



**MATEMATICKO-FYZIKÁLNÍ
FAKULTA**
Univerzita Karlova

DIPLOMOVÁ PRÁCE

Kateřina Boková

Statistický model tvaru obličeje

Katedra softwaru a výuky informatiky

Vedoucí diplomové práce: RNDr. Josef Pelikán

Studijní program: Informatika

Studijní obor: Softwarové systémy

Praha 2019

Ráda bych poděkovala vedoucímu práce za dobré rady a věnovaný čas. A také bych chtěla poděkovat Martinu Pilátovi za korekturu textu.

Prohlašuji, že jsem tuto diplomovou práci vypracovala samostatně a výhradně s použitím citovaných pramenů, literatury a dalších odborných zdrojů.

Beru na vědomí, že se na moji práci vztahují práva a povinnosti vyplývající ze zákona č. 121/2000 Sb., autorského zákona v platném znění, zejména skutečnost, že Univerzita Karlova má právo na uzavření licenční smlouvy o užití této práce jako školního díla podle § 60 odst. 1 autorského zákona.

V dne.....

podpis

Název práce: Statistický model tvaru obličeje

Autor: Kateřina Boková

Katedra / Ústav: Katedra softwaru a výuky informatiky

Vedoucí diplomové práce: RNDr. Josef Pelikán, Katedra softwaru a výuky informatiky

Abstrakt: Cíl této práce je použití metod strojového učení na dataset oskenovaných tváří a vytvořit tak program, který umožní prozkoumat a editovat tváře reprezentované trojúhelníkovou sítí pomocí několika ovládacích prvků. Nejprve jsme museli snížit dimenzi trojúhelníkových sítí pomocí PCA a pak jsme se pokusili předpovídat tvar sítě podle fyzických vlastností jako je váha, výška, věk a BMI. Modelované tváře mohou být pak použity v animacích nebo hrách.

Klíčová slova: model obličeje, trojúhelníková síť, PCA, redukce dimenze

Title: Statistical model of the face shape

Author: Kateřina Boková

Department: Department of Software and Computer Science Education

Supervisor: RNDr. Josef Pelikán, Department of Software and Computer Science Education

Abstract: The goal of this thesis is to use machine learning methods for datasets of scanned faces and to create a program that allows to explore and edit faces represented as triangle meshes with a number of controls. Firstly we had to reduce dimension of triangle meshes by PCA and then we tried to predict shape of meshes according to physical properties like weight, height, age and BMI. The modeled faces can be used in animation or games.

Keywords: face shape model, triangle mesh, PCA, dimension reduction

Obsah

Úvod.....	1
1 Základní pojmy.....	2
2 Předchozí práce a cíle.....	4
2.1 Předchozí práce.....	4
2.1.1 Korespondence sítí.....	4
2.1.2 Statistický model tváře.....	4
2.1.3 Atributy (vlastnosti) jedince a jejich modelování.....	7
2.1.4 Ověření statistického modelu.....	11
2.1.5 Nelineární model.....	13
2.1.6 Využití modelu.....	13
2.1.7 Longitudinální data.....	14
2.1.8 Počet komponent.....	15
2.2 Cíle.....	16
3 Reprezentace povrchu obličeje.....	17
3.1 Získání dat.....	17
3.2 Obsah datasetu.....	18
3.3 Obj formát.....	19
4 Analýza hlavních komponent (PCA).....	21
4.1 Matematický základ.....	21
4.1.1 Vlastní čísla a vektory.....	21
4.1.2 Lagrangeovy multiplikátory.....	22
4.2 Co je PCA a k čemu se používá.....	23
4.3 Výpočet.....	24
4.3.1 Maximalizace rozptylu.....	24
4.4 Výpočet PCA pro data s velkou dimenzí.....	27
4.5 Kolik PCA komponent je třeba?.....	28
5 Regrese.....	30
5.1 Lineární regrese.....	30

5.2	Metoda nejmenších čtverců.....	31
5.2.1	Choleského rozklad.....	33
5.2.2	QR dekompozice.....	34
5.3	Modely jedné vstupní proměnné a polynomiální aproximace.....	35
5.3.1	Lineární model.....	35
5.3.2	Vyšší stupeň.....	36
5.4	Vážená polynomiální regrese s jednou vstupní proměnou.....	37
6	Analýza tvaru obličeje.....	39
6.1	Uživatelské předzpracování.....	39
6.1.1	Normalizace velikosti.....	39
6.1.2	Filtr podle pohlaví.....	39
6.2	Modelování obličeje pomocí regrese.....	40
6.3	Karikatura.....	41
6.4	Vážená regrese.....	41
7	Software pro modelování a vizualizaci tvaru obličeje.....	42
7.1	Menu.....	43
7.1.1	Záložka File.....	43
7.1.2	Záložka View.....	46
7.2	Zobrazující okna.....	48
7.3	Nastavení nad zobrazujícími okny.....	48
7.4	Nastavení pod zobrazujícími okny.....	49
7.5	PCA scores.....	50
7.6	Cartoon.....	51
7.7	Regression.....	52
7.8	Vstupní data.....	53
8	Experimenty, porovnávání a validace.....	54
8.1	Aproximace obličeje pomocí PCA.....	54
8.1.1	Komponenty.....	54
8.1.2	Validace.....	60
8.2	Modelování vlastností.....	63

8.3 Shrnutí.....	69
9 Zajímavé implementační detaily.....	71
9.1 Výkon PCA.....	71
9.2 Track bar.....	72
9.3 Dvě OpenGL okna.....	73
Závěr.....	75
Příloha.....	81
Návrh programu.....	81
Data.....	82

Úvod

Dnešní doba by se mohla s trochou nadsázky označit za dobu datovou. Získávat další a přesnější data s pomocí nových či vylepšených technologií je stále snazší a snazší. A i novodobé počítače již disponují výpočetní silou, jež takové množství dat dokáže v rozumném čase zpracovat a získat tak z těchto dat nějaké zajímavé informace či vzájemné souvislosti, jež lze dále aplikovat v nejrůznějších oborech.

Konkrétně v této práci se budeme zabývat modelováním 3D objektů v závislosti na jejich vlastnostech. Nejdříve se pokusíme snížit počet proměnných reprezentující povrch lidské tváře, což je náš 3D objekt. K tomuto účelu použijeme analýzu hlavních komponent (PCA), která by měla najít novou nekorelovanou reprezentaci dat vyjádřenou menším počtem proměnných. Poté se pokusíme nalézt vztah mezi nově nalezenou reprezentací dat a dalšími vlastnostmi jedinců, pomocí různých typů regrese. Pro tyto účely máme k dispozici dataset 298 tváří skutečných lidí, jež byly naskenované 3D skenerem a následně zpracovány do 3D meshu, která reprezentuje povrch těchto tváří jako síť malých trojúhelníků. Všechny tváře obsahují 15003 vrcholů, tj. každá tvář je reprezentovaná pomocí 45009 reálných čísel.

Cílem této práce bylo vytvořit program, jenž by uživateli umožňoval zobrazit načtené tváře a modifikovat jejich trojúhelníkové sítě pomocí zjednodušeného modelu a spočtených závislostí mezi sítí a dalšími vlastnostmi jedince. Dále by program měl uživateli umožnit pohodlné prohlížení modelovaných tváří a vizuální porovnání rozdílů mezi jednotlivými tvářemi a jejich modifikacemi.

V tomto textu si nejprve popíšeme, jak vznikl náš dataset a jaké má vlastnosti. A jelikož se jedná o poměrně rozsáhlá data, tak si dále popíšeme PCA, její použití, odvození a výpočetní trik, kterým výrazně zvýšíme výpočetní výkon. V následujících kapitolách si pak popíšeme možné propojení této nové reprezentace dat se zbývajícími vlastnosti 3D modelu tváře jako je věk, výška, hmotnost či BMI. V kapitole Uživatelská dokumentace si představíme vzniklý program pro účely modelování tváře a provedeme v něm několik experimentů. V poslední kapitole si pak popíšeme pár zajímavých implementačních problémů, jako byly problémy s časovou výkonností.

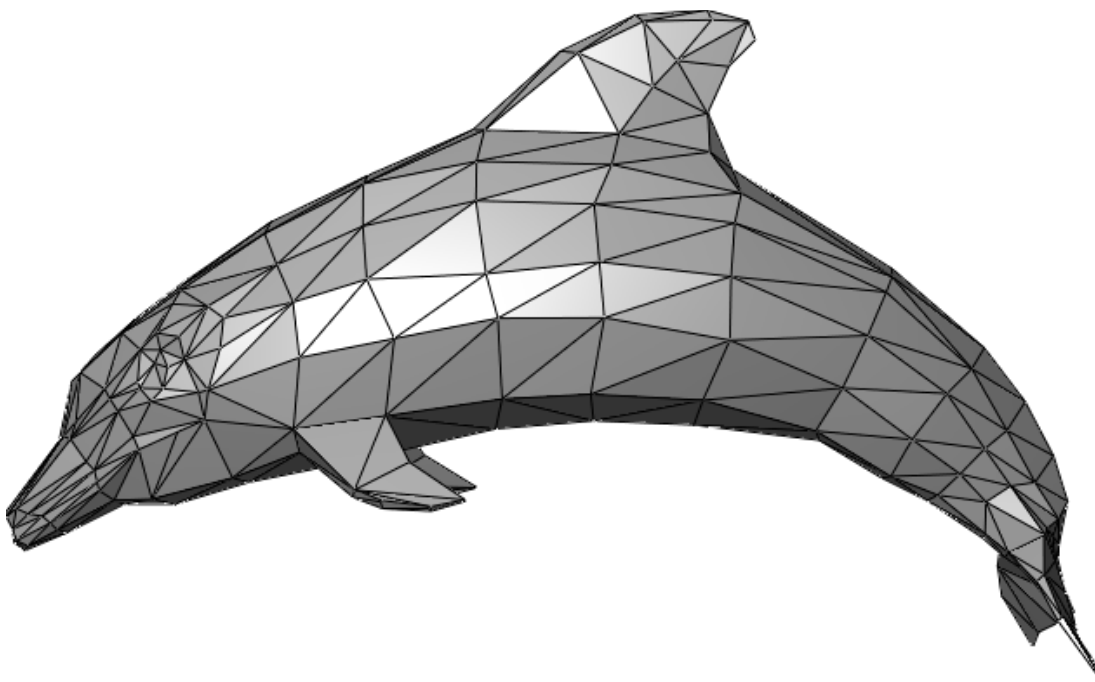
1 Základní pojmy

V této kapitole se seznámíme se základními pojmy, které nás budou provázet celým textem. Mezi tyto pojmy patří:

Vrchol - struktura uchovávající informaci o poloze bodu v 3D prostoru

Trojúhelníková síť/mesh – struktura reprezentující povrch 3D objektů pomocí vrcholů, které vytvářejí trojúhelníky různých velikostí. Tyto trojúhelníky jsou obvykle spojeny společnými hranami nebo rohy a následně pak reprezentují povrch 3D objektu. Obrázek 1.1 zobrazuje objekt pomocí trojúhelníkové sítě.

Textura - fotografie nebo uměle vytvořený soubor barev, který je namapovaný na povrch 3D objektu, aby vytvořil realistický vzhled 3D objektu.

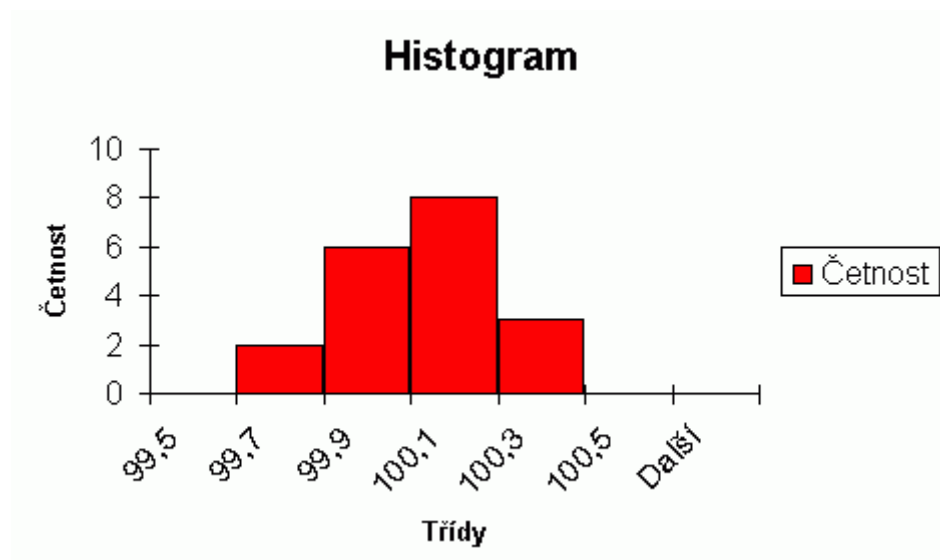


Obrázek 1.1: Ukázka 3D trojúhelníkové sítě(zdroj http://en.wikipedia.org/wiki/File:Dolphin_triangle_mesh.png)

BMI – body mass index (přeložena jako index tělesné hmotnosti) je číslo, které porovnává hmotnost lidí s různou výškou. Vztah je vyjádřen rovnicí 1.1 . Následná interpretace čísla je závislá na pohlaví a věku ale i na rase či zda se jedná o sportovce. Vypočtená hodnota je pak kategorizována do skupin podvýživa, ideální váha, nadváha a obezita.

$$BMI = \frac{\text{hmotnost v kilogramch}}{\text{výška v metrech}^2} \quad (1.1)$$

Histogram – struktura vyjadřující četnost hodnot v daných intervalech. Tyto četnosti se nejčastěji vyjadřují pomocí tabulky nebo sloupcového grafu, kde velikosti sloupců znázorňují četnosti hodnot v intervalech. Obrázek 1.2 znázorňuje histogram pomocí sloupcového grafu.



Obrázek 1.2: Ukázka histogramu pomocí grafu (zdroj <http://www.ikvalita.cz/tools.php?ID=24>)

2 Předchozí práce a cíle

Vytváření statistických 3D modelů na základě reálných dat začalo se vznikem technologií, které toto umožňovaly. Především mluvíme o 3D skenerech, které dokáží zachytit 3D tvar skenovaného objektu a barevnost jeho povrchu a tím vytvořit realistický vzhled obličeje.

2.1 Předchozí práce

V této podkapitole budou v krátkosti zmíněné související práce na poli modelování tvaru obličeje.

Jeden ze statických modelů, který těchto dat využíval, byl zaveden v práci A Morphable Model for the Synthesis of 3D Face (Blanz a Vetter, 1999) a byl pojmenován jako 3D Morphable Model (3DMM). Tvorbu 3DMM doprovází dva základní kroky - vzájemná korespondence mezi sítěmi a texturami a následná redukce dimenze.

2.1.1 Korespondence sítí

Samotná korespondence je zajímavým a obtížným problémem. Existují různé přístupy pro řešení tohoto problému – založené na Thinplate spline funkcích a Coherent point drift algoritmu nebo B-spline. Základní myšlenkou je nalezení vrcholů se stejnou anatomickou vlastností. K tomu je vybrán jeden z objektů jako šablona. Pak se provádí registrace šablony na vícero cílů, což vytvoří konzistentní parametrizaci ve všech naskenovaných datech. Některé přístupy vyžadují manuální vyznačení význačných bodů jako je například špička nosu, koutky očí a dalších, které urychlí konvergenci příslušných algoritmů. Jiné metody jsou plně automatizované. Touto problematikou pro různé reprezentace dat se například zabývá disertační práce Correspondence Problem in Geometrics Morphometric Tasks (Krajíček, 2015). Autor této práce se podílel na tvorbě nástroje Morphome3cs, kterým byla předzpracována data do podoby použitelné pro tuto práci. Nicméně tento problém není cílem této práce, a tak se nyní zaměříme na druhý krok.

2.1.2 Statistický model tváře

V práci Blanze a Vettera (Blanz a Vetter, 1999) bylo pro vytvoření modelu použito

200 oskenovaných obličejů (100 žen a 100 mužů) středního věku a stejné rasy. Po předzpracování měl každý skenovaný obličej zhruba 75 000 vzájemně si korespondujících vrcholů v 3D prostoru. Data byla reprezentována následující strukturou zahrnující tvar tváře s a texturu tváře t .

Tvar tváře je reprezentován následovně

$$s = (x_1, y_1, z_1, x_2, y_2, z_2, \dots, x_n, y_n, z_n)^T \in \mathbb{R}^{3n}, \quad (2.1.1)$$

kde n je počet vrcholů a s je vektor souřadnic vrcholů reprezentující povrch tváře. Vektor s je pak nazýván tvarovým vektorem.

Textura je také reprezentovaná jako vektor, ale tentokrát barev v barevném systému RGB. Dále se předpokládá, že textura má stejný počet hodnot jako je počet vrcholů. Texturu lze vyjádřit takto

$$t = (R_1, G_1, B_1, R_2, G_2, B_2, \dots, R_n, G_n, B_n)^T \in \mathbb{R}^{3n}, \quad (2.1.2)$$

kde n je počet vrcholů a t je vektor reprezentující barvy textury. Vektor t je pak nazýván texturovým vektorem.

Nejběžnější technikou pro redukci dimenze je PCA (Principal Component Analysis), na kterou lze také nahlížet jako na mnohorozměrné Gaussovo rozdělení a která je také použita v práci Blanze a Vettera (Blanz a Vetter, 1999). Model vzniklý z PCA lze pak vyjádřit následujícím vzorcem

$$s_{model} = \bar{s} + \sum_{i=1}^{m-1} \alpha_i u_i = \bar{s} + U \alpha, \quad (2.1.3)$$

kde \bar{s} je průměrný tvar obličeje a u_i jsou vlastní vektory kovarianční matice.

Analogicky je pak definována transformace pro texturu

$$t_{model} = \bar{t} + \sum_{i=1}^{m-1} \alpha_i^t v_i = \bar{t} + V \alpha^t, \quad (2.1.4)$$

kde \bar{t} je průměrný tvar obličeje a v_i jsou vlastní vektory kovarianční matice textur. Mnohé z následujících článků pracují s texturou, ale my se spíše zaměříme na tvar,

jelikož je relevantnější k tématu této práce. Koeficienty α lze z dat získat následujícím způsobem

$$\alpha = U^T (s - \bar{s}). \quad (2.1.5)$$

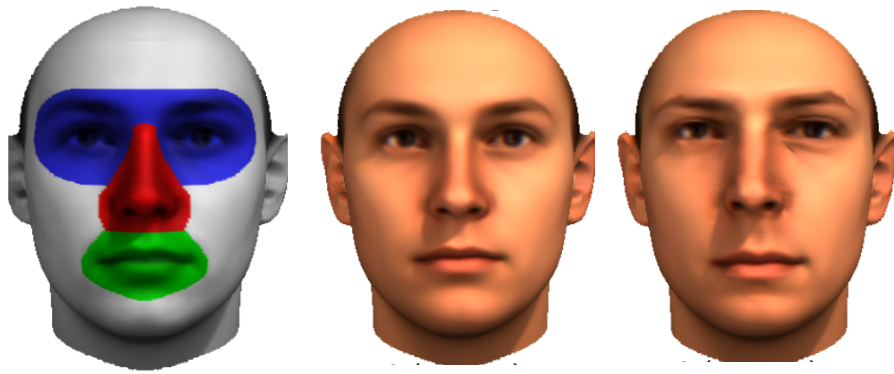
A rekonstrukce $R(s)$ tvaru s je zobrazena na dalším řádku

$$R(s) = \bar{s} + U U^T (s - \bar{s}). \quad (2.1.6)$$

Více o PCA se dozvíme v kapitole 4, kde si popíšeme její princip a výpočetní trik pro urychlení výpočtu nad rozsáhlými daty, kde dimenze prostoru je větší než počet vzorků.

Pro větší rozmanitost obličejů můžeme k modelu obličeje přistupovat po částech, tak jak to bylo popsáno v práci Blanze a Vettera (Blanz a Vetter, 1999). Tam byly vybrány oblasti tváře jako oči, nos, ústa a zbytek (což lze díky vzájemné korespondenci). Na obrázku 2.1.1 jsou tyto části znázorněné. Poté lze vytvořit modely těchto částí nezávisle na sobě. Dané částečné výsledky se pak opět spojí dohromady.

Tento přístup zejména umožňuje modelovat například tváře s jinou velikostí nosu bez závislosti na celkové velikosti hlavy, protože náš celkový vzhled definuje mnoho faktorů, jako je například genetická predispozice. Celkově je to zajímavý způsob jak rozšířit obecný model tváře, ale v této práci se budeme zabývat spíše globálním charakterem závislým na attributech tváře jako je věk, váha, výška a BMI.



Obrázek 2.1.1: Obrázek vlevo znázorňuje segmenty obličeje, které se dají modelovat nezávisle na sobě. Uprostřed je průměrná tvář a vpravo je tatáž tvář, kde byl ponechán tvar hlavy (šedá část z obrázku vlevo) a změněny byly oči, nos a ústa. Model textury je v případě pravé tváře nezměněn. Více variant lze nalézt v práci Blanze a Vettera (Blaiz a Vetter, 1999).

2.1.3 Atributy (vlastnosti) jedince a jejich modelování

Koeficienty modelu α a α^t ze vzorců 2.1.3 a 2.1.4 kódují variabilitu 3D tvaru a barvy čistě statistickým způsobem. Pro intuitivnější ovládání by bylo ideální tyto koeficienty modelovat na základě jiných atributů jako věk, výška nosu a dalších.

Základní myšlenkou nejpoužívanějších přístupů tohoto modelování atributů je, že závislost jednoho atributu $a \in \mathbb{R}$ a tváře $s \in \mathbb{R}^{3n}$ (případně $t \in \mathbb{R}^{3n}$) lze zaznamenat jako funkci, která je tvořena na základě reprezentace tváře α v prostoru tváří. Výsledek funkce je pak hledaný atribut

$$a = f(\alpha). \quad (2.1.7)$$

Funkce f může být libovolný klasifikátor. Nejjednodušším klasifikátorem může být lineární funkce, která byla navržena v Blanze a Vettera (Blaiz a Vetter, 1999). Výhodou lineární funkce je lehká invertibilita a je vhodnější pro menší počet vstupních vzorků. Nevýhodou je její přílišná obecnost. Problémem zejména je, že mnoho parametrů modelu odpovídá jen jednomu atributu. Vezmeme si například věk. Když si vezmeme skupinu lidí stejného věku, tak tváře všech lidí budou zcela rozdílné. Obecně řečeno přiřazení tváří k atributům je jedinečné, zatímco existuje mnoho tváří se stejným atributem.

To nás přivádí k složitějším klasifikátorům. Článek Prediction of Individual Non-Linear Aging Trajectories of Faces (Scherbaum a kol., 2007) se pokouší modelovat věk pomocí nelineární funkce v závislosti na identitě. K tomuto účelu využívá

nelineární Support Vector Regression s RBF jádrem

$$f(\alpha) = \sum_{i=1}^l \delta_i a_i e^{-\gamma \|\alpha_i - \alpha\|^2} + b, \quad (2.1.8)$$

kde δ_i a b jsou reálné číselné hodnoty naučené na trénovací množině. Soubor dat použitých v této práci obsahoval 393 osob ve věku mezi 95 a 360 měsíců. Při křížové validaci se rozdělil dataset na 90% tréninkových dat a 10% testovacích. Support Vector Regression se při učení prokázalo jako lepší klasifikační funkce než lineární regrese. Hlavní myšlenkou algoritmu bylo, že věková trajektorie $z(t)$ by měla být paralelní k gradientu věkové funkce $f(\alpha)$ v každém věku a nalezená trajektorie by také měla projít přes původní tvář α_0

$$\frac{d(z)}{d(t)}(t) = \nabla f(z(t)) \quad (2.1.9)$$

S počáteční testovací podmínkou

$$z(t_0) = \alpha_0, \quad (2.1.10)$$

kde α_0 je původní tvář ve věku t_0 .

V článku Scherbauma a kol. (viz. Scherbaum a kol., 2007) byly výsledné predikce reprezentovány pomocí obrázků. Z fotografie jedince v mladším věku byl odhadnut 3D model. (Tento postup byl popsán v práci Blanze a Vettera (Blanz a Vetter, 1999). Práce obsahuje algoritmus, který z jediné fotografie odvodí parametry již naučeného modelu a tak vytvoří věrohodnou 3D reprezentaci vyfoceného jedince. Algoritmus v rámci běhu také našel parametry jako osvětlení a pozice hlavy.) Následně se pomocí naučeného klasifikátoru nechal obličej zestárnout. Z fotografie toho samého jedince ve starším věku bylo pomocí stejného modelování odhadnuto osvětlení a další parametry jako pozice tváře. Z modelu, kterému byl nastaven věk podle starší fotografie, a s dodatečnými informacemi z analýzy starší fotografie byla následně sestavena fotografie s predikcí pro zkoumaného jedince. Tento postup sloužil zejména k vizuálnímu porovnání výsledků.

Výhodou výše zmíněného přístupu je, že trajektorie stárnutí závisí nelineárním

způsobem jak na věku, tak na individuálním obličejí, což koresponduje s tím, že různí jedinci se mění různými způsoby. Nevýhodou je, že tato metoda zpracovává pouze jeden atribut. V práci Amberga a kol. (Amberg a kol., 2009) se zabývají také manipulací atributů pomocí různých regresních modelů. Cílem bylo umožnit manipulaci s několika parametry zároveň. Základní myšlenkou je hledání takové hodnoty α , která by splňovala následující podmínku

$$\operatorname{argmin}_{\alpha} \|a - f(\alpha)\|^2, \quad (2.1.11)$$

kde a je atribut(y) a $f(\alpha)$ je naučená funkce z prostoru tváří do prostoru atributu z rovnice 2.1.7. Toto samotné ale nestačí, jelikož případných výsledků α může být hodně. Proto se musí přidat další omezení. Typickým příkladem omezení je penalizace za vzdálenost od průměrné tváře $\|\alpha\|^2$. V práci Amberga a kol. (Amberg a kol., 2009) se navíc k tomuto omezení používá i omezení $\|\alpha_{\text{originální}} - \alpha_{\text{hledaná}}\|^2$.

Mezi parametry ovlivňující model nemusí patřit jen zjištěné či naměřené hodnoty nějakých vlastností tváří, ale i samotný jiný model. V článku Face Reconstruction from Skull Shapes and Physical Attributes (Paysan, Lüthi a kol., 2009) a v disertační práci Statistical Modeling of Facial Aging based on 3D Scans (Paysan, 2010) je k dispozici model lebky, který je parametrizován pomocí vektoru β . Takže máme dva modely - model pro povrch tváře, kde α reprezentuje povrch tváře, a model pro lebku, kde β reprezentuje povrch lebky. Z magnetické rezonance a CT snímků bylo vytvořeno párování koeficientů na základě obou naučených modelů (více viz. Paysan, Lüthi a kol., 2009) a tak byl vytvořen dataset odpovídajících korespondujících dvojic (α_i, β_i) , na kterém lze naučit vztah pomocí multivariační lineární regrese. Dále byla každá tvář v datasetu doplněna o atributy $\mathcal{A} \in \mathbb{R}^l$, kde l je počet atributů. Protože každá lebka může mít několik odpovídajících tváří, tak je vhodné tyto atributy vzít v úvahu jako další omezení. Pak už jen zbývá zkombinovat přístup pomocí multivariační regrese a nelineární regrese. V podstatě se hledá α , které by splňovalo následující rovnici s minimální chybou

$$E(\alpha) = E_s(\alpha, \beta) + \lambda_1 E_a(\alpha, \mathcal{A}) + \lambda_2 E_p(\alpha), \quad (2.1.12)$$

kde λ_1 a λ_2 jsou váhy vyvažující vliv podmínek a $E_s(\alpha, \beta)$ je chyba modelu

lebký spočtená následovně:

$$E_s(\alpha) = \|M\alpha - \beta\|^2, \quad (2.1.13)$$

kde M je matice mapující vztah modelu povrchu tváře a lebký pomocí multivariační regrese.

$E_a(\alpha, \vartheta)$ je čtvercová chyba pro odhad parametrů z modelu, tj jak dobře predikuje model tváře zadané parametry. Tato chyba je vyjádřena vzorcem

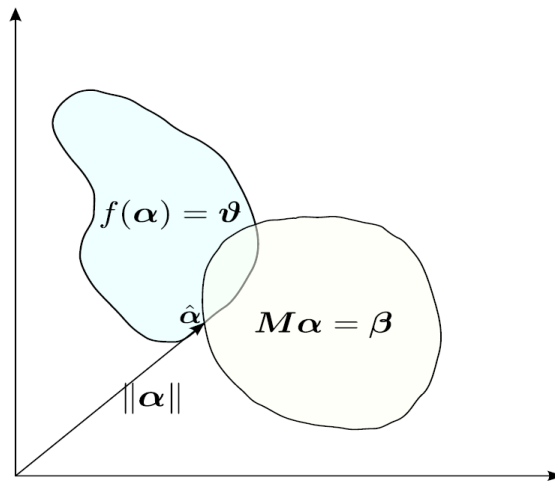
$$E_a(\alpha, \vartheta) = \sum_{i \in I} (w_i (f_i(\alpha) - \vartheta_i))^2, \quad (2.1.14)$$

kde $f_i(\alpha)$ je naučený klasifikátor např. support vector regression a w_i jsou normalizační faktor pro různé rozsahy mezi atributy.

A $E_p(\alpha)$ vynucuje validní tvář, protože se penalizuje větší vzdálenost od průměrné tváře následujícím způsobem:

$$E_p(\alpha) = \|\alpha\|^2. \quad (2.1.15)$$

Rovnice 2.1.12 se dá graficky zobrazit pomocí obrázku 2.1.2.



Obrázek 2.1.2: Zobrazuje průnik podmínek z rovnice 2.1.12 v prostoru tváří

V dalších člancích se experimentovalo s různými druhy klasifikátorů. Například parametrické vyjádření obličeje se objevilo i v článku Tsenga a kol. (Tseng a kol, 2014), kde se modeluje obličej na základě parametrů jako délka obličeje, šířka čelisti

a další. K článku byl vytvořen nástroj pro uživatelské virtuální testování brýlí na vymodelovaných tvářích podle zadaných parametrů. Obličej v softwaru byly modelovány pomocí lineární regrese a ne-lineárního Krigingova modelu, který upravuje poměr podle distribuce bodů, aby se generovaly přesnější odhady.

Další velmi pokročilý klasifikátor byl používán v článku Modeling 3D Facial Shape from DNA (Claes a kol., 2014) V práci se zabývají vztahem mezi tváří a pohlavím, původem a geny mající vliv na tvar obličeje. K tomu účelu měli dataset 600 tváří s západo-africkým a evropským původem ze tří lokalit – USA, Brazílie, Kapverdy. V této práci se používá nově vyvinutý BRIM klasifikátor (bootstrapped response-based imputation modeling), který je rozšířením regresních technik. Využívá závislou (výstupní) proměnnou klasifikátoru k upřesnění, filtrování a transformaci jedné nebo více vysvětlujících (vstupních) proměnných. Výstupem BRIM je nový typ proměnné RIP (response-based imputed predictor). Tato proměnná je přemostující proměnnou, která kombinuje informace jak z nezávislých proměnných, tak ze závislých proměnných. Více o tomto přístupu je napsáno v příloze text S1 v článku (Claes a kol., 2014). Výsledky vlivů genů, pohlaví či původu jsou zobrazeny pomocí tepelných map, kde se kromě Euklidovské vzdálenosti mezi vrcholy zobrazuje poměr plochy, posun normály a rozdíl zakřivení. Zajímavostí je, že jedním z hodnotících kritérií klasifikátoru byl lidský faktor. Lidé i klasifikátor vyplnili dotazník. V dotazníku byli tváře falešně zbarvené, aby se hodnotil pouze tvar obličeje. Dotazník obsahoval následující otázky:

- Je tato tvář ženská či mužská?
- Ohodnoťte ženské rysy od 1 do 7, kde 1 znamená velmi ženské rysy.
- Je tvář typu spíš černého afrického, africko-amerického, bílého, evropského, evropsko-amerického nebo smíšeného.
- Ohodnoťte poměr západo-afrických rysů od 0 do 100.

Hodnocení lidí a klasifikátoru bylo vyhodnoceno jako velmi korelované.

2.1.4 Ověření statistického modelu

Většina dříve zmíněných prací se zakládala jen na několika stovkách trénovacích dat a většinou na omezených datech - například věkem do 30 let, jako tomu bylo v práci Scherbauma a kol. (Scherbaum a kol., 2007). Autorům článku A 3D Morphable Model Learnt from 10,000 Faces (Booth a kol., 2016), se podařilo nasbírat velké množství naskenovaných tváří i s relativně velkou věkovou a rasovou

odlišností. Během návštěvy muzea nechalo oskenovat svůj obličej přes 10000 dobrovolníků. Žen a mužů bylo přibližně stejně a bylo oskenováno 82% bílé rasy a 9% asiátů a 3% černé rasy a 1% ostatních.

Ke zpracování takto rozsáhlých dat implementovali plně automatizované vytváření modelu včetně předzpracování. Vytvořili robustní přístup pro korespondenci mezi tvářemi. Takto velký a rozmanitý dataset umožňuje vytvářet specifické modely na základě například pohlaví nebo etnické příslušnosti, kde je tvarový rozdíl obličejů patrný na první pohled. Tyto specializované modely dosáhly lepších výsledků než globální model obsahující celý dataset. Všechny modely byly podrobeny testům na kompaktnost, obecnost a specifičnost. Tyto testy jsou typické pro tento typ úloh a neprovádí se jen na modelech tváří, ale i například na modech kostí (viz. Wang a Shi, 2017).

Kompaktnost měří variabilitu zachycenou modelem. Kompaktní model má schopnost vyjádřit data pomocí malého počtu proměnných.

Obecnost kvantifikuje schopnost modelu vyjádřit nový tvar, tj. pomocí modelu vytvořeného na tréninkovém datasetu vyjádříme vzorek z testovací sady a spočítáme jeho odlišnost ve smyslu Euklidovské vzdálenosti pro každý vrchol. Tuto vzdálenost následně zprůměrujeme přes všechny vrcholy a přes všechny testovací vzorky. Vyjde tak průměrná chyba na vrchol.

Specifičnost kvantifikuje schopnost modelu generovat náhodné tváře nastavením parametru α_i z rovnice 2.1.3 podle směrodatné odchylky parametru (λ_i určuje rozptyl parametru).

$$\alpha_i \in \langle -3\sqrt{\lambda_i}, 3\sqrt{\lambda_i} \rangle \quad (2.1.16)$$

Tento i -tý vygenerovaný vzorek s'_i se pak porovná s nejbližším vzorkem v testovacích datech tak, aby Euklidovská vzdálenost byla minimální. Přesněji řečeno se nalezne takový vzorek z testovacích dat, který je nejbliž podle průměrné Euklidovské vzdálenosti přes všechny vrcholy. Specifičnost pak určuje průměrná vzdálenost (chyba) přes všechny chyby vygenerovaných vzorků.

Následně článek porovnává účinnost svého globálního oproti jiným veřejným modelům. Stejně tak porovnává věkový klasifikátor. V obou testech má lepší výsledky.

2.1.5 Nelineární model

Nevýhodou 3DMM modelů bylo hlavně v minulosti získání dat, na jejichž základě byl postaven model. Ačkoliv 3D data jsou relativně vzácná, jejich 2D varianta v podobě fotografií je velmi rozšířená. Schopností využít tyto masivní, prakticky neomezené zdroje se zabývá článek Nonlinear 3D Face Morphable Model (Tran a Liu, 2018). V článku se navrhuje naučit nelineární 3DMM model s použitím pouze 2D obrazů. Lineární model se totiž dá považovat za malou neuronovou síť s pouze jednou vrstvou. Pro modelování obrázků pak použili hluboké neuronové sítě, jejichž účelem bylo získání parametrů projekce, tvaru a textury.

2.1.6 Využití modelu

Celkově vzato můžeme všechny výše zmíněné metody použít pro účely rozpoznávání obličejů. U rozpoznávání se řeší několik problémů.

Jedním z problémů jsou různá osvětlení a pozice hlav na fotografiích. Tímto tématem se například zabývá práce A 3D Face Model for Pose and Illumination Invariant Face Recognition (Paysan, Knothe a kol., 2009).

Další záležitostí je výraz obličeje. Na toto téma vznikla publikace Expression invariant 3D face recognition with a Morphable Model (Amberg a kol., 2008). Práce používá modelování výrazu, které bylo navrženo už v práci Blanze a Vettera (Blanz a Vetter, 1999) jako model vektoru rozdílu mezi neutrální tváří a výrazem

$$s_e = s_{neutralní} - s_{výraz} \quad (2.1.17)$$

Celkový model je pak definován následně

$$s_{model}(\alpha; \alpha_e) = \bar{s} + U \alpha + U_e \alpha_e \quad (2.1.18)$$

V práci se předpokládá, že změna výrazu je přenositelná z jedné tváře na druhou a není tedy potřeba mít dataset všech jedinců se všemi výrazy. Výsledkem z tohoto modelu byl lepší klasifikátor pro obličej s výrazem, a zároveň nebyla zhoršená výkonost na datech s neutrálním výrazem.

Jedním z důležitých parametrů osobnosti je věk, který má velký vliv na celkový

vzhled obličeje. Článek *Sexual dimorphism of facial appearance in ageing human adults: A cross-sectional study* (Mydlová a kol., 2015) se jako jeden z mála zabývá změnami tváře u dospělé populace podle PCA modelu. U dospělých se změny tvaru obličeje vyskytují především v měkkých tkáních, na rozdíl od dospívajících, kde lze pozorovat výrazně růstové změny zejména ve tvaru lebky. Tato studie zkoumá celkový tvar a formu obličeje. Forma je model PCA založený na datech, u kterých byla matematickým způsobem odstraněna velikost pomocí těžiště tváře. Přesněji byl zkoumán celkový dimorfismus mezi pohlavími i mezi věkovými skupinami 20 - 40, 41 - 60 a 61 - 82, jak na tvaru tak na formě. Na základě analýzy nejvýznamnějších komponent byli nalezeny například následující poznatky:

- Věk a pohlaví mají významný vliv na tvar a formu obličeje.
- Stárnutí u obou pohlaví probíhalo podobně, ale ženské tváře procházejí menší transformací.
- Značně výrazný pohlavní dimorfismus se projevil hlavně u tvaru obličeje – v každé věkové kategorii byla průměrná mužská tvář větší než ženská.

2.1.7 Longitudinální data

K lepšímu porovnání a i modelování by bylo dobré mít několik naskenovaných obličejů pro každého jedince s různými atributy. K těmto účelům slouží longitudinální data. Tato data se vytvářejí tím, že se ten samý jedinec nechává skenovat například každý rok, stejně jako tomu bylo i v práci *Simulation of facial growth based on longitudinal data: Age progression and age regression between 7 and 17 years of age using 3D surface data* (Koudelová a kol., 2019). Sehnat tyto datasey pro delší časové období je poměrně obtížné – už jen z důvodu, že se 3D modely začaly sbírat relativně nedávno. Dataset použitý v článku Koudelové a kol. (Koudelová a kol., 2019) tvořily dvě skupiny - děti ve věku 7 až 12 let (23 dívek a 17 chlapců) a dospívající od 12 do 17 let (25 dívek a 22 chlapců). K predikování byly vytvořeny trajektorie pro jedince pomocí lineární regrese. Pak se tyto trajektorie zprůměrovaly zvlášť pro dívky a pro chlapce v obou věkových skupinách. Následně se trajektorie z obou věkových skupin spojily pomocí Hermitovské kubické funkce. Tímto způsobem vznikly trajektorie pro obě pohlaví a na jejich základě byly vytvořeny predikce. V každé skupině, tj. pro obě věkové skupiny a obě pohlaví, se pro každého jedince spočetla predikce z jeho nejmladší podoby na nejstarší a ta se

pak porovnávala s originálem ve smyslu Euklidovské vzdálenosti mezi vrcholy. Výsledky byly zobrazeny pomocí tepelné mapy a byla spočtena průměrná chyba na predikci pro danou skupinu.

2.1.8 Počet komponent

Dřívěji zmíněné články často používají PCA modely. A u každého PCA modelu se určuje počet komponent, které mají reprezentovat danou tvář. Čím větší zvolíme počet komponent, tím přesnější bude naše rekonstrukce z rovnice 2.1.6. Když není určen správný počet hlavních složek pro následnou analýzu, tak jsou buď ztraceny relevantní informace (podcenění) nebo je zahrnut do rekonstrukce i šum (nadhodnocení). Výběr počtu komponent je dost závislý i na typu úlohy.

V článku Mydlové a kol. (Mydlová a kol., 2015) bylo pro určení počtu komponent použito pravidlo broken-stick, které je odvozeno na základě průměrné délky kusů tyče zlomené na p částí. Přesněji řečeno, pokud bychom zlomili mnoho tyčí, tak d_1 by byla průměrná délka nejdelších kusů, d_2 by byla průměrná délka druhých nejdelších kusů a tak dále. Pro k -tou komponentu je spočtena hodnota podle následujícího vzorce:

$$d_k = \frac{1}{p} \sum_{i=k}^p \frac{1}{i}, \quad (2.1.19)$$

kde p je počet proměnných. Pokud má k -tá komponenta hodnotu vlastního čísla větší než d_k , tak by měla být zachována.

Pravidlo broken-stick a další jiná pravidla jsou rozsáhle popsána a porovnána v práci How many principal components? stopping rules for determining the number of non-trivial axes revisited (Peres-Neto a kol., 2005).

Jiné přístupy zvolily počet komponent podle vysvětlujícího rozptylu (více viz podkapitola 4.5). V případě ne-lineární predikce věku (Scherbaum a kol., 2007) byly PCA komponenty pro pokusy zvoleny napevno na hodnoty 20, 40, 60 s odůvodněním, že poskytují dostatečný vysvětlující rozptyl. Pro porovnávání modelů v práci s velkým množstvím dat (Booth a kol., 2016) byli spočteny grafy pro prvních 80 komponent. Většina ostatních dříve zmíněných článků volila vyšší vysvětlující rozptyly někde mezi 90 až 100 procenty.

2.2 Cíle

V této práci se pokusíme na základě získaných vstupních dat vytvořit model. Jak už bývá zvykem, k základní redukci dat použijeme analýzu hlavních komponent (PCA). Pro věrohodnější předpovědi bude uživateli umožněno učit model pouze na ženských či mužských datech, protože podle Bootha a kol. (Booth a kol., 2016) mají specializované modely lepší účinnost a také proto, že byly prokázány významné rozdíly mezi pohlavími (více viz Mydlová a kol., 2015).

Na rozdíl od článků, které využívají k predikci klasifikátor a jeho následnou inverzi s různými podmínkami definující podobnost s původní tváří, jsme se rozhodli zvolit trochu odlišný způsob. Pomocí různých metod regrese se podíváme na vztah jednotlivých komponent k nějakému atributu. Tento regresní vztah pak použijeme k modifikaci parametrů modelu tak, abychom zachovali z velké části identitu jedince. A protože vlastnosti (atributy) jako je váha, věk či výška nejsou jediné parametry formující obličej (velký vliv na celkový vzhled má i naše genetická výbava), tak se výsledek predikce pokusíme vylepšit pomocí metod vážené regrese.

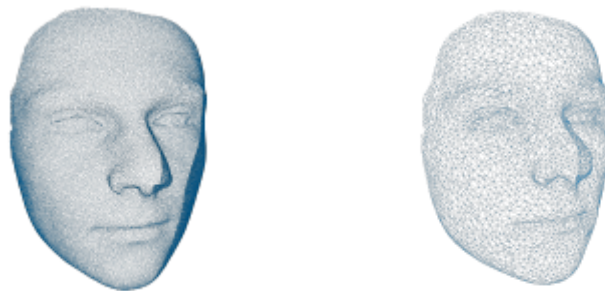
Získané výsledky se pokusíme zhodnotit. Můžeme například porovnávat lineární regresi s kvadratickou nebo zda vážená regrese je lepší než nevážená. Také se podíváme zda specializované modely mají větší účinnost. Cílem je zejména vytvořit uvěřitelné predikce. Co to vlastně znamená a jak toho docílit? V rámci práce vznikne program, který umožní měnit různé vlastnosti jedince a vizuálně v reálném čase zobrazit výslednou predikci. A jaké výsledky očekáváme? Na základě empirické zkušenosti očekáváme například (za předpokladu, že máme jen dataset dospělých jedinců), že pokud zvýšíme váhu, tak se obličej zakulatí a zejména tváře nabudou na objemu (v opačném případě se tváře propadnou), zatímco nos či oči by se měnit neměly. V případě výšky mají vyšší lidé většinou protáhlejší obličej. A s přibývajícím věkem se zejména mění povrch tváře vlivem gravitace, tvář je povadlá a tvoří se na ní různé záhyby.

3 Reprezentace povrchu obličeje

V této kapitole si ve zkratce popíšeme vznik našeho datasetu, jeho úpravu a formát, kterým jsou 3D objekty v datasetu reprezentovány.

3.1 Získání dat

Původní data byla pořízena na Katedře antropologie a genetiky člověka na Přírodovědecké Fakultě Karlovy Univerzity v rámci jiných výzkumných prací. Katedra antropologie a genetiky člověka disponuje 3D skenerem s označením Vectra3D. Tento skener, fungující čistě na optickém principu, je schopen zaznamenat povrch objektu, v našem případě tváře, jako tzv mrak 3D bodů (Obrázek 3.1.1 vlevo). Zároveň je během skenování pořízena pomocí fotoaparátu fotografie skenovaného objektu, která odpovídá příslušnému 3D modelu a vytváří tak texturu, která dotváří kompletní vizuální model naskenovaného objektu. Tento model pak vypadá velice realisticky. Textury v této práci nevyužijeme, jelikož nás spíše zajímá tvar tváře sám o sobě, než zbarvení jeho povrchu.

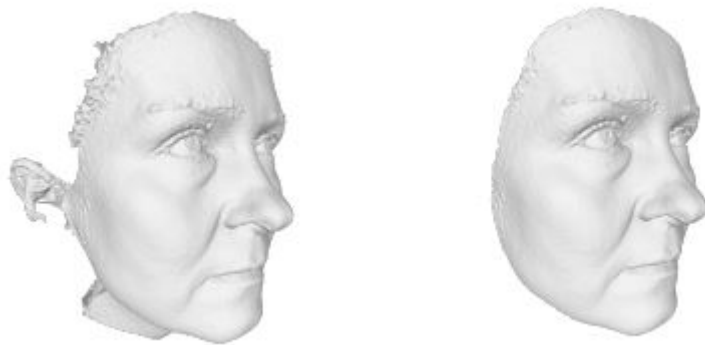


Obrázek 3.1.1: vlevo: tvář zobrazená pomocí mraku bodů, vpravo: tvář zobrazená pomocí trojúhelníkové sítě (zdroj Dvořáková, 2018)

Tato mraková reprezentace byla následně softwarově převedena na trojúhelníkovou síť (Obrázek 3.1.1 vpravo) a vyčištěna od případných chyb během skenování. Tyto chyby mohou vzniknout během skenování lesklých povrchů jako je oko. Ale i skenování řas, obočí, vousů a vlasů může skener vyhodnotit chybně jako součást povrchu, a pak na modelu tváře vznikají různé výstupky a defekty. Z těchto důvodů jsou tváře všech mužů v datasetu dokonale oholeny a vlasy během skenování musely být sepnuty. Dále se během předzpracování musely oříznout okraje tváře,

jelikož skener zaznamenává i části krku a uší (Obrázek 3.1.2). Podrobnější popis čistění sítě je popsán v bakalářské práci Nekvindy (Nekvinda, 2018)

Při poslední fázi předzpracování byla data v programu Morphome3cs topologicky normalizovaná podle jedné z tváří v datasetu. Tím se docílilo toho, že každá tvář je reprezentovaná stejným počtem vrcholů (15003), které si vzájemně topologicky odpovídají a vytváří tak síť 29713 vzájemně si odpovídajících trojúhelníků. Tato topologizace je zvláště výhodná během redukce dimenze dat, kde pomůže přesněji nalézt hlavní příznaky reprezentující data. O redukci dimenze si povíme více v následující kapitole.



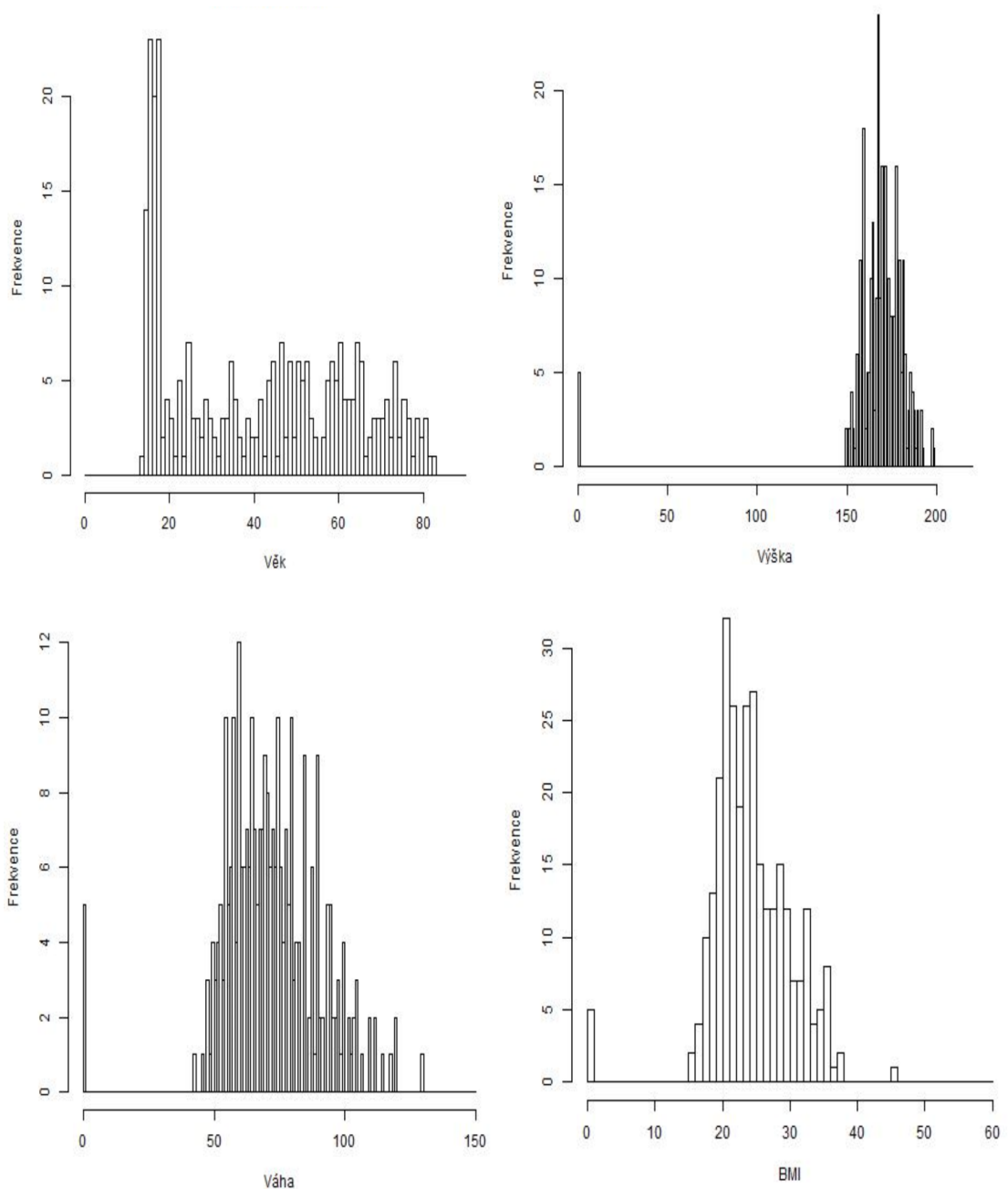
Obrázek 3.1.2: vlevo: tvář bez oříznutí, vpravo: tvář po oříznutí (zdroj Nekvinda, 2018)

3.2 Obsah datasetu

Dataset obsahuje tváře 298 lidí s neutrálním výrazem reprezentující českou populaci. Ke každé tváři je jsou přiloženy následující fyzické údaje jejich majitelů:

- Pohlaví - mužské pohlaví je reprezentované písmenem M (z anglického male) a ženské pohlaví písmenem F (z anglického female)
- Věk - je udán v letech
- Váha - je udána v kilogramech
- Výška - je udána v centimetrech

Zastoupení žen a mužů je přibližně stejné. Obrázek 3.2.1 zobrazuje zastoupení jednotlivých vlastností pomocí histogramů. (U některých osob nejsou všechny údaje známy a jejich hodnota je tedy nastavena na nulu)



Obrázek 3.2.1: histogramy vstupního datasetu následujícím pořadí zleva doprava: váha, BMI, věk, hmotnost

3.3 Obj formát

Obj formát, který vytvořila firma Wavefront Technologies, ukládá různé 3D objekty do textové podoby. Každý řádek začíná písmennou zkratkou (až na komentáře začínající mřížkou #), která určuje typ informací vyskytující se na zbytku řádky. Chceme-li například uložit vrchol 3D objektu, tak řádek bude označen

písmenem v (z anglického vertex) a za ním budou následovat 3 reálná čísla reprezentující vrchol jako 3D souřadnice bodu.

Pokud chceme reprezentovat objekt jako trojúhelníkovou síť, tak kromě vrcholů potřebujeme vyjádřit ještě jejich spojení do trojúhelníků. K tomu nám poslouží písmeno f (z anglického face) a za ním budou následovat indexy odkazující se na již vložený seznam vrcholů.

Tímto formátem se dále reprezentovat i další důležité informace. Mezi nejběžnější patří texturové souřadnice, které bodu přiřadí odpovídající souřadnici v obrázku, nebo vypočítané normály vrcholů. Další informace o tomto užitečném formátu lze najít v jeho oficiální dokumentaci (dokumentace .obj).

```
# Generated in Morphome3cs  
  
# 15003 vertices  
v 73.09008 8.4106 -46.86221  
v 25.12419 88.99742 4.161045  
...  
|  
# 29713 faces  
f 2911 3785 3787  
f 2809 3786 3785  
...
```

Obrázek 3.3.1: ukázka dat v datassetu

4 Analýza hlavních komponent (PCA)

V této kapitole si popíšeme PCA a její účel podle knih o strojovém učení (Bishop, 2006) a (Gareth a kol., 2017). Také si ukážeme odvození výpočtu PCA a nakonec si popíšeme výpočetní trik pro dataset, který obsahuje méně pozorování, než je dimenze vlastností jednoho pozorování. Ale úplně nejdříve si připomeneme pár základních matematických pojmů a metod, které budou zapotřebí pro odvození výpočtu PCA.

V rámci této a další kapitoly budeme používat následující terminologii:

- *Vlastnost pozorování* - číselné vyjádření velikosti nějaké sledované hodnoty pozorování. Pozorování zásadně obsahuje několik různých vlastností. Počet těchto sledovaných hodnot nazveme dimenzí vlastností.
- *Pozorování* - kolekce vlastností reprezentující nějaký objekt či událost
- *Dataset* – kolekce dat, která obsahuje pozorování o různých objektech či událostech stejného typu.

Níže popsaná PCA se stane základním kamenem našeho modelu popsaného v kapitole 6. Díky ní vytvoříme úspornější reprezentaci obličejů a to nám dovolí aplikovat metody z kapitoly 5 k získání důvěryhodných predikcí.

4.1 Matematický základ

Pro odvození výpočtu PCA budou použity následující matematické pojmy.

4.1.1 Vlastní čísla a vektory

Vlastní vektor matice A je nenulový vektor, jehož směr se po transformaci touto maticí nezmění. Přesná definice pak zní:

Definice: Bud' $A \in \mathbb{R}^{n \times n}$, pak $\lambda \in \mathbb{R}$ je *vlastní číslo* matice A a $u \in \mathbb{R}^n$ jemu příslušný *vlastní vektor*, pokud $Au = \lambda u, u \neq 0$.

Věta: Bud' $A \in \mathbb{R}^{n \times n}$. Pak

(1) $\lambda \in \mathbb{R}$ je vlastním číslem A právě tehdy, když

$$\det(A - \lambda I_n) = 0, \quad (4.1.1)$$

kde zkratka \det označuje determinant matice a I_n je jednotková matice.

(2) $u \in \mathbb{R}^n$ je vlastním vektorem příslušný k vlastnímu číslu $\lambda \in \mathbb{R}$ právě tehdy, když $u \neq 0$ a

$$(A - \lambda I_n)u = 0 \quad (4.1.2)$$

Zdrojem definice a věty je kniha o lineární algebře od Milana Hladíka (Hladík, 2017).

4.1.2 Lagrangeovy multiplikátory

Metoda Lagrangeových multiplikátorů je metoda nalézající extrémů diferencovatelné funkce za platnosti diferencovatelných omezujících podmínek, kterou publikoval Joseph-Louis Lagrange.

Věta: Necht' f, g_1, \dots, g_p jsou funkce třídy C^1 na otevřené množině G v Euklidovském prostoru $\mathbb{R}^n, n > p$. Mějme množinu M zadanou

$$M = \bigcap_{i=1}^p \{x \in \mathbb{R}^n \mid g_i(x) = 0\} \subset G. \quad (4.1.3)$$

Dále předpokládejme, že vektory $\text{grad } g_1(x), \text{grad } g_2(x), \dots, \text{grad } g_p(x)$ jsou lineárně nezávislé v všech bodech množiny M . Je-li bod $x_0 \in M$ bodem lokálního extrému funkce f vzhledem k množině M , pak existují taková čísla $\lambda_1, \lambda_2, \dots, \lambda_p \in \mathbb{R}$, že bod x_0 je stacionární bod tzv. Lagrangeovy funkce

$$L(x, \lambda) = f(x) - \sum_{i=1}^p \lambda_i g_i(x) \quad (4.1.4)$$

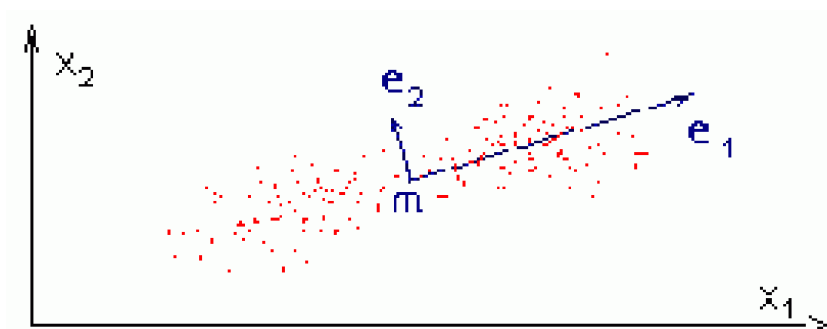
Čísla $\lambda_1, \lambda_2, \dots, \lambda_p$ jsou označovány za Lagrangeovy multiplikátory. Zderivováním Lagrangeovy funkce dostaneme n rovnic a spolu s rovnicemi podmínek tak vznikne soustava $p+n$ rovnic o $p+n$ neznámých. Jejím řešením jsou body podezřelé na extrémní hodnoty. Ty se pak ještě dalším vhodným postupem musí potvrdit a určit zda jde o maximum, minimum a jejich lokální či globální charakter, pokud vůbec jde o bod extrému.

Další informace o hledání extrému lze najít ve skriptech Diferenciální počet funkcí více proměnných (Hamhalter a Tišer, 2005).

4.2 Co je PCA a k čemu se používá

PCA (z anglického principal components analysis), také známá pod názvem Karhunen-Loève rozvoj, patří do skupiny metod tzv. redukce dimenze. Ideou redukce dimenze je snížení dimenze prostoru vlastností, protože ne všechny vlastnosti daného pozorování jsou stejně zajímavé a některé mohou být i na sobě navzájem lineárně závislé. Cílem redukce je tedy najít menší dimenzi reprezentující dané pozorování, která zachytí největší možné množství informace (Gareth a kol., 2017).

Výsledkem je reprezentace dat v jiném ortogonálním prostoru, tj. hledáme na sebe navzájem kolmé vektory v prostoru vlastností, podél kterých jsou původní data nejvíce proměnlivá. Tyto směry tvoří podprostor, který je nejbližší k datům (Obrázek 4.2.1).



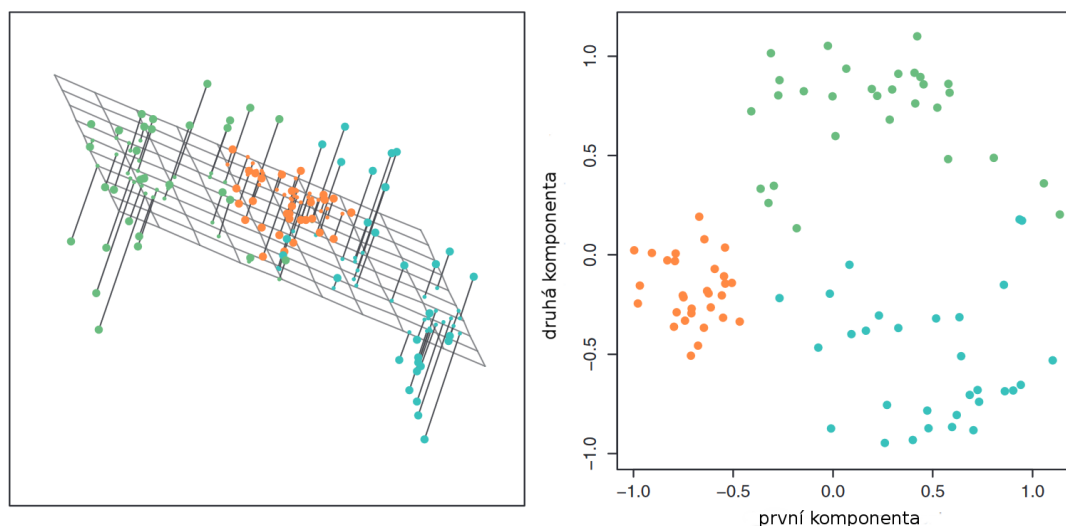
Obrázek 4.2.1: V grafu jsou zobrazená dvojrozměrná data. e_1 reprezentuje první nalezenou hlavní komponentu, e_2 druhou nalezenou hlavní komponentu (zdroj Militký)

PCA se řadí do kategorie učení bez učitele, a tak na rozdíl od učení s učitelem může být interpretace výsledků obtížnější a volba počtu hlavních komponent vhodných pro novou reprezentaci dat je těžší (více v kapitole 4.5)

Definice: Učení bez učitele je soubor metod, kde pro vstupní data není znám odpovídající výstupní údaj.

Definice: Učení s učitelem je soubor metod, kde pro vstupní data je znám odpovídající výstupní údaj.

Využití PCA je mnohostranné. Může posloužit jako nástroj pro vizualizaci vícerozměrných dat pomocí dvourozměrného grafu (Obrázek 4.2.2). Velmi často se používá jako metoda předzpracování před použitím dalších metod strojového učení. V tomto případě často hledáme úspornější vyjádření dat, ať už kvůli odstranění korelovaných hodnot nebo pro výkonnostní zlepšení při následujícím zpracování. PCA se dokonce dá použít pro ztrátovou kompresi dat (snížení velikosti dat, při kterém dojde k lehké ztrátě či poškození informace).



Obrázek 4.2.2: 90 pozorování s původní dimenzí vlastností 3. Vlevo : pozorování zobrazená v původní podobě 3D grafem - mřížka znázorňuje rovinu vytvořenou prvními dvěma hlavními komponentami PCA. Vpravo: ta samá pozorování promítnutá do nových souřadnic této roviny.(zdroj Gareth a kol., 2017)

4.3 Výpočet

Na analýzu hlavních komponent se můžeme dívat dvojím způsobem: buď jako na lineární kombinaci vlastností s největším rozptylem nebo jí můžeme definovat jako minimalizaci čtvercové chyby (5.2) mezi aproximací a původními daty. V obou případech dostaneme stejný výpočet algoritmu.

4.3.1 Maximalizace rozptylu

Nyní si ukážeme odvození pomocí maximalizace rozptylu, které je ukázané v knize Bishopa (Bishop, 2006) a prezentaci Militkého (Militký). Mějme dataset X pozorování x_n , kde $n=1, \dots, N$, N je celkový počet pozorování a x_n je konkrétní

pozorování, jehož velikost vlastností má dimenzi D . (Jedno pozorování má D naměřených hodnot). Cílem je zobrazit pozorování do prostoru s nižší dimenzí než je D . Nyní předpokládejme, že máme prostor s jednou dimenzí $M=1$, tzn že všech D vlastností jednoho pozorování se vyjádří pomocí jedno jediného čísla v novém prostoru s co nejmenší ztrátou informace. Označme u_1 jako D -dimenzionální vektor reprezentující tento prostor. Jelikož nás zajímá jen směr tohoto vektoru, tak bez újmy na obecnosti zvolme u_1 tak, aby platilo, že $u_1^T u_1 = 1$. Každé pozorování je pak zobrazeno na jedno skalární číslo z_n jako lineární kombinace u_1 a x_n .

$$z_n = u_1^T x_n \quad (4.3.1)$$

Dosazením do klasické vzorce pro rozptyl (4.3.2)

$$\sigma^2 = \sum_{i=1}^N [x_i - E(X)]^2 p_i, \quad (4.3.2)$$

kde x_i jsou hodnoty, kterých nabývá veličina X s pravděpodobností p_i a střední hodnotou $E(X)$ dostaneme následující vzorec

$$\sigma^2 = \frac{1}{N} \sum_{n=1}^N \left[z_n - \frac{1}{N} \sum_{j=1}^N z_j \right]^2. \quad (4.3.3)$$

Tento vzorec lze pak vyjádřit jako součin matice S a vektoru u_1 následujícím způsobem

$$\sigma^2 = \frac{1}{N} \sum_{n=1}^N [u_1^T x_n - u_1^T \bar{x}]^2 = u_1^T S u_1, \quad (4.3.4)$$

kde \bar{x} je klasický průměr vektoru

$$\bar{x} = \frac{1}{N} \sum_{n=1}^N x_n \quad (4.3.5)$$

a S je kovarianční matice definovaná takto

$$S = \frac{1}{N} \sum_{n=1}^N (x_n - \bar{x})(x_n - \bar{x})^T. \quad (4.3.6)$$

Nyní, když máme vyjádřený rozptyl projekce, tak už nám zbývá jen nalézt takové u_1 , aby rozptyl byl maximální. K tomu účelu využijeme dříve definované omezení $u_1^T u_1 = 1$ a metodu Lagrangeových multiplikátorů popsanou v kapitole (4.1.2), která tuto podmínku vynutí. Tvar Lagrangeovy funkce L_1 pak vypadá následovně:

$$L_1 = u_1^T S u_1 + \lambda (1 - u_1^T u_1). \quad (4.3.7)$$

Nastavením parciálních derivací na nulu a následnou úpravou rovnic zjistíme, že stacionární bod (podezřelý na extrémy) musí splňovat rovnici 4.3.9

$$\frac{\delta L_1}{\delta u_1} = 2 S u_1 - 2 \lambda_1 u_1 = (S - \lambda_1 E) u_1 = 0. \quad (4.3.8)$$

$$S u_1 = \lambda_1 u_1 \quad (4.3.9)$$

Protože hledáme nenulové řešení u_1 , řešením této rovnice je pak vlastní vektor matice S . Pokud vzorec (4.3.9) vynásobíme zleva u_1^T , tak dle vzorce (4.3.10) lze nahlédnout, že maximální hodnota rozptylu je závislá na velikosti nalezeného vlastního čísla, tj. u_1 odpovídající největšímu nalezenému vlastnímu číslu je řešením naší maximalizační úlohy. (To plyne z faktu, že S je symetrická matice a její největší nalezené vlastní číslo, řeší maximalizační úlohu $\max u_1^T S u_1$ (Hladík, 2017-věta 10.53)

$$u_1^T S u_1 = \lambda_1 \quad (4.3.10)$$

Nalezený vektor u_1 je pak první hlavní komponentou PCA někdy označovanou jako loading vektor a z_n je označován jako score vektor či skóre.

Podobným způsobem můžeme definovat další složku PCA u_2 a tím zobrazíme data do 2-rozměrného prostoru. Vektor u_2 a každý další vektor PCA analýzy musí být navíc kolmý na všechny dříve nalezené hlavní komponenty, což se vyjádří jako další omezující podmínka Lagrangerovy funkce, kterou rovněž maximalizujeme.

4.4 Výpočet PCA pro data s velkou dimenzí

Nyní si ukážeme matematický trik popsany v knize Bishopa (Bishop, 2006), kterým lze výrazně zvýšit efektivnost výpočtu PCA analýzy pro dataset, kde počet pozorování je mnohonásobně menší než počet jejich vlastností. Jako příklad nám poslouží náš dataset tváří, který obsahuje 298 tváří (pozorování). Každá tvář je tvořena trojúhelníkovou sítí s 15003 vrcholy. Každý vrchol je vyjádřen třemi souřadnicemi v 3D prostoru, takže jednu tvář tvoří $3 \cdot 15003 = 45009$ reálných čísel, takže máme 298 pozorování s dimenzí 45009. Už jen uchování kovarianční matice o velikosti $45009 \times 45009 = 2\,025\,810\,081$ by bylo problémové. O výpočetním času ani nemluvě, jelikož hledání vlastních čísel takto obrovské matice by trvalo dlouho. Navíc by se stejně našlo jen 298 vlastních nenulových čísel a příslušných vlastních vektorů. Jak tedy upravíme náš výpočet, abychom se vyhnuli tak obrovské kovarianční matici?

Rozložme si naši kovarianční matici (4.3.6) na součin matic $S = \frac{1}{N} X^T X$, kde

X je matice vycentrovaných vstupních dat dimenze $N \times D$. Nyní hledáme řešení následující rovnice

$$\frac{1}{N} X^T X u_i = \lambda_i u_i \quad (4.4.1)$$

Celou rovnici vynásobíme zleva maticí X .

$$\frac{1}{N} X X^T (X u_i) = \lambda_i (X u_i) \quad (4.4.2)$$

Nyní se opět dostáváme k řešení rovnice pomocí vlastních čísel a vektorů. Pokud si označíme $T = \frac{1}{N} X X^T$ a $v_i = X u_i$ dostaneme rovnici, jejímž řešením jsou stejná

vlastní čísla jako u rovnice s kovarianční maticí (4.3.9).

$$\frac{1}{N} X X^T v_i = \lambda_i v_i \quad (4.4.3)$$

Vynásobíme-li rovnici X^T zprava, zjistíme dle (4.4.1), že $X^T v_i$ je vlastním vektorem u_i kovarianční matice S .

$$\left(\frac{1}{N} X^T X\right)(X^T v_i) = \lambda (X^T v_i) \quad (4.4.4)$$

Tímto postupem mohlo dojít k tomu, že u_i nemusí být normalizované. Normalizaci můžeme za předpokladu, že v_i bylo znormalizované na jednotkovou velikost, zajistit následným přeškálováním

$$u_i = \frac{1}{(N \lambda_i)^{1/2}} X^T v_i \quad (4.4.5)$$

Tímto postupem se nám podařilo konkrétně v našem příkladě ušetřit hodně paměti a výpočetního času, protože nyní počítáme vlastní čísla a vlastní vektory jen z matice $T = X X^T$, jejíž rozměry jsou 298 x 298. Nicméně i tak se musely řešit výkonnostní problémy, jelikož matice T byla získána násobením matice velikosti 298x45009 s její transponovanou verzí (tedy 45009x298), ale o tom si povíme více v kapitole 9.1.

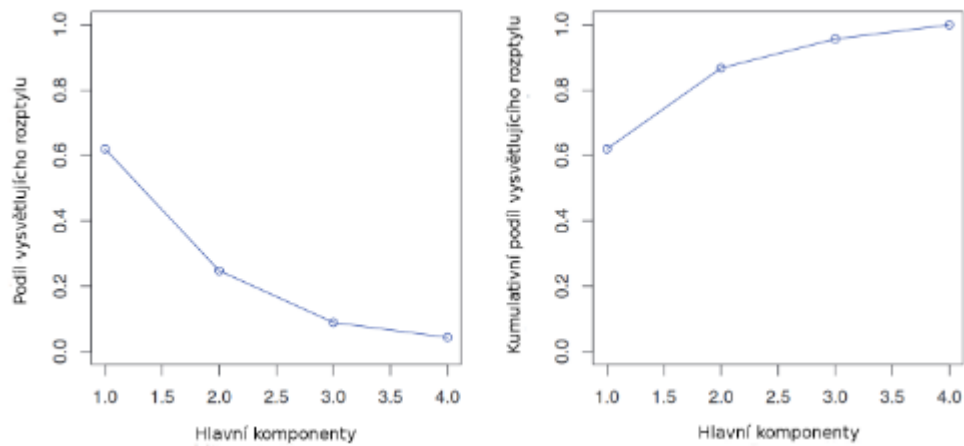
4.5 Kolik PCA komponent je třeba?

Pokud používáme PCA jako nástroj pro vizualizaci dat, tak si vystačíme s dvěma, maximálně třema hlavními komponentami PCA. Pokud jsme používali PCA jako nástroj pro předzpracování dat, tak můžeme využít všechny (pokud došlo k výrazné redukci) nalezené PCA komponenty, a nebo můžeme použít jen ty nejdůležitější. Otázkou je ale jak poznat, které PCA komponenty jsou nejdůležitější? Kde je ta hranice? Pro tento účel si můžeme vyjádřit poměr rozptylu vysvětlující dané komponenty (vzorec 4.5.1) a zobrazit si ho pomocí grafu (Obrázek 4.5.1 vpravo).

$$P_j = \frac{\lambda_j}{\sum_{i=0}^m \lambda_i}, \quad (4.5.1)$$

kde m je celkový počet nalezených vlastních čísel.

Avšak ze samotného grafu nemusí být vidět nějaký výrazný práh, podle kterého bychom mohli prvních n komponent vybrat, aby nám dostatečně reprezentovaly data. Někdy se používá takzvaný kumulativní graf (Obrázek 4.5.1 vlevo), který postupně nasčítává poměr rozptylu P_j pro všechny hlavní komponenty. V tomto grafu je lépe vidět kolik rozptylu je celkově vysvětleno pomocí prvních n komponent. Podle typu úlohy se pak volí nejčastěji počet komponent vysvětlující 65% až 90% celkového rozptylu.



Obrázek 4.5.1: vlevo: graf podílu rozptylu vysvětlujícího čtyři hlavní složky PCA, vpravo: graf kumulativního poměru rozptylu vysvětlujícího čtyři hlavní složky PCA (zdroj Gareth a kol., 2017)

5 Regrese

V této kapitole si popíšeme regresi - jednu z nejzákladnějších metod strojového učení popsanou v knize *An Introduction to Statistical Learning with Applications in R* (Gareth a kol., 2017). Ukážeme si její různé varianty a způsoby výpočtů jejích parametrů. Cílem regrese je co nejlépe aproximovat výstupní data na základě vstupních dat podle zvoleného modelu (typu aproximace), což jí řadí mezi metody učení s učitelem. Jako měřítko nejlepší aproximace se nejčastěji používá metoda nejmenších čtverců, kterou si v této kapitole také popíšeme.

5.1 Lineární regrese

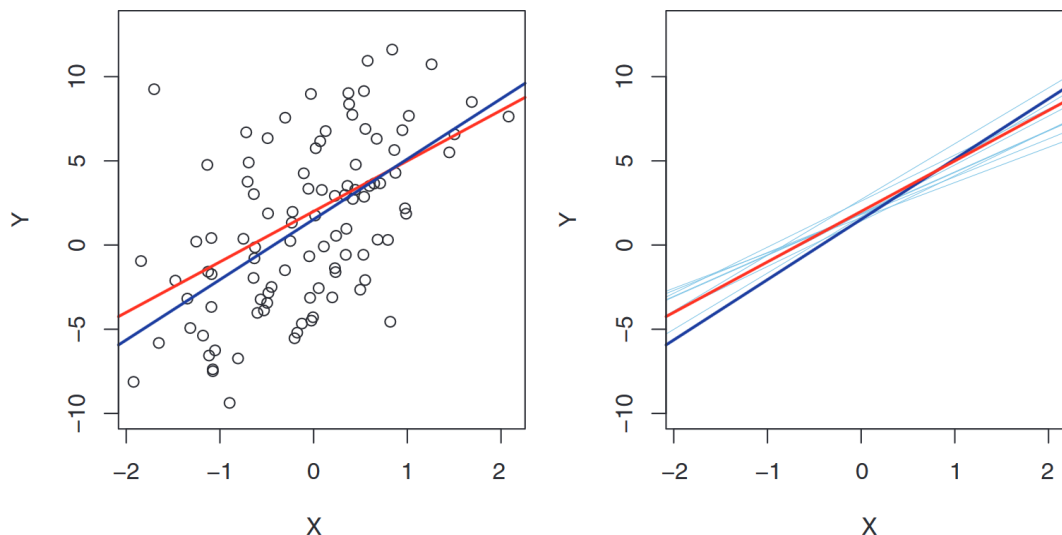
Mějme dataset X vstupních pozorování x_n , kde $n=1, \dots, N$, N je celkový počet pozorování a x_n je pozorování, jehož vlastnosti mají dimenzi D . K nim mějme ještě sadu Y příslušných výsledných hodnot y_n , kde $n=1, \dots, N$, N je celkový počet výsledných hodnot a y_n je výsledná hodnota pro vstupní x_n . Potom můžeme hodnotu y_n vyjádřit lineárním vztahem

$$y_n \approx \beta_0 + \beta_1 x_{n,1} + \dots + \beta_D x_{n,D}. \quad (5.1.1)$$

Jedná se jen o přibližné vyjádření hodnoty y_n . Na základě dostupných dat jsme pak schopni nalézt takové odhady parametrů $\hat{\beta}_1, \dots, \hat{\beta}_D$, aby byl model nejlepší aproximací pro data y_n . Odhad y_n označme \hat{y}_n .

$$\hat{y}_n = \hat{\beta}_0 + \hat{\beta}_1 x_{n,1} + \dots + \hat{\beta}_D x_{n,D} \quad (5.1.2)$$

Obrázek 5.1.1 vlevo aproximuje zobrazená data odhadnutou modrou přímkou, zatímco červená přímka vyjadřuje skutečný vztah mezi daty. Zde je vidět, že odhadnuté přímky mají velmi podobné umístění a směr, takže odhad je celkem dobrý.



Obrázek 5.1.1: vlevo: červenou čarou je vyznačen skutečný vztah mezi X a Y , modrá čára reprezentuje přímku odhadnutou pomocí zobrazených dat z části datasetu vpravo: červená čára reprezentuje skutečný vztah mezi X a Y , modrá čára reprezentuje odhad z levého obrázku a světle modré čáry jsou odhady na základě dat z jiných částí datasetu. Důležité je, že nalezené odhady se podobají skutečnému vztahu. (zdroj Gareth a kol., 2017)

5.2 Metoda nejmenších čtverců

Tato optimalizační metoda hledá odhady parametrů modelu pomocí minimalizace čtvercové chyby. Ta je definovaná jako druhá mocnina z rozdílu skutečné a odhadované hodnoty. V podstatě sčítáme druhé mocniny chyb odhadů.

$$S = \left(\sum_{n=1}^N (y_n - \hat{y}_n)^2 \right) \quad (5.2.1)$$

Rovnice (5.2.2) zobrazuje rozepsanou formu předchozí rovnice pro lineární regresi

$$\min \left(\sum_{n=1}^N (y_n - (\hat{\beta}_0 + \hat{\beta}_1 x_{n,1} + \dots + \hat{\beta}_D x_{n,D}))^2 \right) \quad (5.2.2)$$

Protože hledáme minimum, položíme parciální derivace podle β_1, \dots, β_D rovny nule.

$$\begin{aligned} \frac{\delta S}{\delta \hat{\beta}_0} &= -2(y_1 - \hat{\beta}_0 - \hat{\beta}_1 x_{1,2} - \dots - \hat{\beta}_D x_{1,D}) - \dots - 2(y_N - \hat{\beta}_0 - \hat{\beta}_1 x_{N,2} - \dots - \hat{\beta}_D x_{N,D}) \\ &\quad \vdots \\ \frac{\delta S}{\delta \hat{\beta}_D} &= -2x_{1,D}(y_1 - \hat{\beta}_0 - \hat{\beta}_1 x_{1,2} - \dots - \hat{\beta}_D x_{1,D}) - \dots - 2x_{N,D}(y_N - \hat{\beta}_0 - \hat{\beta}_1 x_{N,2} - \dots - \hat{\beta}_D x_{N,D}) \end{aligned} \quad (5.2.3)$$

Vhodnou úpravou dostaneme soustavu D rovnic o D neznámých, kterou lze vyřešit Gaussovou eliminační metodou.

$$\begin{aligned} \sum_{n=1}^N y_n &= \hat{\beta}_0 \sum_{n=1}^N 1 + \hat{\beta}_1 \sum_{n=1}^N x_{n,1} + \dots + \hat{\beta}_D \sum_{n=1}^N x_{n,D} \\ \sum_{n=1}^N y_n x_{n,D} &= \hat{\beta}_0 \sum_{n=1}^N x_{n,D} + \hat{\beta}_1 \sum_{n=1}^N x_{n,1} x_{n,D} + \dots + \hat{\beta}_D \sum_{n=1}^N x_{n,D} x_{n,D} \end{aligned} \quad (5.2.4)$$

Předchozí rovnice lze zapsat v maticovém zápisu jako

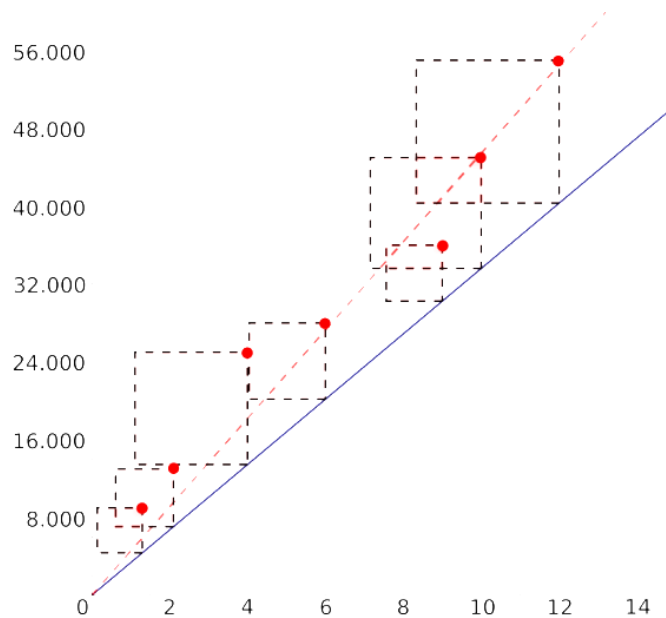
$$A^T y = A^T A \hat{\beta}, \quad (5.2.5)$$

kde A je v případě lineární regrese matice následujícího tvaru

$$\begin{bmatrix} 1 & x_{1,1} & \dots & x_{1,D} \\ 1 & x_{2,1} & \dots & x_{2,D} \\ \vdots & \vdots & \ddots & \vdots \\ 1 & x_{N,1} & \dots & x_{N,D} \end{bmatrix} \quad (5.2.6)$$

Těmto rovnicím se říká *normální rovnice*.

Obrázek 5.2.1 zobrazuje grafickou podobu metody nejmenších čtverců. Je zde zobrazena modrá přímka a k ní zobrazené čtvercové chyby odhadu pomocí přímky. Na první pohled je vidět, že tato přímka není hledané řešení. Představíme-li si čtverce k přerušované červené čáře, tak budou zabírat mnohem menší plochu než vyobrazené čtverce.



Obrázek 5.2.1: Grafické znázornění metody nejmenších čtverců. Z obrázku je jasně vidět, že součet čtverců modré přímky nemůže být minimální, na rozdíl od přerušované červené čáry, která data daleko lépe aproximuje.
zdroj(http://otevreno.fit.cvut.cz/kolik_budu_vydelavat)

Alternativou ke Gaussově eliminaci je o trochu rychlejší Choleského rozklad.

5.2.1 Choleského rozklad

Věta: Pro každou pozitivně definitní matici $A \in \mathbb{R}^{n \times n}$ existuje jediná dolní trojúhelníková matice $L \in \mathbb{R}^{n \times n}$ s kladnou diagonálou taková, že $A = L L^T$.

Rovnici (5.2.5), pak lze napsat jako

$$A^T y = L L^T \hat{\beta}, \quad (5.2.7)$$

protože $A^T A$ je symetrická matice a symetrické matice jsou pozitivně definitní.

Pokud dosadíme $L^T \hat{\beta} = v$, tak rovnici

$$A^T y = L v \quad (5.2.8)$$

lze spočítat dopřednou substitucí a odhady $\hat{\beta}$ spočítáme zpětnou substitucí z $L^T \hat{\beta} = v$.

Gausova eliminace a Choleského rozklad mají jednu velkou nevýhodu. Jejich numerická stabilita není nejlepší, tj. při výpočtu dochází k akumulaci zaokrouhlovací chyby, což může ovlivnit správnost výsledku. Proto se pro přesnější výsledky používají jiné metody pro řešení soustavy rovnic, jako například QR dekompozice.

5.2.2 QR dekompozice

Věta: Pro každou matici $A \in \mathbb{R}^{m \times n}$ existuje ortogonální $Q \in \mathbb{R}^{m \times m}$ a horní trojúhelníková matice $R \in \mathbb{R}^{m \times n}$ s nezápornou diagonálou tak, že $A = QR$.

Rovnici (5.2.5), pak lze napsat jako

$$R^T Q^T y = R^T Q^T Q R \hat{\beta} \quad (5.2.9)$$

Úpravou dostaneme

$$Q^T y = R \hat{\beta}, \quad (5.2.10)$$

protože se můžeme dívat na metodu nejmenších čtverců jako na druhou mocninu normy vektoru chyb

$$S = \left(\sum_{n=1}^N (y_n - \hat{y}_n)^2 \right) = \|e\|^2 \quad (5.2.11)$$

a zároveň platí následující rovnice

$$\|e\|^2 = e^T e = e^T Q Q^T e = (Q^T e)^T Q^T e = \|Q^T e\|^2. \quad (5.2.12)$$

A protože matice R je horní trojúhelníková matice, tak můžeme rovnici 5.2.10 zapsat jako

$$\begin{pmatrix} d_1 \\ d_2 \end{pmatrix} = \begin{pmatrix} R_1 \\ R_2 \end{pmatrix} \hat{\beta}, \quad (5.2.13)$$

kde $R = \begin{pmatrix} R_1 \\ R_2 \end{pmatrix}$ a $Q^T \hat{\beta} = \begin{pmatrix} d_1 \\ d_2 \end{pmatrix}$.

Pak platí

$$\|Q^T e\| = \left\| \begin{pmatrix} R_1 \hat{\beta} - d_1 \\ R_2 \hat{\beta} - d_2 \end{pmatrix} \right\| = \left\| \begin{pmatrix} R_1 \hat{\beta} - d_1 \\ -d_2 \end{pmatrix} \right\| \quad (5.2.14)$$

Z čehož vyplývá, že hledané minimum splňuje rovnici:

$$R_1 \hat{\beta} = d_1. \quad (5.2.15)$$

QR dekompozice bývá pomalejší než Gaussova eliminace, ale řešení je numericky stabilnější.

Výše zmíněné metody řešení soustav rovnic zde byly zmíněny pro úplnost a také proto, že je implementuje použitá matematická knihovna, která v základním nastavení volá pro váženou polynomiální regresi Choleského rozklad a pro neváženou polynomiální regresi QR dekompozici. Zdrojem byla kniha Lineární algebra (nejen) pro informatiky (Hladík, 2017)

5.3 Modely jedné vstupní proměnné a polynomiální aproximace.

5.3.1 Lineární model

Lineární model jedné vstupní proměnné data aproximuje přímkou.

$$\hat{Y} = \hat{\beta}_0 + \hat{\beta}_1 X \quad (5.3.1)$$

Rovnice pro Gausovu eliminaci po vzoru (5.2.4) vypadají následovně.

$$\begin{aligned} \sum_{n=1}^N y_n &= \hat{\beta}_0 N + \hat{\beta}_1 \sum_{n=1}^N x_n \\ \sum_{n=1}^N x_n y_n &= \hat{\beta}_0 \sum_{n=1}^N x_n + \hat{\beta}_1 \sum_{n=1}^N x_n^2 \end{aligned} \quad (5.3.2)$$

Po úpravě vznikne známý vzorec pro lineární regresi.

$$\begin{aligned} \hat{\beta}_1 &= \frac{\sum_{n=1}^N (x_n - \bar{x})(y_n - \bar{y})}{\sum_{n=1}^N (x_n - \bar{x})^2} \\ \hat{\beta}_0 &= \bar{y} - \hat{\beta}_1 \bar{x} \end{aligned} \quad (5.3.3)$$

5.3.2 Vyšší stupeň

Data můžeme aproximovat i polynomem většího stupně než je přímka. Například aproximace polynomem druhého stupně vypadá následovně

$$\hat{Y} = \hat{\beta}_0 + \hat{\beta}_1 X + \hat{\beta}_2 X^2 \quad (5.3.4)$$

A obdobně jako u lineární regrese by se odhady parametrů daly nalézt pomocí parciálních derivací a Gaussovou eliminační metodou z těchto rovnic

$$\begin{aligned} \sum_{n=1}^N y_n &= \hat{\beta}_0 N + \hat{\beta}_1 \sum_{n=1}^N x_n + \hat{\beta}_2 \sum_{n=1}^N x_n^2 \\ \sum_{n=1}^N x_n y_n &= \hat{\beta}_0 \sum_{n=1}^N x_n + \hat{\beta}_1 \sum_{n=1}^N x_n^2 + \hat{\beta}_2 \sum_{n=1}^N x_n^3 \\ \sum_{n=1}^N x_n^2 y_n &= \hat{\beta}_0 \sum_{n=1}^N x_n^2 + \hat{\beta}_1 \sum_{n=1}^N x_n^3 + \hat{\beta}_2 \sum_{n=1}^N x_n^4 \end{aligned} \quad (5.3.5)$$

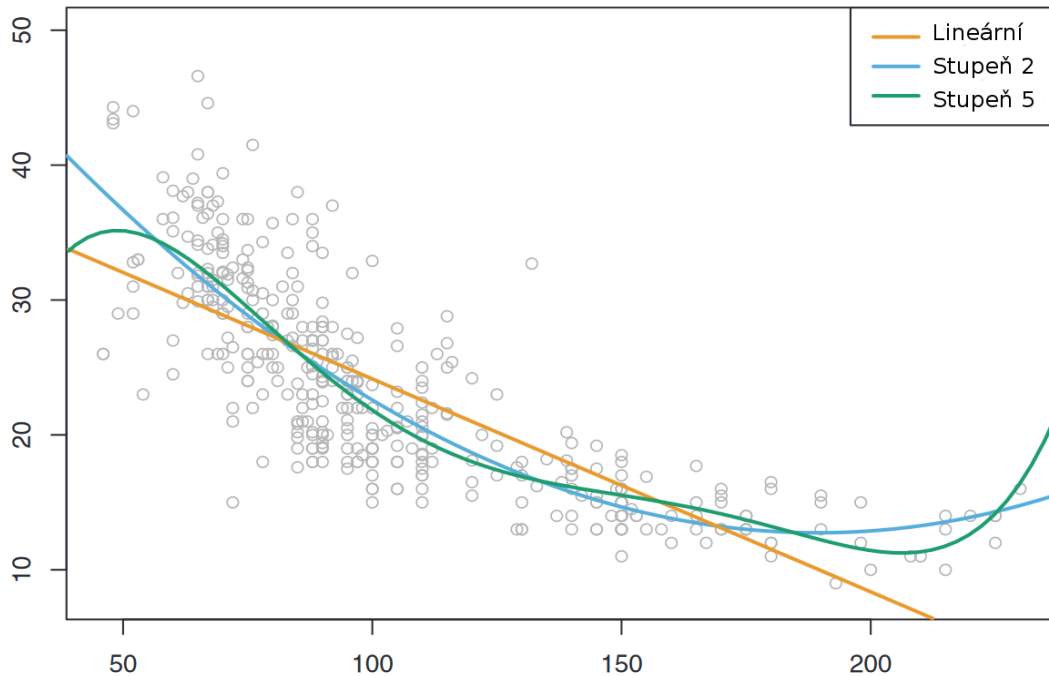
Obecně lze zapsat polynomiální regresi jedné vstupní proměnné pomocí Vandermondovy matice.

Definice: Vandermondova matice je matice $M \times N$, která v každém řádku obsahuje po sobě jdoucí členy geometrické posloupnosti začínající jedničkou.

$$V = \begin{bmatrix} 1 & a_1 & a_1^2 & \dots & a_1^{n-1} \\ 1 & a_2 & a_2^2 & \dots & a_2^{n-1} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & a_m & a_m^2 & \dots & a_m^{n-1} \end{bmatrix} \quad (5.3.6)$$

Hledáme pak řešení následující soustavy rovnic takové, aby čtvercová chyba byla minimální. Použít můžeme například QR dekompozici.

$$\begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_N \end{bmatrix} = \begin{bmatrix} 1 & x_1 & x_1^2 & \dots & x_1^p \\ 1 & x_2 & x_2^2 & \dots & x_2^p \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & x_N & x_N^2 & \dots & x_N^p \end{bmatrix} \begin{bmatrix} \hat{\beta}_0 \\ \hat{\beta}_1 \\ \vdots \\ \hat{\beta}_p \end{bmatrix} \quad (5.3.7)$$



Obrázek 5.3.1: Naobrázku jsou znázorněné tři aproximace polynomem různého stupně. Oranžová přímka představuje odhad pomocí lineární regrese. Modrá křivka zobrazuje odhad pomocí kvadratické funkce a zelená křivka znázorňuje pro ležení dat polynomem 5-tého stupně.

5.4 Vážená polynomiální regrese s jednou vstupní proměnou

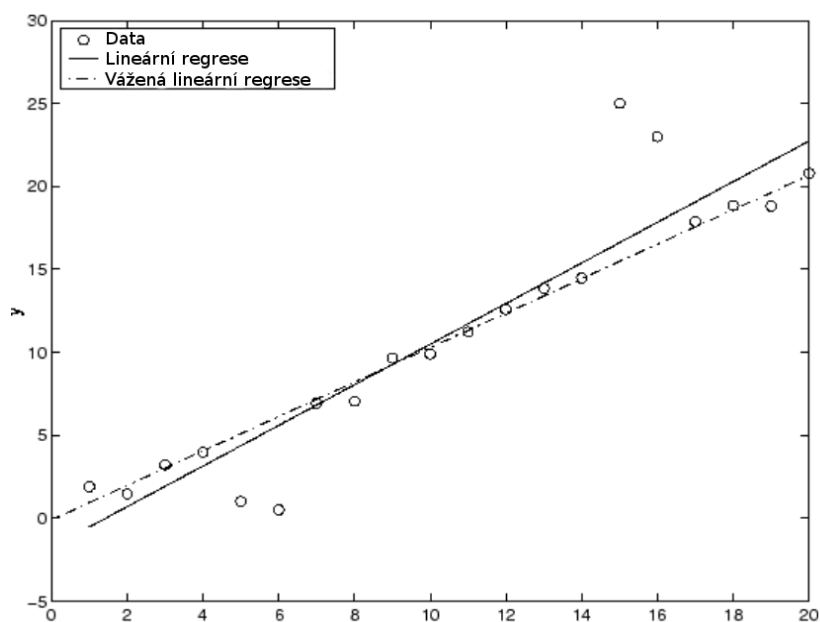
Váženou lineární regresi použijeme v případě, kdy některá pozorování mají mít menší či větší vliv na celkový výsledek. Tyto váhy se dají určit pomocí rozptylových vlastností nebo jsou nějak předem známy či jinak odhadnuty. Při vážené regresi minimalizujeme váženou metodou nejmenších čtverců:

$$S = \left(\sum_{n=1}^N W_{i,i} (y_n - \hat{y}_n)^2 \right), \quad (5.4.1)$$

kde W je diagonální matice vah.

Po zderivování řešíme následující soustavu normovaných rovnic

$$A^T W y = A^T W A \beta \quad (5.4.2)$$



Obrázek 5.4.1: Plnou čarou je znázorněný odhad přímky pomocí lineární regrese a přerušovanou odhad přímky vážené regrese. Vážená regrese bere menší ohled na "odlehlejší data" a tak zbylé data lépe aproximuje, zatímco normální regrese bere všechna data za stejně důležité (zdroj <https://www.mathworks.com/help/curvefit/least-squares-fitting.html>)

6 Analýza tvaru obličeje

V této kapitole si popíšeme, jak jsou v přiloženém programu použity metody z předchozích kapitol 4 a 5. Také si odůvodníme některé kroky, které vedly k těmto způsobům použití.

6.1 Uživatelské předzpracování

Před samotným spouštěním PCA na načtených datech je uživateli umožněno provést filtraci dat podle pohlaví a/nebo normalizaci velikosti tváře.

6.1.1 Normalizace velikosti

Cílem normalizace tváře je odstranit celkovou velikost tváře jako veličinu z PCA komponent, aby nalezená reprezentace aproximovala pouze tvar hlavy bez zatížení velikostí.

Normalizaci velikosti tváře provádíme následujícím postupem

- nalezneme geometrické těžiště T tváře reprezentované n vrcholy v 3D prostoru

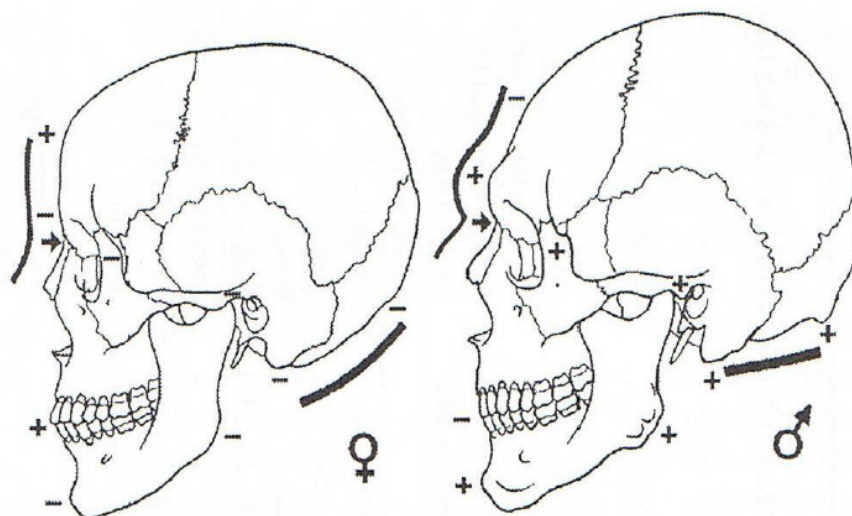
$$\begin{aligned}T_x &= \frac{\sum_{i=1}^n x_i}{n} \\T_y &= \frac{\sum_{i=1}^n y_i}{n} \\T_z &= \frac{\sum_{i=1}^n z_i}{n}\end{aligned}\tag{6.1.1}$$

- najdeme nejvzdálenější bod od těžiště
- všechny vektory z těžiště do vrcholu normalizujeme podle největšího vektoru

6.1.2 Filtr podle pohlaví

Možná filtrace podle pohlaví byla především zavedena kvůli rozdílným tvarovým charakteristikám v tvaru tváře a lebky mezi mužem a ženou (Mydlová a kol., 2015). V bakalářské práci Určení pohlaví na lebce člověka s využitím trojrozměrných

virtuálních modelů (Fialová, 2013) jsou popsány známé faktory ovlivňující tvarové rozdíly lebky mezi mužem a ženou: celková velikost těla, nerovnoměrné tempo růstu, hormonální změny a další. Lebka je druhým nejlepším lidským kosterním pozůstatkem, podle kterého se dá určit pohlaví (prvním je pánev).



Obr. 2.2.: Pohlavní rozdíly na lebce. Zdroj: Stloukal (1999)

Obrázek 6.1.1: Hlavní rozdíly mezi tvarem lebky muže a ženy. (převzato z práce Fialová, 2013 původní zdroj je uveden v rámečku)

6.2 Modelování obličeje pomocí regrese

Po spočtení PCA už máme k dispozici tvář reprezentovanou výrazně menším počtem proměnných. Tyto proměnné se nyní pokusíme predikovat pomocí regrese tak, že vstupní proměnnou bude jedna z vlastností jedince a výstupní proměnnou bude PCA skóre. Samostatnou aproximaci spočteme pro všechny PCA skóre. Hledáme tady přímou souvislost mezi vlastnostmi jedince a novou reprezentací.

Klasické použití regrese, kdy použijeme aproximaci v nově nastaveném vstupním bodě jako predikovanou hodnotu, nedávalo příliš dobré výsledky. Tvář měla tendenci až příliš vizuálně konvergovat k průměrné tváři (zejména při použití lineární regrese) a pak ztrácela svůj původní charakter. Abychom zachovali charakteristické znaky tváře, rozhodli jsem se regresi použít trochu netradičním způsobem. Spočtená aproximace dat slouží spíše jako náznak směru/vývoje hodnoty při změně jedné z vlastností jedince. V podstatě se spočte změna výstupní proměnné vůči změně

vstupní proměnné a o tuto spočtenou hodnotu je hodnota skóre posunuta. Takže pokud v datech není nalezený žádný trend, tak se skóre nezmění.

Tento postup nám navíc umožnil kombinovat změny vícero predikcí tváře najednou. Například lze vytvořit predikci tváře o deset let starší nalezenou pomocí kvadratické funkce a zároveň lze ubrat 20 kilogramů pomocí tendence nalezené lineární váženou regresí. Další výhodou je také možné zkombinování s uživatelským nastavením změn skóre.

6.3 Karikatura

Účelem této funkcionality je zvýraznit již dominantní rysy tváře tím, tak že všechny hodnoty skóre reprezentující tvář přenásobíme nastavenou konstantou (v rozsahu od 0 do 5). Tím pak vznikne jakási matematická obdoba karikatury. Navíc je tak uživateli umožněno hromadně upravit všechny skóre hodnoty o násobek zvolené konstanty.

6.4 Vážená regrese

Váženou regresí chceme docílit, aby data podobných tváří více ovlivnily výsledek aproximace pomocí regrese. Tím by jsme měli docílit lepšího odhadu skutečného trendu. Jako funkci podobnosti jsme si zvolili Euklidovskou metriku, kterou použijeme na PCA skóre.

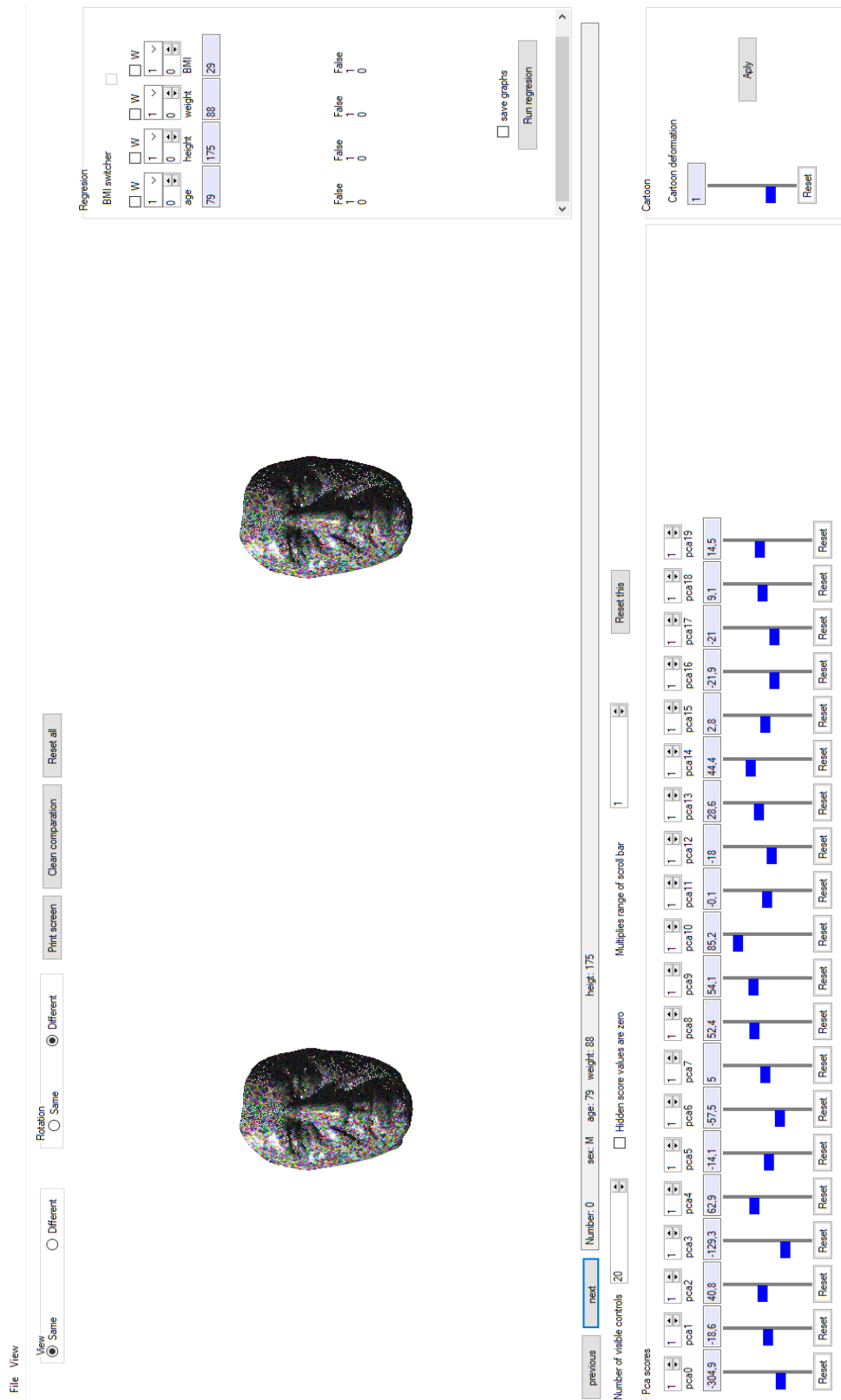
Definice: Euklidovskou metriku dvou vektorů $a, b \in \mathbb{R}^n$ je

$$\sqrt{\sum_{i=1}^n (a_i - b_i)^2}. \quad (6.4.1)$$

Pro každou dvojici tváří tedy vznikne číslo reprezentující jejich podobnost, které pak přetvoříme do váhy následujícím způsobem:

Nejprve si spočteme směrnice rovnicí přímky z bodu [minimální naměřená vzdálenost, 1] do bodu [maximální naměřená vzdálenost * kladná uživatelem nastavená konstanta, 0]. Pak už jen stačí pro danou vzdálenost najít odpovídající hodnotu y z předpisu.

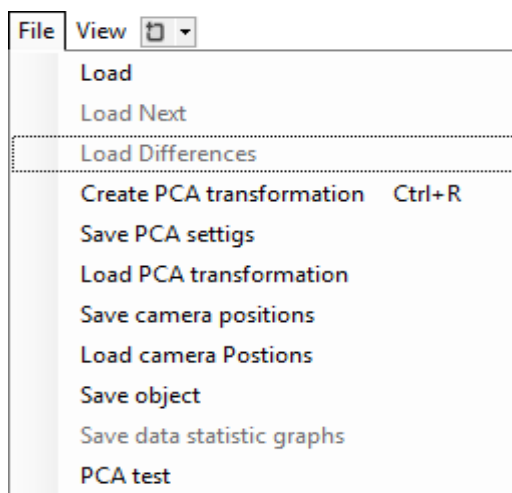
7 Software pro modelování a vizualizaci tvaru obličeje



Obrázek 7.1: Hlavní plocha po spočtení PCA

Zobrazený print screen 7.1 ukazuje celkový vzhled funkčnosti programu po spočtení PCA. Uživatel následně mění zobrazené objekty pomocí různých nastavení. Nyní si postupně popíšeme jednotlivá nastavení. Začne s menu vlevo nahoře.

7.1 Menu



Obrázek 7.1.1: Ukázka menu File

7.1.1 Záložka File

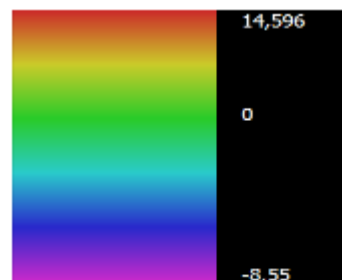
Záložka File (obrázek 7.1.1) v levém horním menu nabízí uživateli následující funkcionální

- Load – zobrazí okno pro výběr souboru a umožní uživateli načíst 3D objekt typu .obj nebo .ply. Načtený 3D objekt je pak zobrazen podobným způsobem, jako jsou zobrazené tváře na úvodním print screenu, tj. objekt je současně zobrazen ve dvou nezávislých oknech. Zobrazenému objektu je náhodně přiřazené obarvení vrcholů.
- Load Next – zobrazí okno pro výběr souboru a umožní uživateli načíst 3D objekt typu .obj nebo .ply. Tento objekt je pak zobrazen přes jiný načtený 3D objekt v levém zobrazovacím okně (obrázek 7.1.2). Takto přes sebe lze načíst celkem tři další objekty. Pro dobré rozlišení překrývajících se částí jsou načtené objekty obarveny v následujícím pořadí: červená, modrá, zelená. Tato funkcionální slouží k porovnání objektů a je jí možné využít až až po spočtení PCA.



Obrázek 7.1.2: Zobrazuje překrytí dvou různých tváří, barevná je původní tvář, červená je druhá načtená

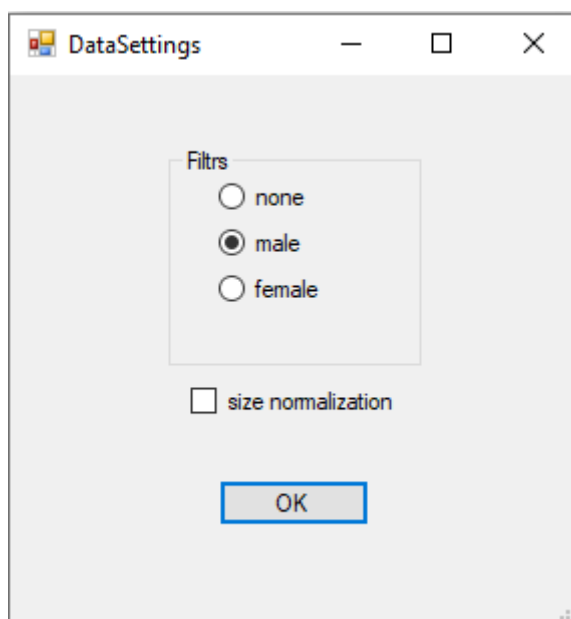
- Load Differences - zobrazí okno pro výběr souboru a umožní uživateli načíst 3D objekt typu .obj nebo .ply., který se použije pro obarvení již jiného načteného objektu (obrázek 7.1.3), který má stejný počet vrcholů, které si navzájem odpovídají. Mezi odpovídajícími se vrcholy je změřena Euklidovská vzdálenost. Této vzdálenosti je navíc přiřazeno kladné nebo záporné znaménko, podle toho zda z-ová souřadnice bodu leží před nebo za bodem již načteného objektu. Zápornou hodnotu dostaneme, pokud by původní bod ležel za nově načteným bodem. Pevně daná barevná škála je pak lineárně naškálovaná na nejmenší a největší vzdálenost. Barevná škála se pak objeví vlevo nahoře v zobrazovacím okně.



Obrázek 7.1.3: Zobrazuje rozdíl mezi dvěma tvářemi pomocí tepelné mapy. Vrcholy původní tváře jsou obarvené podle vzdálenosti odpovídajícího vrcholu druhé tváře. Paleta v pravo učuje význam barev

- Create PCA transformation - zobrazí okno pro výběr složky. Uživatel pak vybere složku s daty popsanou v podkapitole 7.8 Vstupní data. Následně se

zobrazí okno s výběrem předzpracování (obrázek 7.1.4).



Obrázek 7.1.4: Zobrazuje dialog s nastavením předzpracování.

Pomocí boxu Filters je možné nastavit filtr pro načítání dat. Při výběru None se načtou všechny tváře bez rozdílu pohlaví. Při nastavení Male se načtou jen mužské tváře a při nastavení female se načtou jen ženské tváře. Také zde můžeme nastavit normalizaci velikosti, kdy se tváře znormalizují podle svého geometrického těžiště. Po stisknutí tlačítka OK se spustí výpočet a ve zobrazujících oknech se vykreslí průměrná tvář a zpřístupní se další ovládací prvky popsané v následujících podkapitolách.

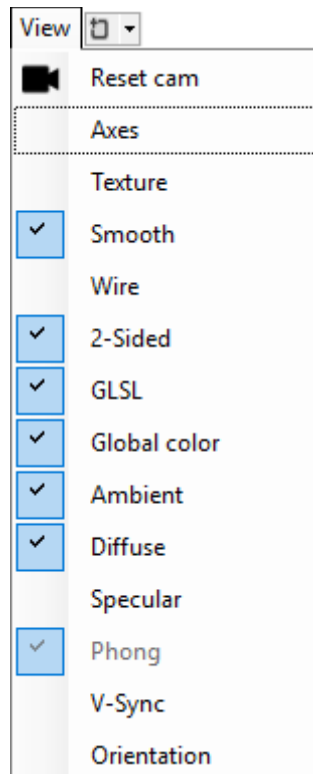
- Save PCA transformation – zobrazí okno pro výběr složky, do které se uloží spočtené parametry PCA pro opětovné načtení. Ve vybrané složce se uloží následující soubory:
 - specs.csv – ukládá vlastnosti jedinců,
 - scores.txt - ukládá matici s hodnotami skóre reprezentující dané obličej.
 - means.txt- ukládá průměrný obličej, loadings.txt- ukládá hlavní komponenty pca, centroid. - ukládá informace o provedené normalizaci velikosti. Pokud je soubor prázdný normalizace nebyla provedena.
 - Indexes.obj - ukládá průměrnou tvář ve formátu .obj, který určuje pomocí indexů vrcholů trojhelny reprezentující povrch tváře.(Tyto soubory nejsou

určené k měnění. Slouží pro uchování vypočtené PCA pro budoucí použití, zvláště pokud výpočet PCA byl zdlouhavý.)

- Load PCA transformation - zobrazí okno pro výběr složky, kde uživatel vybere složku, do které uložil dříve spočtenou PCA a znovu jí načte.
- Load camera position –zobrazí okno pro výběr souboru, kde uživatel vybere soubor s dříve uloženou pozicí kamery pomocí položky Save camera position.
- Save object - zobrazí okno pro uložení souboru a uloží vlevo zobrazeným objekt jako .obj.
- Save data statistics graphs- zobrazí okno pro výběr složky, kam se uloží histogramy ostatních vlastností objektu.
- PCA test – zobrazí okno pro výběr složky s daty a hned poté zobrazí druhé výběrové okno pro uložení spočtených výsledků. Poté se spustí test na globální, mužský a ženský PCA model a výsledek je uložen do souboru Error.txt. Test probíhá následovně: Pro každý objekt se počítá model z dat bez jeho přítomnosti. Poté se objekt aproximuje pomocí tohoto modelu s různým počtem komponent (2, 20, 40, 60 a všechny) a pro každý vrchol se spočítá vzdálenost od originálního objektu. Tyto vzdálenosti se následně zprůměrují přes všechny vrcholy a testovací objekty. Do souboru Error.txt se také uloží průměrná chyba aproximací objektů, které se podíleli na tvorbě modelu. Toto slouží k porovnání. Chyba je vyjádřena v milimetrech.

7.1.2 Záložka View

Záložka View v levém horním menu nabízí uživateli následující funkcionalitu (následující popis vysvětluje zaškrtnutou funkcionalitu)



Obrázek 7.1.5: Záložka View

- Reset cam – resetuje kameru do výchozího nastavení
- Axes – zobrazí osy x, y, z
- Texture – zapne zobrazování textury, pokud je nějaká textura načtená
- Smooth – zapne plynulý přechod barev mezi vrcholy
- Wire – zobrazí načtenou síť jen pomocí hran trojúhelníků. Vznikne drátěný model. V opačném případě je obsah trojúhelníky vybarvený.
- 2-Sided -umožňuje zobrazit povrch z obou stran, jak z vnitřní, tak z vnější
- GLSL- zapne používání shaderů a umožní tvář vystínovat vůči nastavenému zdroji světla. Stínování pak lze ovlivnit zaškrtnutím položek Ambient, Diffuse, Specular, Phong.
- Global Color – obarví celou tvář do žluté barvy. (Výchozí nastavení je náhodně barevné)
- Ambient – přidá do modelu stínování konstantní osvětlení (toto osvětlení simuluje světlo, které by na povrch objektu dopadlo po několikanásobném odražení světelných paprsků od ostatních předmětů).
- Diffuse -zapne zobrazení difusní části světla (rovnoměrně odražená část světla do všech směrů od matného povrchu, která vytváří trojrozměrný

vzhled)

- Specular – zobrazí odlesky a tím učiní povrch lesklý. V opačném případě povrch působí matným dojmem.
- Phong – zapne Phongův světelný model.
- V-sync – experimentální zvýšení rychlosti výpočtu
- Orientation – změni orientaci z-ové souřadnice při exportu dat

7.2 Zobrazující okna

Uživateli jsou k dispozici dvě zobrazovací okna. Ty umožňují uživateli podle zvoleného nastavení zobrazit jeden 3D objekt ze dvou pohledů nebo třeba zobrazit původní objekt s jeho upravenou variantou. V pravém okně se podle nastavení zobrazuje buď původní 3D objekt nebo kopie levého okna. V levém okně se vždy zobrazují veškeré uživatelské změny provedené na objektu po výpočtu PCA. V následujícím textu budou popsány tyto modifikace objektu. Objekty v těchto oknech lze pomocí myši otáčet ze všech stran. Stačí jen levým tlačítkem kliknout a táhnout. Kolečkem myši lze objekty přibližovat a oddalovat: přibližování a oddalování funguje na obě okna zároveň, zatímco otáčení objektem podle nastavení funguje buď pro obě okna zároveň nebo pro každé zvlášť (viz. následující popis).

7.3 Nastavení nad zobrazujícími okny.



Obrázek 7.3.1: Obecné nastavení

Popsané zleva doprava.



Obrázek 7.3.2: Přepínač pohledů

- Same- v obou zobrazujících oknech lze vidět stejný obsah
- Different – v pravém zobrazujícím okně je vidět původní objekt spočtený pomocí PCA. V levém je zobrazený objekt upravený uživatelským nastavením.



Obrázek 7.3.3: Přepínač rotací

- Same – umožní otáčet objekty v obou zobrazujících oknech zároveň. Při přepnutí z módu Different se otáčení nastaví podle levého okna.
- Different- umožní otáčet objekty nezávisle na sobě

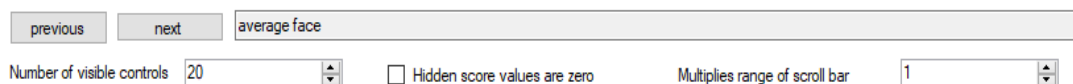
Tlačítka:

- Tlačítko Print screen - zobrazí okno pro uložení print screenu levého zobrazovacího okna jako obrázek typu .bmp

Veškerou následující funkcionalitu je možné použít jen po spočtení nebo načtení PCA transformace.

- Clean comparison – smaže vizuální porovnání tváří, které vzniklo pomocí položek Load next and Load differences v záložce File v levém horním menu.
- Reset all – resetuje veškerá uživatelská nastavení provedená po výpočtu PCA tj. uvede program do stavu, který nastal bezprostředně po výpočtu PCA.
- Compare – spustí test porovnávání metod a výsledky uloží do zvolené složky

7.4 Nastavení pod zobrazujícími okny.



Obrázek 7.4.1: Menu pro ovládání komponent PCA

Tlačítka previous a next umožňují uživateli listovat mezi načtenými 3D objekty z PCA transformace, tj. ve zobrazujících oknech se vykreslí příslušný objekt a v boxech PCA scores a Regression se zobrazí údaje odpovídající tomuto objektu.

- Tlačítko previous - přepne na předchozí objekt
- Tlačítko next – přepne na následující objekt
- Šedé textové okno – zobrazuje informace k vykreslenému objektu. V případě tváří se jedná o index tváře v souboru, pohlaví, věk, váhu a výšku. Je-li tvář velikostně normalizovaná, zobrazí se ještě souřadnice těžiště, maximální nalezená vzdálenost mezi těžištěm a nějakým vrcholem objektu, index vrcholu s maximální nalezenou vzdáleností a průměrná vzdálenost vrcholů od těžiště.
- Number of visible controls - nastavuje počet zobrazených posuvníků v boxu PCA skóre, kterými je pak možné měnit nastavení.
- Hidden score values are zero - zaškrtnutím tohoto zaškrťovacího boxu se

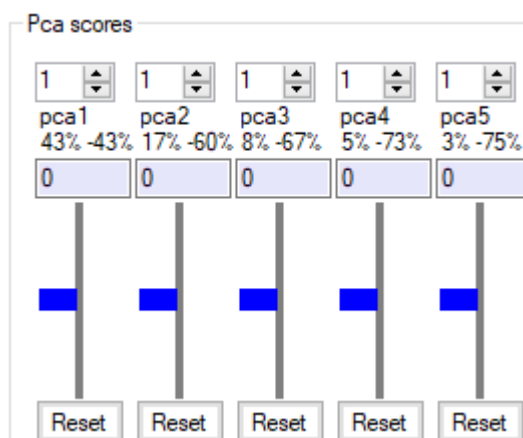
v pravém zobrazovacím okně vykreslí aproximovaná tvář podle počtu PCA skóre nastavených pomocí Number of visible controls.

- Multiplies range of scrollbar – nastavuje rozsah všech posuvníků v boxu PCA scores. Nastavená hodnota slouží jako multiplikativní konstanta nastaveného rozsahu, tj. každý posuvník má nastavený symetrický rozsah hodnot podle maximální či minimální hodnoty (podle největší absolutní hodnoty ze všech objektů pro danou PCA komponentu) a tento rozsah je pak zvětšen násobně krát o nastavenou hodnotu. Pouze hodnoty menší než zadané číslo jsou nastaveny na toto číslo (nastavení větších rozsahů zůstává zachované).

7.5 PCA scores

V této sekci jsou vyobrazeny nalezené skóre pro daný objekt.

Uživateli je umožněno hýbat posuvníkem a tím měnit objekt zobrazený v zobrazovacím okně/oknech (podle nastavení módu view na Same nebo Different).



Obrázek 7.5.1: Nastavení jednotlivých komponent pca

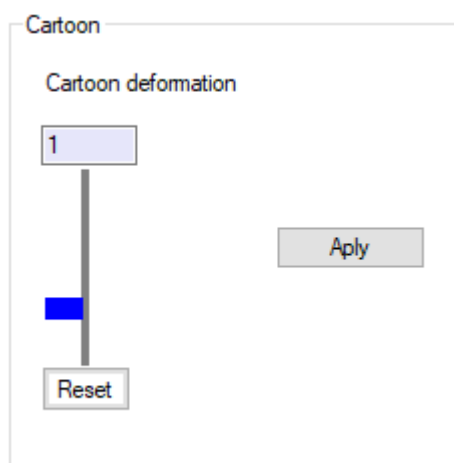
Tento box zobrazuje spočtené PCA skóre a umožňuje jejich editaci.

- V prvním číselném poli (bráno ze shora dolů.) lze nastavit rozsah posuvníku zobrazeného níže následujícím způsobem. Každý posuvník má nastavený symetrický rozsah hodnot podle největší absolutní hodnoty ze všech objektů pro danou PCA komponentu.
- V šedém okně je zobrazena hodnota posuvníku.
- Pohnutím posuvníku se přenastaví hodnota konkrétní PCA komponenty a v zobrazovacím okně/oknech (podle režimu zobrazení View na Same nebo Different) se vykreslí podoba objektu podle nového nastavení. Na změnu

hodnoty posuvníku oproti původní je uživatel upozorněn změnou barvy jezdce na červenou.

- Tlačítko reset restartuje hodnotu, tj. nastaví ji na hodnotu spočtenou PCA
- Poznámka: Skóre PCA je ovlivněno funkcionalitou z boxu Regression. Může se stát, že spočtené hodnoty překročí maximální nastavení posuvníku. V takovém případě je posuvník graficky nastaven na svojí maximální/minimální hodnotu a zobrazená hodnota v šedém okně se vypíše červeně a neodpovídá hodnotě nastaveného posuvníku, ale reálně spočtené hodnotě pomocí regrese. Následkem toho může následné pohnutí jezdce vyvolat velkou změnu.

7.6 Cartoon



Obrázek 7.6.1: Nastavení funkce karikatury

- V šedivém okně je zobrazena hodnota posuvníku
- Posuvníkem se pouze nastavuje hodnota. Až po stlačení tlačítka Apply jsou touto hodnotou vynásobeny všechny hodnoty PCA skóre a dojde k překreslení objektu v zobrazujících oknech.
- Tlačítko reset nastaví posuvník na výchozí hodnotu, kterou je jednička.

7.7 Regression

The image shows a software interface for regression analysis. At the top, there is a 'Regression' title. Below it is a 'BMI switcher' checkbox. There are four columns of controls for variables: 'age', 'height', 'weight', and 'BMI'. Each column has a 'W' checkbox, a dropdown menu (all set to '1'), a slider (all set to '1'), and a 'Reset' button. Below the sliders are four '0' boxes. At the bottom, there is a 'save graphs' checkbox and a 'Run regression' button.

Obrázek 7.7.1: Regression panel

- BMI switcher - umožňuje přepínání mezi BMI proměnnou a proměnnými váha a výška, protože BMI je závislé na váze a výšce. Chceme-li nastavovat BMI proměnnou, pak se nastavení výšky a váhy resetuje - nastaví na výchozí hodnoty. Pokud chceme nastavovat výšku a váhu, tak se BMI nastaví do původní hodnoty.
- Zaškrťovací box W umožní použít váženou regresi.
- Následující výběr určuje stupeň polynomu regrese. Uživatel má na výběr ze tří hodnot: 1 – lineární regrese, 2 – aproximace kvadratickou křivkou 3 - aproximace kubickou křivkou.
- Šedé pole zobrazuje nastavenou hodnotu posuvníku.
- Číslo nastaví sklon přímky, která je použitá k vážení. Přímka prochází bodem

[minimální nalezená Euklidovská vzdálenost mezi aktuální tváří a ostatními tvářemi, 1] a [maximální nalezená Euklidovská vzdálenost mezi aktuální tváří a ostatními tvářemi * nastavená konstanta z tohoto okna,0]. Takže čím větší číslo tím menší sklon.

- Posuvník nastavuje hodnoty vlastností a podle vypočtené regresní křivky mění hodnoty PCA skóre tak, že posune hodnoty PCA skóre ve směru spočtené regrese. V důsledku toho dojde k překreslení zobrazovacího okna/oken podle nového nastavení.
- Vypsání hodnoty zobrazují nastavení (ve stejném pořadí jako jsou nastavovací boxy) posledního běhu regrese. Před prvním spuštěním regrese je vypsáno výchozí nastavení. Změní-li uživatel nastavení, tak se písmo na tlačítku Run regression zbarví dočervena jako signalizace, že vypočtená regrese neodpovídá aktuálnímu nastavení a je potřeba regresi přepočítat. Výpočet nevážené regrese je pro všechny načtené objekty stejný. Na rozdíl od vážené regrese, která je platná jen pro aktuálně zobrazený objekt a tak změní-li se objekt, tak jsou uživatelům zneprístupněny posuvníky, tj. nejsou viditelné pro jiné tváře.
- Zaškrtačací box save graphs umožňuje při zaškrtnutí uložit grafy regresních křivek prvních 30 PCA skóre pro všechny vlastnosti.
- Stisknutím tlačítka Run regression se spočítá regrese podle dříve popsáných nastavení. Byl-li zaškrtnutý box save graphs bude uživatel vyzván k výběru složky, kam se mají grafy uložit. V této složce se vytvoří adresář pojmenovaný podle typu dat, na kterých byl spuštěn (F/M/N, -/normalized) a v něm složky podle typu proměnné a typu regrese (-/weighted, age/weight/height/BMI,1/2/3) a v nich se uloží 30 grafů.

7.8 Vstupní data

Program umí načíst složku s daty v následujícím formátu. Složka s daty obsahuje složky pojmenované číslem, které udává index tváře, a v něm je soubor remesh.obj s trojúhelníkovou sítí. V hlavní složce je navíc soubor specs.csv, ve kterém jsou uloženy vlastnosti jedince v pořadí id, pohlaví, váha, výška a věk. První řádek tohoto souboru slouží jako popis údajů. Na následujících řádcích jsou již zadané údaje.

8 Experimenty, porovnávání a validace

Nyní si ukážeme rozdíly mezi modely PCA pro muže, ženy a globálním modelem (model muže a žen dohromady). Podíváme se, kolik hlavních komponent je potřeba k slušné aproximaci tváře z hlediska vizuálního. Tak nahlédneme zda se podařilo najít uspokojivé propojení s vlastnostmi jedinců a s modelem pomocí různých typů regrese.

8.1 Aproximace obličeje pomocí PCA

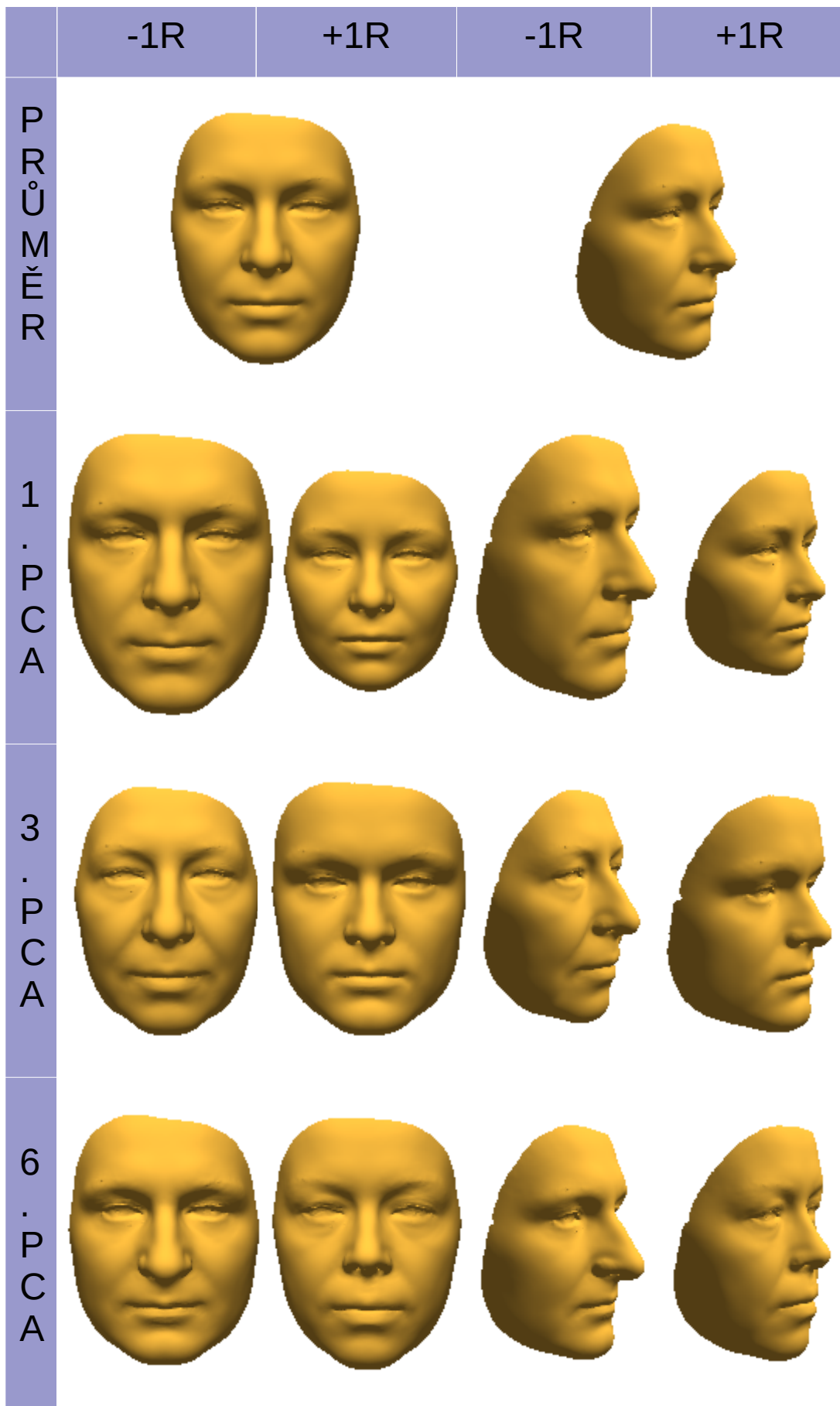
V této podkapitole se zaměříme na model vzniklý pomocí PCA a na jeho ověření.

8.1.1 Komponenty

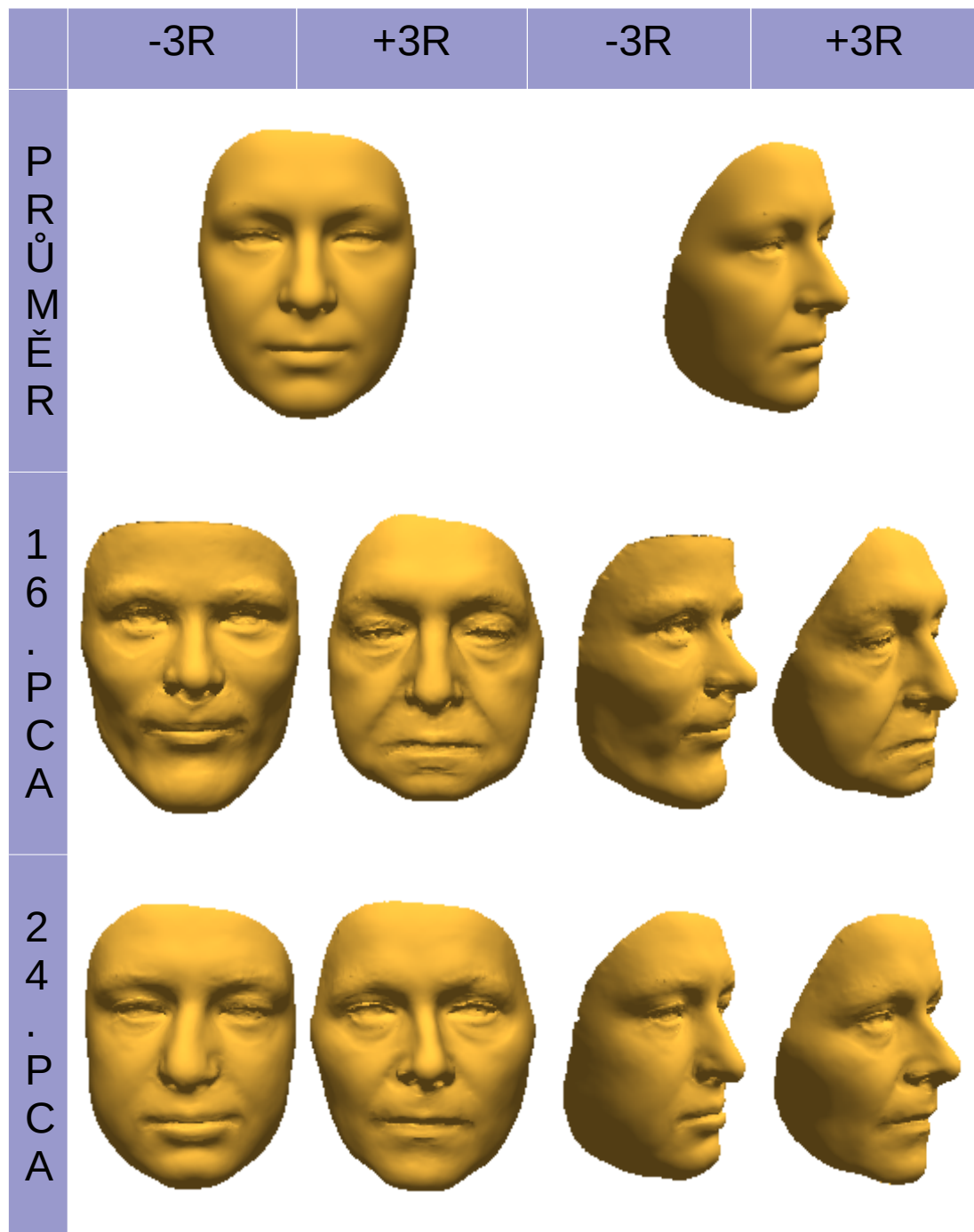
Nejdříve si ukážeme vliv různých PCA komponent na tvar obličeje. Na obrázku 8.1.1 je vidět průměrná tvář ze šikmého profilu a zepředu. Zobrazené tváře byly vytvořené globálním modelem, tj. model tvoří jak ženy, tak muži. Dále jsou na obrázku vyobrazeny změny této průměrné tváře pomocí první (56% vysvětlujícího rozptylu), třetí (7% vysvětlujícího rozptylu) a šesté komponenty (1,8% vysvětlujícího rozptylu). Hodnota skóre byla nastavená na jeho jednonásobek rozsahu, tj. pro danou komponentu se vybere největší absolutní hodnota ze všech jejích skóre (ve všech tabulkách bude R označovat tento rozsah hodnot). Ačkoliv tvář se mění celá, tak je z obrázků vidět výrazná tendence jen k určité změně. Například první komponenta by se dala označit za velikost tváře, třetí komponenta se zaměřuje na čelo a jeho zbytnění, šestá komponenta pro změnu zásadně mění velikost nosu.

Na dalším obrázku 8.1.2 je zobrazená 16. (0,4% vysvětlujícího rozptylu) a 24. komponenta (0,2% vysvětlujícího rozptylu) globálního modelu. Hodnota je opět nastavená na minimální a maximální hodnotu rozsahu R, ale tentokrát je zvětšená na trojnásobek této hodnoty, aby byl zřetelně vidět efekt komponenty na celkový tvar. Komponenty s vyšším pořadovým číslem (nižším vysvětlujícím rozptylem) mají na model menší efekt a jejich vliv není tak zřetelný. Nicméně právě tyto drobné změny dávají obličeji jeho charakter.

Obecně by se z pozorování dalo říct, že prvních několik komponent řeší především tvar jako takový a další komponenty dotvářejí celkový efekt, který je tvořen různými asymetriemi a vráskami.



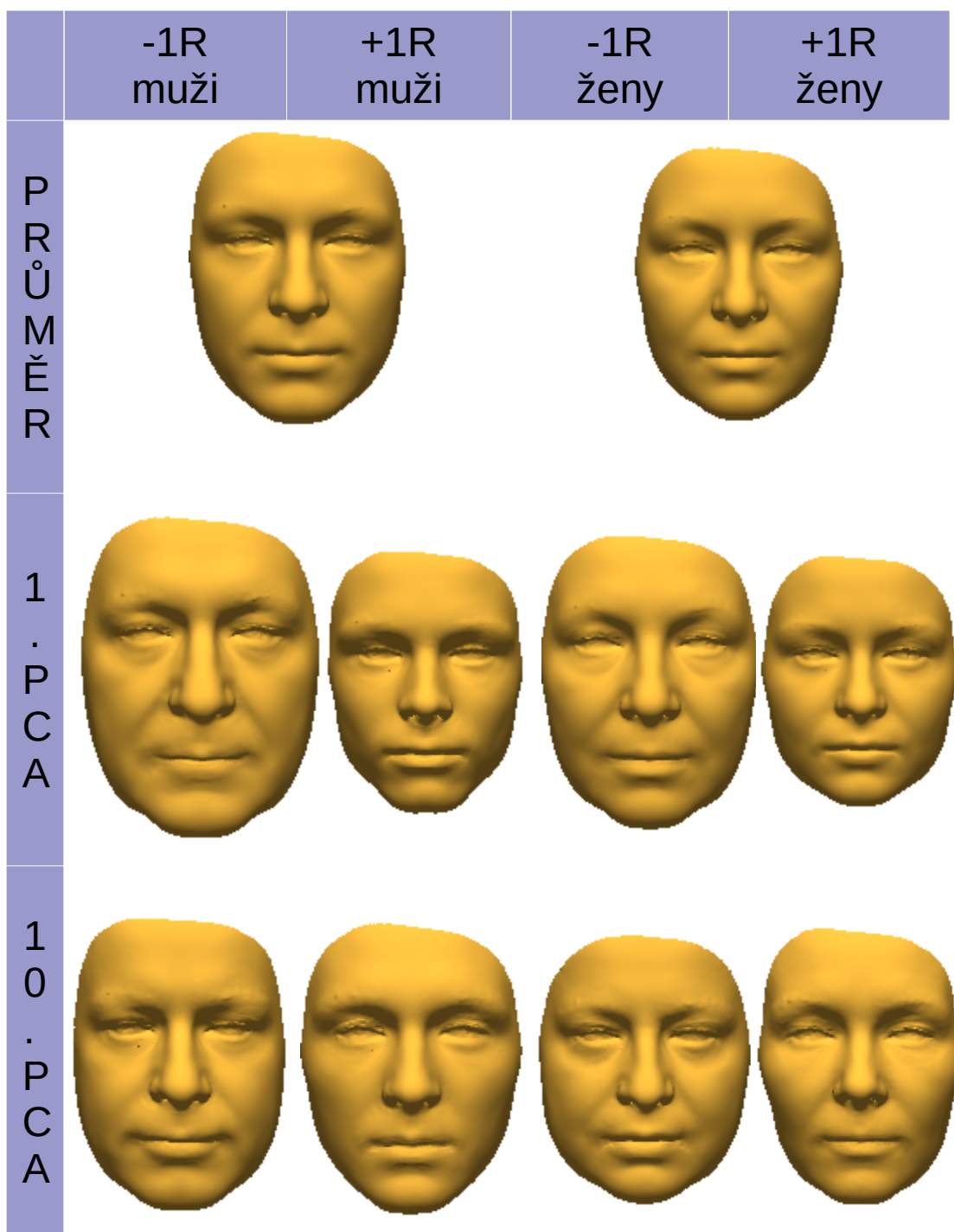
Obrázek 8.1.1: Zobrazuje průměrnou tvář globálního modelu po změně některých PCA komponent na hodnotu jednonásobku jejich rozsahu v datasetu.



Obrázek 8.1.2: Zobrazuje průměrnou tvář globálního modelu po změně některých PCA komponent na hodnotu trojnásobku jejich rozsahu v datasetu.

Na obrázku 8.1.3 je vyobrazený rozdíl mezi mužským a ženským modelem. V levé části tabulky jsou zobrazené mužské tváře a v pravé ženské. První řádek zobrazuje vliv první komponenty a druhý řádek zobrazuje vliv desátého skóre. Ačkoliv první PCA komponenta opět určuje „velikost tváře“, tak už v průměrné tváři je na první pohled vidět rozdíl mezi mužskou a ženskou tváří. U 10. komponenty už je tento rozdíl diametrální – mužská tvář inklinuje k obdélníkovému tvaru zatímco

ženská ke kulatému, což naznačuje rozdílné tendence, I celkově ženské obličejy vypadají ženštěji a mužské mužsky.

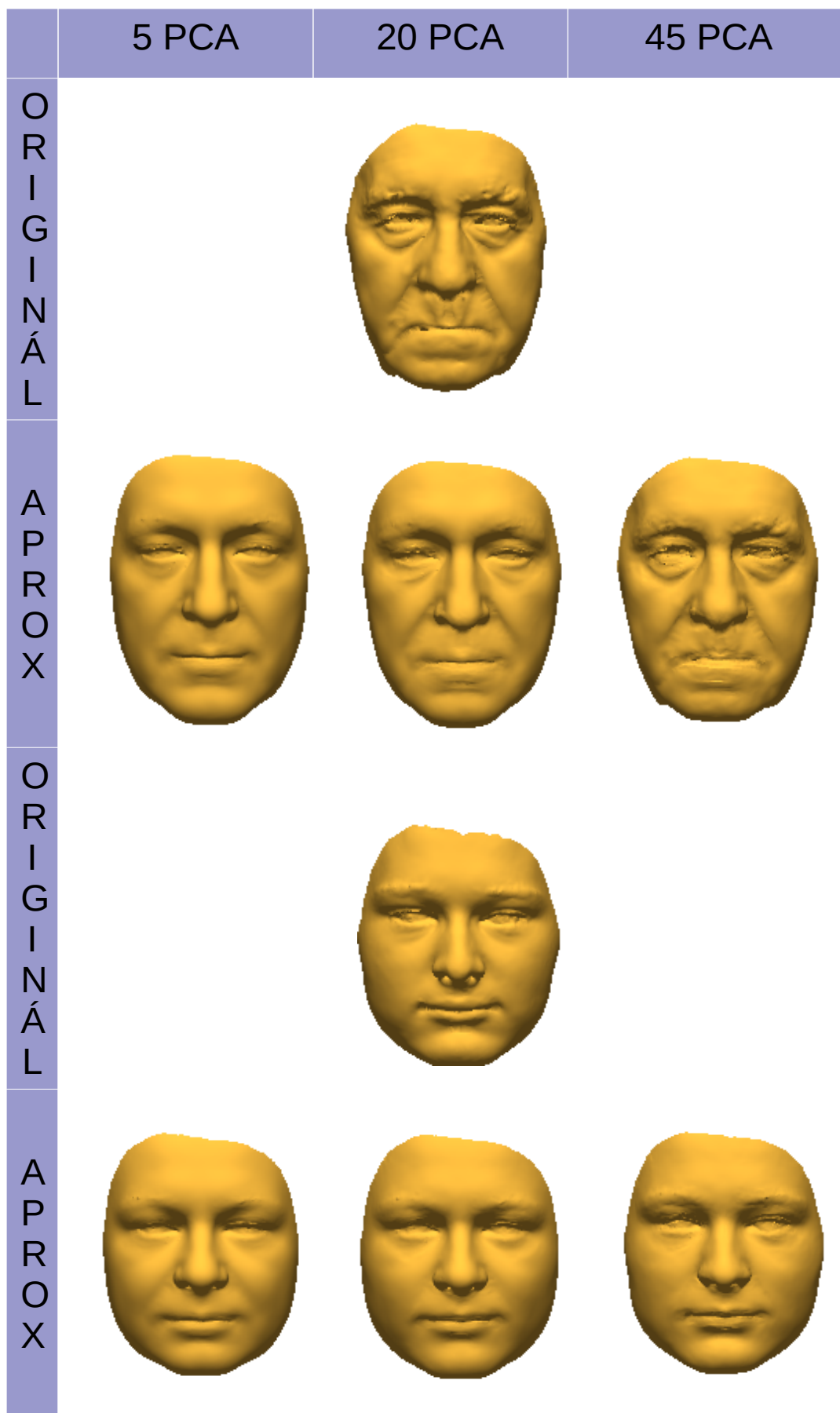


Obrázek 8.1.3: Vizualizace změny 1. a 10. komponenty v mužském a ženském modelu

Kolik komponent je třeba? Na obrázku 8.1.5 jsou zobrazené grafy vysvětlujících rozptylů pro všechny modely a pomocí pravidla broken-stick (Peres-Neto a kol., 2005) je nalezen práh, který určuje kolik hlavní komponent by v modelu mělo určitě zůstat. Nicméně naše úloha je hodně o vizuální stránce, a tak se podíváme na

výsledné aproximace. K porovnání si vezmeme model muže, protože u něj byl nalezen nejnižší práh. Na obrázku 8.1.4 je zobrazen postarší a mladší muž a pod jejich původním vzhledem se zobrazují aproximace pomocí prvních 5 (75% celkový počet vysvětlujícího rozptylu), 20 (necelých 91% celkového vysvětlujícího rozptylu) a 45 (96% celkový počet vysvětlujícího rozptylu) komponent PCA.

Z předchozích pokusů vyplývá, že pro mladší jedince můžeme počet komponent dost zredukovat, což je dáno tím, že PCA komponenty s vyšším pořadovým číslem dotvářejí jemnější rozdíly, zatímco PCA komponenty s nižším pořadovým číslem vytváří celkový tvarový základ obličeje.



Obrázek 8.1.4: Zobrazuje vliv počtu PCA komponent na výsledný vzhled. V řádce originál je zobrazena originální tvář (staršího a pak mladšího muže) a v následujícím řádku jsou zobrazené aproximace s 5, 20 a 45 komponentami.

8.1.2 Validace

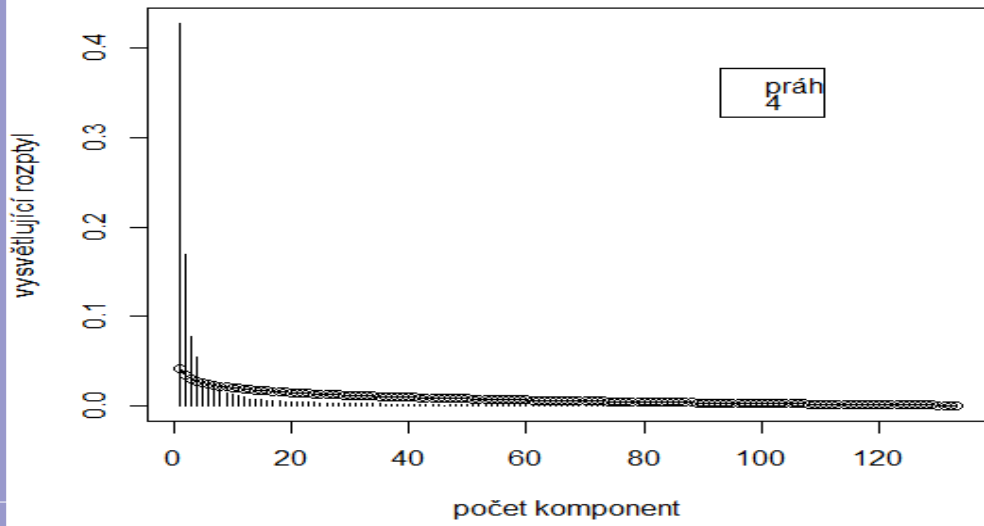
K vyhodnocení PCA modelů jsme použili podobné techniky jako v článku (Booth a kol., 2016). Z grafů v 8.1.5 je vidět, že došlo k velké redukci dimenze, a proto model vyžaduje jen pár desítek komponent. Pro účel validity byly modely podrobeny testu specifičnosti. Test probíhal následovně: Pro každou tvář se počítal model z dat bez její přítomnosti. Poté se tváře aproximovaly pomocí tohoto modelu s různým počtem komponent (2, 20, 40, 60 a všechny) a pro každý vrchol se spočítala vzdálenost od originální tváře. Tyto vzdálenosti se následně zprůměrují přes všechny vrcholy a testovací tváře. K porovnání chyb byli také spočtené i aproximace všech tváří, které tvořily model, a jejich zprůměrované průměrné vzdálenosti.

Ženy	Průměr	Max	Min
5 PCA test	1,2265	4,7222	0,7522
5 PCA všechny	1,1783	4,5063	0,7319
20 PCA test	0,8842	3,1271	0,5237
20 PCA všechny	0,7555	2,6295	0,4358
40 PCA test	0,7367	2,5793	0,4149
40 PCA všechny	0,5579	1,8461	0,3176
60 PCA test	0,6803	2,3041	0,3721
60 PCA všechny	0,4407	1,3733	0,2435
všechny PCA test	0,5918	1,8843	0,3145
všechny PCA všechny	0,0224	0,09419	0,0029

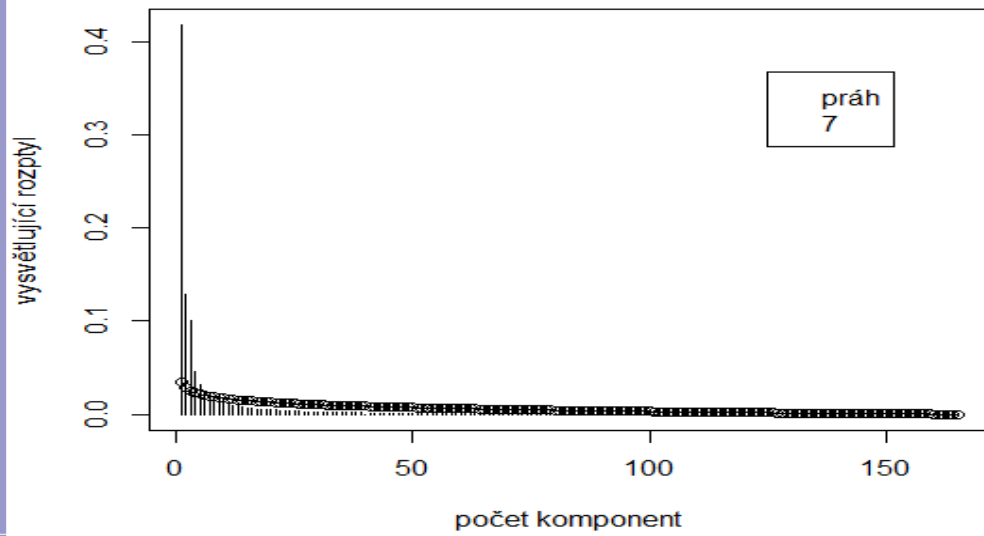
Tabulka 1: Průměrná chyba aproximace pro testovací data a pro všechna data

GRAF

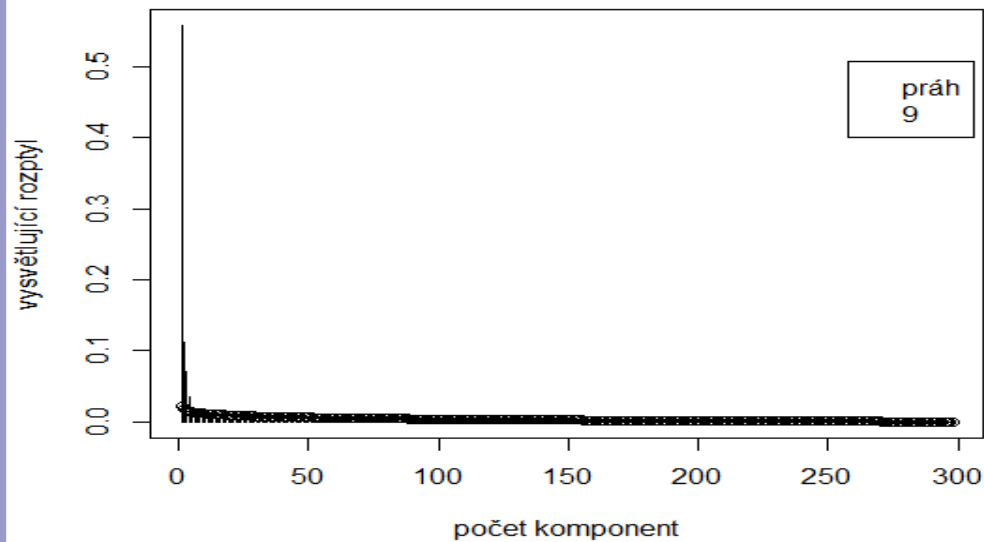
M
U
Ž
Í



Ž
E
N
Y

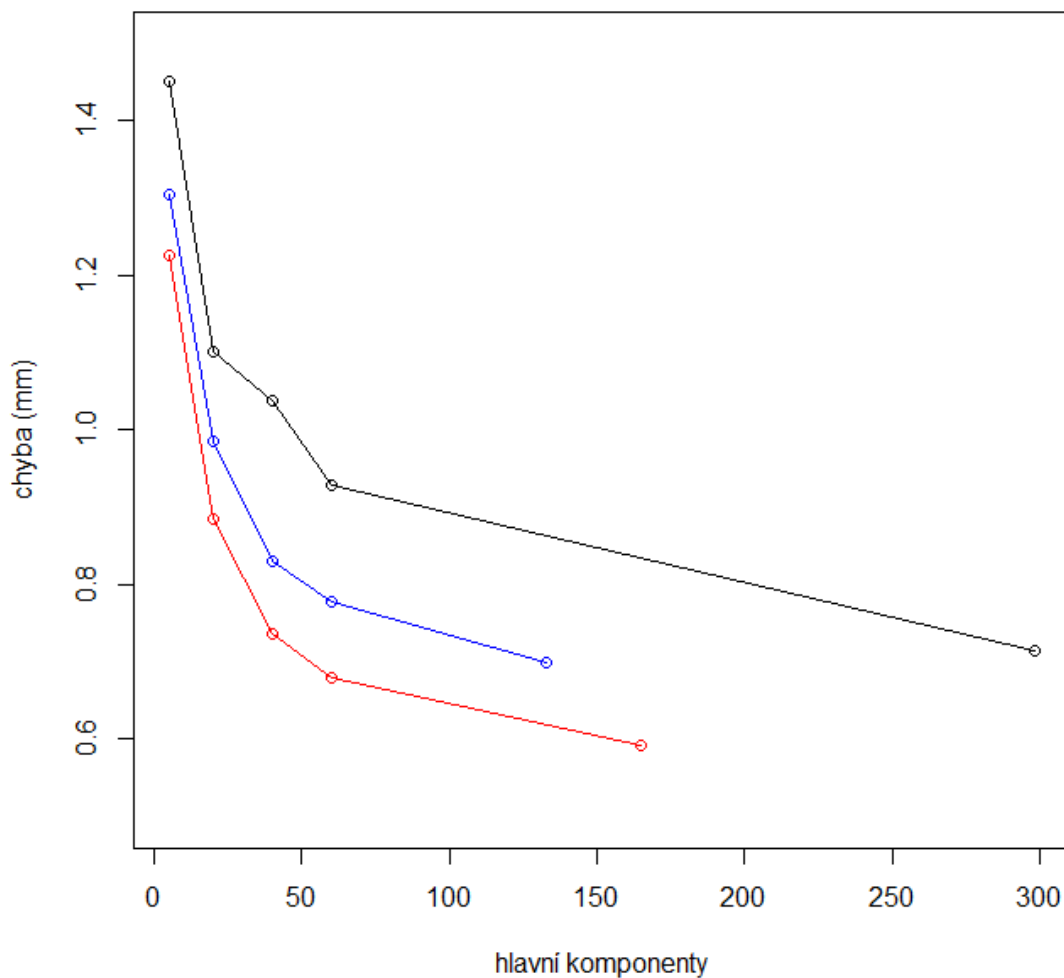


G
L
O
B
Á
L
N
Í



Obrázek 8.1.5: Grafy vysvětlujících rozptylů pro mužský, ženský a globální model. Číslo v rámečku určuje práh podle pravidla broken-stick (zobrazeno kolečkem) - tento počet komponent by určitě měl být zahrnut v daném modelu.

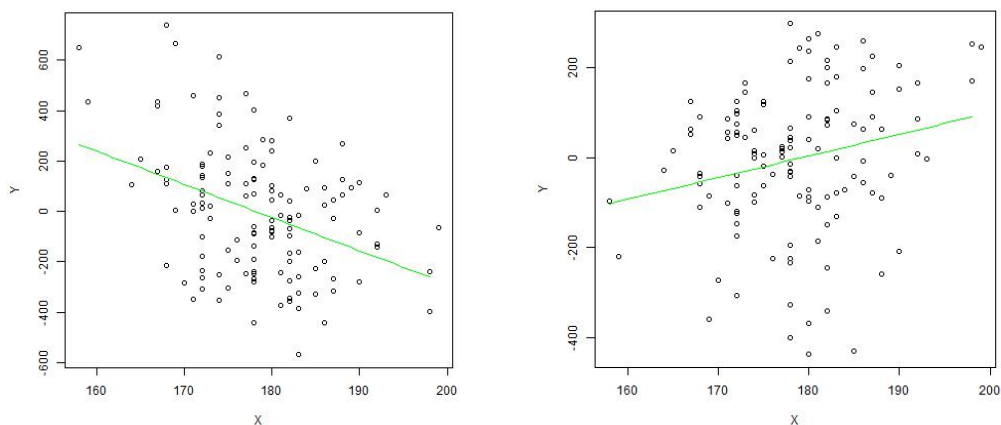
Tabulka žen byla vybrána na základě obrázku 8.1.6, kde se ukázalo, že ženský model má nejlepší aproximace. V práci Booth a kol. (Booth a kol., 2016) sice došly k menším chybám, ale jen v řádu desetin milimetru, zatímco měli několikanásobně větší dataset trénovací dataset (10000 tváří). Ženský model má lepší aproximaci pravděpodobně kvůli menší proměnlivosti v tváři a většímu počtu trénovacích dat (165).



Obrázek 8.1.6: Chyby globálního (černá čára), mužského (modrá čára) a ženského (červená čára) na testovacích datech podle počtu zvolených komponent

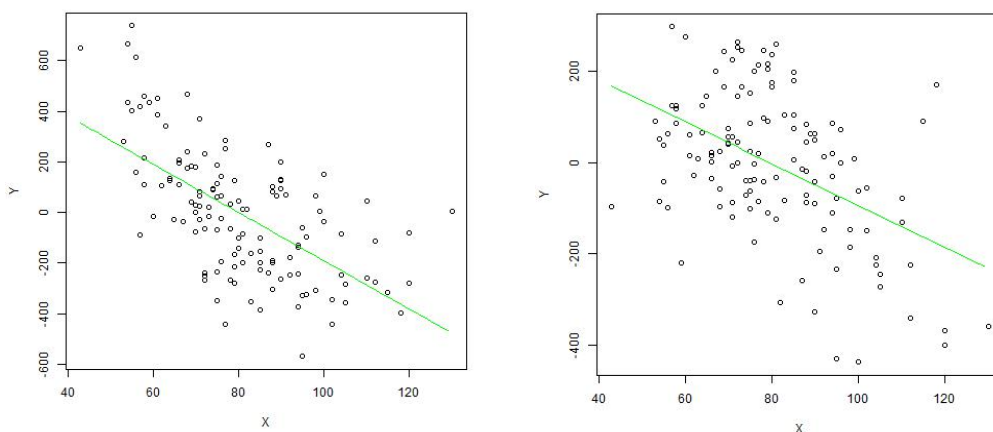
8.2 Modelování vlastností

Na obrázku 8.2.1 je vyobrazena závislost mezi výškou a prvníma dvěma komponentami, které modelují muže. Pokud bychom si z předchozích pokusů první komponentu označili za celkovou velikost a druhou za šířku tváře, dalo by se říci, že s přibývajícím výškou se obličej zvětšuje a zužuje.



Obrázek 8.2.1: lineární závislost vlastnosti výšky u mužů. Vlevo: mezi první komponentou a výškou, vpravo: mezi druhou komponentou a výškou

Na obrázku 8.2.2 jsou grafy s lineární závislostí na váze. Je vidět, že s přibývajícím váhou obličej roste a současně se rozšiřuje.



Obrázek 8.2.2: lineární závislost váhy u mužů. vlevo: mezi první komponentou a váhou, vpravo: mezi druhou komponentou a váhou

Pro základní porovnání využitých metod vznikly tabulky 2, 3 a 4. Jelikož nemáme k dispozici longitudinální data, tak jediné, co můžeme porovnávat, je průměrný obličej z nějaké skupiny s průměrným obličejem z jiné skupiny jedinců. Ve zmíněných

tabulkách porovnááme následující metody:

- lineární regresi a její vážené varianty s parametrem od 1 do 9
- regresi druhého stupně a její vážené varianty s parametrem od 1 do 9
- regresi třetího stupně a její vážené varianty s parametrem od 1 do 9

Postup probíhá následovně: vezmeme si skupinu jedinců a každého jedince pomocí některé z dříve zmíněných metod přetransformujeme do výsledné skupiny. Všechny transformace zprůměrujeme a vznikne tak průměr obličej predikce. Tento nový průměrný obličej porovnáme s průměrným obličejem vzniklým z původních dat s pomocí průměrné chyby na vrchol. Tu spočítáme jako Euklidovskou vzdálenost mezi odpovídajícími vrcholy. Pro srovnání zlepšení je v tabulkách také zmíněný rozdíl mezi průměrným obličejem výchozí skupiny a průměrným obličejem cílové skupiny (viz. informace - skutečný rozdíl v tabulce). Výsledky jsme zkoumali pro následující transformace:

- transformace věku ze skupiny 15-40 let do skupiny 51- 76 let
- transformace výšky ze skupiny 150 – 175 cm do skupiny 171- 196 cm
- transformace váhy ze skupiny 45-80 kg do 81-116 kg.

Skupiny byli vybrány tak, aby zhruba reprezentovali polovinu jedinců s nízkou hodnotou dané vlastnosti a transformovaly se do skupiny s vyšší hodnotou stejné vlastnosti. K testu byly použity všechny PCA komponenty. Z výsledků vyplývá, že věk je nejlepší modelovat pomocí polynomu stupně tři a náš způsob vážení nedává lepší výsledky než samostatná kubická křivka. Toto platí u všech modelů. U výšky pro změnu vychází nejlépe lineární regrese s vážícím koeficientem 1. A u váhy je to rozdílné: pro muže nejlépe vyšla kubická regrese s parametrem 1, u žen kvadratická regrese s parametrem 2 a globální model má nejlepší výsledek s kvadratickou křivkou s parametrem 1. (Celé záznamy lze vygenerovat pomocí programu v příloze a výsledky jsou v milimetrech).

Celkově můžeme říci, že námi zvolený postup tvoření predikce se přibližuje k průměrné tváři cílové skupiny, ať použijeme jakoukoliv ze zmíněných metod. I když by se mohlo zdát, že predikovat u žen výšku pomocí kubické křivky je nesmysl (průměrná chyba je přes 4 mm), výsledek v tabulce je ovlivněn skutečností, že data pro aproximaci neobsahují moc žen s tak velkou výškou (171-196) a polynomy vyšších stupňů mají tendenci pak strmě stoupat či klesat. Aktuální verze programu tento fakt zatím nijak neošetřuje.

	vlastnost	predikce	skutečný rozdíl
informace	věk	Z 15-40 do 51-76	1,2216
stupeň regrese	1	2	3
bez vah	0,303	0,3632	0,1754
1	0,3758	0,4594	0,2931
2	0,3175	0,3888	0,1999
3	0,3109	0,3779	0,1886
4	0,3084	0,3735	0,1844
5	0,3071	0,3711	0,1822
6	0,3063	0,3696	0,1809
7	0,3058	0,3686	0,18
8	0,3054	0,3679	0,1794
9	0,3051	0,3673	0,1789
max	0,3758	0,4594	0,2931
min	0,303	0,3632	0,1754
průměr	0,31453	0,38073	0,19428
	vlastnost	predikce	skutečný rozdíl
informace	výška	Z 150-175 do 171-196	0,8966
stupeň regrese	1	2	3
bez vah	0,8915	1,2097	4,296
1	0,4869	1,3267	4,4757
2	0,7628	1,1981	4,3356
3	0,8162	1,1997	4,3183
4	0,8384	1,2018	4,3115
5	0,8504	1,2032	4,3079
6	0,858	1,2042	4,3056
7	0,8633	1,205	4,3041
8	0,8671	1,2056	4,303
9	0,87	1,206	4,3021
max	0,8915	1,3267	4,4757
min	0,4869	1,1981	4,296
průměr	0,81046	1,216	4,32598
	vlastnost	predikce	skutečný rozdíl
informace	váha	Z 45-80 do 81-116	1,358
stupeň regrese	1	2	3
bez vah	0,6187	0,4557	0,4666
1	0,3587	0,4551	0,4714
2	0,5517	0,4207	0,4455
3	0,5809	0,4336	0,4532
4	0,5924	0,4397	0,4569
5	0,5986	0,4432	0,459
6	0,6024	0,4454	0,4603
7	0,605	0,447	0,4613
8	0,6069	0,4481	0,462
9	0,6083	0,449	0,4625
max	0,6187	0,4557	0,4714
min	0,3587	0,4207	0,4455
průměr	0,57236	0,44375	0,45987

Tabulka 2: Model ženy – porovnání různých typů regresí v různých skupinách.

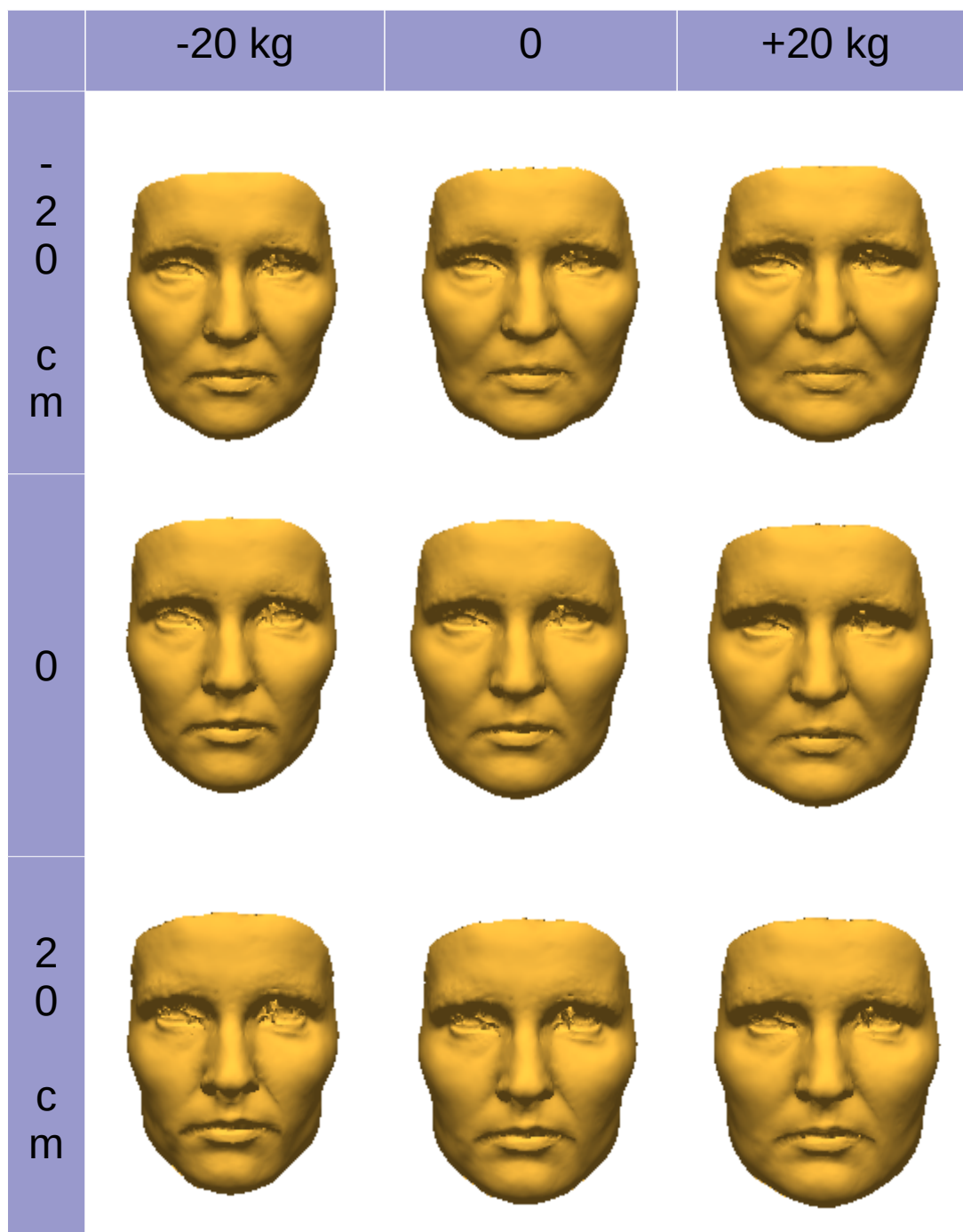
	vlastnost	predikce	skutečný rozdíl
informace	věk	Z 15-40 do 51-76	1,9528
stupeň regrese	1	2	3
max	0,5633	0,596	0,4121
min	0,5255	0,5746	0,2143
průměr	0,53566	0,58738	0,23847
	vlastnost	predikce	skutečný rozdíl
informace	výška	Z 150-175 do 175-200	2,4896
stupeň regrese	1	2	3
max	0,7733	0,7733	0,8758
min	0,5962	0,659	0,7128
průměr	0,73336	0,74412	0,8401
	vlastnost	predikce	skutečný rozdíl
informace	váha	Z 45-80 do 81-116	1,9134
stupeň regrese	1	2	3
max	0,7595	0,5827	0,5375
min	0,4835	0,4228	0,4318
průměr	0,71195	0,54931	0,50893

Tabulka 3: Model muži – porovnání různých typů regresí v různých skupinách.

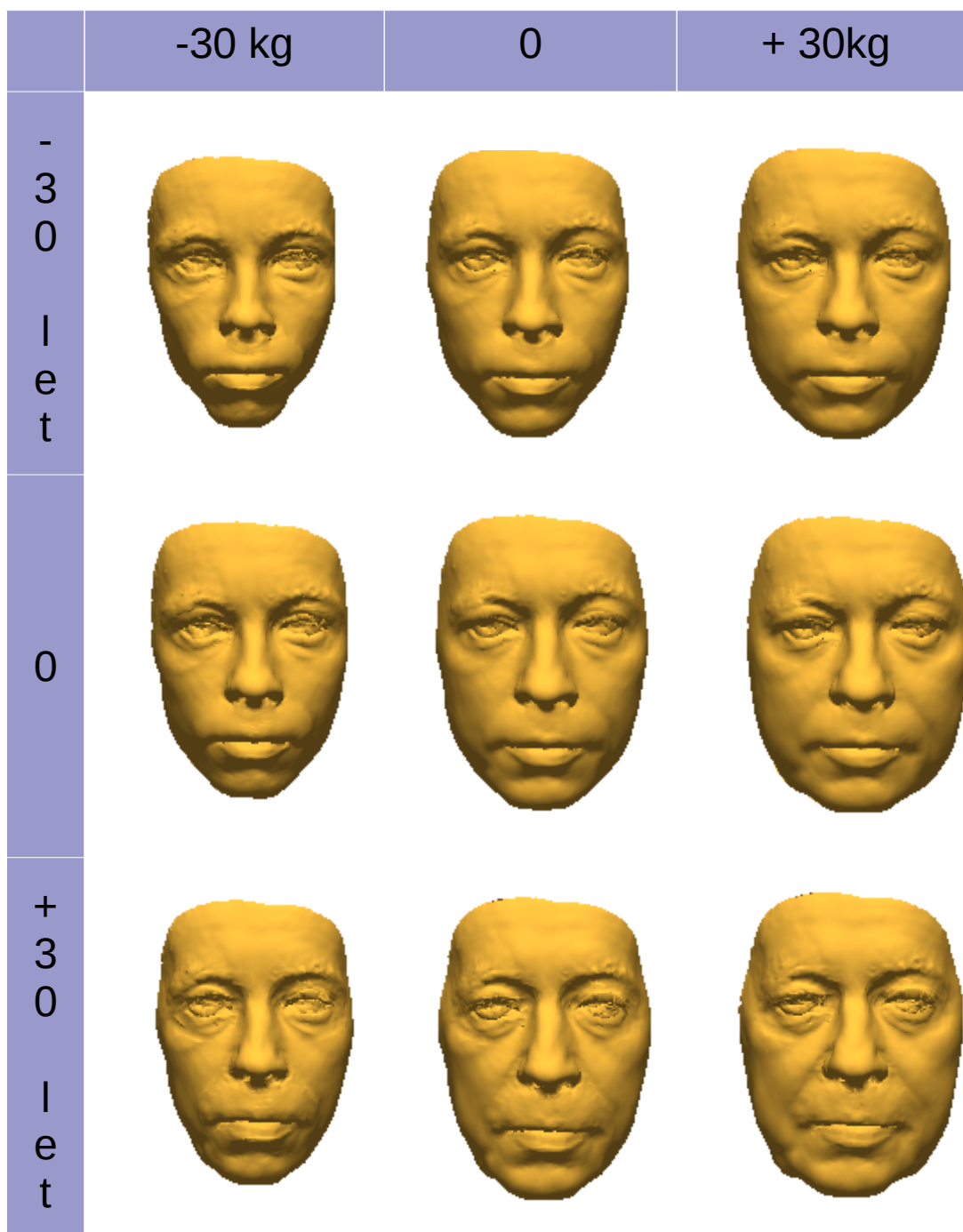
	vlastnost	predikce	skutečný rozdíl
* informace	věk	Z 15-40 do 51-76	1,4038
stupeň regrese	1	2	3
max	0,4106	0,4433	0,243
min	0,332	0,3911	0,1592
průměr	0,3466	0,4001	0,17212
	vlastnost	predikce	skutečný rozdíl
informace	výška	Z 150-175 do 175-200	3,0647
stupeň regrese	1	2	3
max	1,1342	1,2117	1,1957
min	0,5056	0,681	0,59
průměr	1,01581	1,11706	1,08586
	vlastnost	predikce	skutečný rozdíl
informace	váha	Z 45-80 do 81-116	2,3338
stupeň regrese	1	2	3
max	0,8101	0,5866	0,6066
min	0,4016	0,3032	0,3065
průměr	0,73966	0,52255	0,53749

Tabulka 4: Globální model – porovnání různých typů regresí v různých skupinách.

Na obrázcích 8.2.3 a 8.2.4 jsou zobrazené nějaké predikce podle nejlepšího nastavení z tabulek 2 a 3.



Obrázek 8.2.3: Predikce ženy (počáteční věk 44, váha 71, výška 175) pro výšku ± 20 cm a váhu ± 20 kg. Predikce je nastavená podle tabulky pro model ženy (nastavená podle tabulek)

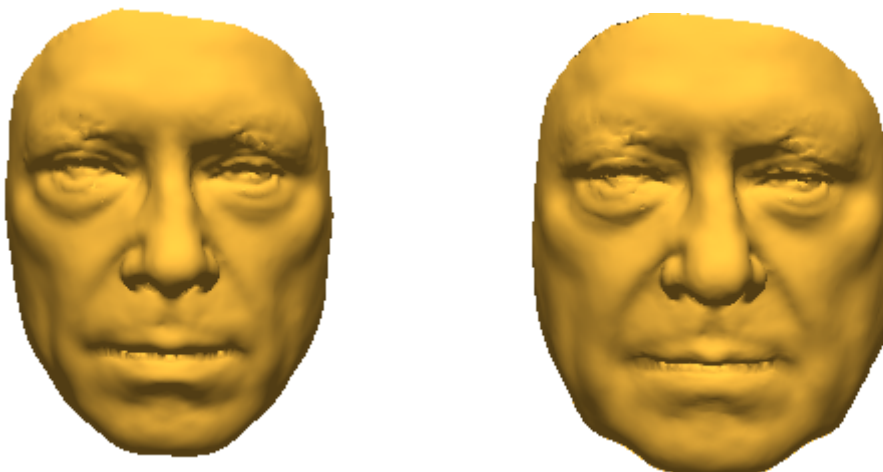


Obrázek 8.2.4: Predikce muže (počáteční věk 50, váha 80, výška 180) pro věk +-30 let a pro váhou +- 30 kilogramů

Pro výraznější viditelný efekt jsou na obrázcích 8.2.5 a 8.2.6 zobrazeny výsledky extrémně dlouhé predikce, taktéž na základě tabulek 2 a 3.



Obrázek 8.2.5: Predikce ženy z věku 19 na 90 let (kubická regrese)



Obrázek 8.2.6: Predikce muže z váhy 66 na 110 kg (kubická regrese s parametrem 1)

8.3 Shrnutí

Celkově shrnuto - výsledky nalezené PCA jsou velmi zajímavé. Jsou zde vidět znatelné tendence, jež se dají pojmenovat (jako například celková velikost obličeje, velikost nosu). Navíc tvář se dá celkem věrohodně aproximovat 45 či menším počtem hodnot. Celkem uspokojivé jsou i nalezené predikce. Tváře vypadají věrohodně (lidsky) a zároveň predikce zachovává z velké části charakteristiku

obličeje. U zvýšeného věku dochází k výraznějším záhybům ve tváři. S přibývajícím
vážou se tvář zakulacuje zejména v oblasti tváří. A při větší výšce se tvář protahuje.

9 Zajímavé implementační detaily

Jako základ programu posloužil projekt 101MeshEditor poskytnutý vedoucím. Tento projekt umožňuje uživateli načíst tváře typu .obj a .ply., a pomocí knihovny OpenTK, která implementuje OpenGL standard pro jazyk C#, umožňuje načtenou tvář uživateli zobrazit v 3D podobě. Projekt navíc umožňuje uživatelsky přívětivé prohlížení tváře ze všech stran a různé režimy nastavení zobrazení popsané v podkapitole 7.1.2 Tento projekt byl následně upraven a rozšířen do podoby popsané v kapitole Software pro modelování a vizualizaci tvaru obličeje 7.

9.1 Výkon PCA

První pokus výpočtu PCA byl zprostředkován pomocí knihovny R.NET, která implementuje použití jazyka R (jazyk vhodný pro statistické výpočty) v rámci .NET. Bohužel výpočet pro data obsažená v našem datasetu nebylo možné dokončit. Proto došlo na vlastní implementaci PCA podle kapitoly Výpočet PCA pro data s velkou dimenzí 4.4, kde R.NET počítalo jen vlastní čísla a vlastní vektory malé matice T zmiňované v kapitole 4.4. Výpočet sice zdárně proběhl, ale doba výpočtu byla nepřijatelná. Měřením se ukázalo, že nejdélní část výpočtu spočívá v násobení matic. Konkrétně výpočet matice T , která vznikla násobením matic velikosti 298×45009 s 45009×298 , byl časově náročný. I po vhodné vícevláknové implementaci násobení matic pomocí Threadpoolu (fronta úloh, které jsou postupně zpracovávány pomocí několika vláken), trval výpočet samotné matice T 1 minutu, což bylo zlepšení, ale ne takové, v které se doufalo. Nakonec pro násobení a výpočet vlastní čísel a vektorů byla zvolena otevřená knihovna Math.NET Numerics, která umožňuje po zavolání funkce

```
Control.UserNativeMKL(),
```

využívat Intel Math Kernel Library (MKL), což je vysoce optimalizovaná matematická knihovna pro procesory Intel. Tato knihovna dokáže matici T spočítat během půl vteřiny, což je výrazné zlepšení. Celý výpočet PCA trvá přibližně jednu sekundu. Výkon byl měřen na Intel Core 7i 1.6Ghz čtyř-jádrový s hyper-threadingem na 8 jader.

9.2 Track bar

Uživatelské rozhraní programu je psané pomocí knihovny Windows Forms. Windows Forms obsahuje základní uživatelské prvky jako tlačítka, zaškrtačací boxy a další. K nastavení číselných hodnot je v programu hojně používán prvek TrackBar. Jedná se grafický prvek posuvného jezdce po stupnici. Nicméně implementace tohoto prvku ve Windows Forms je dosti omezená:

- rozsah hodnot v prvku track bar je reprezentovaný jen pomocí hodnoty int
- hodnota nastavená pomocí prvku track bar není číselně zobrazená
- lze nastavit jen několik grafických módů, které nelze měnit

Pro pohodlí uživatele nezbývalo nic jiného než tyto nedostatky nahradit vlastní implementací. Jelikož napsat si celý vlastní TrackBar s plnou funkcionalitou není úplně časově zanedbatelné. Jako základ nám posloužila implementace prvku TrackBar, od které dědíme. Jelikož chceme uživateli nabídnout funkcionalitu změny barvy jezdce při změně hodnoty, aby bylo zřetelné, které hodnoty se různými nastavením změnily a které ne, tak musíme v poděděné třídě od třídy TrackBaru přepsat metodu OnPaint, která slouží k vykreslování grafického výstupu při jakékoliv změně prvku. Nejprve ale musíme změnit jedno nastavení pro novou poděděnou třídu, protože pro všechny prvky kromě prvku Control není metoda OnPaint defaultně dostupná, tj. bez změny níže zmíněného příznaku nebude metoda OnPaint při překreslování obrazovky volána.

```
SetStyle(ControlStyles.UserPaint, true);
```

Nyní si už můžeme zvolit libovolnou grafickou reprezentaci prvku TrackBar, ale má to jeden háček - Windows Forms API neposkytuje ani informace o poloze jezdce.

A tak se musíme uchýlit ke triku, kde voláme původní implementaci, protože třída TrackBar zastřešuje tento prvek Windows.


```

public Rectangle Slider
{
    get
    {
        RECT rc = new RECT();
        SendMessageRect(this.Handle, TBM_GETTHUMBRECT, IntPtr.Zero, ref rc);
        return new Rectangle(rc.left, rc.top, rc.right - rc.left, rc.bottom - rc.top);
    }
}

private const int TBM_GETTHUMBRECT = 0x400 + 25;
private struct RECT { public int left, top, right, bottom; }
[DllImport("user32.dll", EntryPoint = "SendMessageW")]
private static extern IntPtr SendMessageRect(IntPtr hWnd, int msg, IntPtr wp, ref RECT lp);

```

V tomto bodě můžeme libovolně vykreslit novou grafickou reprezentaci prvku `TrackBar` se zachováním ostatní funkcionality původního `TrackBaru`.

Pokud chceme uživateli poskytnout možnost sledovat nastavené hodnoty, musíme náš nový `TrackBar` propojit například s prvkem `TextBox`, který zobrazuje hodnotu, kde se nachází jezdec. Desetinné číslo je pak reprezentované vhodným vynásobením konstantou určující vybranou desetinnou přesnost. V programu se pak využívají různé typy prvků `TrackBar` s rozšířenou funkcionalitou, které jsou potomkem tohoto našeho “chytrého“ `TrackBaru`.

9.3 Dvě OpenGL okna

Pro zobrazení výstupu spočítaného grafickou kartou se využívá knihovna `OpenTK.GLControl`. Tato knihovna umožňuje přidat okno `OpenGL` jako prvek formuláře. Vytvořením dvou objektů typu `OpenTK.GLControl` se vytvoří dvě na sobě nezávislá okna `OpenGL`. Každé okno má svůj vlastní `OpenGL` kontext, který ukládá veškerá nastavení týkající se `OpenGL`. V podstatě je to něco jako objekt reprezentující `OpenGL`. Chceme-li přenastavit některou vlastnost `OpenGL` s daným kontextem, musíme nejprve tento kontext nastavit jako aktuální. Přepínání kontextu zaručí volání funkce

```
glControl1.MakeCurrent();
```

tj. veškerá následující nastavení `OpenGL` jsou platná pro první okno aplikace označené jako `glControl1`. Ačkoliv lze `OpenGL` okna vykreslovat paralelně, v přiloženém programu jsou jednotlivá uživatelská okna vykreslována jedno po druhém s přepnutím kontextu.

Některé objekty `OpenGL` lze sdílet mezi různými kontexty. Objekty v `OpenGL` lze

rozdělit na dva typy : na běžné objekty a na objekty typu kontejner. Jakýkoliv objekt, který není typu kontejner, může být sdílen mezi kontexty, stejně tak jako Sync Objects a GLSL Objects. Více viz dokumentace OpenGL dokumentace

Závěr

Během práce vznikl uživatelsky přívětivý program, který uživateli umožňuje jednoduše modifikovat trojúhelníkové sítě pomocí několika ovládacích prvků v reálném čase. Trojúhelníkové sítě lze modifikovat proměnnými, které vznikly redukcí dimenze, a také lze predikovat tvar tváře nastavením fyzických vlastností jedinců. K pokusům posloužil dataset 298 tváří, které jsou reprezentované 45009 reálnými čísly. K tvářím byly přiloženy fyzické údaje o jedincích jako je pohlaví, váha, věk, výška.

K redukcí dimenze byla použita PCA. Díky vhodnému předzpracování dat mohla PCA najít optimální novou reprezentaci dat, která funguje překvapivě intuitivně - asi tak, jak by pravděpodobně popsal člověk tvar obličeje. Jako příklad si představíme nějakou situaci, kdy svědek popisuje pachatele policejnímu portrétistovi. Policejní portrétista většinou pokládá svědkovi otázky na tvar hlavy, je-li brada či čelo širší či užší, nebo jak veliké byly rty a nos. A právě takové podobné charakteristiky vykazuje použití PCA na našem datasetu.

Další modifikace trojúhelníkové sítě lze provádět nastavením fyzických hodnot. V tomto případě vytváříme predikce tváře, které zodpovídají například otázku: Jak by vypadala tvář daného člověka, kdyby přibral 20 kilo. Tyto predikce jsou počítány různými typy regrese. I tyto celkem jednoduché aproximační přístupy dokážou v datech nalézt zajímavé vztahy a následné predikce působí věrohodným dojmem. Nicméně za pokus by stálo vyzkoušet pro hledání těchto vztahů neuronové sítě.

Celkem zajímavé by také bylo naše predikce tváří porovnat (ať už numericky nebo vizuálně pomocí metod obsažených v programu, které byly pro účel porovnávání tváří vytvořeny) s reálnými daty a vyhodnotit tak úspěšnost. Ale bohužel taková data nejsou prozatím k dispozici. Nemáme vzorky osob o 10 let starší či o 20 kilo lehčí. A tak můžeme důvěryhodnost zvolených metod hodnotit jen z empirické zkušenosti a nebo jen přibližně porovnat účinnost různých typů metod na základě průměrných obličejů různých věkových, váhových či výškových skupin. Touto metodou porovnávání se došlo k závěrům, že pro věk jsou obecně lepší predikce pomocí kubické regrese, pro výšku lineární a pro váhu s polynomem stupně 2 nebo 3. Výsledné predikce vypadaly důvěryhodně. Dalším poznatkem bylo, že

vážená regrese může zlepšit výslednou predikci. S různými typy vážení by se pak dalo dále experimentovat.

Seznam použité literatury

BLANZ, V. a VETTER, T.(1999). A Morphable Model for the Synthesis of 3D Faces. V *Proceedings of the 26th Annual Conference on Computer Graphics and Interactive Techniques*, 187-194, ISBN: 0-201-48560-5, DOI: 10.1145/311535.311556

KRAJÍČEK, V.(2015). *Correspondence Problem in Geometrics Morphometric Tasks*. Praha. Disertační práce. Matematicko-fyzikální fakulta Univerzity Karlovy, Katedra softwaru a výuky informatiky. Vedoucí disertační práce RNDr. Josef Pelikán.

SCHERBAUM, K., SUNKEL, M., SEIDEL, H.-P., BLANZ, V(2007). Prediction of Individual Non-Linear Aging Trajectories of Faces. V *Computer Graphics Forum*, **26** (3), 285-294, DOI: 10.1111/j.1467-8659.2007.01050.x

AMBERG, B., PAYSAN, P., VETTER, T.(2009). Weight, Sex, and Facial Expressions: On the Manipulation of Attributes in Generative 3D Face Models. V *Advances in Visual Computing*, **5875**, 875-885, ISBN: 978-3-642-10330-8, DOI: 10.1007/978-3-642-10331-5_81

PAYSAN, P., LÜTHI, M., ALBRECHT, T., LERCH, A., AMBERG, B., SANTINI, F., VETTER, T.(2009). Face Reconstruction from Skull Shapes and Physical Attributes. V *Pattern Recognition*, 232-241, ISBN: 978-3-642-03798-6

PAYSAN, P.(2010). *Statistical Modeling of Facial Aging based on 3D Scans*. Basel. Inauguraldissertation zur Erlangung der Würde eines Doktors der Philosophie vorgelegt der Philosophisch-Naturwissenschaftlichen Fakultät der Universität Basel. Dissertationsleiter Prof. Dr. Thomas Vetter

TSENG, C.-Y., WANG, I.-J., CHU, C.-H.(2014). Parametric Modeling of 3D Human Faces using Anthropometric Data. V *IEEE International Conference on Industrial Engineering and Engineering Management, 2015*, DOI: 10.1109/IEEM.2014.7058686

CLAES, P., LIBERTON, D. K., DANIELS, K., ROSANA, K. M., QUILLEN, E. E., PEARSON, L. N., MCEVOY, B., BAUCHET, M., ZAIDI, A. A., YAO, W., TANG, H., BARSH, G. S., ABSHER, D. M., PUTS, D. A., ROCHA, J., BELEZA, S., PEREIRA, R. W., BAYNAM, G., SUETENS, P., VANDERMEULEN, D., WAGNER, J. K., BOSTER, J. S., SHRIVER, M. D.(2014). Modeling 3D Facial Shape from DNA. V *PLOS Genetics*, **10** (3), 1-14, DOI: 10.1371/journal.pgen.1004224

BOOTH, J., ROUSSOS, A., ZAFEIRIOU, S., PONNIAHY, ADUNAWAY, D.(2016). A 3D Morphable Model Learnt from 10,000 Faces. V *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 5543-5552, ISSN: 1063-6919, DOI: 10.1109/CVPR.2016.598

WANG, J., SHI, C.(2017). Automatic construction of statistical shape models using deformable simplex meshes with vector field convolution energy. V *BioMedical Engineering OnLine*, **16** (1), 49, ISSN: 1475-925X, DOI: 10.1186/s12938-017-0340-0

TRAN, L. a LIU, X.(2018). Nonlinear 3D Face Morphable Model. V *CoRR*, **abs/1804.0**, ArXiv ID: 1804.03786

PAYSAN, P., KNOTHE, R., AMBERG, B., ROMDHANI, S., VETTER, T.(2009). A 3D Face Model for Pose and Illumination Invariant Face Recognition.. V *AVSS*, 296-301, ISBN: 978-0-7695-3718-4

AMBERG, B., KNOTHE, R., VETTER, T.(2008). Expression invariant 3D face recognition with a Morphable Model. V *2008 8th IEEE International Conference on Automatic Face Gesture Recognition*, 1-6, DOI: 10.1109/AFGR.2008.4813376

MYDLOVÁ, M., DUPEJ, J., KOUDELOVÁ, J., VELEMÍNSKÁ, J.(2015). Sexual dimorphism of facial appearance in ageing human adults: A cross-sectional study. V *Forensic Science International*, **257**, 519.e1 - 519.e9, ISSN: 0379-0738, DOI: <https://doi.org/10.1016/j.forsciint.2015.09.008>

KOUDELOVÁ, J., HOFFMANNOVÁ, E., DUPEJ, J., VELEMÍNSKÁ, J.(2019). Simulation of facial growth based on longitudinal data: Age progression and age regression between 7 and 17 years of age using 3D surface data. V *PLOS ONE*, **14** (2), 1-16, DOI: 10.1371/journal.pone.0212618

PERES-NETO, P. R., JACKSON, D. A., SOMERS, K. M.(2005). How many principal components? stopping rules for determining the number of non-trivial axes revisited. V *Computational Statistics & Data Analysis*, **49** (4), 974-997, ISSN: 0167-9473, DOI: <https://doi.org/10.1016/j.csda.2004.06.015>

DVOŘÁKOVÁ, G.(2018). *Použití DNN pro analýzu trojúhelníkových sítí v geometrické morfometrii*. Praha. Bakalářská práce. Matematicko-fyzikální fakulta Univerzity Karlovy, Katedra softwaru a výuky informatiky. Vedoucí bakalářské práce RNDr. Josef Pelikán.

NEKVINDA, T.(2018). *Fixing of Facial Triangle Meshes*. Praha. Bakalářská práce. Matematicko-fyzikální fakulta Univerzity Karlovy, Katedra softwaru a výuky informatiky. Vedoucí bakalářské práce RNDr. Josef Pelikán.

Object Files, dokumentace, URL:

https://www.cs.utah.edu/~boulos/cs3505/obj_spec.pdf, navštíveno: březen 2019

BISHOP, Ch. M.(2006). *Pattern Recognition and Machine Learning*. První vydání. Springer. ISBN-10: 0-387-31073-8, ISBN-13: 978-0387-31073-2

GARETH, J., WITTEN, D., HASTIE, T., TIBSHIRANI, R.(2017). *An Introduction to Statistical Learning with Applications in R.*, 7. opravené vydání, Springer, ISBN 978-1-4614-7138-7 (eBook), URL: <http://faculty.marshall.usc.edu/gareth-james/ISL/ISLR%20Seventh%20Printing.pdf>

HLADÍK, M., Lineární algebra(nejen)pro informatiky, 2019, URL:

https://kam.mff.cuni.cz/~hladik/LA/text_la_kindle.pdf, navštíveno: duben 2019

HAMHALTER, J., TIŠER J., Diferenciální počet funkcí více proměnných, 2005,

<https://math.feld.cvut.cz/tiser/web7.pdf>MILITKÝ, J., Analýza hlavníchkomponent -

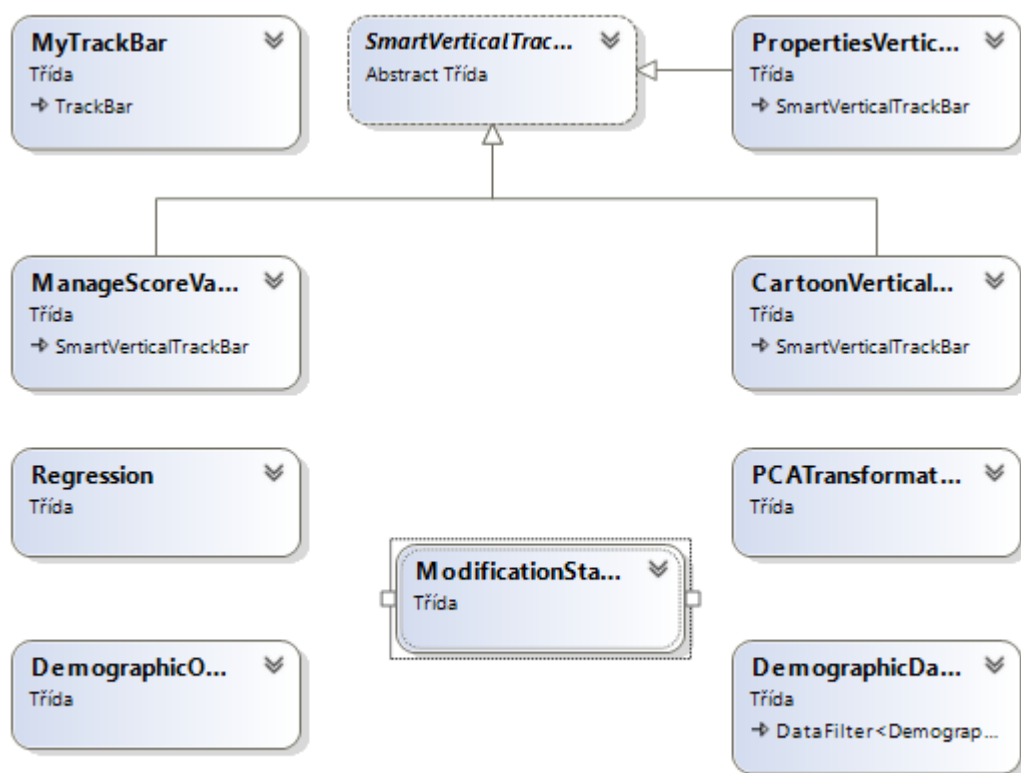
PCA, URL: <https://meloun.upce.cz/docs/license/sylaby-predmetu/militky/pca.pdf>, ,
navštíveno: duben 2019

FIALOVÁ, I.(2013). *Určení pohlaví na lebce člověka s využitím trojrozměrných virtuálních modelů*. Brno. Bakalářská práce. Masarykova univerzita, Přírodovědecká fakulta. Vedoucí práce Petra Urbanová.

OpenGL_Context, dokumentace, URL:
https://www.khronos.org/opengl/wiki/OpenGL_Context, navštíveno: duben 2019

Příloha

Návrh programu



Základním kamenem programu je ModificationStatus.cs. V ní se uchovává, odkaz na spočtenou PCA transformaci, odkaz na načtené fyzické vlastnosti jedince, kopie skóre z PCA - changedScores, sloužící k uchování uživatelského nastavení, kopie vlastností jedince – changedProperties, sloužící také k uživatelskému nastavení. Dále tato třída drží odkazy na veškeré ovládá scrollBar. Stěžejní funkce je changeObj(), která zajišťuje aktualizace, když jsou data změněná.

DemographicObject – uchovává informace o tváři načtené ze souboru

DemographicDataFilter – uchovává seznam načtených demografických objektů a také seznam filtrovaných demografických objektů

PCATransformation – uchovává, ukládá, načítá a počítá PCA transformaci

Regression – slouží k výpočtu různých typů regrese

MyTrackBar - je nová grafická implementace klasického TrackBaru. Tato třída je

obsažená v SmartVerticalTrackBar, který rozšiřuje funkcionalitu o zobrazovací pole, popisek a resetovací tlačítko. Od SmartVerticalTrackBar jsou poděděné tři typy TrackBaru s další následující funkcionalitou.

Program je dělaný ve Windows Formech. K vykreslení 3D objektů je použita knihovna Open.TK, a k numerickým výpočtům slouží knihovna Math.NET Numerics a grafy jsou vykreslovány pomocí R.NET.

Data

Data nelze přidat jako přílohu. Jsou to osobní údaje a nebyl udělen souhlas k jejich veřejnému používání. Slouží jen pro akademické účely. Dataset je možné si vyzvednout u vedoucího práce.