Západočeská univerzita v Plzni
Fakulta aplikovaných věd
Katedra informatiky a výpočetní techniky

Bakalářská práce

Vytváření a úprava geometrických modelů pro tvorbu identikitů

Plzeň 2015 Jana Horová

Prohlášení

Prohlašuji, že jsem bakalářskou práci vypracovala samostatně a výhradně s použitím citovaných pramenů.

V Plzni dne 4. května 2015

Jana Horová

Poděkování

Děkuji Prof. Dr. Ing. Ivaně Kolingerové za odborné vedení, za vstřícnost a rady při zpracování této práce. Dále děkuji Ing. Petru Martínkovi za spolupráci na praktické části a v neposlední řadě policejní analytičce kpt. Martině Kubánkové z oddělení Kriminálních analýz Policie České republiky v Ústí nad Labem za otestování programu a věcné připomínky.

Abstract

Creation and Modification of Geometrical Models for Identikits

The purpose of this bachelor thesis is to create auxiliary software and 3D models for the computer program Identikit, which was made as a part of the thesis by Ing. Petr Martínek and is used for making portraits of people in 3D for police use. The work is focused mainly at the creation of a library for editing texture of the face. The library contains basic functions of graphic software and also functions needed for portrait making. The other part of this work is a creation of a few head models and piercings for the program, and a library for placing the piercings on the head models.

Abstrakt

Cílem této práce je vytvořit pomocné programové vybavení a 3D modely pro program Identikit, který vznikl v rámci diplomouvé práce Ing. Petra Martínka a slouží k vytváření portrétů osob ve 3D pro účely policie. Práce se soustředila především na knihovnu umožňující editaci textury obličeje, která obsahuje základní funkce grafických editorů a funkce potřebné přímo pro tvorbu portrétu. Dále je součástí práce vytvoření několika modelů hlav a piercingů pro program Identikit a knihovny pro umisťování piercingů na tento model.

Obsah

1 Úvod **5**

2 Tvorba identikitů **6**
2.1 Morfování 6
2.1.1 Úsečkový warping (neboli warping pomocí magnetického pole) 7
2.1.2 Barycentrický warping 7
2.2 Skládání z částí 8
2.3 Modelování pomocí deformací 8
2.3.1 Šíření transformace (Transformation Propagation) . . . 8
2.3.2 Deformace založená na skořepině (Shell-Based Deformation) 8
2.3.3 Vícestupňová deformace (Multi-Scale deformation) . . 9
2.3.4 Volná deformace (Freeform Deformation) 9
2.4 Program Identikit 10

3 Modely **11**
3.1 Modely hlav 11
3.2 Piercing 15

4 Knihovna pro kreslení do textury **16**
4.1 Zdrojové soubory 16
4.2 Vybrané postupy 18
4.2.1 Vykreslení 3D modelu hlavy 18
4.2.2 Reprezentace vykreslovaných 2D objektů na textuře a jejich transformace 18
4.2.3 Sférická korekce textury 18
4.2.4 Kreslení 19
4.2.5 Generátor pih 20
4.3 Programátorská dokumentace 24
4.3.1 Základní třídy 24

4.3.2 Knihovna pro úpravu masky vlasů a vousů 25

5 Knihovna pro umisťování 3D objektů 26

6 Dosažené výsledky 28
6.1 Kreslení do textury 28
6.1.1 Příklady funkcí 28
6.2 Knihovna pro umisťování objektů 33
6.3 Možné rozšíření 34
6.4 Uživatelská zpětná vazba 36

7 Závěr 38

Přílohy 39
A Uživatelská dokumentace 39
A 1 Kreslení do textury 39
A 2 Příklady použití 40
A 3 Umisťování 3D modelů 42
B Struktura tříd a objektů 43

1 Úvod

Tato bakalářská práce navazuje na diplomouvou práci Ing. Petra Martínka [1]. V rámci jeho práce vznikl program Identikit, který slouží k vytváření portrétů osob ve 3D pro účely policie . Cílem této práce je vytvořit několik součástí tohoto programu. Především knihovnu umožňující kreslení do textury obličeje. Tato knihovna bude obsahovat jak základní funkce grafických editorů, jako je kreslení, gumování a vkládání obrázků, tak funkce potřebné přímo pro tvorbu portrétu. Mezi ně patří například vkládání mateřských znamének a drobných pih, možnost vytvoření kompletního makeupu ručně i pomocí připravených nástrojů, vkládání tetování, změna barvy pleti a podobně. Dále je součástí práce vytvoření několika modelů hlav a piercingů pro program Identikit a knihovny pro umisťování piercingů na tento model.

V první části této práce se budu zabývat přípravou 3D modelů pro program Identikit. V další části pak dokumentací knihovny pro kreslení do textury a poté knihovny pro vkládání 3D objektů do modelu hlavy. Příloha pak obsahuje stručný manuál pro uživatele těchto knihoven.

2 Tvorba identikitů

Protože moje práce navazuje na diplomovou práci Ing. Petra Martínka [1], v následující části stručně shrnu jeho dosavadní práci.

Jeho cílem bylo vytvořit program pro tvorbu identikitů pro využití v policejní praxi. Identikit je portrét osoby určený pro její identifikaci. Protože policie ČR v době zadání práce využívala zastaralý software (PORIDOS), kde portrét vzniká pouze ve 2D spojováním fotografií jednotlivých částí obličeje (jejichž databáze je taktéž zastaralá a neobsahuje některé věkové kategorie obyvatelstva), rozhodl se společně s pracovníkem Policie České republiky v Ústí nad Labem, se kterým práci konzultoval, pro přípravu softwaru pro vytváření portrétu ve 3D.

Vytvořit portrét v programu PORIDOS trvá pracovníkovi policie cca 45 minut, což je pro oběti trestného činu často nepříjemně dlouhá doba. Cílem bylo vytvořit software, ve kterém by se tato doba zkrátila. Ing. Petr Martínek prozkoumal již existující dostupný software pro tvorbu identikitů, avšak žádný z prozkoumaných programů neodpovídal přímo potřebám policie z důvodu nedostatečné databáze či složitosti ovládání.

Ve zbytku kapitoly shrnu způsoby modelování lidské hlavy ve 3D, které byly zvažovány pro realizaci programu (výsledné modely budou reprezentovány jako trojúhelníková síť). Těmito způsoby jsou morfování z již existujících modelů, skládání z částí a modelování pomocí deformací. Tato kapitola je založena na textu diplomové práce Ing. Petra Martínka [1].

2.1 Morfování

Morfování umožňuje získat z již existujících modelů nový model interpolací. Probíhá ve dvou krocích – morfování trojúhelníkové sítě a textury nebo barvy modelu. Základním požadavkem pro morfování je, aby měly všechny spojované modely stejný počet odpovídajících si vrcholů. Pokud máme například dva modely s odpovídajícími vrcholy, můžeme souřadnice vrcholu z prvního a druhého modelu vynásobit koeficientem 0,5 a tyto souřadnice pak sečíst:

$X = A * w_A + B * w_B$,
kde X je nový vrchol, A je vrchol prvního modelu a B druhého, w_A je váha

určující vliv zdrojového modelu A na cílový a w_B je příslušná váha modelu B. Váhy leží v intervalu <0,1>.

Stejným způsobem je možné morfovat libovolný počet modelů.

Pokud je třeba pracovat i s barvou modelu, je situace o něco složitější. V případě, že modely obsahují barvu přímo ve vrcholech, je možné na výslednou barvu použít stejný vzorec jako na výsledný vrchol. Pokud je barva modelu určena texturou, je třeba použít warpovací algoritmus, jako je například úsečkový warping nebo barycentrický warping.

2.1.1 Úsečkový warping (neboli warping pomocí magnetického pole)

Podstatou tohoto algoritmu je modifikace obrazu na základě úseček, které definují svou velikostí a polohou základní změny v obraze. Úsečky můžeme chápat jako magnety a změnou jejich polohy definujeme změnu tvaru pole. Vzájemná korespondence mezi oběma poli definuje příslušnou transformaci. Warping obrazu se provádí s více úsečkami, přičemž jeden pár úseček určuje jednoznačné přiřazení bodu z obrazu A do bodu v obraze B.

2.1.2 Barycentrický warping

Na obrázku, který chceme warpovat, si zvolíme n-úhelníky, které se mohou vnořovat, ale jejich hrany se nesmí protínat. Pro přesný výsledek je vhodné zvolit více vnořených polygonů, protože největší přesnosti dosáhneme na hranách a vrcholech těchto polygonů. Alternativou je zvolení jednoho velmi členitého polygonu. Principem této metody je pak výpočet barycentrických souřadnic v obecném n-úhelníku. Z barycentrických souřadnic jednotlivých pixelů zdrojového obrazu určíme jejich pozici v obrazu výsledném. Pomocí barycentrických souřadnic zjistíme barvu pixelu ve zdrojovém obrázku tak, že se všechny vrcholy polygonu vynásobí příslušnou barycentrickou souřadnicí a sečtou se. Výsledek zaokrouhlíme na celé číslo a získáme pozici pixelu, který se překopíruje do výsledného obrazu.

Morfování více modelů pro vytvoření identikitu má řadu nevýhod. Hlavní nevýhodou je potřeba rozsáhlé databáze modelů, aby pokryla většinu rozmanitosti lidské populace.

2.2 Skládání z částí

Tato metoda odpovídá původní metodě policie, tedy skládání různých částí obličeje do jednoho. Velkou nevýhodou této metody je stejně jako v případě morfování potřeba velké databáze částí obličeje. Tyto části by se následně spojovaly do jednoho modelu. V případě mezer mezi nimi nebo jejich překrývání by bylo třeba přechody dotvořit, čímž by mohly vzniknout nové nedokonalosti.

2.3 Modelování pomocí deformací

Třetí možností, jak identikit vymodelovat, je pomocí deformací. Ty je možné použít na obecný model (koule), nebo na tvar podobný výslednému modelu (lidská hlava). Modelování hlav z obecného modelu je časově náročné, avšak při použití menšího počtu referenčních modelů stačí pouze upravit detaily. Metoda referenčního modelu je silně závislá na kvalitě tohoto modelu.

Metody deformace:
Tyto metody se dělí například na volné (objemové) deformace a povrchové deformace. V obou případech měníme pouze polohu vrcholů a jejich napojení zůstává neměnné. Výhodou je, že každý vrchol lze individuálně ovládat, což umožňuje vysokou přesnost výsledku.

2.3.1 Šíření transformace (Transformation Propagation)

Oblíbená deformační metoda umožňující uživateli zvolit deformovanou a neměnnou oblast. Transformace se šíří v deformovaném regionu a postupně slábne, což vede k hladkému přechodu mezi neměnnou a deformovanou oblastí.

2.3.2 Deformace založená na skořepině (Shell-Based Deformation)

Deformace založená na předpokladu, že povrch se chová jako kůže nebo list, který se ohýbá podle sil, jež na něj působí. Toto chování může být popsáno

matematickou funkcí pro natahování a ohýbání. I zde je možné deformaci omezit určením neměnné oblasti, řídící oblasti a oblasti přechodu (deformované oblasti). V porovnání s metodou šíření transformace je tato metoda výpočetně náročnější, avšak poskytuje intuitivnější výsledné transformace.

2.3.3 Vícestupňová deformace (Multi-Scale deformation)

Tato deformace je založena na principu rozdělení objektu do frekvenčních pásem, kde každé pásmo odpovídá jiné úrovni detailu (čím vyšší frekvence, tím větší úroveň detailu). Tento princip je možné využít tak, že deformujeme vysoké frekvence dle našich požadavků (např. na tvar hlavy) při zachování detailů v pásmu s vyšší frekvencí (např. tvar nosu). Nejprve se provede rozklad na nízké a vysoké frekvence, poté se nízké frekvence deformují a následuje rekonstrukce – přidání vysoké frekvence zpět.

Všechny doposud popsané způsoby deformace patří do kategorie povrchových deformací s vysokou výpočetní složitostí a možností vzniku artefaktů, kvůli kterým je někdy třeba síť opravit, nebo dokonce znova triangulizovat. Těmto problémům se dá vyhnout použitím volné deformace.

2.3.4 Volná deformace (Freeform Deformation)

Volná deformace deformuje objekt i s jeho okolním prostorem a pracuje s funkcí $d : R^3 \rightarrow R^3$, která ovlivní všechny body původního modelu. Zakřivení prostoru kolem objektu vede i k zakřivení samotného objektu.

a) volná deformace založená na mřížce (Lattice-Based) – Objekt je deformován pomocí manipulace s řídícími body, pro hustou mřížku může být deformace velmi pracná. Naopak pro malý počet kontrolních bodů není možné dosáhnout potřebné přesnosti.

b) volná deformace založená na kleci (Cage-Based) – Jde o zobecnění metody založené na mřížce.

Pro program Identikit Ing. Petr Martínek zvolil využití referenčního modelu a metodu volné deformace tvaru, protože vytvoření rozsáhlé databáze by bylo velmi náročné ve srovnání s deformací referenčních modelů, kterých

nemusí být tolik, avšak čím více jich je, tím snažší pro uživatele je, zvolit model podobný tomu, který chce vytvářet. Pro obarvování modelů zvolil metodu mapování textury na model před metodou obarvování vrcholů, která neumožňuje ostré přechody mezi jednotlivými barvami.

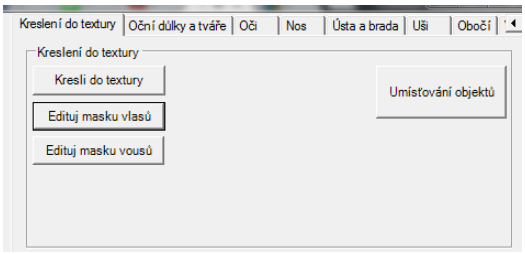
2.4 Program Identikit

Program Identikit , který vznikl v rámci diplomové práce Ing. Petra Martínka funguje na základě volné deformace načteného referenčního modelu hlavy. Obsahuje řadu funkcí. Kromě určování základního tvaru hlavy, nosu, úst, čela a dalších částí obličeje nutných pro identifikaci je v něm možné pomocí textury měnit vzhled očí. Oči se do modelu vkládají samostatně a to z toho důvodu, že přítomnost očí při počátečním popisování pachatele trestného činu odvádí pozornost osoby, která pachatele popisuje, od důležitých detailů. Základní manipulace s částmi obličeje je postavena na principu předdefinovaných maker, avšak tento program umožňuje také ruční transformace konkrétních řídících bodů. Dále program obsahuje možnost vytvoření vlasů a vousů na modelu a práci s obočím.

Mým cílem bylo na tuto práci navázat a vytvořit několik referenčních modelů podle požadavků programu Identikit a tím rozšířit jejich databázi. Dále vytvořit knihovnu pro editaci textury modelu, pro editaci oblastí, ve kterých mohou růst vlasy a vousy, a knihovny pro umisťování drobných 3D detailů na model (například náušnice).

Vytvořené kompletní modely jsou nahrány ve složce s modely programu a je možné je zvolit v nabídce při načítání nového modelu. Nově vytvořené knihovny jsou přístupné z panelu v pravé části okna programu Identikit (obr. 2.1).

Nyní se postupně budu zabývat jednotlivými úkoly.



Obrázek 2.1: Záložka pro spouštění mých knihoven

3 Modely

Zabývala jsem se vytvářením dvou typů modelů pro program Identikit – modelů hlav a modely piercingu.

Použité programové vybavení pro práci s modely bylo následující:
- MeshLab (verze 1.3.3) – open source software pro prohlížení a editaci 3D modelů [6]
- UVMapper (verze 0.30a) – utilita pro vytvoření texturovacích souřadnic modelu [7]
- Blender - software pod licencí GPL, kompletní software pro práci s 3D modely od modelování až po animaci a renderování [8]

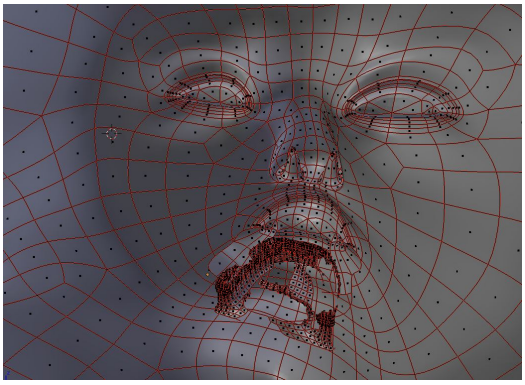
3.1 Modely hlav

Nejprve je třeba vyhledat modely lidských hlav, které se vzájemně liší co nejvíce (věk, rasa, pohlaví...). Důležité je udržet co nejmenší počet vrcholů modelu tak, aby nebyl výsledný program příliš náročný na paměť (zredukovat počet vrcholů pod 30 000 - toto číslo bylo stanoveno na základě experimenů Ing. Petrem Martínkem).

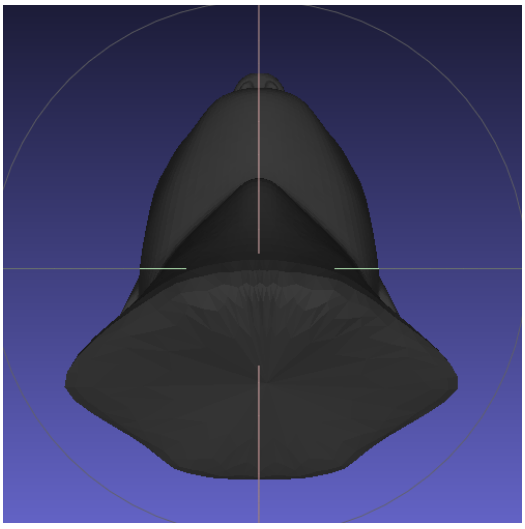
Postup práce byl následující:

1. Získat model hlavy nebo celého těla pod licencí umožňující volné použití.
2. Zbavit model hlavy přebytečných částí – (neviditelné části, zbytek těla, vlasy atd.) (obr. 3.1).
3. Převést plošky, ze kterých se model skládá, na trojúhelníkovou síť.
4. Upravit velikost modelu na interval <0,1> a jeho natočení tak, aby odpovídalo ostatním modelům (obr. 3.2).
5. Zredukovat duplicity a příliš blízké vrcholy.
6. Převést do formátu *.obj, vygenerovat texturovací souřadnice v programu UV Mapper a zároveň si uložit obrázek sítě (obr. 3.3) pro zobrazování při kreslení do textury.

7. Převést zpět do formátu *.ply a uložit včetně texturovacích souřadnic, normál a barvy modelu.



Obrázek 3.1: Vnitřek modelu a zbytky původní čelisti a zubů



Obrázek 3.2: Model v základní poloze

Tímto způsobem vzniklo 5 hlav – 1 černoch a 4 běloši, a to 2 ženy a 3 muži. Tyto modely jsou získány ze zdrojových souborů pod licencí CC-0 (Creative Commons Zero) nebo Creative Commons Attribution 3.0. V případě Creative Commons Attribution 3.0 je nutné uvést autora tohoto modelu. To jsem vyřešila přidáním textového souboru nazev_modelu_LICENSE.txt do složky každého modelu pod touto licencí.

Výsledný model musí odpovídat požadavkům programu, a proto je třeba připravit několik dalších věcí.

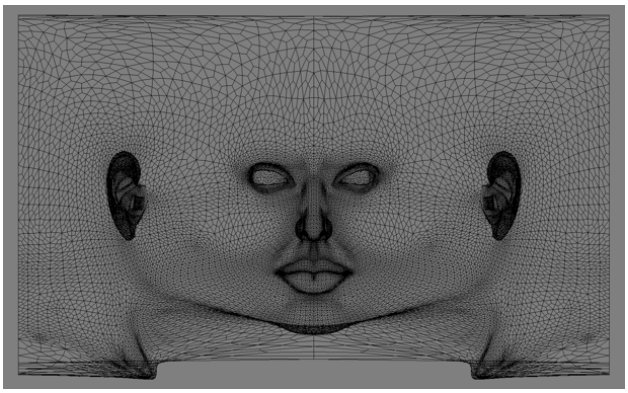
1. Na hotové modely hlav označit místa, kde mohou růst vlasy a kde vousy. To se provádí v programu MeshLab, kde je nutné červeně (RGB 255;0;0) vyznačit místa pro růst vlasů a zeleně (RGB 0;255;0) pro růst vousů (obr. 3.4).

2. Pro tyto modely vytvořit základní texturu a dokreslit ručně na texturu rty. (obr. 3.5)

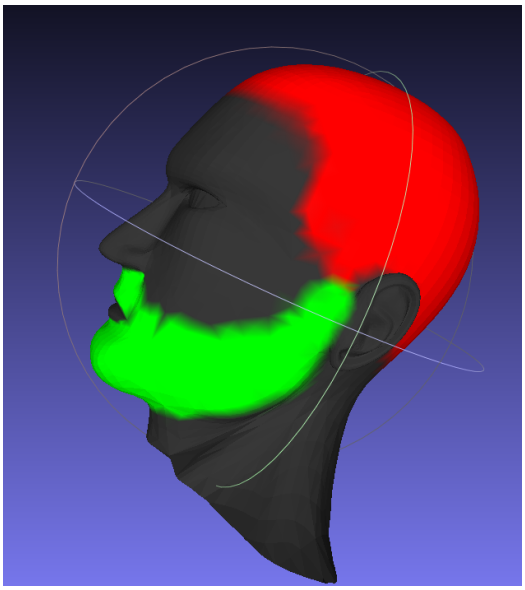
3. Ke každému modelu je třeba přiložit textový soubor stejného názvu jako model, ve kterém jsou vepsány souřadnice význačných bodů. Tyto body, jako například souřadnice koutků očí a úst, slouží poté k transformaci modelů v programu Identikit. K tomuto účelu jsem vytvořila dokument, pro který jsem vyhledala a popsala, kde na hlavě jednotlivé souřadnice leží (na přiloženém CD v podsložce help). Pomocí tohoto souboru lze pak každému modelu přiřadit souřadnice řídících bodů. Takových bodů je na hlavě přibližně 60.

4. Pro každý model vytvořit vlastní složku s různými kombinacemi vlasů a vousů a různými účesy pomocí úpravy červenozelené textury z bodu 1.

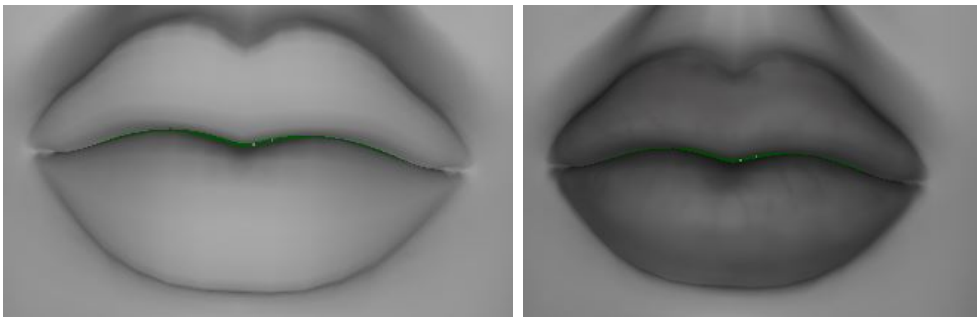
5. Vytvořit složku „zatmeni“, která obsahuje bitmapy s černě označenými oblastmi význačných částí obličeje (nos, ústa oči). Tyto bitmapy obsahují různé kombinace částí obličeje (např. oči + ústa). Tyto bitmapy pak slouží v programu k vykrývání různých částí obličeje, které by mohly odvádět pozornost při popisu pachatele.



Obrázek 3.3: Trojúhelníková síť rozloženého modelu



Obrázek 3.4: Označení míst pro růst vlasů (červená) a vousů (zelená)

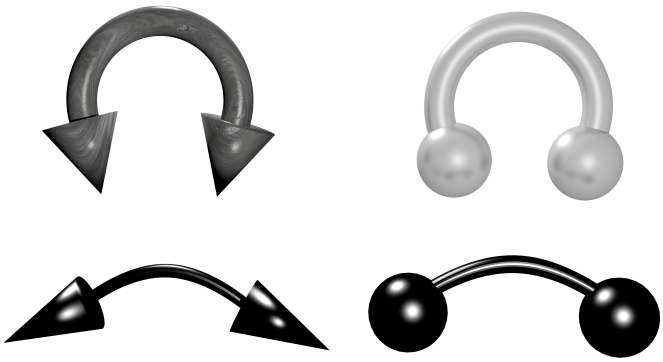


Obrázek 3.5: Rty modelu bez textury (vlevo) a s texturou (vpravo)

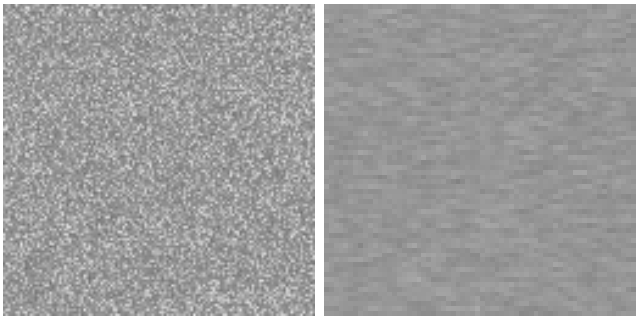
3.2 Piercing

Dalším úkolem bylo vytvořit sadu piercingů, které bude možné použít v programu Identikit. Tyto piercingy je opět třeba upravit podobně jako modely hlav tak, aby neobsahovaly příliš mnoho vrcholů.

Těchto piercingů jsem vymodelovala 7 druhů (příklad vyrenderovaný programem Blender na obr. 3.6) a dále jsem vytvořila dvě jednoduché kovové textury (obr. 3.7), které lze na ně použít.



Obrázek 3.6: Ukázka vyrenderovaných modelů



Obrázek 3.7: Kovové textury vhodné pro piercing

4 Knihovna pro kreslení do textury

Hlavní součástí práce je vytvoření knihovny pro kreslení do textury 3D modelu.

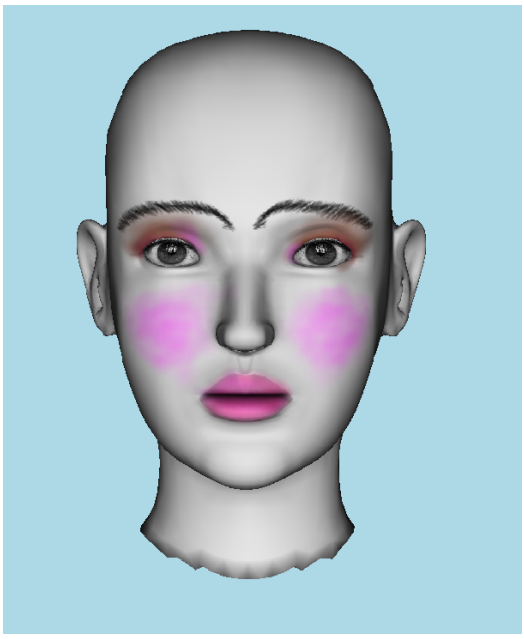
Tato knihovna momentálně umožňuje práci s více než 15 různými nástroji od základních nástrojů, jakými jsou základní geometrické tvary a volné kreslení, po nástroje vhodné přímo pro využití k tvorbě identikitu, jako například automatické oční stíny, vkládání pih a tetování. Po zavření okna knihovny pro kreslení se výsledek odešle na původní model a zobrazí v okně programu Identikit (obr. 4.1).

4.1 Zdrojové soubory

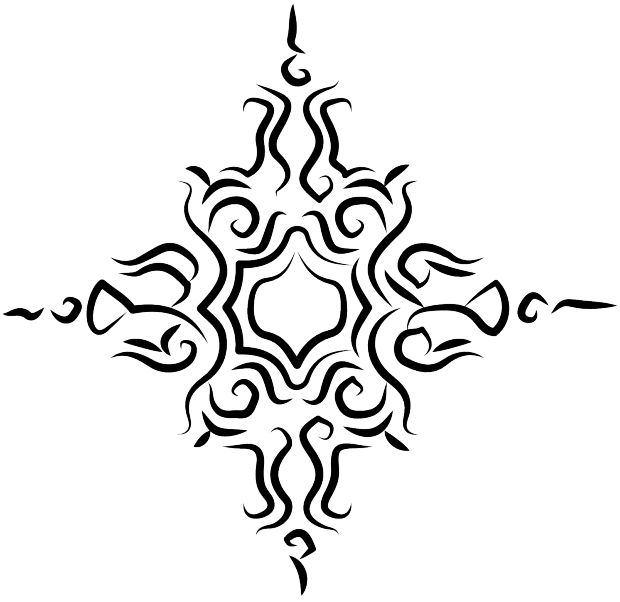
Pro použití v programu bylo třeba získat sadu několika druhů obrázků tetování. Tyto obrázky byly získány z webu [10], kde se dají získat zdarma obrázky v nejnižším rozlišení. Obrázky v malém rozlišení jsem převedla do vektorové podoby a vyrenderovala v lepší kvalitě (obr. 4.2). U některých obrázků došlo při vektorizaci k drobné ztrátě původního tvaru, avšak bez porovnání s originálem je tato ztráta neznatelná. K těmto obrázkům je nutné přiložit údaje o autorovi. Tyto údaje jsou v souboru licence.txt ve složce `tetování` na přiloženém CD.

Tetování jsou nyní dostupná z knihovny kreslení do textury pro umístění na model.

Dále jsem vyfotografovala sadu mateřských znamének a větších pih, která jsou v programu dostupná pro umístění na model. Tato znaménka je možné vkládat na model ze záložky `Znaménka`.



Obrázek 4.1: Ukázka modelu s pokreslenou texturou



Obrázek 4.2: Ukázka vektorizovaného tetování

4.2 Vybrané postupy

V této části shrnu některé problémy, které jsem řešila při vytváření knihovny pro kreslení do textury a postupy jejich řešení.

4.2.1 Vykreslení 3D modelu hlavy

Z hlavního programu jsou knihovně předány instance tříd IndexBuffer a VertexBuffer. Tyto třídy jsou součástí knihovny SlimDX [9], která je v programu použita pro práci s 3D grafikou. Tyto třídy se používají pro uchování pozic vrcholů modelu (VertexBuffer) a pro určení trojúhelníků (IndexBuffer). Těchto instancí je využito pro vykreslení modelu.

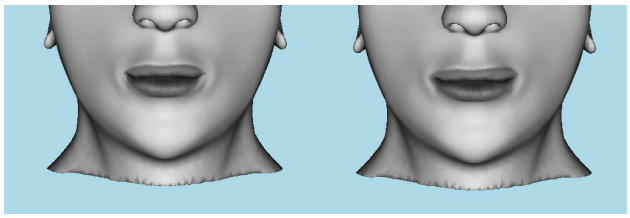
4.2.2 Reprezentace vykreslovaných 2D objektů na textuře a jejich transformace

Všechny objekty vložené na texturu (např. čára, obdélník, obrázek vložený ze souboru...) obsahují vypočítané ohraničení (bounds) určující obdélníkovou obálku objektu. V některých případech je využito jeho vykreslení k snadnější orientaci uživatele (např. při přesunu objektů se zobrazují pouze ohraničení těchto objektů, což urychluje vykreslování v průběhu přesunu). Rotace a velikost objektu je pak závislá na velikosti a rotaci tohoto ohraničení. Je tedy vždy přepočítán posun nebo rotace bodů ohraničení. U jednoduchých tvarů (obdélník, rovná čára) stačí k jejich reprezentaci ohraničení. U ostatních objektů je třeba uchovávat objekt jinak. Například volně kreslená čára je uchována jako seznam bodů. U některých objektů je vykreslovaných bodů příliš mnoho, což by neúměrně zpomalovalo program. Proto objekty, které implementují rozhraní ICacheable, například nástroj sprej, jsou po prvním vykreslení dále reprezentovány jako bitmapy, u kterých jsou transformace rychlejší než při přepočítávání polohy příliš velkého množství bodů.

4.2.3 Sférická korekce textury

Protože při mapování na model hlavy docházelo ke zkreslení v blízkosti „pólů“ modelu, rozhodli jsme se využít sférické korekce textury. Pro vytvoření metody,

která texturu deformuje, jsem využila pseudokódu z webu [4]. Výsledná deformace sice částečně zkreslení odstranila, ovšem vzhledem k tomu, že hlava není přesná koule, další zkreslení vznikalo a bohužel toto zkreslení způsobovalo změnu výrazu obličeje, což je pro tvorbu identikitů zásadní problém (obr.4.3). Rozhodli jsme se tutu možnost uživateli zpřístupnit, avšak ve výchozím nastavení je vypnutá a je možné ji zapnout ze záložky `Další`.



Obrázek 4.3: Porovnání nezdeformované textury (vlevo) a zdeformované (vpravo)

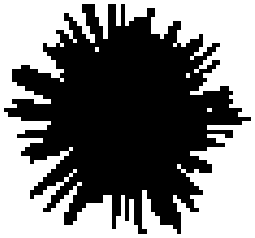
4.2.4 Kreslení

Postup pro ruční kreslení se liší podle použité stopy. V programu jsou použity dva různé postupy.

Základní stopy. Tyto stopy jsou v programu tři a jsou v nabídce pojmenovány jako kulatá, hranatá a špičatá. Využívají základní metody třídy Graphics z knihovny System.Drawing. Metodou DrawLine jsou propojeny jednotlivé body, které vrací událost MouseMove. Využití této knihovny umožňuje u hranatého profilu čáry zaoblené napojení při změně směru tahu. Nevýhodou této metody je, že je nutné při překreslování vždy vykreslovat celou čáru, což zpomaluje reakce programu při kreslení delších čar narozdíl od využití obrázkové stopy, kde není třeba v průběhu kreslení překreslovat celé plátno.

Obrázkové stopy. Obrázkové stopy fungují na principu zdrojového obrázku vkládaného podél tahu myši. Pro tyto stopy byly vytvořeny bitmapy obsahující pouze černou barvu a průhlednost (obr.4.4). To umožňuje přebarvit černé pixely na jakoukoli barvu při zachování tvaru stopy díky průhlednosti. Tyto bitmapy jsou načteny a vkládají se do jednotlivých bodů kreslené čáry. Protože událost MouseMove produkuje souřadnice bodů v nestejných vzdálenostech (v závislosti na rychlosti tažení myši), jsou v určitých intervalech dopočítány body mezi jednotlivými úseky.

Pokud na konci intervalu zbyde nějaký úsek menší než daný interval, je zbytek tohoto úseku připočítán na další interval, aby byly rozestupy stále stejné. Velkou výhodou této metody je možnost vytvořit nespočet různých druhů stop v jakémkoli grafickém editoru a ty pak v programu zpřístupnit. Tato metoda je však nevhodná pro základní ostře ohraničené stopy, protože rozestupy jednotlivých vzorků stopy by musely být velmi malé, aby nebyly při vyšší průhlednosti rozpoznatelné jednotlivé vzorky stopy, což by se projevilo na rychlosti. Při takto malých rozestupech by se vzorky navrstvily tolik, že bychom přicházeli o velkou část průhlednosti.



Obrázek 4.4: Ukázka bitmapy určující tvar stopy

4.2.5 Generátor pih

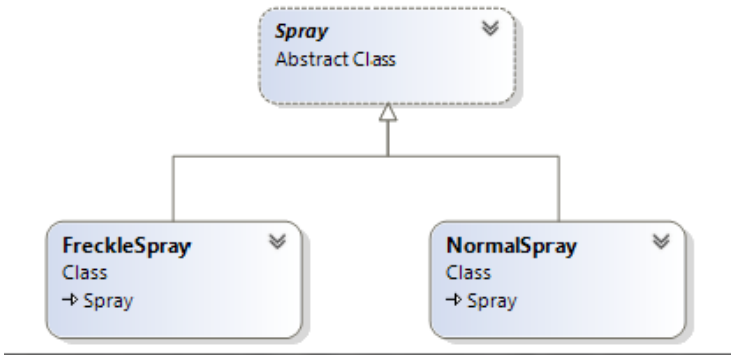
Funkce generování pih zaplňuje danou oblast pihami automaticky, čímž výrazně urychluje vytváření textury obličeje s velkým množstvím malých pih. Pozice pih se generuje náhodně a jejich průhlednost je ovlivněna vzdáleností od středu zvolené oblasti -– čím větší vzdálenost od středu, tím průhlednější. Uživatel může zvolit oblast pro pihy kreslením volné oblasti. V tomto případě získáme nejprve obdélníkovou obálku tohoto polygonu (nejkrajnější body). V této obálce generujeme pihy a vždy kontrolujeme, jestli leží přímo ve zvoleném polygonu pomocí algoritmu Ray casting [2]. Generátor pih se nachází ve třídě TFreckleGenerator.

Sprej

Pokud uživatel drží stisknuté tlačítko myši a nehýbe kurzorem, je po určitých intervalech časovače volána událost MouseMove, čímž dojde ke zhuštění vyplnění oblasti. Vzhledem k tomu, že při kreslení čáry sprejem (mění se jeho ohraničení) a při přesouvání celého spreje by bylo nutné procházet všechny

body (nebo pihy), je využito zjednodušení, kdy nenahlížíme na jednotlivé vybarvené body, ale pouze na pozice jednotlivých koleček, která body obsahují. Absolutní pozice každého bodu se spočítá součtem levého horního okraje ohraničení, pozice konkrétního kolečka a relativní pozice bodu vůči kolečku.

Tento nástroj pracuje se souborem OSpray.cs. V tomto souboru je třída OSpray, která ukládá seznam instancí třídy Spray (jednotlivá kolečka spreje). Abstraktní třída Spray obsahuje lokaci konkrétního kolečka a metodu pro vykreslení. Od této třídy dědí třídy NormalSpray (klasický sprej generující náhodně jednobarevné body) a FreckleSpray (zde se nastavení bodů řídí nastavením v generátoru pih)(obr.4.5). Celkové vazby jsou zobrazeny na obrázku (příloha B3) (diagram závislostí).



Obrázek 4.5: třída Spray

Obarvování pleti a rtů

Jediným problémem při obarvování pleti je, že na textuře, kterou chceme přebarvit, už jsou umístěny rty a obočí. Tento problém byl vyřešen tak, že je nejprve zaznamenána v rohu textury původní barva. Poté porovnáváme barvu každého pixelu s barvou z rohu původní textury, a pokud je rozdíl těchto barev větší než určitý práh, jedná se o rty nebo obočí. V tomto případě se obrázek přebarví v režimu násobení barev. Jinak se barva pouze přepíše.

Při obarvování rtů postupujeme obdobně, pouze s tím rozdílem, že procházíme texturu odspodu a od prvního rozdílu v barvě pokračujeme jen k poslednímu řádku, který se liší od barvy pozadí. Obarvení rtů je v samostatné vrstvě a je reprezentováno jako obrázek (OImage). Uživatel si může zvolit režim přebar-

vování. Pokud je režim přebarvování zapnutý, je oblast rtů pouze obarvena danou barvou bez ohledu na podklad, jinak je použit režim násobení, aby se i u tmavších barev zachovala textura rtů.

Automatické oční stíny

Pro rychlejší práci s programem jsem vytvořila funkci automatických očních stínů, která umožňuje uživateli zvolit dvě barvy (světlejší a tmavší) a jeden ze tří připravených tvarů. Výsledkem jsou oční stíny připravené pro umístění na texturu.

Automatické oční stíny fungují na principu ručně předkreslené dvoubarevné bitmapy (zdroje) (obr. 4.6). Bitmapa je vytvořena ve dvou barvách - červené a zelené. Zelená barva je pak nahrazena tmavší barvou zvolenou uživatelem a červená světlejší.

Výsledná barva je pak tvořena podle následujícího vzorce:

(červená_složka_zdroje/255)*světlejší + (zelená_složka_zdroje/255)*tmavší

```
1 //ncolor = nova barva konkretniho pixelu
  //ocolor = puvodni barva konkretniho pixelu
3 //lighter, darker - barvy na ktere se obarvuje
  ncolor = Color.FromArgb(ocolor.A,
5     (int)Math.Min(255, (ocolor.R / 255.0)
          * lighter.R + (ocolor.G / 255.0) * darker.R),
7     (int)Math.Min(255, (ocolor.R / 255.0)
          * lighter.G + (ocolor.G / 255.0) * darker.G),
9     (int)Math.Min(255, (ocolor.R / 255.0)
          * lighter.B + (ocolor.G / 255.0) * darker.B));
```

Zdrojový kód 4.1: Obarvení očních stínů

Výhodou tohoto postupu je možnost ručního vytvoření velkého počtu dvoubarevných „šablon“ pro různé tvary očních stínů.

Obrázek 4.6: Zdrojový obrázek reprezentující konkrétní tvar očních stínů

Historie akcí a kroky zpět

Program obsahuje dva seznamy provedených akcí - lokální a globální. Globální (globalActionLog) ukládá všechny akce, které byly pro tuto instanci provedeny, lokální (actionLog) ukládá pouze ty, které byly provedeny od posledního otevření okna Kreslení do textury. Pokud uživatel zavře okno bez uložení, celý lokální zásobník se smaže, přepíše globálním a provedou se kroky vpřed až do stavu při posledním uložení. Pokud zvolí uložení, globální zásobník akcí se smaže a nahradí lokálním.

V instanci třídy Painter je uložen lokální a globální seznam akcí. Akce jsou reprezentovány abstraktní třídou Action, od které jednotlivé akce (například přesun, vložení, rotace...) dědí. Každá akce má definován krok vpřed a zpět.

Textura pleti

Protože modely s jednobarevnou pletí působily příliš uměle, obsahuje knihovna dvě funkce pro lepší texturu pleti. Funkce je možné povolit při změně barvy pleti. První z nich dodává pleti jemnou texturu připomínající póry (obr.6.7). Tato textura vzniká pomocí náhodných malých změn odstínu pro jednotlivé pixely. Funkce označená v programu jako `skvrny na pleti` vytváří na pleti větší skvrny s vysokou průhledností (obr. 6.8), zde je využito již připraveného generátoru pih se specifickým nastavením (větší flíčky s vysokou průhledností).

4.3 Programátorská dokumentace

Knihovna byla naprogramována v jazyce C# pomocí „balíku“ Microsoft Visual Studio Ultimate 2012 + Visual C# 2012. Knihovna se skládá přibližně z 50 tříd, jejichž strukturu znázorňuje diagram tříd (příloha B4).

4.3.1 Základní třídy

KresleniDoTex: Třída, která spojuje hlavní program Identikit s knihovnou. Zde probíhá komunikace s hlavním programem.

Painter: Vykresluje na prázdnou texturu všechny objekty. Obsahuje seznam provedených akcí pro kroky zpět a vpřed. Metody pro nastavení pořadí objektů, výběr, odstranění, správu akcí atd. Painter je vytvořen podle návrhového vzoru Singleton.

Důležitou metodou je statická metoda MakePositiveRectangle, která převádí obdélník se zápornou výškou nebo šířkou na obdélník s kladnými parametry.

Zoom: Uchovává nastavení zoomu – pozici a faktor přiblížení. Obsahuje metody pro nastavení faktoru a polohy.

ToolManager: Jedináček (nebo také Singleton - tento návrhový vzor se používá, pokud chceme v programu pouze jednu instanci dané třídy), který obsahuje instance všech nástrojů a umožňuje jejich přepínání. Spojka mezi událostmi (myš) a aktuálním nástrojem. Získává nově vytvořený objekt aktuálního nástroje (metoda getObject()).

ToolSettings: Jedináček, který uchovává všechna aktuální nastavení nástrojů.

PenBrushSettings a FreckleSettings: Uložené parametry jednotlivých objektů (např. síla čáry). Mohou se lišit od aktuálního nastavení.

Model3D: Uchovává instanci a parametry 3D objektu.

RotRectangle: Třída umožňující pracovat s obdélníkem, který může být natočen. Je reprezentován čtyřmi body a úhlem.

Tool: Abstraktní třída, od které dědí všechny nástroje (Tužka, Guma...). Obsahuje všechny metody, které je nutné u každého nástroje implementovat.

Objekt: Abstraktní třída, od které dědí všechny objekty (Čára, Obrázek...).

ICacheable: Rozhraní zaručující, že ICacheable objekt je také reprezentován jako bitmapa kvůli rychlejšímu vykreslování. Dědí od ISavable.

ISavable: V případě nepoužívání objektu se ISavable objekt (jeho bitmapa) může uložit na disk.

Action: Abstraktní třída, od které dědí všechny akce (události), které jsou zaznamenány pro pozdější pohyb v historii.

ImageProcessor: Má za úkol převádění obrázků do černobílé podoby. Pro převod byl použit algoritmus ze zdroje [3]. Již přebarvené obrázky si uchovává ve slovníku pro urychlení pozdějšího použití téhož obrázku.

4.3.2 Knihovna pro úpravu masky vlasů a vousů

Jedním z požadavků na tuto práci bylo vytvořit knihovnu, která by umožňovala editovat texturu pro rozmístění vlasů a vousů na model (obr. 3.4). Tato knihovna byla vytvořena z kopie základního kreslení do textury s tím rozdílem, že lze kreslit pouze červenou, zelenou a základní šedou barvou a jsou povoleny jen vybrané nástroje a funkce.

Při volbě barvy stopy je namísto volby průhlednosti volba intenzity. Ta má vliv na červenou a zelenou barvu a pohybuje se v rozmezí 0-255, přičemž tato hodnota odpovídá hodnotě odpovídající složky výsledné barvy. Například pro kreslení červenou s intenzitou 50% je výsledná barva RGB(128,0,0), pro zelenou (0,128,0). Při nastavení intenzity na 0 je výsledná barva černá, tudíž nemá na růst vlasů a vousů žádný vliv.

Informace, jestli je pro danou instanci požadován tento režim, je předávána z hlavního programu pomocí proměnné `bool texturaKuze`, která je false pro mód úpravy vlasů a vousů.

5 Knihovna pro umisťování 3D objektů

Poslední součástí práce bylo vytvoření knihovny pro umisťování menších 3D modelů na model hlavy. Tato knihovna zatím slouží pro umisťování modelů piercingu, které jsem v rámci práce vytvořila, avšak je možné ji využít pro načítání a umisťování jakýchkoli modelů pro dotvoření identikitu (např. další piercingy a drobné šperky).

Pro celý program Identikit zvolil Ing. Petr Martínek pro ukládání 3D modelů formát PLY. Tento formát byl vyvinut na Stanford University a jeho struktura je volně přístupná (popis např. článek [5]). Pro načtení souboru v tomto formátu a pro vytvoření této knihovny mi byly poskytnuty od Ing. Petra Martínka třídy (PlyLoader.cs, Vertex.cs, ColorVertex.cs) a metody pro převod souřadnic pozice kurzoru myši na 3D souřadnice na modelu.

Po tom, co je uživatelem zvolen soubor ve formátu *.ply, je tento soubor načten třídou PlyLoader.cs a načtená data se uloží do třídy Model3D. Dále knihovna obsahuje třídu Objekt, která uchovává samotný model a zároveň parametry tohoto modelu, jako jsou pozice, rotace a zvětšení.

Tento objekt je přidán do seznamu všech objektů, které jsou do scény načteny. Aktuálně vybraný model je obarven na červeno pro snadnější orientaci uživatele.

Při umisťování objektu na model je využita metoda pro převod souřadnic kurzoru na souřadnice modelu, která mi byla poskytnuta. Pokud bude uživatel měnit polohu již umístěného objektu, přesune se kurzor myši na pozici objektu a poté vše funguje stejně jako při prvním umístění. Pokud by model byl při přemisťování objektu na odvrácené straně modelu hlavy, je kvůli algoritmu převodu souřadnic myši na 3D souřadnice (hledá nejbližší průsečík od pozorovatele), přesunut na přivrácenou stranu.

Tato knihovna dále umožňuje posun vloženého modelu v jednotlivých osách. Tento problém jsem vyřešila pomocí promítnutí rovnoběžky se zvolenou osou procházející modelem do souřadnic obrazovky. Tím vznikne přímka ve 2D, na kterou lze promítat pohyb myši. Ve chvíli, kdy uživatel zvolí osu, podle které bude objekt posouvat, jsou zvoleny dva body na přímce procházející aktuální pozicí objektu a rovnoběžné se zvolenou osou. Souřadice objektu a dvou bodů

na přímce jsou promítnuty do souřadnic obrazovky a je možné vytvořit vektor určující směr vzniklé přímky. Vzdálenost těchto dvou promítnutých bodů na obrazovce je pak využita jako parametr (d) pro korekci výsledného posunu (projeví se zde zkrácení délky natočené úsečky při promítání z 3D do 2D, které způsobuje příliš rychlý posun modelu).

Uživatel tažením myší určuje pohyb po ose, tím jsou získány souřadnice původní a výsledné pozice kurzoru myši, které jsou přes kolmé promítnutí na přímku v souřadnicích obrazovky převedeny na dva body této přímky. Z těchto dvou bodů je získán parametr rovnice přímky procházející aktualní pozicí kurzoru vůči původní a je provedena korekce výsledku podle parametru d. Vysledný posun je přičten k té 3D souřadnici objektu, která odpovídá uživatelem zvolené ose. Promítnutím pozice posunutého objektu na obrazovku a porovnáním s původní pozicí je získáno znaménko posunu a v případě potřeby je směr posunu otočen.

6 Dosažené výsledky

V této kapitole shrnu výsledky práce a uvedu několik obrázkových příkladů. Všechny vytvořené soubory a samotný program jsou dostupné na přiloženém CD.

V rámci práce jsem získala a upravila 6 modelů hlav podle požadavků programu. Jeden z nich musel být bohužel vyřazen z toho důvodu, že měl příliš výrazné oční bulvy, které by překrývaly oči. Oči jsou v programu Identikit vkládány samostatně, bulvy se mi nepodařilo odstranit tak, aby vznikly dobré řídící body pro vkládání očí programem Identikit.

Dále jsem v programu Blender vymodelovala 7 jednoduchých modelů piercingu, které je možné umisťovat na model.

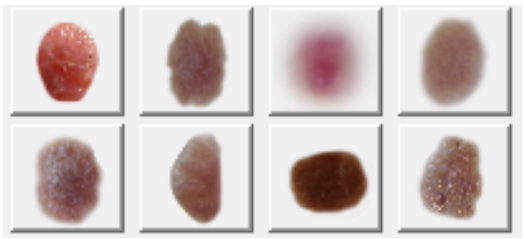
6.1 Kreslení do textury

Hlavní součástí práce bylo vytvoření knihovny pro kreslení do textury obličeje a do textury určující růst vlasů a vousů.

Knihovna pro kreslení do textury nyní umožňuje kromě základních funkcí, jako jsou základní geometrické tvary, volné kreslení, přesun a rotace také funkce vhodné konkrétně pro tvorbu identikitů. Dále uvedu několik příkladů těchto funkcí.

6.1.1 Příklady funkcí

Znaménka a tetování Knihovna obsahuje sadu mateřských znamének a pih, vytvořenou z vlastních fotografií, které je možné na modely vkládat, a sadu tetování vektorizovaných z obrázků získaných v malém rozlišení (obr. 6.1 a 6.2).



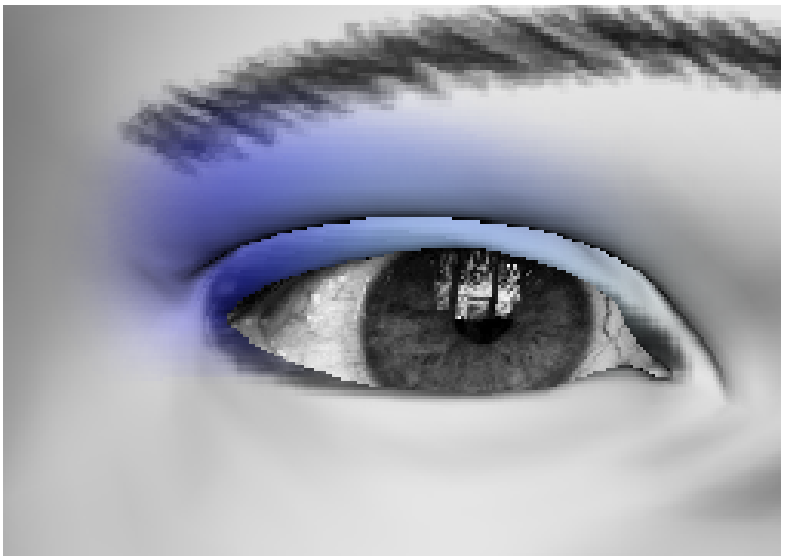
Obrázek 6.1: Sada znamének dostupných přímo z programu

Obrázek 6.2: Ukázka sady tetování

Oční stíny a rtěnka Pro účely tvorby identikitu jsem vytvořila funkce pro

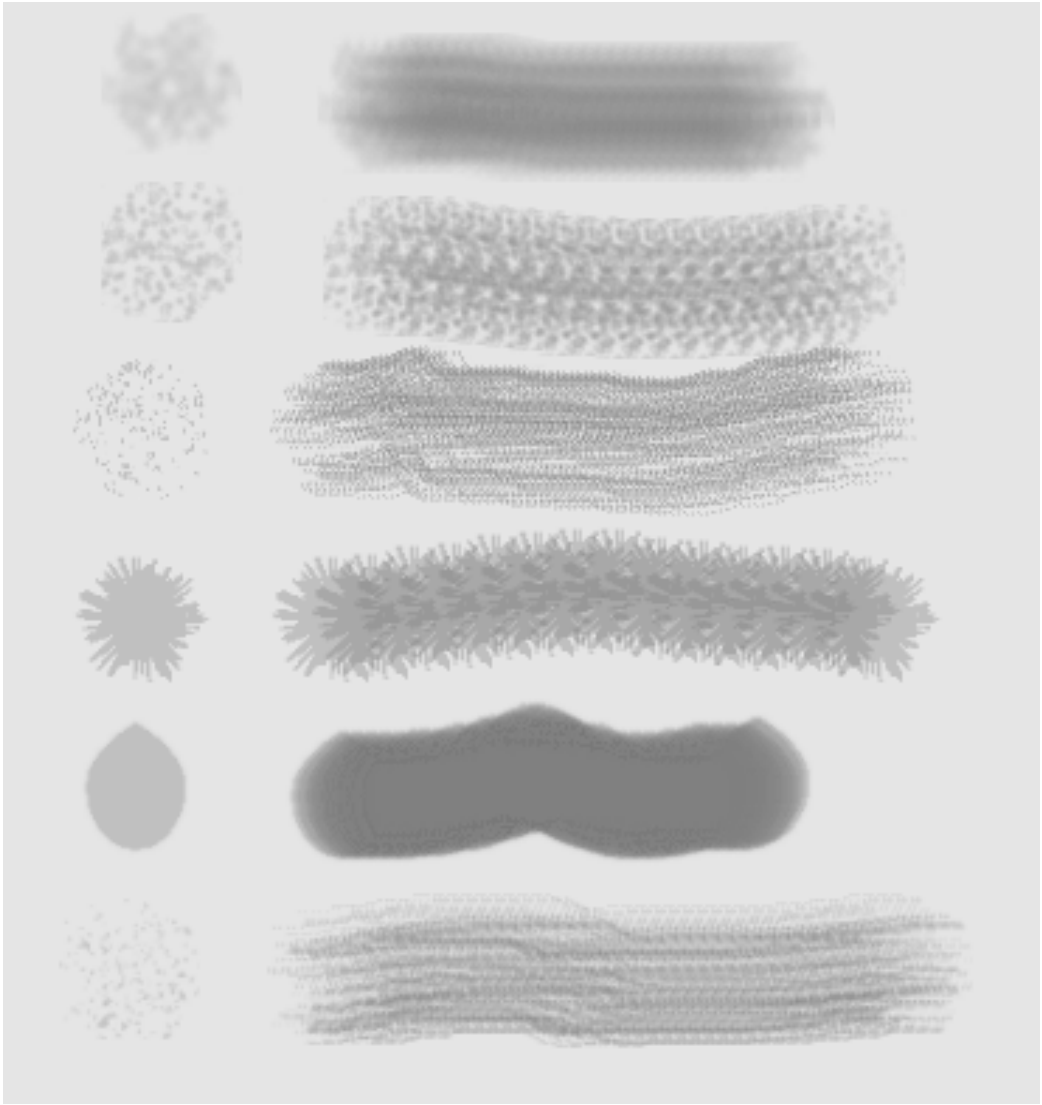
automatické generování očních stínů a obarvování rtů (obr. 6.3 a 6.4). Funkce automatického obarvení rtů občas přetáhne skutečný obrys rtů, což je dáno odlišnou kvalitou ručně malovaných textur. Tento problém je možné vyřešit ručním vymazáním přebytečné části.

Stopy Nástroj volné kreslení obsahuje kromě základních stop také obrázkové stopy, které jsem navrhla (obr. 6.5).



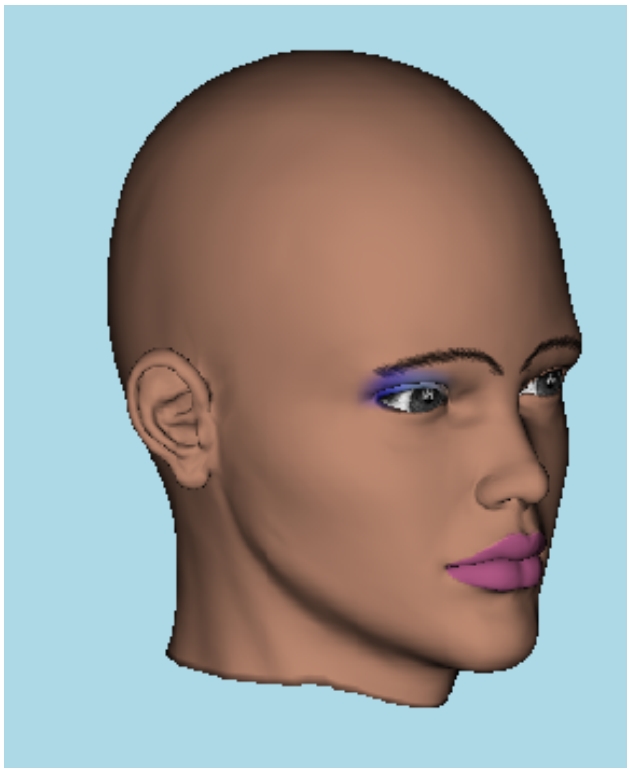
Obrázek 6.3: Oční stíny na modelu

Obrázek 6.4: Automaticky obarvené rty



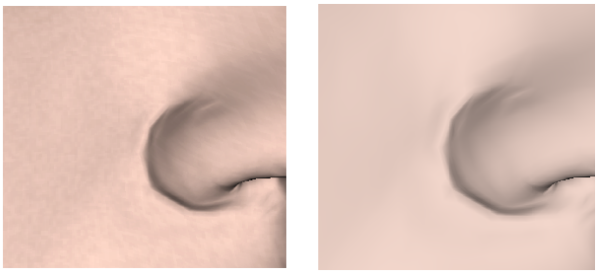
Obrázek 6.5: Obrázkové stopy

Barva pleti Jednou z důležitých funkcí pro tvorbu identikitů je možnost změny barvy pleti bez destrukce původní textury obsahující rty a objektů na texuře umístěných (obr.6.6).

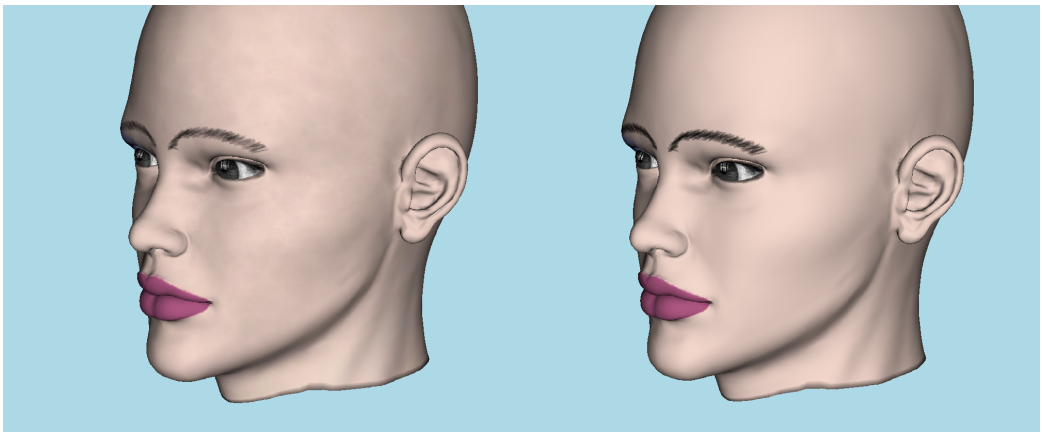


Obrázek 6.6: Přebarvená pleť

Textura pleti Protože modely s jednobarevnou pletí působily příliš uměle, vytvořila jsem dvě funkce pro lepší texturu pleti. Funkce je možné povolit při změně barvy pleti. První z funkcí dodává pleti jemnou texturu připomínající póry (obr.6.7). Funkce označená v programu jako `skvrny na pleti` vytváří na pleti hrubší skvrny (obr. 6.8)



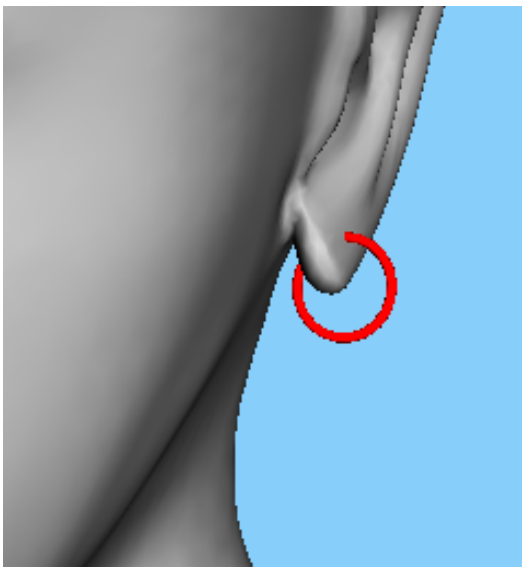
Obrázek 6.7: Model s jemnou texturou pleti (vlevo), původní model (vpravo)



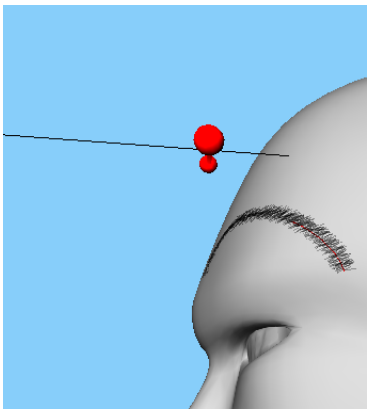
Obrázek 6.8: Pleť se skvrnami (vlevo) a původní model (vpravo)

6.2 Knihovna pro umisťování objektů

Knihovna umožňuje uživateli načíst ze souboru 3D model a ten umístit na model hlavy (obr. 6.9). Těchto vložených modelů může být více a jejich seznam je zobrazen v okně programu. Právě zvolený model je obarven pro lepší orientaci uživatele. Modely je možné rotovat a přesouvat. Knihovna také umožňuje pohyb pouze ve zvolené ose (obr. 6.10) .



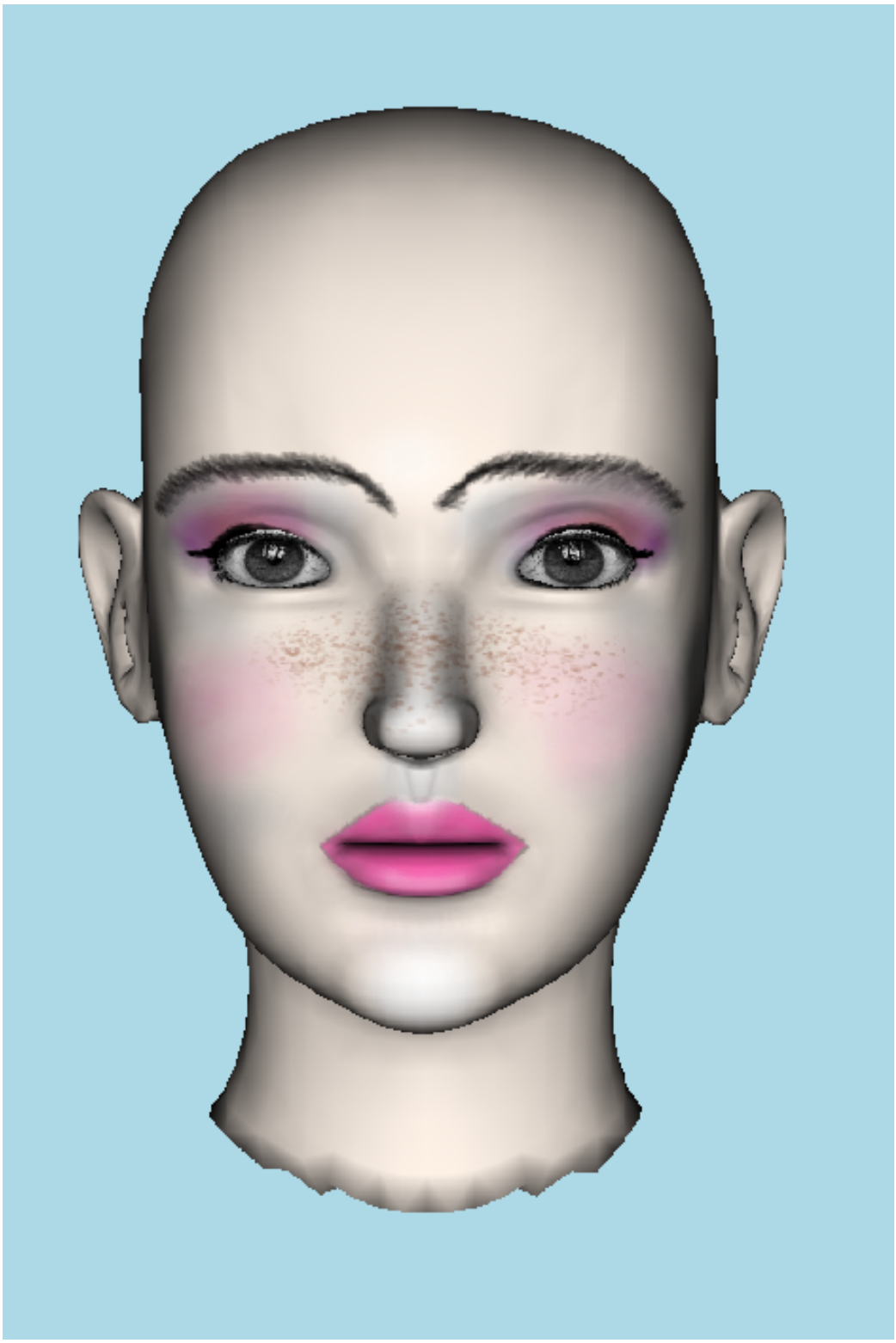
Obrázek 6.9: Model s vloženým piercingem



Obrázek 6.10: Pohyb v jedné z os

6.3 Možné rozšíření

Možnost rozšíření práce vidím například v přidání většího počtu šablon pro oční stíny a stopy štětce, více druhů tetování, modřiny atd. Dále by bylo možné zpřístupnit uživateli práci s vrstvami. Samotná knihovna už s vrstvami pracuje, avšak vrstvy nejsou zahrnuty v grafickém rozhraní. Malý problém vidím v tom, že tím, že se modelům v hlavním programu vkládají oči odděleně od textury a využívají vlastní textury, je zde občas nepřirozený přechod mezi očním makeupem a samotnýma očima, což je vidět i na obrázku 6.11. Knihovna kreslení do textury by se také bez větších úprav dala využít pro upravování textur očí.



Obrázek 6.11: Ukázka makeupu vytvořeného v knihovně pro kreslení do textury

6.4 Uživatelská zpětná vazba

Ing. Petr Martínek spolupracoval v rámci své diplomové práce s pracovištěm oddělení Kriminálních analýz Policie České republiky (dále již PČR) v Ústí nad Labem. Na tomto pracovišti nechal otestovat mou práci policejní analytičkou.

Nyní tedy shrnu problémy, na které paní analytička narazila a doplním, jak byly tyto problémy vyřešeny.

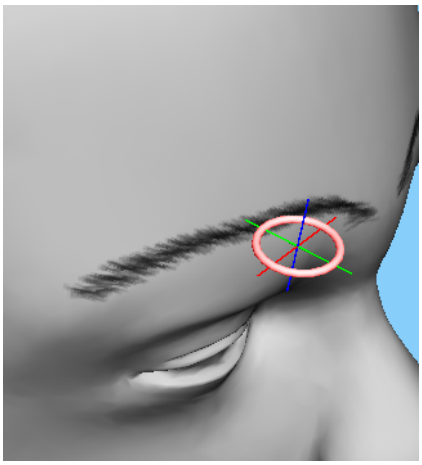
1. Prvním a zásadním problémem bylo, že během testování došlo k několika selháním programu. To je u programu pro vytváření portrétu z popisu svědka nepřípustné, protože není možné vytváření portrétu po náhlém ukončení programu od začátku opakovat.

   Tato závada byla kromě ošetření několika chyb v programu vyřešena tak, že kopie každého vloženého objektu je ukládána do souboru. Pokud je program regulérně ukončen, tyto soubory jsou smazány. Pokud ne, je při příštím spuštění knihovny pro kreslení do textury uživateli oznámeno, že existují záložní soubory a je možné je načíst. Je ovšem nutné mít v hlavním programu načten správný model, ke kterému textura patří, pokud uživatel omylem zvolil jiný, je možné si texturu načtenou ze zálohy uložit do souboru a načíst pro správný model. Do programu byla z toho důvodu k funkci uložení textury přidána i možnost jejího načtení.

2. Další věcí, která se projevila jako problémová, byl výběr několika vložených objektů pomocí nástroje `Výběr`. Nástroj `Výběr` vybere všechny objekty, u kterých byla při tažení myší zasažena jejich obdélníková obálka. To není úplně ideální u objektů, které zabírají malou část plochy této obálky (například šikmá čára z jednoho rohu obálky do druhého). V některých případech to způsobovalo výběr příliš mnoha objektů.

   Jako jednoduché a rychlé řešení tohoto problému byla zvolena možnost zpřísnění podmínek výběru a byly tak vytvořeny tři volby pro tento nástroj. Kromě původní metody, kdy je objekt vybrán při jakémkoli zasažení obálky, jsou doplněny možnosti výběru objektu pouze pokud je obsažen střed obálky, nebo že je obálka ve výběrovém obdélníku obsažena kompletně. Pro usnadnění výběrů bylo také přidáno zaškrtávací políčko umožňující kdykoli zobrazit obdélníkové obálky objektů vložených na texturu. Tato funkce je také užitečná k tomu, aby uživatel mohl smazat objekty, které sice na plátno vložil, ale jsou příliš malé nebo průhledné, než aby je později viděl.

3. Dále byly navrženy menší změny v knihovně pro umisťování 3D objektů na model. První z nich je možnost změny barvy piercingu. Dále označení jednotlivých os tak, aby bylo zřejmé, kam se při posunu táhel pro konkrétní osu bude model v 3D prostoru pohybovat. Pro základní modely piercingů, které vznikly v rámci této práce, bylo navrženo, aby byly přístupné přímo z uživatelského rozhraní (namísto načítání pomocí ručního vyhledání souboru v počítači). Všechny tyto funkce byly do knihovny následně zahrnuty. Ukázka zobrazení os ve 3D prostoru je na obrázku 6.12.



Obrázek 6.12: Zobrazení os v 3D prostoru

4. Při testování byla také navržena funkce přidávání modřin do textury. Pro vytvoření této funkce by bylo třeba získat pod volnou licencí dostatečný počet kvalitních obrázků modřin, případně vytvořit algoritmus, který by modřiny přímo generoval. Vzhledem k velkému množství typů modřin a nedostatečnému počtu dostupného obrazového materiálu pod volnou licencí by tato funkce vyžadovala hlubší studium problematiky a pravděpodobně také vytvoření vlastního obrazového materiálu nebo získání licence k již existujícímu. Pod volnou licencí se mi podařilo získat obrázek monoklu, který jsem upravila tak, aby ho bylo možné vkládat do textury, a zpřístupnila jeho vkládání z programu.

I přes tyto problémy program paní analytičku zaujal, a to hlavně realističtějším výsledkem ve srovnání s programem PORIDOS, který i nadále používají, a počtem možností práce s modelem i texturou.

7 Závěr

Naučila jsem se připravovat modely pro program Identikit a několik jich připravila. Dále jsem vytvořila poměrně složitý grafický editor, ale také jsem zjistila, že takto složitý program by vyžadoval velmi výraznou optimalizaci, aby se mohl rovnat komerčním editorům. Nejobtížnější na této práci bylo nastudovat již rozpracovaný program a při tvorbě modelů se mu přizpůsobit, zatímco v něm ještě probíhaly změny. Přestože jsem vytvářela svou vlastní knihovnu, bylo také třeba relativně často změny rozebrat s Ing. Martínkem a pro synchronizaci kódu využít verzovací systém.

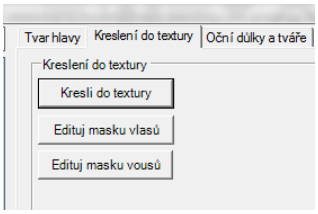
Přílohy

A Uživatelská dokumentace

Požadavky pro spuštění odpovídají hlavnímu programu Identikit [1].

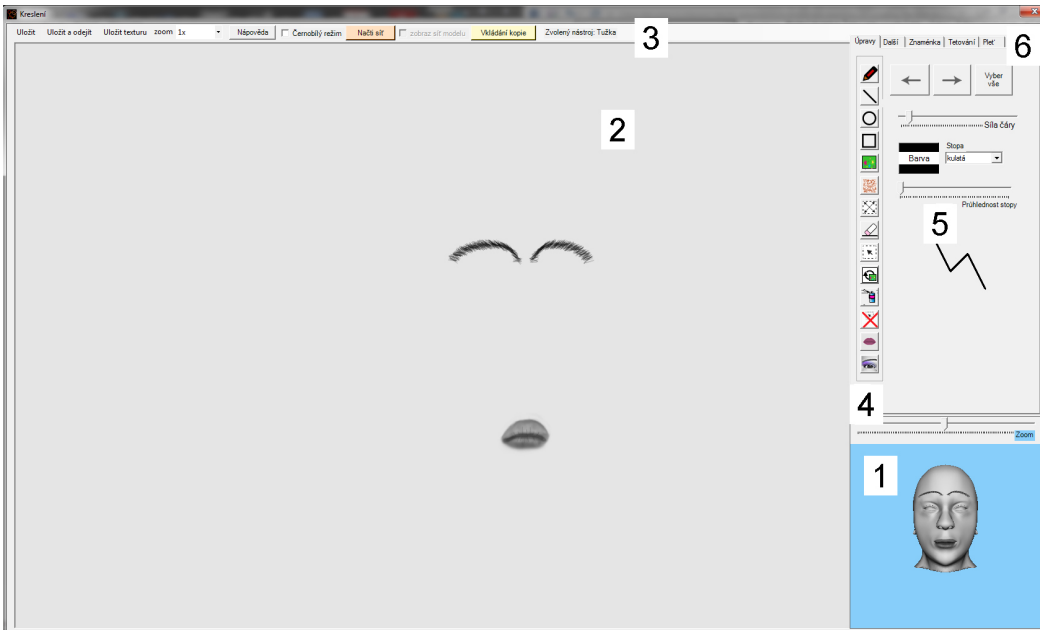
A 1 Kreslení do textury

Po načtení 3D modelu hlavy do programu Identikit zvolíme možnost „Kresli do textury“ (obr. A1).



Obrázek A1: přístup ke knihovně z hlavního programu

V pravé dolní části najdeme okno s náhledem textury na modelu (obr. A2, č.1). Model lze pomocí posuvníku zvětšovat a zmenšovat. Kliknutím levým tlačítkem myši na náhled je možné si ho zvětšit. Většinu pracovní plochy zabírá samotná textura (obr. A2, č.2), kterou je možné přiblížit pro detailní kreslení v horním menu (obr. A2, č.3). Horní menu dále slouží k ukládání pokreslené textury a mimo jiné zde najdeme tlačítko Nápověda s krátkou nápovědou k ovládání. Vpravo nahoře je zobrazen seznam nástrojů (obr. A2, č.4) s možnostmi aktuálně zvoleného nástroje (obr. A2, č.5). Podržením kurzoru myši nad ikonou nástroje si zobrazíme název nástroje a případně také klávesovou zkratku, která nástroj vyvolá. Při přepnutí záložky (obr. A2, č.6) se dostaneme k méně používaným funkcím, jako je například vkládání tetování a změna barvy pleti.



Obrázek A2: Náhled hlavní obrazovky

A 2 Příklady použití

Základní manipulace

Po otevření knihovny je textura připravena pro kreslení. Jako výchozí nástroj je zvolena tužka. V pravém dolním rohu se okamžitě zobrazuje malý náhled výsledku. Tento náhled lze zvětšit stisknutím levého tlačítka myši nad náhledem. Model v tomto náhledu lze zvětšit nebo zmenšit pomocí táhla `Zoom`. V horním menu se nachází oranžové tlačítko `Načti síť`. Po kliknutí na toto tlačítko je možné vybrat ze složky trojúhelníkovou síť aktuálního modelu, která se poté zobrazí na pracovní ploše. Tato síť slouží jako vodítko pro nalezení význačných bodů modelu na textuře a do výsledné textury se neukládá. Její průhlednost lze změnit v záložce `Další`. V horním menu je dále možné uložit celý dokument, uložit si texturu do samostatného souboru a přiblížit si texturu pro detailnější kreslení. Oblast přiblížení je možné měnit tak, že za stisknutí mezerníku přetáhneme myší obdélník určující oblast zvětšení. Tlačítko `Vkládání kopie` umožňuje režim rychlého vkládání aktuálního (vybraného) objektu. Tento režim ukončíme zvolením jiného nástroje. `Černobílý režim` (pod záložkou `Další`) automaticky převádí všechny barvy do odstínů

šedi, avšak zachovává i barevnou verzi, ke které je možné se po odškrtnuí vrátit. Pod tlačítkem `Nápověda` najdeme několik základních tipů pro ovládání knihovny. Při zavření celého okna je zobrazena nabídka umožňující uložení nebo neuložení rozpracovaného projektu.

Transformace

Pro transformace již vložených objektů je nejprve nutné zvolit nástroj `Výběr` a kliknutím myši zvolit požadovaný objekt. Pokud chcete zvolit skupinu objektů, je možné myší táhnout přes požadovanou oblast. Při držení klávesy Shift je možné z vybrané skupiny objektů odebírat. Dále zvolíme nástroj `Úprava`. Nyní je možné tažením myši vybrané objekty přesouvat. Pokud je zvolen jen jeden, je také možné měnit jeho velikost. Pokud je při přesunu stisknuta klávesa Alt, přesouváme pouze kopii zvolených objektů a původní objekty zůstávají na místě. Tato klávesa musí být stisknuta ještě před začátkem tažení objektů. Při zvolení nástroje `Rotace` je možné libovolně rotovat jeden zvolený objekt. Po kliknutí pravým tlačítkem na objekt vložený do textury je možné jej odstranit, přesunout více dozadu nebo dopředu, a u obrázkových objektů (Obrázek, oční stíny...) je zde navíc také možnost zrcadlení.

Sprej

Ovládání nástroje `Sprej` odpovídá klasickému kreslení myší, avšak při podržení myši na jednom místě se body dále generují a oblast je hustěji vyplněna. Při zaškrtnutí možnosti `Sprej na pihy` se místo jednobarevných bodů pomocí spreje generují pihy dle nastavení nástroje `Pihy`. Tímto způsobem je možné vytvořit jakoukoli oblast posetou pihami (viz. obr. 6.11).

Generátor pih

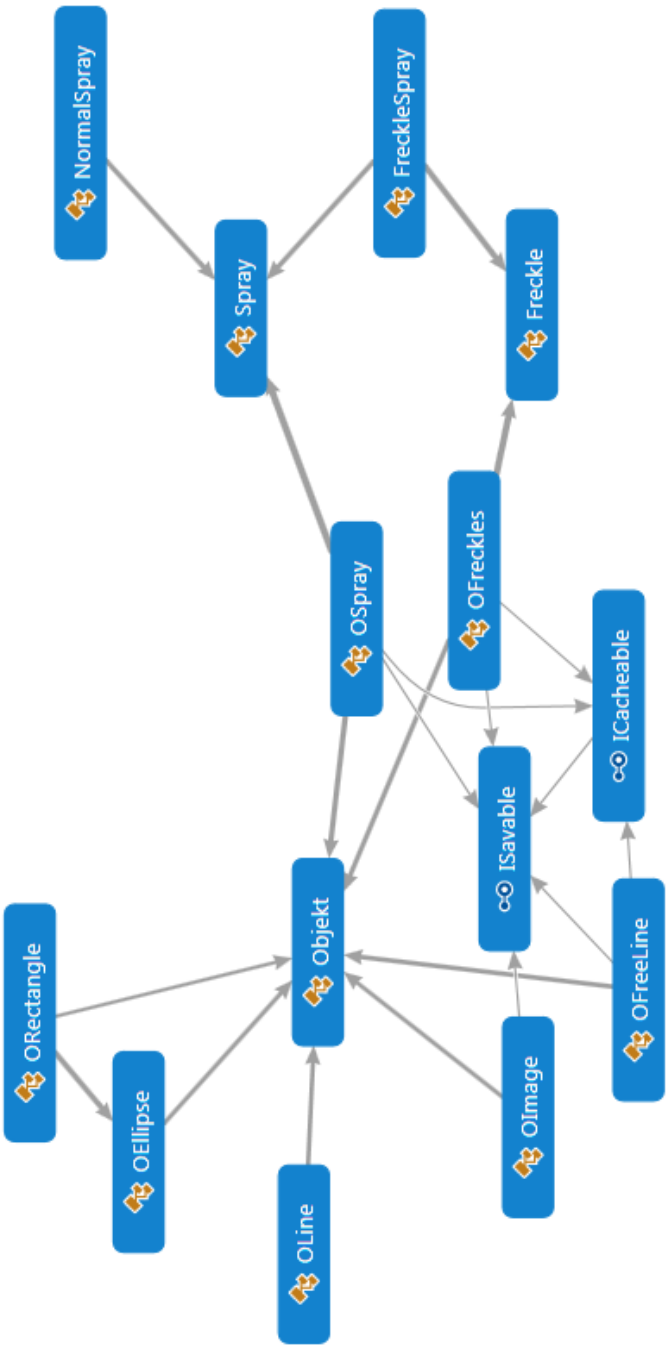
Po zvolení nástroje `Pihy` se v levém panelu zobrazí detailní nastavení tohoto nástroje. Nyní je nástroj v základním nastavení a je možné zvolit na textuře tažením oblast, do které se pihy vygenerují. Z aktuální oblasti je možné pihy i odebírat, a to tažením nové obdélníkové oblasti za držení klávesy Ctrl. Při tažení myší je odečítaná oblast ohraničena červeným rámečkem. Pokud chceme zvolit jiný tvar oblasti, do které se pihy budou generovat, zaškrtneme v pravém menu možnost `Volný výběr oblasti`. Nyní je možné tuto oblast nakreslit.

Tato oblast musí být uzavřená. Pokud čáru nenakreslíme až k původnímu bodu, doplní se oblast automaticky.

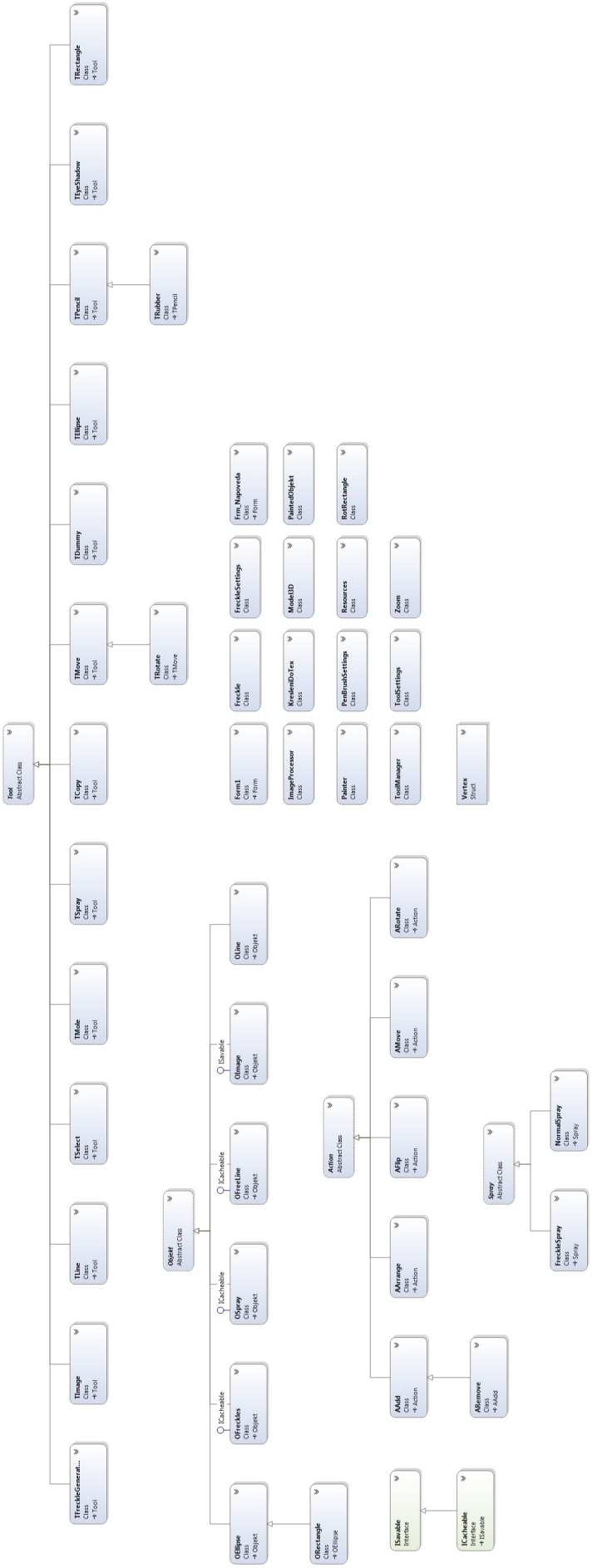
A 3 Umisťování 3D modelů

Pro spuštění knihovny pro umisťování objektů zvolíme v hlavním programu v záložce `Kreslení do textury` možnost `Umisťování objektů`. Po kliknutí na tlačítko `Přidat` lze vybrat soubor ve formátu *.ply, který bude načten. Takto je možné přidat do seznamu i více modelů a jednotlivé modely pojmenovat. Model, který je v seznamu aktuálně zvolen, je v hlavním okně obarven červeně. Tažením myši při stisknutém levém tlačítku je rotován model hlavy, při stisknutém pravém tlačítku je rotován aktuálně zvolený objekt. Zvoleným modelem lze pohybovat pomocí tlačítka `Umístit na model` nebo pomocí jezdců v jednotlivých osách. Objekt lze také přesouvat myší pouze v jedné ose. Zvolením osy v nabídce `Přesun podle osy` a stisknutím tlačítka `Přesunout` je spuštěn režim, který převádí tažení myši na pohyb v jedné z os.

B Struktura tříd a objektů



Obrázek B3: Diagram závislostí pro objekty



Obrázek B4: Diagram tříd knihovny pro kreslení do textury

Literatura

[1] MARTÍNEK, Petr. *Programové vybavení pro sestavování identikitů*. Plzeň, 2012. Diplomová práce. Západočeská univerzita v Plzni. Vedoucí práce Kolingerová Ivana, Prof. Dr. Ing.

[2] FINLEY, Darel Rex. Point-In-Polygon Algorithm: Determining Whether A Point Is Inside A Complex Polygon. FINLEY, Darel Rex. *Alien Ryder Flex* [online]. 1998, 2006, 2007 [cit. 2015-02-24]. Dostupné z: `http://alienryderflex.com/polygon/`

[3] BUTT, Asad. Convert an Image to Grayscale. *Stack Overflow* [online]. 2010, č. 1 [cit. 2015-03-03]. Dostupné z: `http://stackoverflow.com/questions/2265910/convert-an-image-to-grayscale`

[4] BOURKE, Paul. Texture Map Correction for Spherical Mapping. *Polar texture mapping* [online]. 2001, č. 1 [cit. 2015-03-26]. Dostupné z: `http://web.archive.org/web/20060808212243/http://local.wasp.uwa.edu.au/~pbourke/texture_colour/polargrid/`

[5] TURK, Greg. The PLY Polygon File Format. *The PLY Polygon File Format* [online]. 1994, č. 1 [cit. 2015-04-14]. Dostupné z: `http://www.dcs.ed.ac.uk/teaching/cs4/www/graphics/Web/ply.html`

[6] MeshLab `http://meshlab.sourceforge.net/`

[7] UVMapper `http://www.uvmapper.com/index.html`

[8] Blender `http://www.blender.org/`

[9] SlimDX `http://slimdx.org/`

[10] Freedigitalphotos.net `http://http://www.freedigitalphotos.net/`

Seznam obrázků

2.1 Záložka pro spouštění mých knihoven 10

3.1 Vnitřek modelu a zbytky původní čelisti a zubů 12

3.2 Model v základní poloze 12

3.3 Trojúhelníková síť rozloženého modelu 13

3.4 Označení míst pro růst vlasů (červená) a vousů (zelená) 14

3.5 Rty modelu bez textury (vlevo) a s texturou (vpravo) 14

3.6 Ukázka vyrenderovaných modelů 15

3.7 Kovové textury vhodné pro piercing 15

4.1 Ukázka modelu s pokreslenou texturou 17

4.2 Ukázka vektorizovaného tetování 17

4.3 Porovnání nezdeformované textury (vlevo) a zdeformované (vpravo) 19

4.4 Ukázka bitmapy určující tvar stopy 20

4.5 třída Spray 21

4.6 Zdrojový obrázek reprezentující konkrétní tvar očních stínů . . 23

6.1 Sada znamének dostupných přímo z programu 29

6.2 Ukázka sady tetování 29

6.3 Oční stíny na modelu 30

6.4 Automaticky obarvené rty 30
6.5 Obrázkové stopy 31
6.6 Přebarvená pleť 32
6.7 Model s jemnou texturou pleti (vlevo), původní model (vpravo) 32
6.8 Pleť se skvrnami (vlevo) a původní model (vpravo) 33
6.9 Model s vloženým piercingem 33
6.10 Pohyb v jedné z os 34
6.11 Ukázka makeupu vytvořeného v knihovně pro kreslení do textury 35
6.12 Zobrazení os v 3D prostoru 37

A1 přístup ke knihovně z hlavního programu 39
A2 Náhled hlavní obrazovky 40
B3 Diagram závislostí pro objekty 43
B4 Diagram tříd knihovny pro kreslení do textury 44