	7
1.6 Počítačová analýza tvaru křivky	7
1.7 Průběh počítačové analýzy	7
2 Segmentace obrazu	8
2.1 Určení koncových bodů A,B křivky	8
2.2 Metody segmentace	8
2.3 První metoda segmentace – prahování	8
2.3.1 Určení výřezu (ROI)	8
2.3.2 Globální prahování určené histogramem	9
2.3.3 Detekce hran v binárním obrázku	9
2.3.4 Hledání nejkratší cesty	9
2.3.5 Asymptotická časová složitost	10
2.4 Druhá metoda segmentace – obrazové filtry	11
2.4.1 Preprocessing – tři filtry	11
2.4.2 Vytvoření cesty z A do B	12
2.4.3 Asymptotická časová složitost	12
2.5 Srovnání obou segmentačních metod	12
3 Různé reprezentace křivek	14
3.1 Normalizace křivky	14
3.2 Semilandmarkové metody	15
3.2.1 Určení semilandmarků podle úhlu	15
3.2.2 Určení semilandmarků podle délky křivky	16
3.2.3 Určení semilandmarků normálovou konstrukcí křivky ⁹	16
3.3 Posloupnosti funkcí	16
3.3.1 Fourierova řada	17
3.3.2 Legendreovy polynomy	19
3.3.3 Poznámka k terminologii	20
3.4 Srovnání reprezentací	20
4 Implementace	23
4.1 Program pro testování segmentace	23
4.2 Prostředí Morphome3cs	23
4.2.1 Postup určování pohlavního dimorfismu v prostředí Morphome3cs	23
5 Závěr	24
Dodatek I – Uživatelská dokumentace	25
I.1 Specimen Editor	25
I.2 Curve Detection Editor	25
I.3 Python, R, filtry	26

Dodatek II – Programátorská dokumentace	28
II.1 Curve Detection Editor	28
II.1.1 Vstupní data editoru	28
II.1.2 Formát textového souboru	29
II.1.3 Ukládání/načítání textového souboru	30
II.1.4 Export hodnot v rámci programu	30
II.2 GUI	30
II.3 Segmentace křivky	31
II.4 Výpočet aktivní reprezentace křivek	32
II.4.1 Normalizace a „denormalizace“ křivky	35
II.4.2 Vykreslení aktivní reprezentace	36
Literatura	38
Internetové odkazy	38

Abstrakt

Název práce: Reprezentace křivek pro morfometrii
Autor: Tomáš Zámečník
Katedra: Kabinet software a výuky informatiky
Vedoucí bakalářské práce: RNDr. Josef Pelikán
e-mail vedoucího: pepca@cgg.mff.cuni.cz

Abstrakt: Tato práce se zabývá zkoumáním, vývojem a implementací různých metod pro reprezentaci křivek v antropologii a metod pro detekci těchto křivek v digitální fotografii. Užití metody spadají do oboru počítačové analýzy obrazu.

Nedílnou součástí práce je softwarový projekt – editor křivek, začleněný do prostředí Morphome3cs, vyvíjeného stejnojmenným týmem studentů na MFF UK. Motivací a přímou aplikací práce je zkoumání pohlavního dimorfismu velkého zářezu kosti pánevní a nahrazení dosud užívaných manuálních metod. Výzkumem pohlavního dimorfismu se zabývá katedra antropologie PřF UK, pro níž je projekt vyvíjen.

Klíčová slova: morfometrie, incisura ischiadica major, křivka, segmentace křivky

Title: Curve Representations for Morphometrics
Author: Tomáš Zámečník
Department: Department of Software and Computer Science Education
Supervisor: RNDr. Josef Pelikán
Supervisor's e-mail address: pepca@cgg.mff.cuni.cz

Abstract: This paper deals with analysis, research and implementation of various methods of curve representation in anthropology and methods for such curves detection in digital photography. Used methods appertain to computer image processing and analysis.

Fundamental part of this work is software project – curve editor, included in Morphome3cs framework, which was developed by same-named team on Faculty of Mathematics and Physics on Charles University in Prague.

As a motivation for this work, and simultaneously as a first application, served analysis of sexual dimorphism of the greater sciatic notch and replacement of contemporary used manual methods. Faculty of Science on Charles University in Prague deals with such sexual dimorphism research and this project is developed for it.

Keywords: morphometry, incisura ischiadica major, curve, curve segmentation

Kapitola 1

Úvod

1.1 Křivky v antropologii

Motivací pro studium tvarů křivek byla práce antropologů na Přírodovědecké fakultě Univerzity Karlovy. Tato práce spočívala ve zkoumání pohlaví jedince z úlomků nalezených kostí.

Na následujících řádcích se pokusíme ve stručnosti uvést části antropologického výzkumu, které předcházely této bakalářské práci.

1.2 Určení pohlaví jedince na základě zkoumání tvaru kosti pánevní

Vhodným, pro posuzování pohlaví, se ukázal velký zářez pánevní (lat. incisura ischiadica major). Bylo tomu jednak z toho důvodu, že (na rozdíl od jiných kostí, jež jsou u obou pohlaví tvarově rozdílné) se pánev často nacházela v dostatečně zachovalém stavu a dále proto, že podle antropologických studií velmi dobře odděluje mužské a ženské vzorky – zachycuje pohlavní dimorfismus (zevrubně rozebráno např. v [7]).



Obrázek 1:
Příklad pohlavního dimorfismu. Nalevo je velký zářez pánevní, patřící nálezů ženských ostatků, pravá fotografie patří muži.

1.3 Incisura ischiadica major

Velký zářez pánevní mívá u mužů a žen značně odlišný tvar. Při bočním pohledu si můžeme všimnout následujících rozdílů: U mužských pánví bývá zářez úzký a protáhlý, kdežto u ženských pánví je výrazně „kruhovitější“ (neboť je součástí porodní cesty).

1.4 Vstupní vzorky antropologického výzkumu

S ohledem na uvedené skutečnosti se jakožto vstupní vzorek pro určení pohlaví na základě tvaru stala fotografie kosti pánevní z bočního pohledu, vyfotografovaná na tmavém pozadí. Jednotlivé tmavé pozadí usnadňuje identifikaci obrysu kosti na fotografii. Při bočním pohledu

vidíme právě konturu velkého zářezu pánevního. Tuto křivku lze snadno identifikovat a obkreslit.

1.5 Ruční analýza tvaru křivky

Praxe při převádění křivky do matematické podoby obnášela ruční obkreslení křivky na vytištěné fotografii vzorku, pomocí speciálního elektronického pera, které přenášelo souřadnice kreslené čáry do počítače. Nevýhodou této metody je její pracnost a možnost zanesení chyby při ručním obkreslování.

1.6 Počítačová analýza tvaru křivky

Na podnět dr. Jany Velemínské z katedry antropologie PřF UK, která se výzkumem zabývá, a dr. Josefa Pelikána z MFF UK měla být vyvinuta nová metoda zpracování vstupních vzorků, která by co nejvíce využívala počítače a byla do značné míry automatizovaná.

Zkoumání takovýchto metod a spolupráce při jejich softwarové implementaci je hlavní náplní mé bakalářské práce.

Některé výsledky práce už byly použity dr. Janou Velemínskou na konferenci pořádané společností American Association of Physical Anthropologists v Albuquerque v Novém Mexiku, ve dnech 14.-17.dubna 2010. (plakát vystavený na konferenci [10])

1.7 Průběh počítačové analýzy

Tím, jak počítačová analýza křivky velkého zářezu pánevního probíhá, se zabírají kapitoly 2 a 3. Na následujících řádcích si zhruba nastíníme, z jakých kroků se celý proces skládá.

Nejprve je třeba v rastrovém obrázku křivku detekovat – vybrat množinu pixelů, která křivku pokrývá. Tento proces se nazývá segmentace obrazu a je zevrubně popsán v kapitole 2.

Vstupem segmentace je fotografie vzorku na homogenním tmavém pozadí a souřadnice koncových bodů křivky, výstupem posloupnost pixelů, pokrývajících křivku.

Dalším krokem zpracování, je převedení křivky do nějaké vhodné matematické reprezentace. Zkoumané reprezentace a jejich podrobný popis obsahuje kapitola 3. Obecně lze říci, že v tomto kroku dojde k převodu množiny pixelů (jejich souřadnic) na uspořádanou množinu čísel, odpovídajících dané reprezentaci.

Posledním krokem je statistické zpracování výsledné množiny čísel (popsáno v závěru třetí kapitoly). Zkoumá se především, jak daná matematická reprezentace křivky odděluje mužské a ženské vzorky. Ze statistických metod je možno aplikovat např. PCA nebo LDA. Vstupem zpracování je množina vzorků v nějaké vybrané reprezentaci (obecně n-rozměrná tabulka čísel). Typ výstupu záleží na použité statistické metodě. Může se jednat například o graf zachycující rozdělení mužských a ženských vzorků.

Kapitola 2

Segmentace obrazu

V této kapitole se budeme zabývat detekcí (segmentací) křivky velkého zářezu pánevního z digitální fotografie.

Segmentace je proces, při němž vybereme podmnožinu pixelů obrazu, která je objektem našeho zájmu. My budeme zkoumat tvar části obrysu kosti. Tzn. budeme hledat body, které danou část obrysu pokrývají.

2.1 Určení koncových bodů A,B křivky

Abychom mohli jednotlivé vzorky mezi sebou porovnávat, je třeba jasně a obecně určit, kde křivka začíná a kde končí. To je anatomická záležitost. Antropolog je schopen rozpoznat na fotografii význačné body, které určují začátek a konec. V testovací množině vzorků od dr.Velemínské byla místa, kde má křivka začínat a končit, vyznačena červenými křížky (obrázek 1).

Ať už jsou křížky ve fotografii zadány nebo ne, vždy koncové body určuje člověk. Jak přesnost zadání ovlivňuje výslednou statistiku, je rozebráno v kapitole 3.4.

2.2 Metody segmentace

Tato kapitola popisuje dvě různé metody segmentace křivky. První z nich využívá globálního prahování, druhá pak některé morfologické operace a filtry. Druhá metoda, kterou navrhl a implementoval dr.Pelikán, se nakonec ukázala jako lepší, neboť na rozdíl od první metody dávala výsledky na všech testovaných obrázcích.

Pro úplnost a srovnání uvádím metody obě.

Vstupními parametry segmentace obou metod jsou souřadnice bodů A,B a zkoumaná fotografie převedená do stupňů šedi (každý pixel má pouze jednu hodnotu - intenzitu).

2.3 První metoda segmentace – prahování

2.3.1 Určení výřezu (ROI)

Nejprve se z obrazu vybere obdélník, na nějž se omezíme při všech dalších operacích¹. Tím, že se omezíme na výřez, neměníme kvalitu segmentace, zrychlí se však výpočet (hlavně v případě větších obrázků, kde křivka zabírá jen malou část – např. je-li na snímku celá kost pánevní).

Díky tomu, že všechny křivky si jsou podobné, můžeme určit velikost výřezu obecně, jako nejmenší obdélník pokrývající definovanou oblast – např. kruh se středem vprostřed úsečky AB a poloměrem $k \cdot |AB|$ nebo obdélník, jehož jedna strana je rovnoběžná s AB, střed leží vprostřed AB a délky jeho stran jsou nějakými násobky délky $|AB|$. (schéma ROI je znázorněno na obrázku 12 v 2. dodatku této práce)

¹ Běžně se užívá anglického termínu Region of interest, zkráceně ROI.

2.3.2 Globální prahování určené histogramem

Dalším krokem je základní segmentace obrazu, která ho rozdělí na oblast popředí – kosti a pozadí – zbytku obrázku. Toto jednoduché rozdělení můžeme provést díky tomu, že kosti jsou světlé a jsou fotografovány na homogenním tmavém pozadí.

Určíme intenzitu I (práh), která o každém pixelu rozhodne, zda se nachází v popředí nebo na pozadí. Popředí je tvořeno všemi pixely, s intenzitou alespoň I . Této metodě se říká prahování (angl. thresholding) a je dobře popsána v literatuře (např. [3] nebo [11]).

Hodnota prahu vhodně oddělující popředí a pozadí je pro každý vzorek specifická.

K určení hodnoty prahu bylo využito histogramu. Aby se omezil negativní vliv těch částí fotografie, které nepřísluší ani kosti, ani černému pozadí (v některých fotografiích bylo např. vloženo ještě pravítko), je lepší nabírat hodnoty pro histogram pouze z okolí kontury, tedy třeba z malého čtvercového či kruhového okolí bodů A,B.

Díky tomu, že kost má světlou barvu a pozadí barvu tmavou, jsou hodnoty v histogramu rozděleny do dvou velkých celků. Potřebujeme získat hodnotu intenzity, která tyto celky od sebe oddělí. K jejímu nalezení byla použita metoda Otsu.

Prahováním rozdělíme histogram na dvě části 1 a 2. Metoda Otsu se snaží minimalizovat hodnotu σ_w^2 – vážený průměr rozptylů intenzit obou částí histogramu.

$$\begin{aligned}\sigma_w^2 &= p_1\sigma_1^2 + p_2\sigma_2^2 \\ p_1 &= \frac{1}{T} \sum_{i=0}^{T-1} h(i) \\ p_2 &= \frac{1}{256-T} \sum_{i=T}^{255} h(i)\end{aligned}\tag{1}$$

σ_1^2 σ_2^2 jsou rozptyly první a druhé části.

Metoda Otsu je popsána např. v [3].

2.3.3 Detekce hran v binárním obrázku

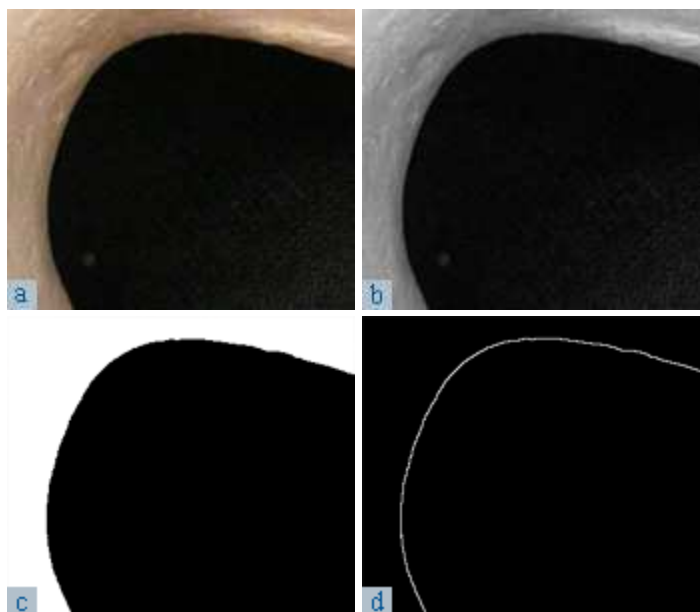
Prahováním jsme získali binární obrázek². Nyní v něm ponecháme pouze hrany. Všechny pixely, které náležejí hranám, budou mít maximální intenzitu, pixely, které neleží na hranách budou mít intenzitu rovnu nule. O pixelu prohlásíme, že leží na hraně právě tehdy, když má nenulovou hodnotu a alespoň jeden z jeho osmi sousedů má hodnotu nulovou.

2.3.4 Hledání nejkratší cesty

V hranovém binárním obrázku najdeme nejkratší cestu jdoucí z bodu A do bodu B pouze přes pixely s nenulovou intenzitou. K nalezení nejkratší cesty můžeme použít například průchod do šířky nebo Dijkstrův algoritmus (oba dva způsoby uceleně popsány např. v [9]).

Výsledkem je posloupnost pixelů, která je výchozí reprezentací křivky pro další zpracování (další zpracování v kapitole 3).

² Pixely binárního obrazu nabývají právě jedné z nějakých dvou hodnot. Obvykle se rozlišují nulové a nenulové hodnoty. Nulové = pozadí, nenulové = popředí.



Obrázek 2:
Průběh segmentace křivky zářezu kosti pánevní pomocí prahování
 a) Originální obrázek b) Převod do stupňů šedé
 c) Prahování s užitím histogramu d) Detekce hran

2.3.5 Asymptotická časová složitost

Určení velikosti výřezu probíhá v konstantním čase. Další rozbor složitosti bude vůči počtu pixelů výřezu, pojmenujme ho N .

Vytvoření histogramu obnáší zjištění hodnoty všech pixelů v nějakém malém okolí bodů A, B . To znamená určitě nejvýše $O(N)$ operací.

Určení prahu se provádí v lineárním čase vzhledem k velikosti histogramu. Ta je vždy 256, tedy celé stanovení prahu proběhne v čase $O(1)$.

Prahování spotřebuje $O(N)$ operací – pro každý pixel se v konstantním čase určí jeho nová hodnota.

Nalezení hran je opět nejvýše lineární vůči N , neboť pro každá pixel se vypočte hodnota z okolí konstantní velikosti.

Pro většinu obrázků dává dobré výsledky hledání nejkratší cesty pomocí průchodu do šířky. Pokud bychom ho použili, mělo by časovou složitost $O(N)$ – každý pixel projdeme nejvýše jednou a celá metoda segmentace by tedy byla $O(N)$. Při průchodu do šířky však nemůžeme použít ohodnocení hran, diagonální hrany tedy jsou stejně cenné, jako svislé a vodorovné. To odpovídá metrice

$$\rho(a, b) = \max \{ |a_x - b_x|, |a_y - b_y| \} \quad (2)$$

kde $\rho(a, b)$ je vzdálenost mezi body a, b . Pokud chceme mít klasickou euklidovskou metriku, která odpovídá skutečné vzdálenosti dvou bodů v rovině, je třeba zavést ohodnocení hran a použít např. Dijkstrův algoritmus. Použijeme-li implementaci Dijkstrova algoritmu se setříděným seznamem, (která se ukázala být dostačující), bude hledání nejkratší cesty $O(N^2)$, neboť v každém kroku algoritmu prohlásíme délku cesty do jednoho z celkového počtu N vrcholů za konečnou a uvnitř tohoto kroku přidáme do setříděného seznamu nejvýše osm sousedů daného vrcholu (přidání do setříděného seznamu probíhá v čase $O(N)$).

Celková asymptotická složitost segmentační metody je tedy rovněž $O(N^2)$.

2.4 Druhá metoda segmentace – obrazové filtry³

2.4.1 Preprocessing – tři filtry

Vstupem preprocessingu je obrázek ve stupních šedi (nebo opět pouze jeho výřez), výstupem pak obrázek obsahující pouze hrany⁴.

Nejprve se aplikuje Sobelův hranový filtr, postupně ve čtyřech různě natočených variantách. Každá varianta slouží k detekci hran v určitém směru. Sobelův filtr je konvoluční jádro⁵ velikosti 3x3.

1	2	1
0	0	0
-1	-2	-1

1	0	-1
2	0	-2
1	0	-1

2	1	0
1	0	-1
0	-1	-2

0	-1	-2
1	0	-1
2	1	0

Obrázek 3:

Sobelův filtr ve čtyřech variantách pro detekci horizontálních (1.malice), vertikálních (2.malice) i diagonálních hran (3. a 4. matice)

Dalším filtrem je eroze. Postupuje se tak, že se hodnota každého pixelu nahradí minimální hodnotou z jeho „čtyřokolí“, tzn. pokud $P[x,y]$ je hodnota pixelu se souřadnicemi x,y , pak nová hodnota $P'[x,y]$ se vypočte:

$$P'[x,y] = \min\{ P[x, y - 1], P[x - 1, y], P[x, y], P[x + 1, y], P[x, y + 1] \} \quad (3)$$

Eroze zde poslouží k „vyčištění“ obrázku. Odstraní se např. „roztřepené hrany“.

Posledním filtrem, který se v preprocessingu použije, je Gaussův filtr velikosti 3x3. Opět se jedná o konvoluční jádro. Používá se k odstranění šumu pomocí mírného rozmazání obrázku (míra rozmazání roste s velikostí konvolučního jádra).

1	2	1
2	4	2
1	2	1

Obrázek 4:

Konvoluční jádro Gaussova filtru velikosti 3x3.

Pozn.: hodnota dělitele pro konvoluci je v tomto případě 16 (aby byl součet roven 1).

Popis všech uvedených obrazových filtrů lze nalézt např. v [3].

³ Tuto metodu navrhl a v prostředí Morphome3cs implementoval Josef Pelikán

⁴ Tzn. tam, kde v původním obrázku byla ostrá změna intenzity, bude ve výsledku vysoká intenzita, tam kde byla malá nebo žádná změna intenzity, bude nízká intenzita.

⁵ Konvoluce je jedním ze základních způsobů zpracování obrazu. Pro každý pixel přepočte jeho intenzitu podle nejbližších sousedních pixelů. Způsob přepočtu a okolí, které započítává určuje konvoluční jádro (angl. kernel). Více o konvoluci např. v [3]

2.4.2 Vytvoření cesty z A do B

Nyní máme obrázek obsahující zvýrazněné hrany a s utlumenými pixely, kde žádné hrany nebyly. Je velmi pravděpodobné, že kontura kosti, kterou hledáme, bude mít též vysokou intenzitu.

Hledanou křivkou budiž nejlevnější cesta z A do B, přičemž cena jednotlivých hran se určí z intenzity pixelů a délky hrany. Cena c hrany byla zvolena:

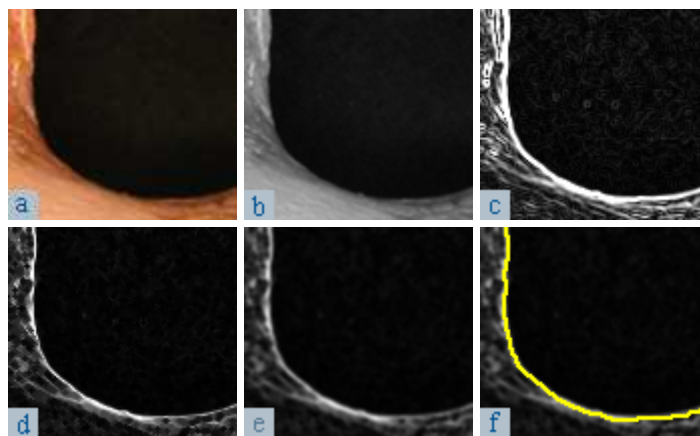
$$c = (255 - i)^3 \cdot l$$

l je délka hrany (většinou je 1, neboť hrany vedou mezi sousedními pixely)

i je intenzita pixelu v rozsahu 0..255

Křivka tedy bude vedena po pixelech s co nejvyšší intenzitou, pokud možno nejkratší cestou.

K nalezení nejlevnější cesty byl použit Dijkstrův algoritmus (popsán v [9]).



Obrázek 5:

Průběh segmentace křivky zářezu kosti pánevní pomocí obrazových filtrů

a) Originální obrázek b) Převedení do stupňů šedé c) Sobelův hranový filtr velikosti 3x3

d) Eroze na čtyřokolí e) Gaussův filtr velikosti 3x3 f) Cesta nalezená Dijkstrůvým algoritmem

2.4.3 Asymptotická časová složitost

Stejně jako u předešlé metody, budeme analyzovat časovou složitost v závislosti na počtu pixelů pracovního výřezu N .

Všechny tři použité preprocessingové filtry procházejí celý obrázek a pro zpracování každého pixelu potřebují konstantní počet operací. Jejich časová složitost je tedy $O(N)$.

K hledání nejkratší cesty použijeme Dijkstrův algoritmus se setříděným polem, který má časovou složitost $O(N^2)$ (rozebráno v analýze složitosti první metody).

Celý algoritmus segmentace má tedy asymptotickou složitost $O(N^2)$.

2.5 Srovnání obou segmentačních metod

Hlavním důvodem, proč byla použita druhá metoda je, že pro některé obrázky se první metodou nepodařilo křivku detekovat. Jedná se o obrázky, kde je pozadí nehomogenní nebo přesevřtené. Důvodem je globální prahování – pro nehomogenní obrázky nemusí existovat globální práh, bylo by tedy lepší zvolit lokální prahování. Vzniklý binární obrázek může být vlivem špatného prahování „roztržený“ (objekt popředí není souvislý) a tudíž cesta z bodu A do bodu B neexistuje.

Rozdílnost přístupu druhé metody a také hlavní výhoda proti metodě první tkví v tom, že nehledá nejkratší cestu v binárním obrázku (která nemusí existovat, neboť pomyslné grafové hrany vedou pouze mezi pixely s nenulovou intenzitou), ale nejlevnější cestu v „šedotónovém“ obrázku, která existuje vždy (hrany jsou definovány z každého pixelu do všech sousedů).

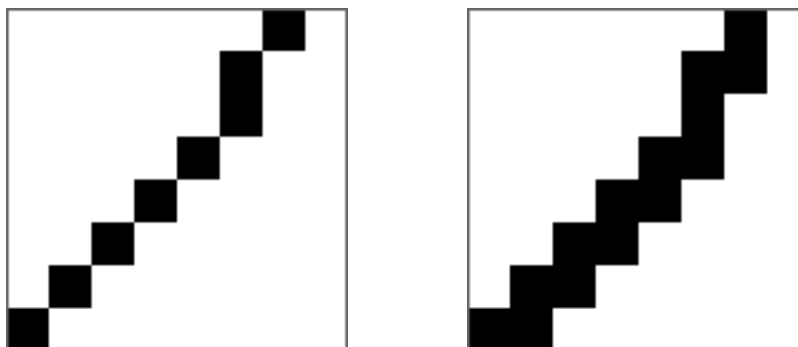
Z hlediska asymptotické časové složitosti jsou na tom obě metody stejně (uvedeno v rozbořech složitosti jednotlivých metod).

Ani jedna metoda nemůže zaručit stoprocentní korektnost výsledku, tzn., že detekuje obrys vzorku správně, druhá metoda však zaručuje, že se křivka vždy vytvoří.

Kapitola 3

Různé reprezentace křivek

Základní reprezentací digitalizované křivky je 8-souvislá⁶ posloupnost pixelů (obrázek 6, vlevo), získaná při segmentaci. Křivka v „pixelové“ reprezentaci není ničím jiným než podmnožinou všech pixelů obrázku. V závislosti na rozlišení fotografie se může jednat o stovky až tisíce pixelů. Pro další zpracování je tato reprezentace nevhodná. Obsahuje nespecifikované množství bodů, které nejsou nijak normovány a mohou obsahovat šum⁷.



Obrázek 6:
Srovnání základních typů souvislosti rastrových křivek.
Vlevo 8-souvislá křivka, vpravo 4-souvislá křivka.

3.1 Normalizace křivky

Před dalším zpracováním je třeba transformovat křivku do nějakého normovaného systému souřadnic. Převod sestává ze tří základních geometrických transformací – posunutí, otočení, změna měřítka.

K určení parametrů transformace nám poslouží úsečka koncových bodů AB.

Jako střed souřadného systému vezmeme střed úsečky. Kladný směr osy X ztotožníme se směrem vektoru AB a nakonec přiřadíme úsečce AB velikost 1.

Na závěr přidáme ještě jednu transformaci, kterou je zrcadlení křivky, kde osou symetrie je osa X. Překlápíme pouze křivky, jejichž body jdou od A do B podle směru hod. ručiček.

Díky těmto transformacím budou všechny křivky začínat na souřadnici $(-0,5; 0)$, končit na souřadnici $(0,5; 0)$ a všechny jejich body budou ležet v polorovině určené osou X a kladným směrem osy Y (tzn. „nad osou X“).

Díky normalizaci můžeme porovnávat tvar křivky nezávisle na způsobu, jakým byla pořízena fotografie kosti, tzn. nezávisle na natočení vzorku, případném překlopení, vzdálenosti od objektivu a na rozlišení fotoaparátu.

Dalším důsledkem je, že nezáleží ani na fyzické velikosti vzorku. Pokud bychom chtěli tuto velikost zohlednit, bylo by třeba zavést fyzické měřítko (zohlednit skutečnou vzdálenost bodů

⁶ 8-souvislost znamená spojení sousedních pixelů v libovolném z osmi základních směrů. 8-souvislá křivka je tedy taková posloupnost čtvercových bodů, které na sebe navazují svými vrcholy nebo stranami. Příznačné pro 8-souvislou křivku je, že neobsahuje pravouhlé spojení žádných tří po sobě jdoucích pixelů. Právě takovéto křivky (bez pravouhlých přechodů) jsou výsledkem segmentace z předešlé kapitoly.

⁷ Šum je zde myšlen jako nepřesnost v umístění bodů, k níž může dojít v průběhu segmentace křivky například z důvodu mírně rozmazané fotografie. Určitá míra šumu se dá očekávat vždy, proto není možné pracovat s přesností na 1 pixel a tedy s posloupností všech pixelů. Pro určení hrubého tvaru, jakým se zabýváme, to ani není třeba.

A,B na vzorku) namísto pevného přiřazení jednotkové délky. Toto by šlo realizovat přiložením pravítka ke všem fotografovaným vzorkům a jeho změřením při zpracování fotografie. „Pixelové jednotky“ by se následně převedly na naměřené fyzické jednotky⁸.

Po normalizaci máme křivku ve tvaru posloupnosti dvourozměrných bodů nad tělesem racionálních čísel. Tento tvar použijeme jako výchozí pro převod na všechny použité reprezentace křivky.

3.2 Semilandmarkové metody

V antropologii se používá výraz landmark pro význačný bod na kostře, který určuje antropolog. V našem případě se jedná o koncové body A,B. Dále můžeme na povrchu vzorku určit jemnější členění pomocí tzv. semilandmarků. Podle vzájemné pozice všech těchto bodů můžeme určovat tvar křivky. Následuje popis testovaných semilandmarkových metod:

3.2.1 Určení semilandmarků podle úhlu

Z vhodného bodu povedeme N paprsků. Semilandmarky budou ležet v průsečících paprsků s křivkou.

Při této metodě využijeme skutečnosti, že křivka má vždy konvexní obloukovitý tvar. Z počátku souřadnic (střed úsečky AB) vedeme v pravidelných úhlových rozestupech N paprsků (polopřímek). Pro každý paprsek vytvoříme právě jeden semilandmark tam, kde paprsek protne křivku. Průsečík je vždy jeden, díky tvaru křivky a určení počátku souřadnic.

Výhodou této metody je především snadná implementovatelnost.

Body se nejprve převedou do polárního souřadného systému. Poté se hledá průsečík křivky s každým paprskem. V obecném případě paprsek protne křivku mezi některými dvěma, po sobě jdoucími, body C, D . Necht' f je funkce, která přiřadí bodu velikost úhlu, který svírá s osou AB . Potom C, D jsou jediné dva po sobě jdoucí body, pro které platí: $f(A) \leq \alpha \leq f(C)$, kde α je úhel, který svírá paprsek s osou AB .

Pokud jsou body křivky dostatečně husté, můžeme použít konstantní interpolaci pro výpočet průsečíku. Vzdálenost semilandmarku od počátku bude tedy rovna vzdálenosti některého z bodů C, D .

Přesnější je však použít lineární interpolaci, zejména, jsou-li zadané body daleko od sebe. V tom případě leží semilandmark v průsečíku paprsku s úsečkou CD . Průsečík vypočteme pomocí parametrické rovnice paprsku a přímky určené úsečkou CD .

V této podobě se metoda semilandmarků podle úhlu však ukázala nevhodná pro statistické zpracování, neboť výsledek obsahuje redundantní informace. (Více o úspěšnosti metody v kapitole 3.4)

⁸ Tato metoda byla testována ještě před přenesením programu do prostředí Morphome3cs. V jejím používání se ale dále nepokračovalo, namísto toho byla zvolena metoda pevného měřítka (pokud zkoumáme pouze tvar, je to dostačující)

3.2.2 Určení semilandmarků podle délky křivky⁹

Křivku rozdělíme na N stejně dlouhých částí. Mezi každými dvěma částmi bude ležet jeden semilandmark.

K implementaci této metody je třeba umět zjistit vzdálenost bodu křivky od jejího počátku. Přesněji řečeno, pokud AB jsou koncové body křivky a X je libovolný bod křivky, potřebujeme umět určit délku křivky AX .

Můžeme-li se spolehnout na to, že body normalizované křivky jsou zadány relativně hustě, vystačíme si opět s lineární interpolací. Délku křivky tedy aproximujeme jako délku lomené čáry. Chceme-li znát délku AX , sečteme délky všech úseček, z nichž se AX skládá.

Při určování semilandmarků potřebujeme rozdělit křivku na N stejně dlouhých křivek. Zjistíme tedy nejprve délku křivky AB (označíme $|AB|$). Potom k -tý semilandmark je bod křivky, ležící ve vzdálenosti $k \cdot |AB|/N$ od počátku křivky.

Zpravidla bude semilandmark (stejně jako u předchozí reprezentace) ležet mezi dvěma zadanými body. Můžeme použít buď jeden z nich nebo jejich střed nebo dopočítat polohu pomocí lineární interpolace.

3.2.3 Určení semilandmarků normálovou konstrukcí křivky⁹

Křivku budeme konstruovat pomocí vhodného vyplňování prostoru trojúhelníky. Semilandmarky pak najdeme v některých jejich vrcholech.

První semilandmark bude ležet na průsečíku osy úsečky AB s křivkou. Pojmenujme si tento semilandmark např. C . Další dva semilandmarky budou ležet na průsečíku křivky s osou úsečky AC a průsečíku křivky s osou úsečky BC . Pro určení dalších semilandmarků postup rekurzivně opakujeme až do hloubky N (pro nějaké pevně zadané $N \geq 1$).

Počet semilandmarků je tedy $1 + 2 + 4 + \dots + 2^{N-1} = 2^N - 1$.

V každém kroku algoritmu máme zadány dva body trojúhelníku (buďto body AB nebo body vypočtené v některém z předchozích kroků algoritmu). Třetí bod vypočteme jako průsečík přímky s křivkou. Opět použijeme aproximaci křivky pomocí lomené čáry a hledáme průsečík přímky s touto lomenou čarou, tedy s každou z úseček, které ji tvoří.

Normálová konstrukce je detailně popsána např. v [4], ve 2. kapitole.

3.3 Posloupnosti funkcí

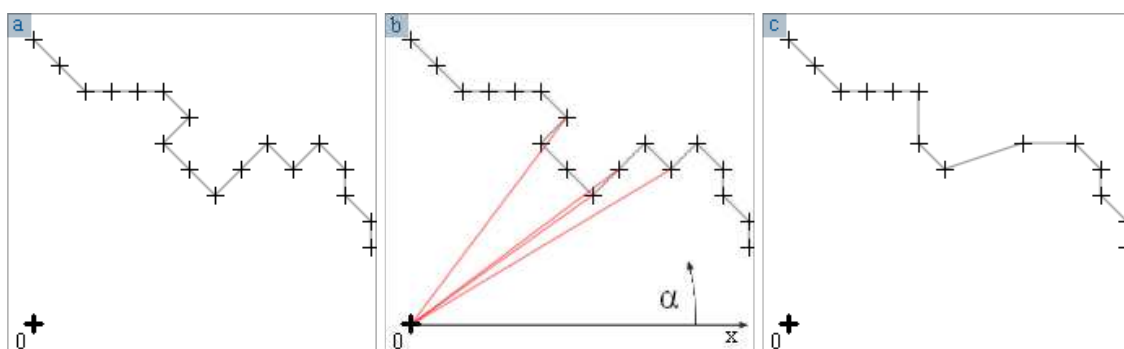
Následující dvě reprezentace pohlížejí na křivku jako na spojitou funkci úhlu s definičním oborem $[0, \Pi]$.

Než přistoupíme k převodu na jednu z těchto reprezentací, je třeba nejprve převést body křivky do polárních souřadnic. Bodu A v kartézských souřadnicích odpovídá bod A' v polárních souřadnicích, který se vypočte následovně:

⁹ Tuto metodu implementoval v projektu Morphome3cs dr. Josef Pelikán.

$$\begin{aligned}
A &= (x, y) \\
A' &= \left(a \tan\left(\frac{y}{x}\right), \sqrt{x^2 + y^2}\right), \quad x \neq 0 \\
A' &= \left(\frac{\Pi}{2}, \sqrt{x^2 + y^2}\right), \quad x = 0, y \geq 0 \\
A' &= \left(\frac{3}{2}\Pi, \sqrt{x^2 + y^2}\right), \quad x = 0, y < 0
\end{aligned}
\tag{4}$$

Dále musíme zajistit, aby se skutečně jednalo o funkci úhlu. Toho dosáhneme sekvenčním průchodem posloupnosti a vymazáním některých bodů – zachováme pouze takové body, kde hodnota úhlu roste (v daném bodě je vyšší než v bodě předcházejícím) a leží v rozmezí definičního oboru (tzn. $[0, \Pi]$).



Obrázek 7: Odstranění některých bodů tak, aby posloupnost tvořila funkci úhlu.
a) Výsledek segmentace křivky - posloupnost bodů. (lomenou čarou znázorněna lin. interpolace)
b) Průchod posloupností a odstranění bodů, kde při hodnota úhlu neroste.
c) „Pročištěná“ posloupnost bodů. Nyní je funkcí úhlu.

Po tomto kroku posloupnost bodů splňuje definici funkce, to znamená, že pro každou hodnotu z definičního oboru existuje nejvýše jeden bod posloupnosti a bude to platit i pokud množinu bodů doplníme o úsečky spojující každé dva sousední body, což je vlastnost, kterou budeme pro obě následující reprezentace křivky potřebovat.

Vynecháním některých bodů sice změníme tvar křivky, ale v praxi je tato změna jen velmi malá a na hrubý tvar křivky, který budeme zkoumat nemá prakticky žádný vliv. Jinak by tomu mohlo být, pokud by rozlišení křivky bylo příliš nízké. Kdyby například její délka byla pouze několik jednotek nebo desítek pixelů.

3.3.1 Fourierova řada

Každou periodickou funkci F s délkou periody 2Π lze rozložit na součet spočetně mnoha harmonických funkcí tvaru:

$$F(\alpha) = \frac{1}{\sqrt{2\Pi}} a_0 + \sum_{l=1}^{\infty} a_l \frac{\cos(l\alpha)}{\sqrt{\Pi}} + b_l \frac{\sin(l\alpha)}{\sqrt{\Pi}}
\tag{5}$$

$\{ a_0, a_1, a_2, a_3, \dots, b_1, b_2, b_3, \dots \}$ je množina koeficientů.

Doplníme-li pomocí středové symetrie posloupnost bodů na úhlový rozsah $[0, 2\Pi]$, získáme jednu periodu z požadované 2Π -periodické funkce a můžeme ji tedy rozložit na součet spočetně mnoha harmonických funkcí.

Tento proces se nazývá Fourierova transformace. Vstupem Fourierovy transformace je množina dvojic $(x, y(x))$, v našem případě je x úhel a y vzdálenost bodu od počátku souřadného systému. Výstupem transformace je množina dvojic $(n, k(n))$, kde n je číslo harmonické funkce a k je hodnota koeficientu této funkce.

Před tím, než převedeme posloupnost bodů na posloupnost koeficientů, zvolíme číslo N , které bude určovat, kolik koeficientů budeme počítat. Zbylé koeficienty prohlásíme za nulové.

Čím větší zvolíme N , tím větší frekvence započteme do výsledné křivky. Můžeme tedy velmi snadno určovat přesnost, s jakou chceme křivku aproximovat. S nekonečnou přesností samozřejmě pracovat nemůžeme. Naopak, je výhodné přesnost značně omezit, tím se nám podaří odstranit případný šum, obsažený ve vysokých frekvencích. Volba N je záležitostí čistě experimentální. Pro účely zkoumání tvaru křivky zářezu kosti pánevní bylo použito N v řádu desítek či jednotek.

K Fourierově transformaci existuje inverzní transformace, tzv. zpětná Fourierova transformace, jejímž výstupem je posloupnost bodů odpovídající dané posloupnosti koeficientů. Zpětnou Fourierovu transformaci použijeme, když potřebujeme křivku vykreslit do obrázku a zhodnotit, jak věrně aproximuje tvar kontury.

Výpočet koeficientů

Při výpočtu koeficientů se budeme na věc dívat algebraickým pohledem. Množina všech 2π -periodických funkcí tvoří vektorový prostor se spočetnou bází. Prvky báze jsou harmonické funkce ze vzorce (5).

Koeficienty vypočteme projekcí vstupní funkce na jednotlivé bazické funkce. Hodnota k -tého koeficientu se vypočte následovně:

$$w_i = \int_0^{2\pi} F(\theta) B_i(\theta) d\theta \quad (6)$$

Integrál vypočteme obdélníkovou metodou. Odpovídá to konstantní interpolaci mezi každými dvěma sousedními body (jako kdyby křivka byla tvořena „kruhovými schody“¹⁰). Přesnost aproximace originálního tvaru tedy velmi závisí na vzorkování¹¹.

Výpočet integrálu:

$$w_i = \sum_{j=1}^K F_j B_i(\theta_j) d \quad (7)$$

K je počet vzorků „signálu“, tzn. počet bodů křivky, které použijeme pro výpočet. F_j je hodnota j -tého vzorku (vzdálenost j -tého bodu od počátku souřadného systému). θ_j je úhel j -tého bodu, který svírá s kladným směrem osy x a d je vzdálenost mezi dvěma sousedními vzorky. Pokud funkce není navzorkována v konstantních intervalech, použije se místo konstanty k hodnota $\theta_j - \theta_{j-1}$.

¹⁰ Je důležité mít na paměti, že počítáme v polárních souřadnicích. Pokud je funkce mezi dvěma body konstantní, bude tam mít křivka tvar části kružnice. Pokud je funkce mezi dvěma body lineární, bude mít křivka mezi těmito body tvar části spirály.

¹¹ Pro dostatečnou aproximaci (a proto, aby na křivce nevznikaly obloučky (viz poznámka 10)), je vhodné množinu bodů před výpočtem Fourierovy transformace zahustit pomocí interpolace (byla použita lineární), která ovšem musí být provedena v kartézských souřadnicích, nikoli v polárních (jinak by vznikaly nežádoucí obloučky (viz poznámka 10)).

Komprimovaná množina koeficientů

Ježto je funkce F Π -periodická, vyjdou hodnoty některých koeficientů nulové. Budou to koeficienty s těmito indexy: $\{3, 4, 7, 8, 11, 12, 15, 16, \dots\}$, což jsou koeficienty, kterým odpovídají členy báze ve tvaru:

$$\frac{\cos((2i+1)\alpha)}{\sqrt{\Pi}} \quad \text{nebo} \quad \frac{\sin((2j+1)\alpha)}{\sqrt{\Pi}} \quad i, j = 0, 1, 2, 3, 4, 5, \dots \quad (8)$$

Neboť je F Π -periodická, platí následující rovnosti (a tím pádem jsou zmíněné koeficienty nulové):

$$\begin{aligned} \int_0^{\Pi} F(\theta) \sin((2i+1)\theta) d\theta &= - \int_{\Pi}^{2\Pi} F(\theta) \sin((2i+1)\theta) d\theta \\ \int_{\frac{\Pi}{2}}^{\frac{\Pi}{2}} F(\theta) \cos((2i+1)\theta) d\theta &= - \int_{\frac{\Pi}{2}}^{\frac{3\Pi}{2}} F(\theta) \cos((2i+1)\theta) d\theta \end{aligned} \quad (9)$$

Zmíněné koeficienty jsou nulové pro všechny počítané křivky a tím pádem nenesou žádnou informaci. Nebudeme si je proto pamatovat a budeme ukládat pouze vybranou podposloupnost takovou, která bude obsahovat všechny původní koeficienty, kromě těch, které jsou vždy nulové. Těto posloupnosti budeme říkat komprimovaná posloupnost(množina) koeficientů.

Zpětná transformace

Křivku jako množinu bodů v polárních souřadnicích získáme pomocí lineární kombinace báze. Vzdálenost bodu křivky od počátku, pro daný úhel, se vypočte:

$$F(\theta) = \sum_{i=0}^N w_i B_i(\theta) \quad (10)$$

Více o Fourierově transformaci se lze dočíst v [1], [2] nebo [3].

3.3.2 Legendreovy polynomy

Tato reprezentace křivky je alternativou k reprezentaci předcházející. Funkci úhlu zde převedeme opět na spočetný součet jistých známých funkcí, tentokrát polynomů.

Celý postup dopředné i zpětné transformace je stejný jako v předchozí metodě, rozdíl je pouze v použitých bázevých funkcích B_i .

Transformací tedy opět získáme množinu reálných koeficientů a zpětnou transformací znovu funkci F . Tedy ve skutečnosti funkci, která se blíží F tím více, čím více vypočteme při dopředné transformaci koeficientů.

Použitá báze je následujícího tvaru:

$$\begin{aligned}
B_0(x) &= 1 \\
B_1(x) &= x \\
B_i(x) &= \frac{(2i-1)x B_{i-1}(x) - (i-1) B_{i-2}(x)}{i}
\end{aligned}
\tag{11}$$

Bazická funkce B_i je i -tý Legendreův polynom. Pro Legendreovy polynomy platí, že pokud chceme funkci F rozložit na jejich lineární kombinaci, může parametr x nabývat pouze hodnot z intervalu $[-1, 1]$. Před projekcí na bazické funkce je tedy ještě nutné převést definiční obor funkce F na interval $[-1, 1]$. Integrace rovněž probíhá na $[-1, 1]$, namísto intervalu $[0, 2\pi]$. Legendreovy polynomy jsou popsány např. v [8], [12] nebo [14].

3.3.3 Poznámka k terminologii

Pro obě metody (Fourierovy řady i Legendreovy polynomy) se lze v literatuře setkat s pojmem „Circular harmonics“. U Fourierových řad má tento název to opodstatnění, že se jedná o součet harmonických funkcí. U Legendreových polynomů zase vychází z analogie s obecnější prostorovou variantou zvanou Spherical harmonics (popsáno v [12]), kde se rovněž používají Legendreovy polynomy. Z důvodu nejednoznačnosti se proto v této práci termínu Circular harmonics pokoušíme vyhýbat.

3.4 Srovnání reprezentací

Několik různých reprezentací bylo testováno proto, aby se zjistilo, která je nejlépe využitelná pro počítačovou analýzu tvaru incisura ischiadica major. Jako základní kritérium byla zvolena schopnost oddělit od sebe množinu ženských a mužských vzorků a s co největší pravděpodobností určit u daných vzorků, zda se jedná o ženské či mužské pohlaví.

Na výstupu transformací do jednotlivých reprezentací je vždy nějaká uspořádaná n -tice hodnot (koeficienty nebo souřadnice).

Nad množinou 114 vzorků, u nichž bylo dopředu známo, o jaké pohlaví se jedná, byla provedena detekce křivek, výpočet všech reprezentací a následně LDA a PCA analýza¹².

Statistická analýza se vždy prováděla nad množinou příslušných vektorů (n -tic).

Pomocí LDA bylo provedeno rozdělení množiny na dvě podmnožiny. Toto rozdělení bylo provedeno pouze na základě naměřených hodnot křivek.

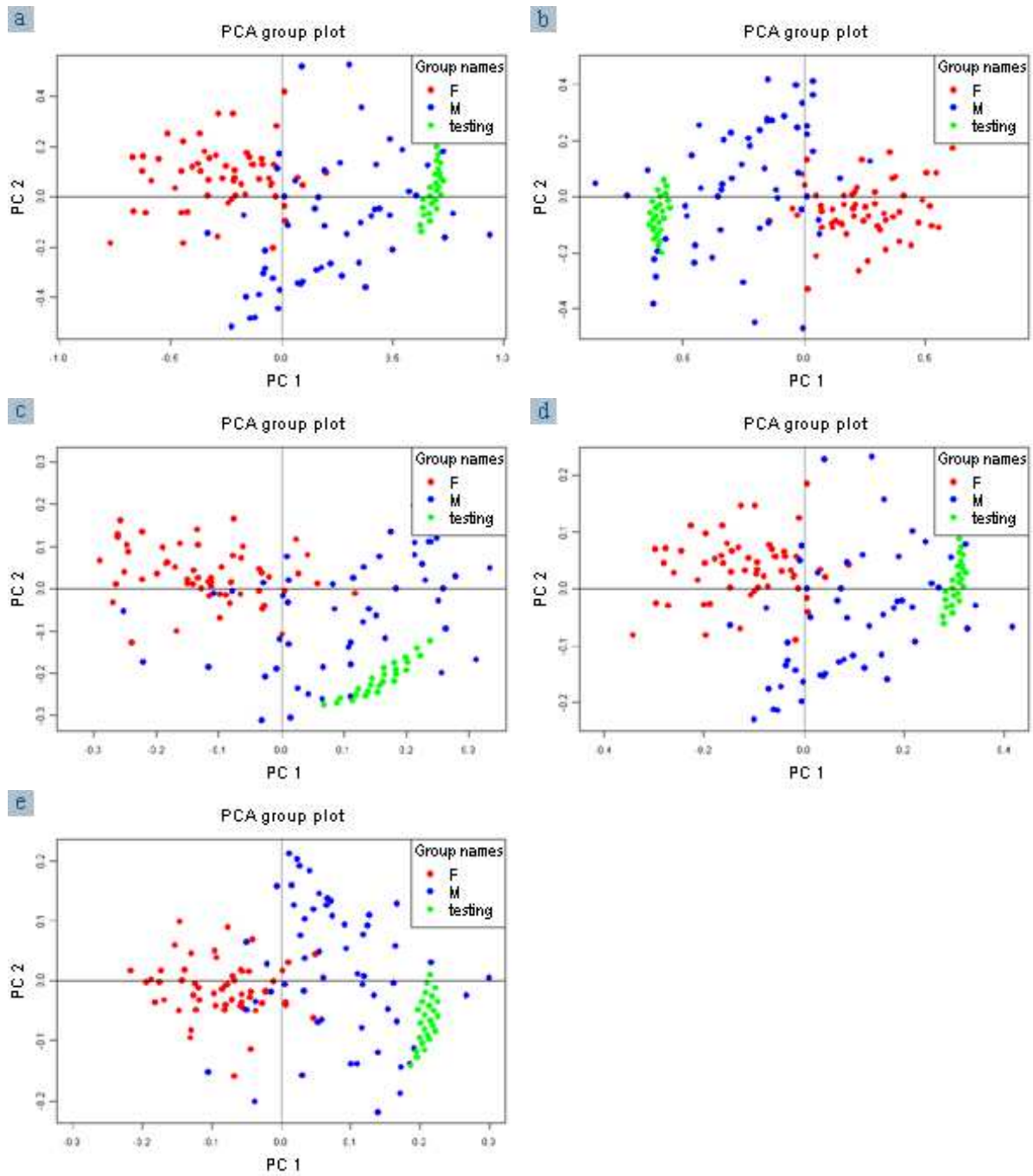
My však pro každou křivku víme, jakému pohlaví přísluší, můžeme proto ověřit, nakolik podmnožiny získané pomocí LDA odpovídají podmnožinám určeným na základě pohlaví. Množina LDA, která více odpovídá mužské, bude brána jako mužská, druhá pak jako ženská. Následující tabulka uvádí procento úspěšného zařazení. Úspěšně jsou zařazeny ty vzorky, jimž ruční i LDA oddělení přiřadilo stejné pohlaví.

¹² LDA a PCA analýzu v prostředí Morphome3cs implementoval Jakub Kratochvíl. Více o těchto metodách v [5] a [6].

Metoda	Úspěšnost
Vzdálenosti podle úhlu	96,5%
Semilandmarky podle délky křivky	97,4%
Normálová konstrukce	96,5%
Fourierovy koeficienty	96,5%
Legendrový koeficienty	96,5%
Semilandmarky podle úhlu	90,4%

Tabulka pro srovnání uvádí v posledním řádku metodu měření Semilandmarků podle úhlu, která byla nahrazena metodou měření vzdálenost podle úhlu, právě z důvodu nižší úspěšnosti, způsobené redundancí dat (je-li zadán implicitně úhel, není třeba počítat souřadnice x,y, stačí pouze vzdálenost – podrobněji v rozboru dané metody v kapitole o reprezentacích křivek).

Při dalším testování vyvstala otázka, jak moc záleží na chybě uživatele při zadávání koncových bodů. K posouzení vlivu uživatelské chyby byl proveden orientační test. Ze zmíněné množiny 114 vzorků byl náhodně vybrán jeden, u kterého se kromě zadání koncových bodů podle vyznačených křížků provedl výpočet reprezentací ještě pro posunuté koncové body. Každý z koncových bodů byl posunut na pět různých míst v okolí křížku do vzdálenosti 5% z celkové délky křivky. Pro každou dvojici bodů se pak vypočetly všechny reprezentace křivek, tzn. pro každou reprezentaci 25 alternativních zadání koncových bodů. Tato alternativní zadání se přidala do PCA analýzy. Na následujících PCA grafech (obrázek 8), zobrazujících na osách x a y první dvě PC komponenty, jsou modrou a červenou barvou vyznačeny mužské a ženské vzorky a zelenou barvou 25 přidaných testovacích vzorků. Tento orientační test ukazuje, že „malá“ nepřesnost v zadání nezmění výrazně výsledek a pokud je vzorek např. výrazně mužského tvaru(jako na obr.8), zůstane jím i při nepřesném zadání koncových bodů.



Obrázek 8:

PCA grafy 114 vzorků křivek incisura ischiadica major. Červenou barvou jsou znázorněny ženské vzorky, modrou barvou mužské. Zelené body odpovídají 25 pozměněným zadáním koncových bodů u jednoho vzorku.

- a) Vzdálenosti podle úhlu b) Semilandmarky podle délky křivky
- c) Normálová konstrukce d) Fourierovy koeficienty
- e) Legendrovy koeficienty

Kapitola 4

Implementace

4.1 Program pro testování segmentace

Pro vývoj segmentace křivky byl vytvořen testovací program. Je napsán v jazyce C++, využívá knihovny Win32API pro práci s okny a uživatelským rozhraním a knihovny GDI+ pro práci s obrázky JPG.

Program umožňuje otevřít obrázek, nastavit koncové body křivky, spustit segmentaci, nastavit měřítko¹³ a uložit výslednou posloupnost pixelů do csv souboru. Ukládané hodnoty se transformují ztotožněním úsečky AB s osou x kartézského souřadného systému. Navíc se provede změna velikosti tak, aby hodnoty odpovídaly nastavenému měřítku.

4.2 Prostředí Morphome3cs¹⁴

Morphome3cs je aplikace vyvíjená na MFF UK stejnojmenným týmem studentů, pod vedením dr. Josefa Pelikána. Jedná se o framework, umožňující snadno implementovat nejrůznější programy pro měření antropologických vzorků a spojovat tyto programy popřípadě jejich výsledky s vestavěnými statistickými nástroji, sloužícími k vyhodnocení měření.

Určování pohlavního dimorfismu pomocí porovnávání tvarů incisura ischiadica major se mělo stát jednou z prvních aplikací zavedených do frameworku Morphome3cs, a tak veškeré další implementace, vývoj a testování probíhalo v tomto prostředí.

Byl vytvořen modul pro zpracování křivek nazvaný CurveEditor, který obsahuje veškeré uvedené algoritmy. Tento editor umožňuje načítat fotografie vzorků, detekovat křivku zadáním koncových bodů a převést ji do všech uvedených reprezentací s nastavením příslušných potřebných parametrů pro každou jednotlivou reprezentaci. Křivky ve všech vypočtených reprezentacích se uloží do samostatného textového souboru.

4.2.1 Postup určování pohlavního dimorfismu v prostředí Morphome3cs

Morphome3cs umožňuje naimplementovat celý postup zpracování vzorků a interpretace výsledků. Základní experiment, který byl uskutečněn, byla detekce křivek u 114 vzorků, u nichž bylo dopředu určeno, o jaké pohlaví se jedná. U všech vzorků se vypočetly všechny reprezentace křivek. Pro každou reprezentaci se pak udělala lineární diskriminační analýza¹⁵ (LDA), jež rozdělila množinu vzorků na dvě disjunktní podmnožiny (muži a ženy).

Pro každou reprezentaci křivek se pak zjišťovalo, v kolika procentech bylo pohlaví určeno správně.

Uživatelská a programátorská dokumentace implementovaných metod jsou uvedeny v dodatcích I a II této práce.

¹³ U fotografií bylo přiloženo pravítko určující reálnou velikost vzorku. Neboť ale v této práci šlo především o studium tvaru, bylo od používání měřítka upuštěno, namísto toho se používají normalizované souřadnice (viz kapitola o reprezentacích křivek)

¹⁴ Dokumentaci k projektu Morphome3cs lze nalézt na [13]

¹⁵ LDA v prostředí Morphome3cs naimplementoval Jakub Kratochvíl z MFF UK.

Kapitola 5

Závěr

V této bakalářské práci jsme se zabývali možnostmi počítačové analýzy pohlavního dimorfismu vzorků pánevních kostí. Klíčovým k určení dimorfismu z těchto vzorků, je tvar velkého zářezu pánevního. V průběhu práce jsme zkoumali metody, jak zachytit a matematicky reprezentovat křivku tohoto zářezu, aby co nejlépe vypovídala o pohlaví jedince.

Cílem bylo zautomatizovat proces detekce a zpracování křivky incisura ischiadica major (velký zářez pánevní) a nahradit tak dosud užívané ruční metody.

Navrhli jsme a implementovali způsob detekce křivky velkého zářezu pánevního z digitální fotografie vzorku. Dále jsme navrhli a implementovali pět matematických reprezentací těchto křivek. Při vývoji všech metod byly použity (a v textu práce citovány) známé algoritmy z oboru zpracování obrazu.

V práci jsme se dále zabývali srovnáním navržených metod z hlediska úspěšnosti oddělování ženské a mužské části vzorků. Všechny metody jsme otestovali na množině 114 fotografií velkého zářezu pánevního s dopředu známým pohlavím. Dále byly provedeny testy vlivu nepřesného zadání koncových bodů křivky uživatelem na určení pohlaví.

Uvedené algoritmy byly naimplementovány včetně uživatelského rozhraní jako součást aplikace Morphome3cs, vyvíjené studenty MFF UK. Kompletní zdrojový kód v jazyce C# je volně dostupný na internetových stránkách projektu Morphome3cs (na adrese [13]) včetně dokumentace a setupu pro instalaci programu.

Z budoucích kroků bude nepochybně zajímavé a přínosné vytvoření klasifikátoru, který na základě zjištěného dimorfismu z testovací (učící) množiny vzorků bude schopen automaticky klasifikovat další jednotlivé vzorky jako mužské či ženské.

Dodatek I

Uživatelská dokumentace

V tomto dodatku bude vysvětlena práce uživatele při analýze tvaru křivek množiny vzorků v prostředí Morphome3cs. První částí je načtení obrázků a detekce křivek. Druhou částí pak statistické zpracování.

I.1 Specimen Editor

Specimen Editor je komponenta prostředí Morphome3cs určená pro načítání a prohlížení jednotlivých vzorků a doplnění jejich vstupních parametrů (přidání jména, landmarků, ...).

Při otevření editoru se zobrazí tabulka vzorků. Každý řádek odpovídá jednomu jedinci, každý sloupec pak nějakému uživatelem definovanému parametru. Počet řádků i sloupců lze měnit prostřednictvím panelu nástrojů. Dále je možno měnit typ sloupců (parametrů). Kromě základních datových typů (boolean, integer, real, string) lze pro parametr nastavit typ FILE, který má následující další podtypy: CurveDetectionEditor, Landmark2DEditor, Landmark2DEditor.

K parametru typu FILE se vážou data vytvořená příslušným editorem. V našem případě se budeme zabývat pouze typem CurveDetectionEditor.

Příklad:

Při detekci křivky byly zvoleny následující parametry:

Název parametru	Typ parametru	Komentář
no.	integer	pořadové číslo; unikátní identifikátor jedince; defaultní parametr ¹⁶
selected	boolean	příznak výběru jedince pro další zpracování; defaultní parametr ¹⁶
name	string	jméno jedince; unikátní identifikátor
sex	string	pohlaví jedince; hodnoty F nebo M
image	FILE	fotografie kosti pánevní a textový soubor s naměřenými hodnotami (křivka v různých reprezentacích)

Hodnoty parametrů základních datových typů se zadávají přímo do tabulky. Při poklepnání myši na buňku tabulky obsahující parametr typu FILE se otevře příslušný editor.

I.2 Curve Detection Editor

Curve Detection Editor je komponenta přidaná do prostředí Morphome3cs za účelem studia tvaru křivek incisura ischiadica major. Naprostá většina zpracování popsáná v této práci se odehrává právě v editoru křivek.

Základní chování a vzhled Curve Detection Editoru vychází z dříve vestavěné komponenty Landmark2DEditor – editoru umožňujícího otevírat obrázky a vkládat do nich landmarky.

¹⁶ Tento parametr je povinný – sloupec nelze odstranit

Při prvním otevření Curve Detection Editoru pro daného jedince se zobrazí dialogové okno pro zadání názvu obrázku a textového souboru, do nějž se budou ukládat detekované křivky. Potvrzením zadání se dialog uzavře a zobrazí se okno editoru s načteným obrázkem.

V horní části okna je lišta panelu nástrojů s několika ovládacími prvky. V její levé části jsou tři tlačítka pro práci s landmarky – jejich přidávání, odebrání a zobrazování/skrývání popisků. Následuje panel pro zvětšování a zmenšování obrázku (zoom). Až potud je ovládání shodné s Landmark2DEditorem.

Další část panelu nástrojů umožňuje ovládat segmentaci křivky a výpočty jednotlivých reprezentací. Z vyklápěcí nabídky (combo box) je možno vybrat jednu z reprezentací křivky. Vpravo od této nabídky je textové pole pro zadání parametru dané reprezentace (všechny použité reprezentace mají právě jeden parametr). Posledním ovládacím prvkem je tlačítko, umožňující přepočít křivky při změně reprezentace.

Zadá-li uživatel koncové body křivky (landmarky), provede se automaticky detekce křivky. Do obrázku se vykreslí vizualizace křivky v dané reprezentaci, do textového souboru asociovaného se vzorkem se uloží číselné vyjádření reprezentace (koeficienty, souřadnice, ...).

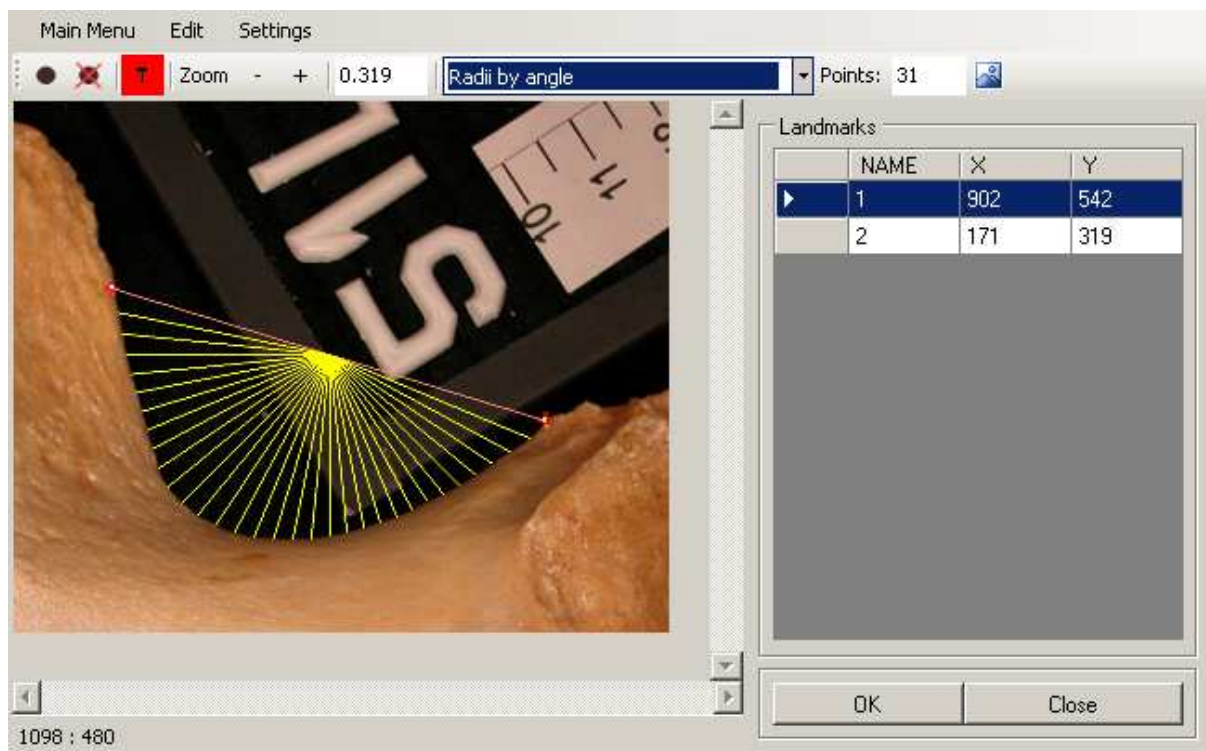
Tlačítkem v pravém dolním rohu okna se Curve Detection Editor zavře a aplikace se přepne zpět do Specimen Editoru.

Po úpravě všech jedinců zavřeme i Specimen Editor a uložená data můžeme dále zpracovat.

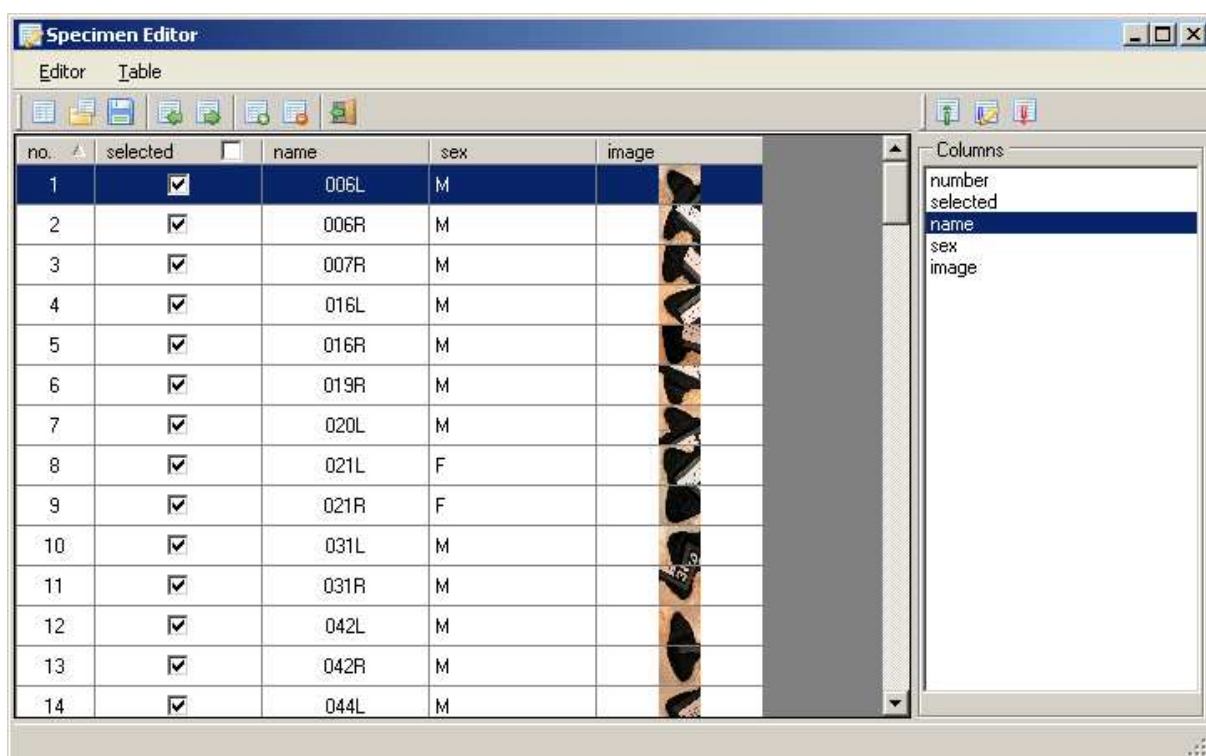
I.3 Python, R, filtry

Další zpracování dat je záležitostí uživatele. Do prostředí Morphome3cs lze totiž programovat nové filtry pomocí skriptovacího jazyku Python a přidávat statistické výpočty využívající známý statistický program R.

Jak pracovat s filtry a vytvářet nové lze nalézt v dokumentaci projektu Morphome3cs. (Možno nalézt na adrese [13])



Obrázek 9:
Práce s Curve Detection Editorem v prostředí Morphome3cs



Obrázek 10:
Práce se Specimen Editorem v prostředí Morphome3cs.

Dodatek II

Programátorská dokumentace

Metody pro segmentaci křivky a její transformaci do různých reprezentací byly začleněny do prostředí Morphome3cs.

Základem začlenění bylo vytvoření nové komponenty frameworku Morphome3cs nazvané Curve Detection Editor – uživatelského rozhraní pro práci s křivkami.

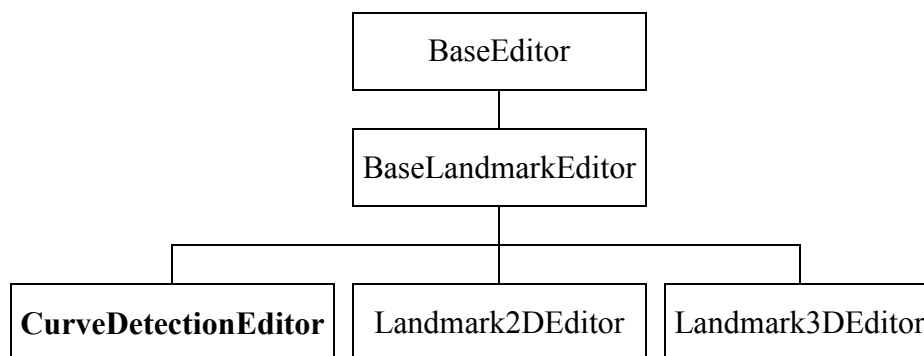
Zdrojový kód byl napsán v jazyce C#.

II.1 Curve Detection Editor

Komponenta přidaná do frameworku Morphome3cs, sloužící k detekci křivek a jejich převodu do několika různých reprezentací.

CurveDetectionEditor je třída odvozená od abstraktní třídy BaseLandmarkEditor, která implementuje interface BaseEditor.

BaseLandmarkEditor i BaseEditor patří mezi základními třídy/interfacy prostředí Morphome3cs. Program je navržen tak, že umí pracovat s editory odvozenými od těchto tříd, proto k začlenění křivkového editoru stačilo správně implementovat pouze několik základních virtuálních metod zděděných od třídy BaseLandmarkEditor.



Obrázek 11:
Hierarchie editorů v prostředí Morphome3cs

Editory odvozené od BaseLandmarkEditor slouží pro práci s landmarky (význačnými body vzorku, které zadává uživatel). V případě CurveDetectionEditoru jsme se omezili na pouhé dva landmarky (uživateli je zakázáno zadat jich více), určující koncové body křivky.

II.1.1 Vstupní data editoru

CurveDetectionEditoru dostane na vstupu jméno souboru, v němž je uložena fotografie vzorku a jméno textového souboru, v němž jsou uloženy zadané landmarky a data jednotlivých reprezentací křivek (pod označením textový soubor, budeme dále rozumět tento soubor).

II.1.2 Formát textového souboru

Textový soubor je rozdělen na jednotlivé záznamy. Každý záznam je na jednom samostatném řádku. Záznamy jsou sdruženy do sekcí. Každá sekce je jednoznačně identifikována svým prvním záznamem (řádkem), který tvoří jakýsi nadpis a ukončena speciálním ukončovacím řádkem (obsahuje nějaký konstantní řetězec).

Interpretace záznamu závisí na tom, do které sekce náleží. Program dokáže číst a zapisovat následující sekce:

Pixels

Ukládá nasegmentovanou křivku v pixelové reprezentaci. Každý záznam odpovídá právě jednomu pixelu a obsahuje jeho souřadnice.

Ending Points

Ukládá první a poslední pixel křivky (zadané landmarky). Na prvním řádku jsou souřadnice prvního landmarku, na druhém jsou souřadnice druhého.

Semilandmarks by angle

Ukládá libovolný počet bodů křivky v kartézských souřadnicích s normalizovanými jednotkami. Jedná se o průsečíky křivky s paprsky vedenými ze středu úsečky AB v konstantním úhlovém rozestupu.

Každý záznam obsahuje souřadnice jednoho bodu.

Radii by angle

Úspornější varianta předchozí reprezentace. Průsečíky jsou uloženy v polárních souřadnicích. Jeden záznam sekce odpovídá vzdálenosti jednoho průsečíku v normalizovaných jednotkách. Záznamy jsou seřazeny vzestupně podle úhlu, který daný paprsek svírá s úsečkou AB, takže se hodnota úhlu nemusí ukládat (je dána implicitně počtem průsečíků a pořadím záznamu).

Semilandmarks by arc

Ukládá libovolný počet bodů křivky v kartézských souřadnicích, přičemž délka křivky mezi každými dvěma body je shodná. Body jsou seřazeny podle své vzdálenosti na křivce od prvního landmarku.

Každý řádek obsahuje souřadnice jednoho bodu v normalizovaných jednotkách.

Normal Polyline

Ukládá křivku sestavenou normálovou konstrukcí.

Každý řádek obsahuje jeden koeficient. Koeficient je poměr výšky(normály) trojúhelníka k délce jeho základny (délky v normalizovaných souřadnicích).

Záznamy jsou uloženy v pořadí určeném procházením stromové struktury normálové konstrukce „do šířky“. Tedy po jednotlivých úrovních/vrstvách.

Circular Harmonics

Ukládá Fourierovy koeficienty. Na každém řádku je jeden koeficient. Záznamy jsou seřazeny vzestupně podle čísla koeficientu.

Circ. Harmonics - Legendre Polynomials

Ukládá koeficienty Legendrových polynomů. Na každém řádku je jeden koeficient. Záznamy jsou seřazeny vzestupně podle čísla koeficientu.

II.1.3 Ukládání/načítání textového souboru

Výstup do textového souboru je realizován pomocí třídy CurveDetectionExporter, odvozené od interface frameworku Morphome3cs ILandmarkExporter.

CurveDetectionExporter implementuje následující metodu interface ILandmarkExporter:

```
void exportSpecimenDataToFile(Hashtable table,  
string filePath)
```

table – hashovací tabulka; klíčem je název sekce textového souboru, hodnotou její data.
filePath – jméno souboru, do něž budou data uložena.

Načtení textového souboru se provádí prostřednictvím třídy CurveDetectionImporter, odvozené od interface frameworku Morphome3cs ILandmarkImporter.

CurveDetectionImporter implementuje následující metodu interface ILandmarkExporter:

```
void importSpecimenDataFromFile(Hashtable table,  
string filePath)
```

table – hashovací tabulka; klíčem je název sekce textového souboru, hodnotou její data.
(Metodě se předává odkaz na prázdnou tabulku, po úspěšném návratu z funkce obsahuje tabulka sekce uložené v textovém souboru)
filePath – jméno souboru, z něž budou data načtena.

II.1.4 Export hodnot v rámci programu

Pokud chceme, aby editor odvozený od třídy BaseLandmarkEditor exportoval nějaké hodnoty (na žádost nějaké jiné komponenty prostředí Morphome3cs), musí implementovat následující metodu:

```
IData ExportLandmarks(int mode)
```

V ostatních landmarkových editorech slouží tato metoda k exportu landmarků a jako návratovou hodnotu vrací tabulku landmarků. V CurveDetectionEditoru však navíc využívá parametr mode k určení toho, co se bude exportovat. Exportují se data určující jednotlivé reprezentace křivky. Každé reprezentaci odpovídá jedna hodnota parametru mode. Hodnoty jsou exportovány v jednorozměrné (jednosloupcové) nebo dvourozměrné tabulce (dvourozměrná tabulka, pokud se jedná o dvojice souřadnic).

Pokud je odněkud z aplikace zavolána tato metoda, příslušná instance CurveDetectionEditoru vrátí volajícímu subjektu reprezentaci křivky určenou parametrem mode.

II.2 GUI

Uživatelské rozhraní sestává z dialogu/formuláře vytvořeného pomocí editoru dialogů Microsoft Visual Studio 2008. Třída dialogu CurveDetectionEditorForm je odvozena od třídy frameworku Morphome3cs BaseLandmarkEditorForm.

CurveDetectionEditor je řízen událostmi (eventy), vyvolanými ovládacími prvky dialogu. Hlavní část okna zabírá fotografie vzorku. Jedná se o aktivní oblast, do níž se pomocí myši zadávají landmarky. Funkce pro detekci křivky a výpočet jejích reprezentací se spouštějí tlačítkem v panelu nástrojů v horní části okna. (Více o GUI v části Uživatelská dokumentace)

II.3 Segmentace křivky

Při vyvolání události pro detekci křivky se zavolá následující metoda CurveDetectionEditoru:

```
void CurveDetection()
```

Tato funkce zkontroluje, zda byly zadány oba dva landmarky a v kladném případě vyvolá samotný výpočet segmentace.

Segmentace se počítá pomocí metod třídy CurveSegmentation. Tato třída obsahuje tři statické metody:

```
Point[] DetectIncisuraCurve(Image img, Point A, Point B)
```

img – fotografie vzorku

A, B – koncové body křivky (landmarky)

Tato funkce vrací křivku v pixelové reprezentaci. Detekce se provádí pomocí prahování. (Algoritmus výpočtu uveden v kapitole 2.3)

```
Point[] DetectInsisuraCurve2(Image img, Point A, Point B)
```

img – fotografie vzorku

A, B – koncové body křivky (landmarky)

Tato funkce vrací křivku v pixelové reprezentaci. Detekce se provádí pomocí obrazových filtrů. (Algoritmus výpočtu uveden v kapitole 2.4)

```
Rectangle CountRoiForSegmentation(Point A, Point B,  
int iImageWidth, int iImageHeight, int iBorder)
```

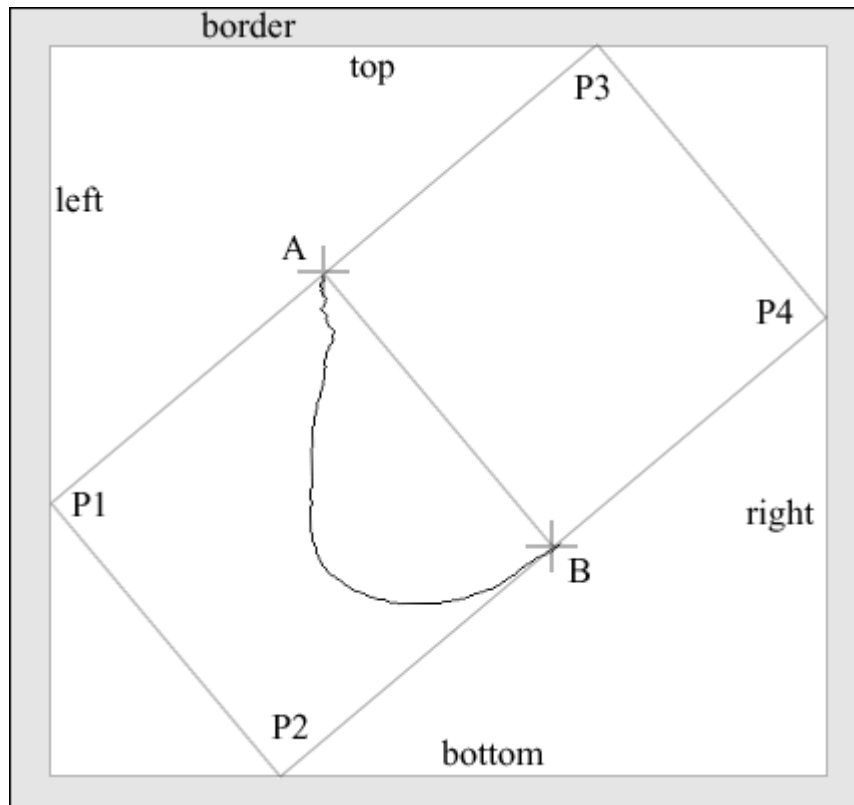
A, B – koncové body křivky (landmarky)

iImageWidth – šířka obrázku

iImageHeight – výška obrázku

iBorder – konstanta, o níž se má ROI rozšířit (do výšky i do šířky)

CountRoiForSegmentation počítá výřez (region of interest) pro detekci křivky. Funkce vypočte výřez tak, aby bezpečně obsahoval všechny části křivky. Předpokládáme, že křivka bude ležet v obdélníku P1P2P3P4 (P1P2AB a ABP4P3 jsou stejně velké čtverce). ROI potom bude nejmenší obdélník rovnoběžný s osou x, obsahující P1P2P3P4. Neboť některé vzorky mají tvar křivky velmi protáhlý k jednomu z bodů A, B a někdy i částečně přes, přičteme ještě k šířce i výšce výřezu parametr iBorder (jakási volitelná bezpečnostní hodnota zadaná).



Obrázek 12:
Výřez části obrázku (ROI) v závislosti na poloze bodů A, B.

II.4 Výpočet aktivní reprezentace křivek

CurveDetectionEditor má členskou proměnnou, určující aktivní reprezentaci křivek. Tzn. reprezentaci, která se zobrazuje a která se počítá.

Voláním následující funkce CurveDetectionEditoru

```
void ComputeActiveCurveRepresentation(int iParam)
```

iParam – parametr, který se předá funkci pro výpočet konkrétní reprezentace.

začíná proces výpočtu. Nejprve je nutno normalizovat pixelovou křivku (popsáno v odstavci Normalizace a „denormalizace“ křivky)

Podle toho, jaká je aktivní reprezentace, se zavolá jedna z následujících funkcí s parametrem iParam:

```
void ComputeSemilandmarksByAngle(int iPoints)
```

iPoints – počet semilandmarků (paprsků / průsečíků s křivkou)

Tato funkce vypočítá pozice semilandmarků protínáním křivky paprsky vedenými ze středu úsečky AB.

Stěžejní výpočty probíhají v následujících dvou statických metodách třídy PointSequence:


```
PointPolar[] MakeFunctionOfAngle(PointPolar[] pts,  
double dMin, double dMax)
```

pts – posloupnost bodů v polárních souřadnicích

dMin – nejnižší hodnota definičního oboru

dMax – nejvyšší hodnota definičního oboru

MakeFunctionOfAngle vytvoří z posloupnosti bodů pts posloupnost bodů, která je rostoucí v hodnotě úhlu. Tím je myšleno, že pokud $i > j$, pak $pts[i].angle > pts[j].angle$. Všechny vrcholy, které tuto podmínku nesplňují jsou vyřazeny. Navíc jsou vyřazeny všechny body, jejichž hodnota úhlu je mimo rozsah [dMin, dMax].

Pokud chybějí body s hodnotou úhlů dMin a dMax, pak se doplní na začátek a na konec posloupnosti pomocí konstantní extrapolace (přiřadí se jim stejná délka, jakou měly dosavadní koncové body).

Posloupnost bodů splňující všechna uvedená omezení má tu vlastnost, že pokud se sousední body spojí úsečkami, je výsledná lomená čára (v polárních souřadnicích) funkcí úhlu na intervalu [dMin, dMax]. To znamená, že pro libovolnou hodnotu úhlu z intervalu [dMin, dMax] existuje právě jeden bod lomené čáry, který má tuto hodnotu úhlu.

```
PointPolar[] InterpolateFunctionOfAngle(PointPolar[] pts,  
int iSamples)
```

pts – posloupnost bodů v polárních souřadnicích

iSamples – počet vzorků, kde se bude interpolovat hodnota funkce

InterpolateFunctionOfAngle získá posloupnost bodů v polárních souřadnicích, které jsou prvky funkce určené posloupností bodů pts. Nezávislou proměnnou této funkce je úhel, závislou potom vzdálenost od počátku. Pro doplnění definičního oboru na všechny racionální úhly se posloupnost bodů doplní na lomenou čáru (počítají se průsečíky s úsečkami).

```
void ComputeRadiiByAngle(int iPoints)
```

iPoints – počet semilandmarků (paprsků / průsečíků s křivkou)

Tato metoda je obměnou funkce ComputeSemilandmarksByAngle. Místo posloupnosti průsečíků vezme pouze posloupnost jejich vzdáleností. Úhel je určen implicitně pořadím každého bodu a celkovým počtem bodů.

```
void ComputeSemilandmarksByArc(int iPoints)
```

iPoints – počet semilandmarků rozdělujících křivku (včetně dvou krajních landmarků)

ComputeSemilandmarksByArc vypočítá pozice semilandmarků, které mají mezi sebou stejně dlouhé úseky křivky.

Nejprve se zjistí vzdálenost každého bodu posloupnosti od bodu A (vzdálenost po křivce). Následně se najde (interpoluje) n bodů ($n = iPoints$), které mají mezi sebou konstantní vzdálenost, tedy vzdálenost i-tého bodu od počátku křivky je $i \cdot l$, kde l je vzdálenost mezi dvěma sousedními body.

K výpočtu pole vzdáleností všech bodů slouží následující statická metoda třídy PointSequence:

```
double[] CurveArc(PointD[] pts)
```

pts – normalizovaná posloupnost bodů

CurveArc vrátí posloupnost vzdáleností od počátku křivky, tedy nějaké pole q, splňující následující: pokud je i indexem pole pts, pak q[i] je vzdálenost bodu pts[i] od začátku křivky. (Více o vzdálenosti bodu od počátku křivky v kapitole 3.2.2)

```
void ComputeCircularHarmonicsFourier(int iCoeffs)
```

iCoeffs – počet potenciálně nenulových fourierových koeficientů

Stěžejní část práce funkce ComputeCircularHarmonicsFourier se uskutečňuje v metodě třídy CircularHarmonics:

```
double[] CountCoefficients(int iCoefficients, PointD[] pts,  
Ebasis enBasis)
```

iCoefficients – počet koeficientů, které se budou počítat

pts – normalizovaná posloupnost bodů

enBasis – typ báze (Fourierovy řady nebo Legendrovy polynomy)

CountCoefficients nejprve převede posloupnost pts na polární souřadnice. S vypočtenými body se zavolá funkce MakeFunctionOfAngle (popsaná výše) a poté InterpolateFunctionOfAngle (též popsaná výše). Dostaneme navzorkovanou posloupnost bodů q v polárních souřadnicích, v konstantních úhlových rozestupech, která je navíc funkcí úhlu.

V dalším kroku vypočteme projekci funkce q na bázi b.

Pokud má enBasis hodnotu EBasis.eLegendre, pak b je množina Legendrových polynomů. Pokud má enBasis hodnotu EBasis.eFourier, pak b je množina harmonických funkcí (část Fourierovy posloupnosti).

Jiné hodnoty enBasis nejsou přípustné.

Projekce se vykoná ve funkci FourierBasis.CountCoefficients, která vrátí vypočtenou množinu koeficientů.

Neboť je vstupní posloupnost Π -periodická, bude polovina koeficientů nulová (rozebráno v kapitole 3.3.1). Budeme s tím dopředu počítat a vrátím „komprimovanou množinu koeficientů“ – nulové koeficienty vynecháme (neboť nenesou žádnou informaci).

(Více o Fourierových řadách v kapitole 3.3.1)

```
void ComputeCircularHarmonicsLegendre(int iCoeffs)
```

iCoeffs – počet potenciálně nenulových koeficientů legendrových polynomů

ComputeCircularHarmonicsLegendre zajišťuje výpočet koeficientů Legendrových polynomů. Výpočet probíhá, stejně jako při výpočtu Fourierových koeficientů, v metodě třídy CircularHarmonics CountCoefficients, popsané výše, tentokrát ale s parametrem enBasis nastaveným na hodnotu EBasis.eLegendre, která říká, že se použije báze sestávající z Legendrových polynomů.

Samotná projekce na bázi Legendrových polynomiálních funkcí se vykoná v metodě LegendreBasis.CountCoefficients, která vrátí vypočtenou množinu koeficientů.

(Více o Legendrových polynomech v kapitole 3.3.2)

```
void ComputeNormalPolyline(int iCoeffs)
```

iCoeffs – počet koeficientů, který se bude počítat

Tato funkce zajišťuje výpočet křivky pomocí normálové konstrukce. Výpočet probíhá ve statické metodě `PointSequence.NormalPolylineSegment`:

```
void NormalPolylineSegment (ref PointD[] pts, int starti,  
int endi, PointD A, PointD B, ref double[] result,  
int resulti, int resultSize)
```

pts – vstupní posloupnost bodů v normalizovaných souřadnicích

starti – počáteční index vybrané podposloupnosti odkazující na bod z pts

endi – první index za koncem vybrané podposloupnosti

A – počáteční bod

B – koncový bod

result – výsledná množina koeficientů

resulti – index do result, první prvek, který bude přepočítán

resultSize – počet prvků result, které se mají přepočítat

`NormalPolylineSegment` převede vybranou podposloupnost pts, určenou parametry starti a endi na podmnožinu koeficientů result, určenou parametry resulti a resultSize.

Funkce vypočítává koeficienty rekurzivně. Při prvním volání z vnějšku je vybraná podposloupnost bodů i koeficientů rovna celé posloupnosti a body A, B odpovídají koncovým bodům křivky.

Provede se iterace (výpočet prvního trojúhelníku), parametry se rozdělí podle vypočtené normály a pokračuje se rekurzí do dvou nových větví, dokud se nedojde k počtu koeficientů 1. (Více o normálové konstrukci v kapitole 3.2.3)

II.4.1 Normalizace a „denormalizace“ křivky

K převedení křivky z pixelové reprezentace do normalizovaného tvaru slouží funkce:

```
PointD[] NormalizePointSequence(Point[] pts)
```

pts – posloupnost pixelů (celočíslné hodnoty)

Tato funkce vypočte a vrátí normalizovaný tvar posloupnosti pixelů. Hodnoty vrácených souřadnic jsou typu double.

Pro výpočet potřebujeme znát souřadnice landmarků (body A, B). Bod A je prvním bodem sekvence pts, bod B posledním bodem.

`NormalizePointSequence` transformuje systém obrazkových souřadnic do souřadného systému s počátkem ve středu úsečky AB, osou x ve směru orientované úsečky AB a jednotkovou délkou rovnou délce |AB|. (Postup výpočtu uveden v kapitole 3.1)

Zpětnou transformaci na pixelovou reprezentaci provedeme pomocí funkce:

```
Point[] FitToBaseLine(PointD[] pts, PointD A, PointD B,  
bool bFlip)
```

pts – posloupnost normalizovaných bodů
A, B – koncové body křivky (landmarky)
bFlip – příznak pro překlopení křivky

FitToBaseLine převede systém souřadnic, uvedený v popisu předchozí funkce, do obrazkových souřadnic (střed v levém horním rohu obrázku, osa x rovnoběžná s horním okrajem obrazovky, pixel délky 1) a zaokrouhlí hodnoty na celá čísla.

Převod sestává z inverzních transformací použitých ve funkci NormalizePointSequence v opačném pořadí (tedy v pořadí otočení, změna měřítka, posun).

Pokud je parametr bFlip true, předřadí se navíc těmto transformacím ještě operace zrcadlení.

Parametr bFlip se v aplikaci nastavuje na true, pokud je posloupnost nasegmentovaných pixelů seřazena proti směru hodinových ručiček. (Tuto posloupnost máme vždy k dispozici, neboť se ukládá do textového souboru)

II.4.2 Vykreslení aktivní reprezentace

Do obrázku se kromě landmarků dále vykresluje znázornění aktivní reprezentace křivky.

Pixels

Všechny pixely křivky se zvýrazní žlutou barvou.

Semilandmarks by angle

Nejprve se provede denormalizace semilandmarků, pak se všechny vykreslí jako křížky, navíc se znázorní úsečky spojující počátek a průsečíky.

Radii by angle

Vykreslí se všechny paprsky vycházející z počátku a protínající křivku. Denormalizace je zde jednodušší, stačí pouze změnit délku paprsku. Poloha středu a poloha bodů A, B je známá, takže se paprsky pouze vykreslují ve správném pořadí po konstantním úhlovém rozestupu, který se vypočítá z jejich počtu.

Semilandmarks by arc

Posloupnost semilandmarků se nejdříve nenormalizuje, jednotlivé semilandmarky se pak vykreslí jako křížky. Navíc se ještě vykreslí křivka v pixelové reprezentaci.

Normal Polyline

Vykreslování, stejně jako výpočet, probíhá rekurzivně. Nejprve se tenkou úsečkou spojí koncové body křivky A, B. Poté se přepočte délka normály (výšky trojúhelníku) na pixelové souřadnice a normála se vykreslí jako zvýrazněná úsečka.

Dále se pokračuje ve vykreslování v dceřiných větvích s novými koncovými body. Místo bodů A, B jsou pro nové úseky koncové body A, C a C, B, kde C je vrchol náležející normále.

Fourierovy a Legendreovy koeficienty

Nejprve se provede inverzní transformace – z posloupnosti koeficientů do posloupnosti bodů v normalizovaných souřadnicích. Následně se provede denormalizace a vykreslení křivky (nyní už posloupnosti pixelů). Inverzní transformace se provádí pomocí následující statické metody třídy CircularHarmonics:

```
PointD[] ApproximateCurve(double[] arCoeffCompressed,  
Basis enBasis)
```

arCoeffCompressed – množina koeficientů (komprimovaná, jsou-li Fourierovy)
enBasis – typ báze (Fourierovy řady nebo Legendrovy polynomy)

Tato funkce vypočte posloupnost bodů v polárních souřadnicích jako lineární kombinaci koeficientů a příslušné báze. V případě Fourierových koeficientů proběhne ještě proložení posloupnosti arCoeffCompressed nulovými koeficienty.

Vypočtené body se převedou do kartézských souřadnic (normalizovaných) a pošlou volající funkci jako návratová hodnota.

Pozn. neboť jsou bazové funkce spojité na celém definičním oboru, lze vypočítat hodnotu pro jakýkoli úhel. Funkci ApproximateCurve používáme pouze pro vykreslení křivky na obrazovku, takže vzorkování přizpůsobíme tak, aby byla výsledná křivka dostatečně hustá. Byla zvolena konstantní hodnota 1000 vzorků (pro většinu testovaných obrázků jde vzdálenost dvou sousedních bodů pod hranici velikosti pixelu, což je pro vykreslení dostačující).

Literatura

- [1] Brigham E. O.(1974): The Fast Fourier Transform. Englewood Cliffs, Prentice Hall, New Jersey
- [2] Čížek V. (1981): Diskrétní Fourierova transformace a její použití, SNTL, Praha
- [3] Gonzalez R. C., Woods R. E. (2008): Digital Image Processing, Third Edition, Pearson Prentice Hall, New Jersey
- [4] Guskov I., Vidimče K., Sweldens W., Schröder P. (2000): Normal Meshes, SIGGRAPH '00 proceedings, ACM Press, New York
- [5] Jolliffe, I. T. (1986): Principal Component Analysis, Springer, New York
- [6] Martinez A. M., Kak A. C. (2001): PCA versus LDA, IEEE Transactions on Pattern Analysis and Machine Intelligence vol. 23
- [7] Pretorius E., Steyn M., Scholtz Y. (2006): Investigation Into the Usability of Geometric Morphometric Analysis in Assessment of Sexual Dimorphism, American Journal of Physical Anthropology
- [8] Rieger B., Van Vliet L. J.(2003): Representing Orientation in N-Dimensional Spaces, Springer, Berlin
- [9] Töpfer P. (1995): Algoritmy a programovací techniky, Prometheus, Praha
- [10] Velemínska J. et al. (2010): Geometric Morphometric Approach to Sex Determination, Using the Greater Sciatic Notch of the Maxwell Museum Documented Skeletal Collection, to appear in AMAAPA '10, Albuquerque, New Mexico
- [11] Žára J., Beneš B., Sochor J., Felkel P. (2004): Moderní počítačová grafika, 2. vydání, Computer press, Praha

Internetové odkazy

- [12] Circular & spherical harmonics, http://commons.wikimedia.org/wiki/Spherical_harmonic
- [13] Documentation of the Morphome3cs project, <http://cgg.mff.cuni.cz/trac/morpho>
- [14] Legendre polynomials, http://en.wikipedia.org/wiki/Legendre_polynomials