3D-Gesichtsmodellierung mit bilddatenbasierter Modifikation eines generischen Gesichtsmodells

Entwicklung einer Benutzeroberfläche und Modifikation des generischen Gesichtsmodells anhand bereitgestellter Daten

Bachelor-Thesis

zur Erlangung des akademischen Grades B.Sc.

Gunnar Burmeister

2021915

Hochschule für Angewandte Wissenschaften Hamburg
Fakultät Design, Medien und Information
Department Medientechnik

Erstprüfer: Prof. Dr. Torsten Edeler

Zweitprüfer: Prof. Dr. Andreas Plaß

Hamburg, 15. 08. 2016

Zusammenfassung

Das Ziel dieser Arbeit ist es, den Prozess der 3D-Modellierung eines menschlichen Gesichts zu vereinfachen. Als Ansatz wird dazu eine Anwendung mit grafischer Benutzeroberfläche entwickelt, die ein ausgewähltes Frontalbild basierend auf Benutzereingaben analysiert und anschließend ein generisches 3D-Gesichtsmodell vollautomatisch modifiziert. Die Umsetzung der Bildanalyse, auf die in dieser Arbeit zurückgegriffen wird, erfolgt in der Bachelorarbeit von Kreutz (2016). Für die Modifizierung werden Funktionen der 3D-Grafiksuite Blender verwendet.

Für die Bewertung der Qualität dieser vereinfachten Modellierung wurden im Versuch mehrere Testläufe mit Bildern verschiedener Personen durchgeführt. Das Ergebnis dieser Bewertung zeigt, dass der gewählte Ansatz noch nicht ausgereift genug ist, um ein 3D-Modell zu erhalten, welches das jeweilige Gesicht zufriedenstellend abbildet. Es werden jedoch Möglichkeiten formuliert, mit denen der Ansatz verbessert werden kann.

Abstract

The aim of this thesis to simplify the process of 3d-modeling a human face. As approach an application with a graphical user interface is developed, that analyses a choosen frontal image based on user input and fully automatically modifies a generic 3D model. The image analysis, that is used in this thesis, is implemented in the thesis of Kreutz (2016). The modification is realised with functions of the 3D computer graphics software Blender.

Tests with images of various individuals were passed to assess the quality of this simplified modeling. The result of this assessment shows that the choosen approach is not yet sophisticated enough to receive a 3d model which satisfactorily represents the respective face. Potential ways to optimise the approach are mentioned though.

Ich versichere, die vorliegende Arbeit selbstständig ohne fremde Hilfe verfasst und keine anderen Quellen und Hilfsmittel als die angegebenen benutzt zu haben. Die aus anderen Werken wörtlich entnommenen Stellen oder dem Sinn nach entlehnten Passagen sind durch Quellenangaben eindeutig kenntlich gemacht.

Ort, Datum Gunnar Burmeister

Inhaltsverzeichnis

1 Einleitung

Dieses Kapitel beschäftigt sich mit der Heranführung an das Thema dieser Arbeit, siehe Punkt 1.1, und der Zielsetzung. Diese Arbeit ist im Kontext mit der Arbeit von Kreutz (2016) zu betrachten. Die Zielsetzung besteht aus zwei Teilfragen, welche in 1.2 formuliert sind und von denen eine in dieser und eine in der Arbeit von Kreutz (2016) behandelt wird. Das kombinierte Ergebnis dieser beiden praktischen Arbeiten ist eine Softwareanwendung.

1.1 Aktueller Bezug

Das Thema der digitalen Darstellung von Objekten, insbesondere von Menschen, ist in der heutigen Zeit aktueller denn je. In Bereichen wie dem Film, Videospielen, der Medizin oder der Gesichtserkennung werden hierfür unter anderem 3D-Modelle verwendet. Aufgrund der gestiegenen technischen Leistung von Endgeräten ist es heute möglich, 3D-Darstellungen immer realistischer, wenn nicht gar fotorealistisch zu gestalten. Während teilweise ein ganzes Modell benötigt werden, beispielsweise ein Reiter, werden an anderer Stelle nur Teile wie beispielsweise ein Gesicht benötigt. Für die Erzeugung eines Modells gibt es verschiedene Verfahren, etwa den 3D-Scan oder aber die mehr oder weniger automatisierte Modellierung von Hand. Während die realistische Darstellung nicht immer gewünscht ist, muss in den Bereichen, in denen sie gewünscht oder sogar erforderlich ist, besonders darauf geachtet werden, dass sie authentisch ist (zum Beispiel in der Medizin) oder zumindest so wirkt (zum Beispiel in Videospielen).

Das Gesicht spielt dabei eine wichtige Rolle, denn in der Regel ist das Gesicht eine der Regionen des Körpers, auf die der Betrachter zuerst schaut; auf jeden Fall ist es eines der wichtigsten Körperteile, um einen Menschen zu identifizieren. Wirkt das Gesicht unecht, gilt dies zumeist automatisch auch für die ganze Figur.

Die gute Modellierung eines Gesichtes ist deshalb umso wichtiger, aufgrund seiner individuellen und detailreichen Form zugleich aber schwieriger als andere Körperteile. Für den Laien, der beispielsweise eine 3D-Modell seines eigenen Gesichtes erstellen möchte, ist dies ohne Vereinfachung nicht möglich. In Videospielen, die dem Benutzer das Individualisieren eines Gesichtes ermöglichen, geschieht dies etwa durch das Vorgeben eines Gesichtsprofils, das dann mit Schiebereglern individuell angepasst werden kann. Dies beansprucht immer noch relativ viel Zeit, sofern der Benutzer ein seinen Vorstellungen entsprechend genaues Ergebnis erzielen möchte und das Modell

ist auf die Verwendung im jeweiligen Spiel beschränkt. Software-Produkte, welche die Erstellung eines von anderen Anwendungen losgelösten Modells ermöglichen, haben zumeist Nachteile; so sind diese entweder recht teuer und/oder sie haben eine kompliziertere Bedienung als die eben genannte mit Schiebereglern in Videospielen. Ein Beispiel für eine solche Software ist *Faceworx*. Hier lädt der Benutzer ein Frontal- und ein Profilbild und passt dann detaillierte Schablonen so an, dass sie den jeweiligen Merkmalen entsprechen und die Software modifiziert ein Start-Modell automatisch. Dies ist jedoch immer noch relativ aufwendig.

Für die 3D-Modellierung eines Gesichtes anhand einer Bildvorlage sind prinzipiell zwei Schritte notwendig, zum einen eine Analyse des Bildes und zum anderen die Modellierung selbst. Bei einer manuellen oder teilautomatisierten Modellierung geschieht beides parallel; so betrachtet der Modellierende das Bild und vergleicht es mit dem Modell, während er an diesem arbeitet. *Manuell* bedeutet dabei, dass die Modellierung komplett von Hand und direkt am Modell erfolgt und *teilautomatisiert* meint dabei eine vereinfachte Modellierung, wie etwa bei Faceworx, bei welcher der Benutzer de facto nur ein bestehendes Modell modifiziert.

1.2 Zielsetzung

Für den „normalen“, ungelernten Benutzer ist eine möglichst einfache Bedienung und vorzugsweise auch ein möglichst geringer Kostenaufwand wünschenswert, weil dieser in der Regel nur kurzen Gebrauch von einer solchen Anwendung macht („Just for fun“).

Der Prozess der Modellierung eines 3D-Gesichtsmodells kann dabei hauptsächlich an zwei Stellen vereinfacht werden, der Analyse des Bildes und der Modellierung, unterstützt durch eine Benutzeroberfläche. Beide Schritte können grundsätzlich manuell, teilautomatisch oder vollautomatisch ablaufen. Für eine Analyse in Echtzeit beziehungsweise nahezu in Echtzeit ist es beispielsweise nötig, dass beide Schritte vollautomatisiert sind. Durch die hierbei fehlende „lenkende Hand“ des Benutzers steigt zugleich das Fehlerrisiko. Für den „normalen“ Benutzer ist Echtzeit nicht erforderlich, für ein potentiell besseres und robusteres Ergebnis, kann also auf die Unterstützung durch den Benutzer zurückgegriffen werden. Der Aufwand wird dabei für den Benutzer am geringsten gehalten, wenn sich die Unterstützung dabei auf Eingaben für die Bildanalyse beschränkt. Dieser Schritt im Prozess ist der wichtigere, denn Fehler bei der Bildanalyse machen Folgefehler bei der anschließenden Modellierung sehr wahrscheinlich. Eine gute Bildanalyse hingegen ermöglicht eine vollautomatisierte Modellierung. Wichtig ist nun erstens, wie weit die Bildanalyse automatisiert werden kann und wie viele zusätzliche Eingaben durch den Benutzer noch erforderlich sind, um mit der anschließenden Modellierung ein gutes Resultat zu erzielen, und zweitens, wie die Modellierung umgesetzt wird. Für die Modellierung gibt es die Möglichkeit,

eigene Algorithmen zu entwickeln oder aber auf die Funktionen einer bestehenden 3D-Grafiksuite zurückzugreifen, die auch bei einer manuellen Modellierung zum Einsatz kommen würde. Diese 3D-Grafiksuites sind in ihrem Funktionsumfang zumeist sehr komplex und ausgereift und werden auch im professionellen Bereich eingesetzt. Somit bietet es sich also an, auf eine solche zurückzugreifen.

Für diese beiden Schritte, Bildanalyse und Modellierung, werden im Rahmen der beiden Arbeiten folgende Fragen behandelt:

- Kann eine teilautomatisierte, zielgerichtete Analyse eines geeigneten Frontalbildes eines Gesichts eine ausschließlich manuelle Datenerfassung gleichwertig ersetzen, und dies in einer Art und Weise, die den Benutzeraufwand reduziert?
- Inwiefern kann die vollautomatische Nutzung von komplexen Funktionen einer schon existierenden Grafiksuite sowohl den Modellierungsprozess vereinfachen als auch ein zufriedenstellendes Ergebnis liefern?

Hierbei wird, wie in der ersten Frage formuliert, nur von einem frontalen Bild ausgegangen, weshalb eine Modellierung in die Tiefe nur schlecht möglich ist. Dies soll als erster Ansatz dienen, der später bei erfolgversprechenden Ergebnissen erweitert werden kann. Hierzu wird dann zumindest noch ein weiteres Bild benötigt, welches Tiefeninformationen liefert; hierfür eignet sich beispielsweise ein Profilbild.

In dieser Arbeit wird die zweite Frage behandelt. Die Anwendung, die als Ergebnis dieser beiden Arbeiten entsteht, besteht hauptsächlich aus drei Teilen, einer grafischen Benutzeroberfläche, der Bildanalyse und der Modellierung. Davon sind die grafische Benutzeroberfläche und die Modellierung Bestandteil dieser Arbeit und die Bildanalyse ist Bestandteil der Arbeit von Kreutz (2016). Beide Arbeiten sind in sich geschlossen, greifen aber jeweils auf Daten der anderen Arbeit zurück oder stellen Daten für die andere bereit.

2 Grundlagen

Zunächst erfolgt eine Erläuterung der Themenbereiche dieser Arbeit. Diese sind grafische Benutzeroberflächen und die 3D-Gesichtsmodellierung.

2.1 Grafische Benutzeroberflächen

Grafische Benutzeroberflächen sind in der heutigen Zeit nicht mehr aus der modernen Technik wegzudenken. Waren diese am Anfang der digitalen Technik noch rudimentär und umständlich zu bedienen, so sind sie heute elementarer Bestandteil der meisten elektronischen Geräte, die für die Bedienung durch den Menschen gedacht sind, wie beispielsweise Computer, Smartphones, Tablets, Navigationsgeräte oder Bedienungsanzeigen in Fahrzeugen. Von einfachen Anwendungen bis hin zu komplexen Systemen müssen sie zuverlässig die Kommunikation zwischen dem Benutzer und der Software ermöglichen und sollen gleichzeitig möglichst einfach zu verstehen und zu bedienen sein. In der Software-Ergonomie, die sich speziell damit beschäftigt, wird von grafischer Benutzerschnittstelle, englisch ***G****raphical* ***U****ser* ***I****nterface* (GUI), gesprochen.

Im Folgenden werden zunächst die Merkmale eines GUI beschrieben und anschließend die C++-Klassenbibliothek *Qt* (Version 5.4), die in dieser Arbeit für das GUI verwendet wird.

2.1.1 Merkmale einer grafischen Benutzeroberfläche

Die europäische Norm (ISO 1996: EN ISO 9421-110 ff.) regelt die Anforderungen, die an ein GUI gestellt werden. Sie definiert folgende Merkmale[1], ergänzt durch Beispiele, die sich auf einen Onlineshop beziehen:

- *Aufgabenangemessenheit:* „Ein Dialog ist aufgabenangemessen, wenn er den Benutzer unterstützt, seine Arbeitsaufgabe effektiv und effizient zu erledigen.“
  Die Führung durch das Bestellformular ist klar. Die erforderliche Eingabe des Benutzers ist simpel und, zumindest in Bezug auf die Pflichtangaben, auf die relevanten Daten beschränkt. Bei einer eventuell nötigen Korrektur wird klar auf die Fehlerstelle verwiesen.

- *Selbstbeschreibungsfähigkeit:* „Ein Dialog ist in dem Maße selbstbeschreibungsfähig, in dem für den Benutzer zu jeder Zeit offensichtlich ist, in welchem Dialog,

[1] Entnommen 9421-10, ergänzt durch Änderung in 110

an welcher Stelle im Dialog sie sich befinden, welche Handlungen unternommen werden können und wie diese ausgeführt werden können."
Es gibt eine Anzeige, die Aufschluss darüber gibt, in welchem Schritt des Bestellvorgangs der Nutzer sich befindet, welcher der nächste ist und wie viele es insgesamt gibt.

- *Steuerbarkeit:* „Ein Dialog ist steuerbar, wenn der Benutzer in der Lage ist, den Dialogablauf zu starten sowie seine Richtung und Geschwindigkeit zu beeinflussen, bis das Ziel erreicht ist."
  Der Benutzer hat die Möglichkeit, den Bestellvorgang jederzeit abzubrechen oder einen Schritt zurückzugehen, etwa um Angaben noch einmal zu ändern.

- *Erwartungskonformität:* „Ein Dialog ist erwartungskonform, wenn er konsistent ist und den Merkmalen des Benutzers entspricht, z.B. seinen Kenntnissen aus dem Arbeitsgebiet, seiner Ausbildung und seiner Erfahrung sowie den allgemein anerkannten Konventionen."
  Bezeichnungen im und um den Bestellvorgang und der Aufbau, etwa die Position bestimmter Elemente, sind identisch. Beispielsweise ist die Schaltfläche, mit der der Benutzer zum nächsten Schritt gelangt, immer auf der rechten Seite unterhalb des aktuellen Schrittes.

- *Fehlertoleranz:* „Ein Dialog ist fehlertolerant, wenn das beabsichtigte Arbeitsergebnis trotz erkennbar fehlerhafter Eingaben entweder mit keinem oder mit minimalem Korrekturaufwand seitens des Benutzers erreicht werden kann."
  Es werden die Angaben des Benutzers geprüft, etwa ob eine angegebene Email-Adresse ein At-Zeichen (@) enthält und der Benutzer wird im Fehlerfall eindeutig auf die entsprechende Stelle verwiesen.

- *Individualisierbarkeit:* „Ein Dialog ist individualisierbar, wenn das Dialogsystem Anpassungen an die Erfordernisse an die Arbeitsaufgabe sowie an die individuellen Fähigkeiten und Vorlieben des Benutzers zulässt."
  Der Benutzer kann seine Daten hinterlegen, sodass er sie bei einer anderen Bestellung nicht erneut eingeben muss oder die Möglichkeit hat, ein früher gekauftes Produkt einfach wieder zu bestellen.

- *Lernförderlichkeit:* „Ein Dialog ist lernförderlich, wenn er den Benutzer beim Erlernen des Dialogsystems unterstützt und anleitet."
  Der Benutzer bekommt Hinweise, wie bestimmte Abläufe vonstattengehen oder aus welchen Zeichen sein Passwort zusammengesetzt sein kann.

Ein GUI besteht aus einem Hauptfenster, in welches die einzelnen Elemente, auch Widgets genannt, integriert sind. In der folgenden Abbildung 2.1 ist das Hauptfenster von Microsoft Paint zu sehen. Integriert sind etwa die Werkzeugleiste im oberen

Bereich und ein Bereich zur Darstellung des Bildes im Hauptbereich.

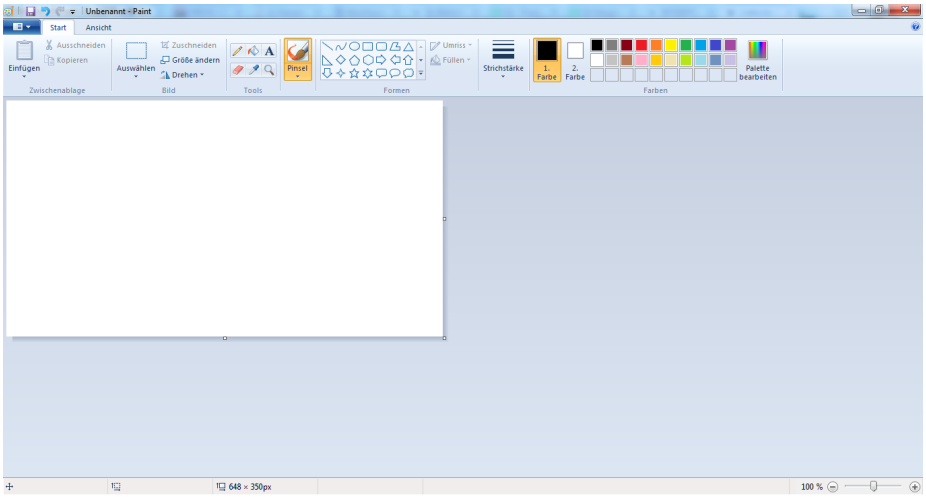

Abbildung 2.1: Grafische Oberfläche von Microsoft Paint

Alle Elemente zusammen bilden das ***W**indows-**I**cons-**M**enus-**P**ointer*-Modell (WIMP), welches aktuell das wichtigste Grundkonzept von GUIs darstellt. Pointer bezieht sich dabei auf den Mauszeiger. Um die Bedienung auch ohne Zeigegerät zu ermöglichen, hat immer ein GUI-Element den *Fokus*, dass bedeutet, die nächste Aktion des Benutzers mit einem zeigerlosen Eingabegerät, etwa der Tastatur, wird die Funktion dieses Elements auslösen. Hervorgehoben wird das entsprechende Element beispielsweise durch eine Umrandung bei Schaltflächen oder einen blinkenden Cursor bei Textfeldern. Der Benutzer kann den Fokus beispielsweise mit den Pfeiltasten oder der Tabulatortaste wechseln. Für die Maus wird ein *Capture* verwendet, mit dem bestimmt wird, welches Element die nächste Mauseingabe verarbeitet. In der Regel ist dies das Element unter der Maus, zum Beispiel bei Tooltips oder Mausklicks. Ausnahmen sind häufig mit einer gedrückt gehaltenen Maus verbunden, etwa Drag und Drop. Wird die Maus dabei etwa kurzzeitig über eine Schaltfläche bewegt, wird diese nicht betätigt.

Die Implementierung dieser Funktionalitäten geschieht meist durch darauf ausgelegte Bibliotheken wie Qt.

2.1.2 C++-Klassenbibliothek Qt

Qt ist eine C++-Klassenbibiliothek, die für die plattformübergreifende Programmierung von GUIs gedacht ist. Dies schließt neben Computern mit verschiedenen Betriebssystemen auch andere Geräte wie Smartphones ein. Die zahlreichen C++-Klassen sind in einzelne Module, zum Beispiel *Qt Core*, *Qt Gui* und *Qt Widgets*,

aufgeteilt, wodurch die Größe und die Performance von Anwendungen optimiert werden kann, weil nur die Module eingebunden werden müssen, die tatsächlich benutzt werden. Die explizite Programmierung kann mit der Verwendung des *QtDesigners* kombiniert werden, der es dem Benutzer ermöglicht, die Oberfläche in Echtbilddarstellung („What you see is what you get“) zu erstellen.

Ein wichtiges Werkzeug der Qt-Bibliothek ist der ***M**eta-**O**bject-**C**ompiler* (moc), mit dem Funktionen wie *Introspektion* beziehungsweise *Reflexion*, die kein nativer Bestandteil von C++ ist, oder das von Qt geprägte *Signal-Slot-Konzept*, das einen ereignisgesteuerten Programmfluss ermöglicht, implementiert werden. Bei letzterem wird im Zuge eines bestimmten Events, zum Beispiel dem Drücken einer „Speichern“-Schaltfläche, ein Signal ausgesendet (*emittiert*) und ein oder mehrere Empfänger (Slots), für sich genommen einfach Funktionen, die mit dem Signal verbunden (*connected*) sind, werden aufgerufen. Bei einer „Speichern“-Schaltfläche kann sich etwa ein Dateidialog öffnen, in dem der Benutzer eine Datei an der gewünschten Stelle mit dem gewünschten Namen speichern kann. Das Konzept stellt eine Implementierung des Beobachter-Entwurfsmusters dar und ist in Bezug auf Typsicherheit einfacher und flexibler zu handhaben als *Callbacks*[2], wenngleich diese, zumindest theoretisch, etwas performanter sind.

Für die Darstellung und Interaktion von beziehungsweise mit selbst programmierten 2D-Elementen bietet Qt, unter anderem, das *Graphics View Framework*. das einen item-basierten Ansatz der *Model-View-Programmierung*[3] bereitstellt. Die drei Hauptbestandteile sind die Szene (QGraphicsScene), *Scene*, ein Betrachungsfenster (QGraphicsView), *View*, sowie Objekte (QGraphicsItem), *Items*. Dabei beinhaltet die Scene die Items und das View stellt das Widget bereit, in dem die Scene dargestellt wird. Die einzelnen Komponenten können verschieden Ereignisse unterstützen, beispielsweise Drag und Drop, Doppelklick, Zoom, Rotation, Kollision oder Skalierung von Items. Dieses Framework wird in dieser Arbeit für die in 4.1.2 beschriebene Auswahlmaske verwendet.

2.2 3D-Gesichtsmodellierung

Bei der 3D-Modellierung im Allgemeinen und der 3D-Gesichtsmodellierung im Speziellen geht es darum, ein Objekt aus der realen Welt in digitaler Form durch ein 3D-Modell zu beschreiben und in der Regel auch darum, es zu visualisieren. Die folgenden Abschnitte befassen sich damit, was ein 3D-Gesichtsmodell beschreiben muss,

[2] Callbacks sind Zeiger auf Funktionen, die als Parameter an eine andere Funktion übergeben und dann, unter entsprechenden Bedingungen, aufgerufen werden können.

[3] View und Controller des Model-View-Controller Entwurfsmusters (MVC) werden hier zusammengezogen. Die Gliederung passt inhaltlich jedoch nur bedingt zur ursprünglichen Definition von MVC durch Trygve Reenskaug, 1979.

wie es dies tut, wie man sie grundsätzlich erstellt und wie dies in der 3D-Grafiksuite *Blender* (Version 2.72b), die für diese Arbeit verwendet wird, im Speziellen möglich ist. Zuletzt werden einige Beispiele für den Einsatz von 3D-Gesichtsmodellen aufgeführt.

2.2.1 Struktur von Gesichtern

Um ein Gesicht möglichst real mit einem 3D-Modell darzustellen, ist es zunächst wichtig, die Beschaffenheit von Gesichtern zu betrachten. Relevant sind für die visuelle Abbildung die grundsätzlichen Formen wie Kieferbreite, Ausprägung der Wangenknochen oder Größe der Nase sowie das Verhältnis von Bereichen zueinander, etwa der Abstand der Augen. Diese Merkmale unterscheiden sich je nach Mensch und zusätzlich je nach Volkszugehörigkeit, es lassen sich jedoch gewisse Zusammenhänge beziehungsweise Relationen erkennen, zum Beispiel, dass die Augen sich etwa auf halber Höhe des Kopfes befinden. Der *goldene Schnitt* [4] kann hierzu verwendet werden. Einige der so gebildeten Verhältnisse können der Abbildung 2.2 unten entnommen werden.

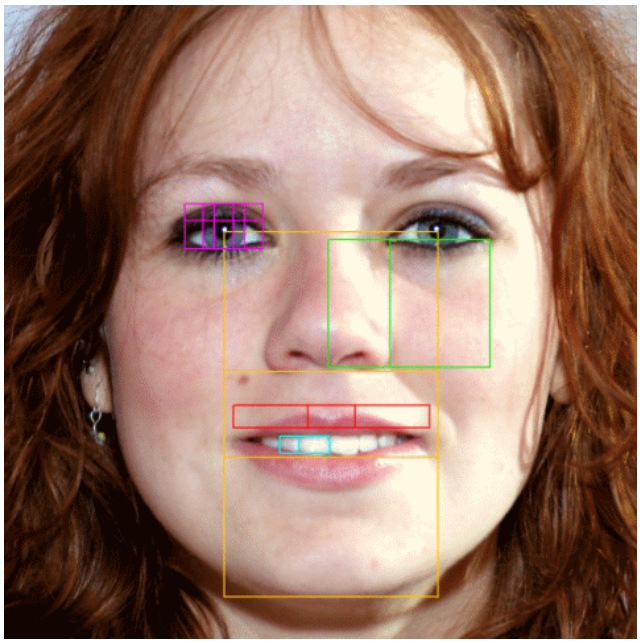

Abbildung 2.2: Relationen im Gesicht auf Basis des Goldenen Schnitts, entnommen Phi-Point Solutions

[4] Der goldene Schnitt ist ein besonderes, asymmetrisches Teilungsverhältnis (1:1,618), das in verschiedenen Bereichen zum Einsatz kommt, unter anderem für die Beschreibung der „Schönheit“ von Menschen.

In der Praxis wird nur bedingt exakt nach diesen Verhältnissen gearbeitet. Üblicherweise nimmt der Betrachter beziehungsweise Modellierer sie intuitiv wahr und unter Umständen fließt der eine oder andere Wert mit ein.

Neben der äußeren Form des Gesichts ist auch der Verlauf der Muskulatur wichtig. Die Muskulatur ist einerseits für die Animation wichtig, wobei diese im Rahmen dieser Arbeit nicht vorgenommen wird, andererseits auch für die Modellierung beziehungsweise Modifikation, damit keine unnatürliche Verzerrungen auftreten, wenn Bereiche des Modells bewegt werden.

Wichtige Muskelpartien befinden in der Stirn, für die Bewegung der Augenbrauen, und in den Wangen, um den Mund, die Wangen selbst, aber auch die Augen zu bewegen oder zu verformen, beispielsweise beim Lächeln. Der Verlauf der Muskulatur kann der folgenden Abbildung 2.3 entnommen werden.

Abbildung 2.3: Struktur der Muskulatur eines menschlichen Gesichts, entnommen Von Rolbeck

2.2.2 3D-Gesichtsmodell

Ein 3D-Modell, das, wie für diese Arbeit, beispielsweise im Rahmen einer geometrischen Modellierung entsteht, dient dazu, ein reales Objekt digital zu beschreiben. Dabei gibt es verschiedene Beschreibungsmöglichkeiten, die bei Volumenmodellen, die

häufig verwendet werden, grundsätzlich in *direkte* und *indirekte Darstellungsschemata* unterteilt sind.

Direkte Darstellungsschemata verwenden einfache Grundkörper, um einen komplexen Körper zu konstruieren. Hier werden beispielsweise Kugeln oder Würfel durch verschiedene Operatoren wie Addition oder Subtraktion miteinander verbunden. ***C****onstructive* ***S****olid* ***G****eometry* (CSG) etwa ermöglicht die Darstellung eines Modells in Form eines Baumes, der die einzelnen Schritte für die Darstellung des Objektes widerspiegelt (siehe Abbildung 2.4).

Indirekte Darstellungsschemata definieren ein 3D-Modell über seine Hülle. Dies kann beispielsweise durch ein *Drahtgittermodell* geschehen, welches das Objekt über Außenkanten beschreibt. Das Problem hierbei ist, dass Drahtgittermodelle aufgrund von fehlenden Flächen nicht unbedingt eindeutig sind.

Eine andere, verbreitete Möglichkeit ist das *Boundary Representations*-Modell (B-rep). Hierbei wird das Objekt mit Punkten (*Vertices*), Kanten (*Edges*), die diese Punkte verbinden und Flächen (*Faces*), die die Kanten füllen, eindeutig beschrieben. Indirekte Verfahren sind sehr effizient darzustellen und eine spezielle Version der B-reps, *Polygonnetze*, ist in der Computergrafik sehr verbreitet. Polygonnetze werden üblicherweise mit drei- oder viereckigen Flächen gebildet (siehe Abbildung 2.4).

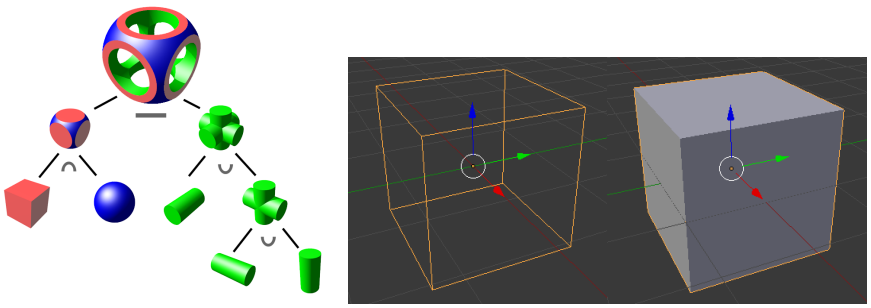

Abbildung 2.4: Darstellungsschemata: CSG-Baum (links), entnommen Wikipedia (2012), Drahtgittermodell (Mitte), Polygonnetz (rechts)

3D-Modelle sind für sich genommen reine Datensätze. Um sie visuell zu veranschaulichen, müssen aus diesen Rohdaten Bilder berechnet (*gerendert*) werden, die für den Betrachter dann auf Monitoren dargestellt werden können. Dieser Vorgang wird *Bildsynthese* genannt.

Zu beachten ist dabei, dass ein Modell zunächst nicht texturiert ist, häufig wird es dann grau dargestellt.

2.2.3 Modellierungsverfahren

Für die eigentliche Modellierung des Modells gibt es mehrere Verfahren, die grundsätzlich in zwei Verfahrensarten eingeteilt werden können. Eine Möglichkeit besteht darin, reale Objekte zu erfassen und zu vermessen und dann mit den daraus gewonnen Daten ein 3D-Modell zu generieren. Hierbei kommen beispielsweise Scanner oder Kameras zum Einsatz; man spricht hier etwa von *3D-Scan* oder *3D-Rekonstruktion*. Das Objekt, das erfasst werden soll, wird für den Vorgang dabei entweder Stück für Stück mit einem Laser abgetastet oder durch eine Vielzahl von Bildern aus verschiedenen Winkeln beschrieben. Das anschließend automatisch berechnete Modell muss in der Regel noch von Hand ausgebessert werden, besonders reflektierende Oberflächen sind problematisch. Auf diese Art der Modellierung wird im Rahmen dieser Arbeit nicht weiter eingegangen, weil sie nicht verwendet wird. Stattdessen erfolgt eine *geometrische Modellierung* in einer 3D-Grafiksuite.

Bei der Modellierung in einer 3D-Grafiksuite wird das Modell durch mathematische Operationen erstellt. Wie in 2.2.2 beschrieben, gibt es die Möglichkeit, das Modell entweder durch sein Volumen oder durch seine Oberfläche zu beschreiben. Die hier verwendete und im nächsten Abschnitt näher beschriebene 3D-Grafiksuite *Blender* arbeitet mit der Oberflächenbeschreibung, deshalb bezieht sich die weitere Ausführung auf diese Art der Modellbeschreibung.

Eine gebräuchliche Möglichkeit, ein Modell zu erstellen, ist das *Box Modeling*. Dabei wird von einer Grundform ausgegangen, zumeist von einem Würfel, die dem gewünschten Objekt immer weiter angenähert wird. Eine oder mehrere Vorlagen und eine gewisse Planung der Vorgehensweise sind dabei wichtig. Die Bearbeitung des Modells selbst erfolgt mit verschiedenen Funktionen, mathematischen Transformationsoperatoren. Beim Polygonnetz können dabei Punkte, Kanten oder Flächen einzeln oder in Gruppen transformiert, skaliert, oder unterteilt werden. Die folgende Abbildung zeigt beispielhaft einige Schritte während des Box Modeling. Hier wird zusätzlich für die linke Hälfte des Modells ein sogenannter *Mirror-Modifier* verwendet, der die rechte Hälfte exakt repliziert.

Abbildung 2.5: Zwischenschritte einer 3D-Modellierung mit Box Modeling, entnommen Rosver (2009)

Eine andere Möglichkeit der Modellierung stellen ***n**on-**u**niform **r**ational **B**-**s**plines* (NURBS) dar. Hierbei handelt es sich um Kurven oder Flächen, die durch bestimmte Punkte, sogenannte Kontrollpunkte, beeinflusst werden. Diese NURBS beschreiben den Verlauf des Modells, wobei die Kontrollpunkte die Parameter der Polynome steuern, welche die Kurven oder Flächen definieren. Aufgrund der in Korrelation zur Größe zunehmenden Komplexität der Formeln und dem damit einhergehenden Rechenaufwand werden die Modelle zumeist aus mehreren Teilen, sogenannten *Patches*, zusammengefügt. NURBS sind insbesondere im industriellen Bereich der Standard für die mathematisch exakte Darstellung von Objekten.

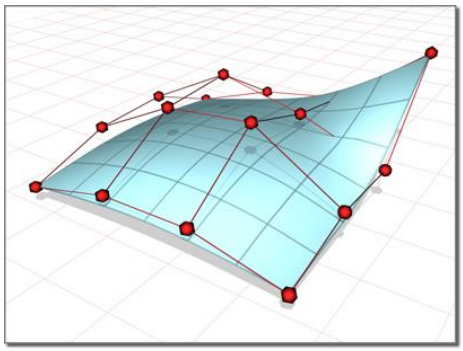

Abbildung 2.6: Modell in NURBS-Darstellung, entnommen 3dmax-Tutorials

Neben dem bereits erwähnten Mirror-Modifier gibt es weitere Modifier und Funktionen, die dem Modellierer Arbeit abnehmen. *Smooth-Shading* etwa verändert das Modell nicht, aber die Oberfläche wird bei der Darstellung und für das Rendern interpoliert und nicht exakt wiedergegeben. Durch diesen vorgetäuschten hohen Detailgrad entstehen Rundungen und das Modell sieht glatter und organischer aus.

2.2.4 3D-Grafiksuite Blender

3D-Grafiksuites verfügen neben der Möglichkeit der Modellierung auch über weitere Funktionen, welche mindestens die Texturierung und die Bildsynthese ermöglichen. Namhafte Software sind etwa *3DS-Max*, *Blender* oder *Maya*, wobei 3DS-Max und Maya kostenpflichtig[5] sind. Blender hingegen steht Nutzern kostenlos zur Verfügung, was als das entscheidende Kriterium für seine Verwendung im Rahmen dieser Arbeit gilt, denn die benötigten Funktionen und eine Schnittstelle für Skripte, mit denen eine automatisierte Bedienung möglich ist, bieten alle drei Grafiksuites und in allen Fällen ist die dafür verwendete Skript-Sprache *Python*.

[5]Beide erhältlich über Autodesk, eine einjährige Lizenz kostet, stand 11.07.2016, in beiden Fällen 1904 €.

Blender bietet neben der Modellierung, Texturierung und Bildsynthese diverse weitere Funktionalitäten wie Animation, Postproduktion oder eine eigene 3D-Spieleengine. Die Oberfläche ist sehr flexibel und kann an die Vorlieben des Benutzers und gerade benötigte Werkzeuge angepasst werden.

Als Hauptdarstellungsform von 3D-Modellen nutzt Blender Polygonnetze, andere Formen wie NURBS werden jedoch auch unterstützt.

Wie bereits erwähnt, wird als Skriptsprache Python verwendet. Blender hat einen Python-Interpreter eingebettet, der im Hintergrund läuft. Die Programmierschnittstelle (englisch ***A**pplication **P**rogramming **I**nterface*, API) stellt Blender dem Interpreter über das Hauptmodul *bpy* bereit. Unter diesem Modul sind weitere Module gegliedert; für diese Arbeit sind die Folgenden von besonderer Bedeutung:

- *bpy.data:* Dieses Modul bietet den Zugang zu den internen Daten von Blender wie Szenen, Objekten, Vertices, Vertexgruppen.
- *bpy.ops:* Dieses Modul bietet den Zugang zu den Operatoren von Blender wie Translation, Rotation, Skalierung.

Neben bpy gibt es noch das *bmesh*-Modul. Dieses bietet erweiterte Funktionalitäten, die benötigt werden, hauptsächlich, um für die Modifikation die Merkmalskorrespondenzpunkte einzeln auswählen zu können, siehe hierzu 4.3.3.

2.2.5 Beispiele für den Einsatz von 3D-Gesichtsmodellen

In der Einleitung wurden bereits einige Bereiche aufgeführt, in denen 3D-Gesichtsmodelle zum Einsatz kommen. An dieser Stelle folgen dazu einige explizite Beispiele.

Im Bereich der *Videospiele* werden individualisierte Gesichtsmodelle seit einigen Jahren immer mehr verwendet, besonders in Videospielen, die einen „Karrieremodus“ bieten, in dem der Spieler einen personalisierten Charakter spielt. Vorwiegend sind dies Sportspiele wie *Fifa* (Fußball) oder *NBA Live* (Basketball). Hier hat der Spieler die Möglichkeit, ein sogenanntes *Game Face* zu erstellen. Hierbei wird ein Modell beziehungsweise dessen Gesicht an ein Bild des Spielers angepasst. Das Game Face muss dabei nicht übermäßig genau gestaltet sein, weil die Spielfigur meist nicht von Nahem zu sehen ist.

Ein anderes Beispiel ist besonders im Rollenspielbereich vertreten. Hier hat der Spieler die Möglichkeit, ein vorgegebenes Modell mit Schiebereglern zu individualisieren. Üblicherweise betrifft die Individualisierung hier das komplette Charaktermodell; das Hauptaugenmerk liegt jedoch in der Regel auf dem Kopf. Spiele wie etwa *The Elderscrolls V: Skyrim* lassen dabei der Kreativität des Spielers viel Spielraum und wie realitätsnah das Aussehen dabei im Ergebnis ausfällt, bleibt dem Spieler

überlassen.

Im *Filmbereich* werden beispielsweise für ***C**omputer **G**enerated **I**magery* (CGI) Gesichtsmodelle verwendet. In *Terminator: Genisys* etwa wird eine junge Version Arnold Schwarzeneggers auf einen anderen Schauspieler, Brett Azar, projiziert, Gesicht inklusive. Dafür wurde auf Basis von zahlreichen Bildern, Mimikaufnahmen des aktuellen Gesichts und einer Dokumentation (*Pumping Iron* von 1977) eine digitale Version von Schwarzenegger nachgebildet. Dies war mit großem Aufwand verbunden und hat etwa zwölf Monate Arbeit für die nicht einmal fünf Minuten, die die junge Version von Schwarzenegger im Film zu sehen war, in Anspruch genommen. Dieses Beispiel stellt vom Aufwand her jedoch einen Sonderfall dar.

In der *Medizin* werden Modelle von Gesichtern etwa für die Erstellung von Gesichtsprothesen verwendet, die mittels eines 3D-Druckers produziert werden. Hierfür wird in der Regel zunächst ein 3D-Scan des Gesichts oder eines Teilbereichs erstellt und mit dem daraus resultierenden 3D-Modell kann dann die benötigte Prothese hergestellt werden. Auf diesem Wege können zum Beispiel körperliche Verunstaltungen als Folge eines Unfalls kaschiert werden.

Ein weiterer wichtiger Bereich ist die sogenannte *Gesichtserkennung*. Hier geht es darum, ein Gesicht, zum Beispiel aus einem Überwachungsvideo, mit bekannten Gesichtern, etwa denen aus einer Datenbank, abzugleichen und im optimalen Fall ein passendes Gesicht und damit Informationen zu der Person zu finden. Dies kann bei der Aufklärung oder Prävention von Verbrechen von Interesse sein, aber auch in anderen Fällen. *Facebook* etwa hat eine System, genannt *DeepFace*, entwickelt, mit dem es möglich sein soll, Menschen in verschiedenen Bildern wiederzuerkennen, auch wenn die Haltung und Mimik nicht die gleiche sind, also genauso, wie der Mensch selbst es kann. Nach eigenen Angaben wird dabei eine Genauigkeit von über 97% erreicht, Facebook (2014).

Abbildung 2.7: Beispiele für den Einsatz von Gesichtsmodellen: Charaktererstellung in Skyrim (links), entnommen Shepard (2014), B. Azar als A. Schwarzenegger in Terminator:Genisys (rechts), entnommen Gonzales (2015)

3 Konzept

Dieses Kapitel beschreibt, welche Anforderungen an das GUI gestellt werden und wie der Modellierungsprozess insgesamt für den Benutzer einfacher und schneller gestaltet werden soll.

3.1 Anforderungen an die grafische Benutzeroberfläche

Es gibt mehrere Anforderungen, welche die entwickelte Anwendung erfüllen soll. Dies sind zum einen grundsätzliche Merkmale eines GUI wie in 2.1.1 beschrieben:

- Die Anwendung soll möglichst einfach zu bedienen sein und dem Benutzer, wenn nötig, Hilfestellung leisten.
- Bei Fehlern, entweder aufgrund von fehlerhaften Eingaben seitens des Benutzers oder aufgrund von Fehlern im Ablauf der Anwendung, soll der Benutzer darüber informiert werden, welcher Fehler vorliegt und wie er diesen beheben kann.
- In den Schritten, in denen der Benutzer Nichts tut, beispielsweise bei der Analyse und der Modellierung, soll der Benutzer erkennen können, wie der dort Fortschritt ist.

Zum anderen gibt es auch einige anwendungsspezifische Anforderungen. Diese ergeben sich aus dem, was die Bildanalyse an Ausgangsdaten erwartet und aus dem, wie der Modellierungsprozesses an sich vereinfacht werden soll (siehe hierzu 3.2):

- Es wird ein Bild erwartet, der Benutzer muss also die Möglichkeit haben, ein Bild auszuwählen, das dann geladen wird.
- Es wird eine grobe Vormarkierung der Bereiche, in denen sich die Gesichtsmerkmale befinden, die in der Bildanalyse gefunden werden sollen, durch den Benutzer vorausgesetzt, siehe hierzu Kreutz (2016). Es muss für den Benutzer also möglich sein, auf dem Bild Bereiche zu markieren und er muss wissen, welche Bereiche er markieren soll. Darüber hinaus muss sichergestellt werden, dass diese Vormarkierung nicht erkennbar falsch ist, etwa dass die Markierung eines Auges außerhalb der Markierung für das Gesicht selbst liegt.

3.2 Vereinfachung des Modellierungsprozesses

Die detailgetreue Modellierung eines menschlichen Gesichts ist sehr aufwendig. Das Gesicht ist eines der wichtigsten, wenn nicht sogar das wichtigste visuelle Merkmal eines Menschen, so zum Beispiel, wenn es darum geht, einen Menschen wiedererkennbar und auch realistisch wirkend darzustellen. Hierbei können schon leichte Deformierungen beim Betrachter das Gefühl hervorrufen, dass etwas „nicht richtig“ ist. Dabei muss dieser nicht einmal unbedingt sagen können, was genau nicht stimmt. Darüber hinaus werden meist Modelle für mehrere verschiedene Personen beziehungsweise Charaktere benötigt, zum Beispiel für ein Videospiel. Deswegen ist es erstrebenswert, den Modellierungsprozess insgesamt zu vereinfachen.

Neben der Optimierung und Vereinfachung der Arbeitsumgebung lässt sich auch die Modellierung selbst vereinfachen. Hier bietet es sich an, nicht jedes Mal ein komplett neues Modell zu konstruieren, sondern, ähnlich den Editoren in Videospielen, von einem Grundmodell auszugehen und dieses zu modifizieren. Das Modell muss dabei möglichst allgemein (generisch) sein, um jede Gesichtsform erreichen zu können. Durch die Verwendung eines Ausgangsmodells reduziert sich die Modellierung mehr zu einer Modifikation.

Im Rahmen dieser Arbeit soll diese Modifikation eines generischen Modells anhand eines Bildes erfolgen. Dafür wäre es möglich, dem Benutzer einen Editor zur Verfügung zu stellen, in dem er manuell dieses generische Gesichtsmodell anhand einer Bildvorlage modifizieren kann. Für den ungeübten, nicht professionellen Benutzer wäre dies jedoch immer noch recht aufwendig und würde relativ viel Zeit in Anspruch nehmen. Der geringste Aufwand für den Benutzer entstünde, wenn dieser das Bild mit dem gewünschten Gesicht einfach in einer Anwendung auswählen kann, auf eine Schaltfläche („Weiter“) drückt und die Software alle weiteren Schritte automatisch ausführt. Daraus ergibt sich jedoch der Nachteil, dass Fehler in der Analyse des Bildes nicht vom Benutzer korrigiert werden können, was ein Bild, zumindest erst einmal, unbrauchbar machen würde. Und auch die Reliabilität der Anwendung wäre so schwierig zu gewährleisten. Eine gewisse Einflussnahme durch den Benutzer ist deshalb sinnvoll. Um diese möglichst einfach, aber trotzdem zielführend zu gestalten, bietet sich eine Vormarkierung auf dem Bild an. Eine ebenso simple wie trotzdem weitestgehende ausreichende Vormarkierung ist durch das Umranden der relevanten Merkmale mit einfachen geometrischen Formen wie Rechtecken zu erzielen. Diese lassen sich verhältnismäßig einfach setzen und, falls nötig, nachjustieren.

Zusammengefasst soll der Modellierungsprozess deshalb durch folgende Punkte vereinfacht werden:

- Die Verwendung eines generischen 3D-Gesichtsmodell. Hierdurch wird die eigentliche Modellierung zu einer Modifikation.

- Die Vormarkierung von wichtigen Gesichtsmerkmalen auf einem Gesichtsbild, das als Vorlage dient, durch den Benutzer.
- Die automatisierte Analyse des Gesichtsbildes unter Berücksichtigung ausgehend von den Benutzermarkierungen.
- Die automatisierte Modifikation des generischen Modells anhand der Ergebnisse der Analyse.

Der dritte Punkt wird dabei in der Arbeit von Kreutz (2016) behandelt.

4 Umsetzung

Die Anwendung ist überwiegend in C++ entwickelt. Eine Ausnahme bildet die Modifikation des 3D-Modells. Diese erfolgt durch Python-Skripte, um Blender ansprechen zu können. Für das GUI wird die *Qt*-Klassenbibliothek verwendet und für die Bildanalyse *OpenCV*, siehe dazu Kreutz (2016). Der schematische Aufbau der Anwendung kann Abbildung 4.1 entnommen werden.

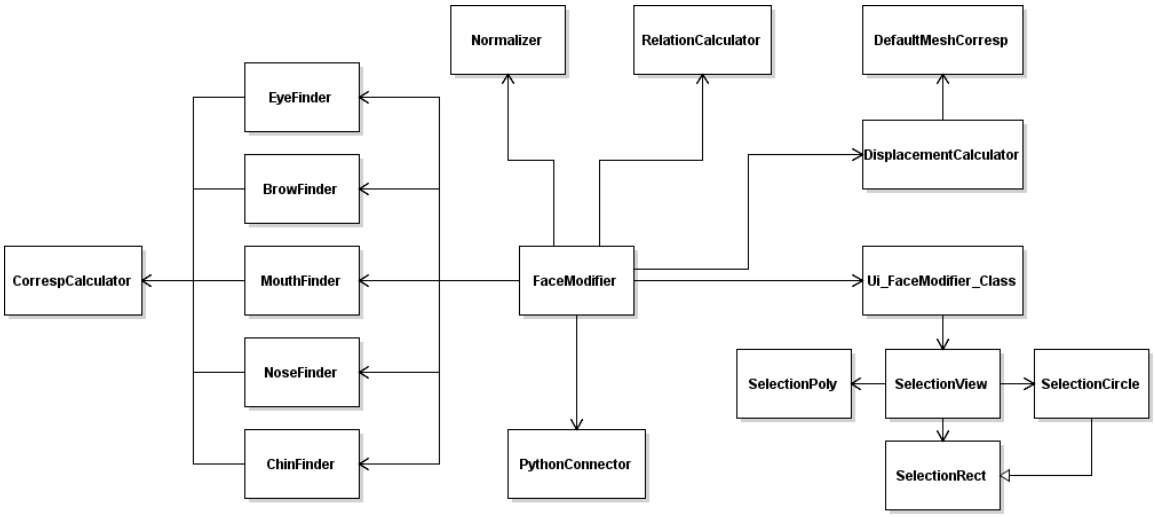

Abbildung 4.1: Struktur der Anwendung

Die Anwendung arbeitet mit einem Ausgabeordner, der bei der Installation standardmäßig mit dem Pfad `C:\Users\%username%\Documents\FaceModifier\Output\` angelegt wird.

Im Folgenden wird die Umsetzung des GUI, das generische 3D-Gesichtsmodell und dessen Modifikation beschrieben.

4.1 Gestaltung der grafischen Benutzeroberfläche

Die Anwendung hat wenige Teilschritte, weshalb das GUI nur aus wenigen Elementen besteht. Diese ergeben sich aus den in 3.1 ausgeführten Anforderungen, dass der Benutzer die Möglichkeit haben muss, ein Bild auszuwählen (siehe 4.1.1) und

anschließend Bereiche auf diesem Bild zu markieren (siehe 4.1.2). Die danach folgenden Programmschritte sind rein programmatischer Art. Damit der Benutzer einen Überblick hat, was im Anwendungsablauf gerade passiert, wann die Modifikation fertig ist und auch, wenn Fehler auftreten, wird auf verschiedene Arten mit dem Benutzer kommuniziert. Die Schritte der Anwendung werden auf drei aufeinander folgende Seiten verteilt, wobei der Benutzer den Wechsel von einer Seite zur nächsten per Schaltfläche vollzieht. Jegliche Interaktion, zum Beispiel Hinweise, Bestätigungen oder Fehlermeldungen, zwischen Anwendung und Benutzer wird, mit Ausnahme der Vorauswahlmaske, in 4.1.3 erläutert. Texte und Bezeichnungen von Schaltflächen und dergleichen sind dabei in englischer Sprache gehalten.

4.1.1 Bildauswahl

Für die Bildauswahl stehen zwei gebräuchliche Varianten zur Verfügung, zum einen eine Auswahl per *Datei Dialog*; hierbei öffnet sich, in der Regel verbunden mit einer „Laden“-Schaltfläche, eine Ansicht der Ordnerstruktur, in der der Benutzer zum gewünschten Bild navigieren und dieses dann auswählen kann. Zum anderen die Auswahl per *Drag and Drop*; hierbei zieht der Benutzer einfach das gewünschte Bild aus einer nebenher geöffneten Ordnerstruktur oder vom Desktop in einen bestimmten Bereich im GUI der Anwendung. Letzteres ist häufig intuitiver und wird deshalb in dieser Arbeit verwendet.

Auf der Startseite der Anwendung hat der Benutzer die Möglichkeit, ein Bild in einen hervorgehobenen Bereich zu ziehen, um es auszuwählen, siehe Abbildung 4.2.

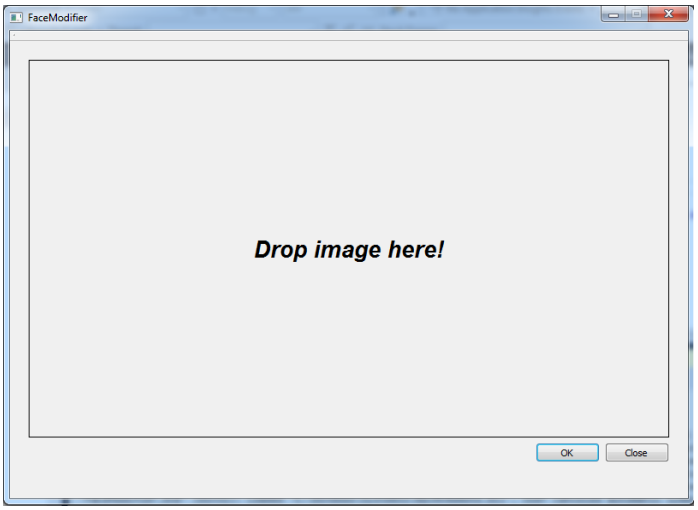

Abbildung 4.2: Startseite der Anwendung mit Drag and Drop-Bereich für die Bildauswahl

4.1.2 Auswahlmaske

Die Auswahlmaske dient der Vormarkierung der Merkmalsbereiche durch den Benutzer. Die relevanten Merkmalsbereiche werden durch die Analyse vorgegeben und sind die *Augen* beziehungsweise deren *Iris*, die *Augenbrauen*, die *Nase*, der *Mund*, das *Kinn* und der Ausschnitt des Bildes, in dem sich das Gesicht befindet. Bis auf das Kinn lassen sich alle diese Merkmale durch eine rechteckige Auswahl einfach und sinnvoll markieren. Auch die Iris als Kreis lässt sich mit einem *bounding rect*, einem minimal umgebenden Rechteck, beschreiben.

Für die Analyse wird eine Summe von Punkten erwartet, die die Kinnkurve beschreibt. Für eine Vormarkierung ist deshalb die Verwendung eines *offenen Polygonzuges*, eine Folge von miteinander verbundenen Punkten, sinnvoll. Mit diesen Punkten kann ein *Polynom* als Ausgang für die Analyse berechnet werden, wobei sich fünf Vormarkierungspunkte als ausreichend erwiesen haben.

In diesem Schritt, der Vormarkierung (Seite zwei der Anwendung), wird das ausgewählte Bild im Hintergrund dargestellt und eine, vom Bildinhalt unabhängige, vordefinierte Maske, bestehend aus einem Rechteck für jedes Merkmal sowie das Gesicht und dem Polygonzug für das Kinn, platziert. Der Benutzer kann diese dann, ähnlich den Auswahlwerkzeugen in Bildbearbeitungsprogrammen, verschieben und skalieren. Um zu gewährleisten, dass die Zuordnung von Rechteck zu Merkmal nicht durcheinander gerät, gibt es ein Kombinationsfeld, in dem der Benutzer ein Merkmal auswählen kann, dass er verändern möchte. Der entsprechende, aktive Auswahlbereich wird farblich hervorgehoben. Die Rechtecke lassen sich, mit Ausnahme von denen für die Iriden, an den Eckpunkten und an den Seitenkanten anfassen und skalieren. Die Iriden sind kreisförmig, weshalb hier nur eine gleichmäßige Skalierung über die Eckpunkte sinnvoll ist, um die quadratische Grundform beizubehalten, und die Punkte der Kinnlinie lassen sich nur verschieben.

Die Merkmale im Bild sind unterschiedlich groß, die Augen sind beispielsweise deutlich kleiner als die Kinnlinie, und bei einigen Merkmalen ist es notwendig, den entsprechenden Bildausschnitt zu vergrößern, damit eine gute Markierung möglich ist. Dafür wurde eine *Zoom*-Funktionalität implementiert, verbunden mit einer *Pan-* beziehungsweise *Schwenk*-Funktionalität, mittels welcher der Benutzer das Bild verschieben kann, um einen anderen Bildausschnitt zu erreichen, wenn er hineingezoomt hat. Der Zoom erfolgt dabei stufenlos.

Die Vormarkierung erfolgt mit der Maus in einem QGraphicsView und der damit verbundenen QGraphicsScene. Mit der linken Maustaste kann der Benutzer den aktiven Auswahlbereich skalieren, wenn er sich mit der Maus an einer der Seitenkanten befindet. Dabei wird der Mauszeiger zu einem Doppelpfeil in die entsprechende Richtung. Befindet sich der Mauszeiger im inneren Bereich des aktiven Auswahlbereiches beziehungsweise über einem Punkt der Kinnlinie, kann der Benutzer den Bereich

oder Punkt, ebenfalls mit der linken Maustaste, verschieben. Der Mauszeiger wird hier zu einem gekreuzten Doppelpfeil. In beiden Fällen drückt der Benutzer die linke Maustaste, bewegt dann bei gehaltener Taste die Maus und lässt sie an der gewünschten Stelle los (*Click and Drag*). Zoom und Panning werden mit dem Mausrad gesteuert. Wenn der Benutzer hoch- oder runterscrollt, zoomt er hinein oder heraus. Das Verschieben des Bildes funktioniert wie das Verschieben eines Auswahlbereiches per Click and Drag, jedoch mit dem Mausrad als Taste.

Für eine Erklärung der Steuerung und ein Bild mit einer beispielhaften Markierung steht dem Benutzer eine „Hilfe“-Schaltfläche zur Verfügung.

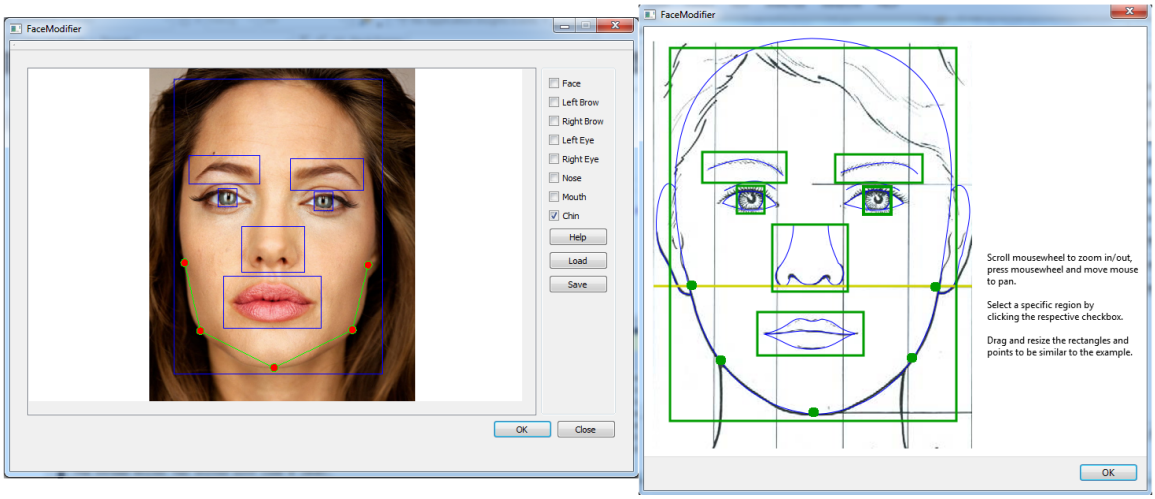

Abbildung 4.3: Auswahlmaske der Anwendung (links) und Hilfefenster (rechts)

Als letzte Funktionalität hat der Benutzer die Möglichkeit, seine aktuelle Vormarkierung zu diesem Bild abzuspeichern oder eine Vormarkierung zu laden. Dabei wird im Ausgabeordner eine Textdatei angelegt oder diese geladen. Dies geschieht ohne Dateidialog, der Benutzer muss also die Datei per Hand tauschen und den vorgegebenen Namen berücksichtigen. Ein nur einmaliges Verwenden dieser Funktion ist etwas umständlicher als das Verwenden eines Dateidialoges, weil der Benutzer die Datei so manuell sichern muss; bei wiederholter Verwendung des selben Bildes, etwa um das Ergebnis zu optimieren, wird ohne den Dateidialog jedoch etwas Bearbeitungszeit eingespart.

4.1.3 Interaktion mit dem Benutzer

Die Interaktion mit dem Benutzer erfolgt an verschiedenen Stellen in der Anwendung. Einen großen Teil davon macht die in 4.1.2 beschriebene Auswahlmaske und einen kleineren Teil die in 4.1.1 beschriebene Bildauswahl aus. Daneben gibt es eine

Hauptdialog, der den Benutzer Seite für Seite durch die Anwendung führt. Dies erfolgt durch Schaltflächen, mit denen der Benutzer weiter gelangt oder mit denen er die Anwendung beenden kann beziehungsweise, auf der letzten Seite, wenn die Anwendung fertig durchgelaufen ist, zurück zum Anfang gelangt und mit einem anderen oder mit dem selben Bild erneut beginnen kann. Soweit nötig, sind diese Schaltflächen mit Überprüfungen verbunden, zum Beispiel, ob die Vormarkierung des Benutzers grundsätzlich sinnvoll ist, dass etwa alle Vormarkierungen für die Merkmale innerhalb der Vormarkierung des Gesichtes liegen oder dass der Benutzer ein Bild ausgewählt hat.

Informations- und Fehlermeldungen erfolgen in einem kleinen, sich öffnenden Fenster, einem *Popup*. Dieses erfordert entweder eine einfache Bestätigung oder, im Fehlerfall, die Entscheidung, wie der Benutzer weiter vorgehen will. Hier kann er entweder die Anwendung beenden oder in einen Schritt zurückgehen, in welchem er versuchen kann, den Fehler zu beheben. Dies kann etwa durch die Veränderung der Vormarkierung geschehen, wenn die aktuelle Auswahl zu nicht gefundenen Merkmalen geführt hat.

Nach der Vormarkierung durch den Benutzer erfolgen die restlichen, rein programmatischen Schritte der Anwendung. Damit der Benutzer einen Überblick über deren Fortschritt hat, befindet sich auf der dritten Seite ein Fortschrittsbalken. So ist für den Benutzer auch erkennbar, wann die Anwendung durchgelaufen ist. Hier erfolgt dazu noch der Hinweis, dass der Benutzer das Ergebnis absichern muss, weil es automatisch in einem festgelegten Ausgabeordner angelegt wird und bei einem erneuten Durchlauf überschrieben werden würde.

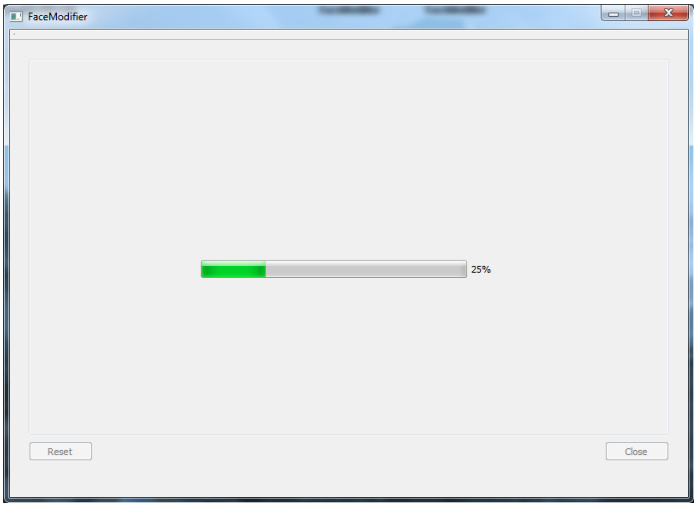

Abbildung 4.4: Letzte Seite der Anwendung mit Fortschrittsanzeige

4.2 Generisches 3D-Gesichtsmodell

Wie in 3.2 erläutert, soll für die Vereinfachung des Modellierungsprozesses ein generisches 3D-Gesichtsmodell verwendet werden, welches modifiziert wird. Dieses generische Modell muss dabei möglichst einen Durchschnitt in Bezug auf Gesichtsmerkmale bilden, damit der Unterschied zum Eingabebild immer recht gering ist. Das ist möglich, weil, trotz aller Unterschiede, Gesichter grundsätzliche Ähnlichkeiten aufweisen. Für die Erstellung eines solchen generischen Modells ist die Mittlung zahlreicher Gesichter nötig. Waters (1987) hat dies in seiner Arbeit getan und für dieser Arbeit wurde die durch Hähnel (2008) leicht abgewandelte Form jenes Modells mit 415 Vertices, soweit möglich, reproduziert (siehe Abbildung 4.5). Die Änderungen von Hähnel bestehen dabei hauptsächlich in einer Symmetrisierung des Polygonnetzes.

Für die Modifikation werden Korrespondenzpunkte auf dem Modell benötigt, um ein Verhältnis von Bildvorlage zu 3D-Gesichtsmodell berechnen zu können. Diese Korrespondenzpunkte bilden den Ausgangspunkt für die Bildanalyse in Kreutz (2016). Die Modifikation des Modells erfolgt mit einer Verschiebefunktion von Blender, siehe hierzu 4.3.3, die das proportionale Mitverschieben von benachbarten Vertices ermöglicht, weshalb es nötig ist, die Punkte um die Korrespondenzpunkte einem Korrespondenzpunkt zuzuordnen. Dies erfolgt über *Vertexgruppen*, eine Funktionalität, die Blender bietet.

4.2.1 Merkmalskorrespondenzen

Die Korrespondenzpunkte auf dem 3D-Gesichtsmodell müssen die prägnanten Punkte des Gesichtes markieren, damit die Modifikation ein zufriedenstellendes Ergebnis liefern kann. Deshalb markieren die Punkte die wichtigen Merkmale des Gesichtes, den Zug der Augenbrauen, die Form der Augen, die Nasenspitze und die Nasenflügel, die Mundkontur und die Mundlinie sowie die Kinnlinie. Die insgesamt 53 Korrespondenzpunkte (siehe Abbildung 4.5 auf der nächsten Seite) und auch die im nächsten Abschnitt beschriebenen Vertexgruppen sind dabei, wie das Modell, symmetrisch.

4.2.2 Vertexgruppen

Nur die Merkmalskorrespondenzpunkte alleine zu verschieben, würde zu einem unrealistischen Ergebnis führen, weil so lediglich ein kleiner Teil der Vertices des Modells verschoben würde. Sollen die übrigen Punkte proportional mit verschoben werden, ist es nötig, sie einem bestimmten Korrespondenzpunkt zuzuordnen, damit sie nur in Verbindung mit diesem verschoben werden, andernfalls können Verzerrungen entstehen. Für die Bildung der Zusammenhänge wird deshalb der in 2.2.1 beschriebene Verlauf der Muskeln, soweit sinnvoll möglich, berücksichtigt, siehe Abbildung 4.6 auf der nächsten Seite.

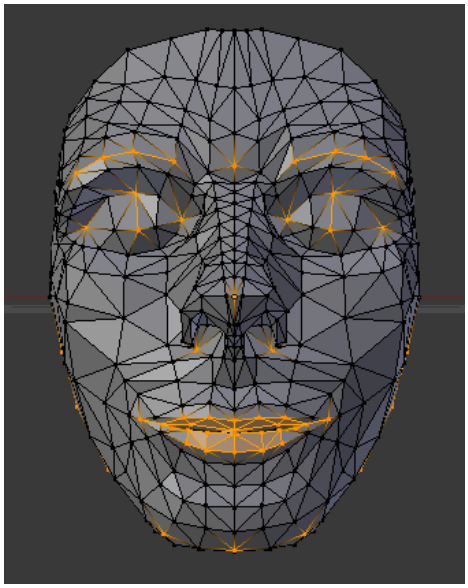

Abbildung 4.5: Generisches 3D-Modell mit Korrespondenzpunkten

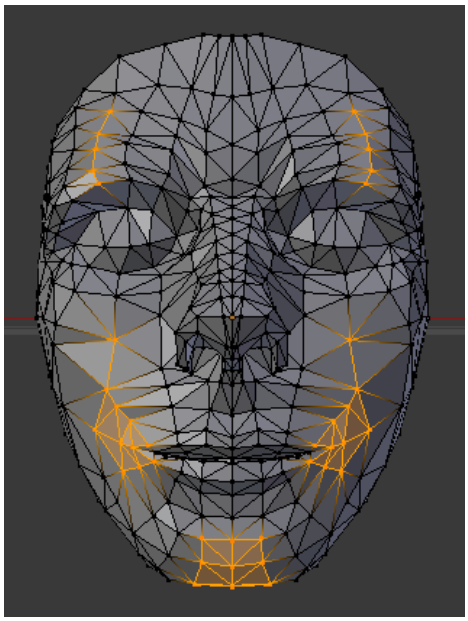

Abbildung 4.6: Verschiedene Vertexgruppen des Modells: Ein Punkt der Augenbrauen, der Mundwinkel, die Kinnspitze

4.3 3D-Gesichtsmodifikation

Im Folgenden wird beschrieben, wie die eigentliche Modifikation des generischen 3D-Gesichtsmodells durchgeführt wird. Die Basis für die Modifikation bilden die bereitgestellten Ergebnisse der Bildanalyse. Um Modell und Bild in Relation setzen zu können, müssen die Ergebnisse der Bildanalyse gewisse Anforderungen erfüllen, die in 4.3.1 näher ausgeführt werden. Anhand dieser Ergebnisse wird berechnet, wie das Modell modifiziert werden muss, um dem Gesicht im Bild zu entsprechen. Anschließend kann dann die eigentliche Modifikation des generischen 3D-Gesichtsmodells unter Verwendung bestimmter Funktionen von Blender erfolgen.

4.3.1 Anforderungen an die Analysedaten

Die Modifikation des Modells soll durch die Verschiebung der Merkmalskorrespondenzen und den damit verbunden Vertexgruppen erfolgen. Dafür ist es erforderlich, dass die Ergebnisse der Bildanalyse Daten für jeden einzelnen Korrespondenzpunkt liefern. Diese Datenwerte sind in isolierter Form jedoch nicht verwertbar, weil die Korrespondenz der einzelnen Punktpaare nicht besteht. Um diese Korrespondenz zu bilden, muss ein Korrespondenzpaar in Bild und Modell gewählt werden, dass als Referenz dient und über das sich die anderen Punkte des Modells und des Bildes ins Verhältnis setzen lassen. Als eine solche Referenz eignet sich ein Punktpaar, das in Bild und Modell nur wenig voneinander abweicht. Hier bietet sich die Nasenspitze an, die zentral im Gesicht und üblicherweise auf der Mittelachse liegt.

In Bild und Modell lassen sich alle Punkte in Relation zu diesem Referenzpunkt beziehungsweise als Richtungsvektor von diesem Referenzpunkt beschreiben und die nötige Verschiebung der Modellpunkte kann berechnet werden. Die Ergebnisse der Bildanalyse müssen also grundsätzlich in Relation zur Nasenspitze des Gesichtes im Bild vorliegen.

4.3.2 Berechnung der Verschiebung

Die Modifikation des Modells erfolgt in zwei Schritten, zunächst in einer *globalen* und anschließend in einer *lokalen* Anpassung. Beide erfolgen auf Basis des Vergleichs der Relationen des generischen 3D-Gesichtsmodells mit den Ergebnissen der Bildanalyse.

Die globale Anpassung passt die grundsätzlichen Proportionen des Modells an. Diese sind die Gesichtsform und das Verhältnis von einzelnen Gesichtsbereichen zueinander, etwa Stirn- zu Kieferbreite, sowie die Rotation des Modells. Rotationen nach links oder rechts werden jedoch nur leicht korrigiert. Stärkere Rotationen führen zu Verzerrungen des Modells, weil prinzipiell Frontalaufnahmen erwartet werden.

Die Faktoren und Winkel für Skalierung und Rotation werden in der folgenden Reihenfolge berechnet:

- Eine horizontale Skalierung *SH* des Modells auf Basis des horizontalen Abstands vom ersten zum letzten Kinnpunkt, der Kieferbreite. Der Faktor wird für nachfolgenden Berechnungen berücksichtigt.

- Eine vertikale Skalierung *SV* des ganzen Modells auf Basis des vertikalen Abstands vom mittleren Kinnpunkt, der Kinnspitze, zum inneren Punkt des linken Auges. Der Faktor wird für nachfolgenden Berechnungen berücksichtigt.

- Eine Skalierung *ST* in die Tiefe. Obwohl die räumliche Modifikation in der Tiefe nicht Ziel dieser Arbeit ist, erfolgt in 4.3.3 ein einfache Skalierung auf dieser Achse, weil hier ein akzeptabler Näherungswert durch den Mittelwert von SH und SV gebildet werden kann. So bleibt die grundsätzliche Tiefe des Gesichts erhalten und es wird nicht zu flach oder tief, abhängig davon, ob das Modell deutlich kleiner oder größer als das Gesicht im Bild ist.

- Eine vertikale Skalierung *SVU* der unteren Hälfte des Modells auf Basis des vertikalen Abstands von der Nasen- zur Kinnspitze.

- Eine vertikale Skalierung *SVO* der oberen Hälfte des Modells auf Basis des vertikalen Abstands von der Nasenspitze zum linken inneren Augenpunkt.

- Eine horizontale Skalierung *SHU* der unteren Kieferpartie auf Basis des horizontalen Abstands vom zweiten zum vorletzten Kinnpunkt.

- Eine Rotation *RV* um die vertikale Achse, also nach rechts oder links. Dafür wird der Arkustangens mit

  $$Gegenkathete = \frac{(linker Kinnpunkt + halbe Gesichtsbreite)}{2}$$

  und

  $$Ankathete = skalierte Nasentiefe$$

  gebildet, wobei die Nasentiefe des generischen 3D-Gesichtsmodells, abhängig von zwei Nasenpunkten im Modell, vordefiniert ist.

Eine horizontale Skalierung des oberen Gesichtsbereichs hat keine positiven Auswirkungen mit sich gebracht, weil die Breite in der Regel einheitlich ist und dieser Bereich bereits mit SH skaliert wird.

Die lokale Anpassung passt die Details des Modells an, die durch die Merkmalskorrespondenzpunkte repräsentiert werden. Für jeden Korrespondenzpunkt wird ein Verschiebungsvektor berechnet. Hierfür wird zunächst jeder Korrespondenzpunkt K_b

des Bildes im Modell-Raum rekonstruiert, indem die Richtungsvektoren, die als Ergebnis der Bildanalyse vorliegen, auf den Referenzpunkt R des 3D-Modells addiert werden. Anschließend wird für jeden Korrespondenzpunkt K_m des Modells der Verschiebungsvektor berechnet, indem jeweils der Richtungsvektor K_m zu K_b mit

$$K_b - K_m$$

gebildet wird.

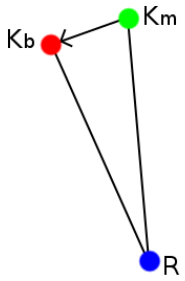

Abbildung 4.7: Schematische Darstellung für die Berechnung der Verschiebungsvektoren

4.3.3 Modifikation des generischen 3D-Gesichtsmodells

Zunächst erfolgt die globale Anpassung des Modells. Für die Skalierung wird jeweils das *scale*-Attribut des Modells selbst oder der einzelnen Vertices angesprochen:

bpy.data.objects['Kopf'].scale.x/y/z

Für die Rotation wird eine *rotate*-Funktion von Blender verwendet:

bpy.ops.transform.rotate(value, axis)

Die Parameter geben dabei den Winkel (*value*) und die Achsen (*axis*), um die rotiert werden soll, an.

Die in 4.3.2 berechneten Faktoren und der Rotationswinkel werden dabei wie folgt angewandt:

- *SH:* Auf alle Vertices des Modells, auf der horizontalen Raumachsen.
- *SV:* Auf alle Vertices des Modells, auf der vertikalen Raumachse.

- *ST:* Auf alle Vertices des Modells, auf der Tiefenachse.
- *SVU:* Auf alle Vertices des Modells von der Nasenspitze an abwärts, auf der vertikalen Raumachse.
- *SVO:* Auf alle Vertices des Modells oberhalb der Nasenspitze, auf der vertikalen Raumachse.
- *SHU:* Auf alle Vertices des Modells von der Mundlinie an abwärts, auf der horizontalen Raumachse.
- *RV:* Um die vertikale Raumachse.

Im Anschluss an die globale Modifikation werden die 3D-Vektoren der Vertices ausgelesen und diese werden dann für die in 4.3.2 beschriebene Berechnung der lokalen Verschiebung benutzt.

Für die Verschiebung der Korrespondenzpunkte des Modells und den über die Vertexgruppen zugeordneten, benachbarten Vertices wird eine *translate*-Funktion von Blender verwendet:

bpy.ops.transform.translate(value, constraint_axis,
constraint_orientation, mirror, proportional,
proportional_edit_falloff, proportional_size)

Die Parameter haben dabei folgende Bedeutung:

- *value:* Die Verschiebung als dreidimensionaler Richtungsvektor. In der üblichen Ansicht von Blender ist y dabei die Tiefe, für die Verschiebung hier werden deshalb x und z angesprochen und y bleibt unverändert.
- *constraint_axis:* Die Beschränkung der Verschiebung auf bestimmte Achsen, in diesem Fall auf x und z.
- *constraint_orientation:* Die Verschiebung kann in verschiedenen Zusammenhängen interpretiert werden, etwa *LOCAL*, was bedeutet, dass die *lokalen* Achsen des Modells als Bezugspunkt genutzt würden. Die Berechnung bezieht sich jedoch auf den Ursprung des Koordinatensystems, in dem das Modell erstellt wurde, den *globalen* Raum. Deswegen wird hier *GLOBAL* verwendet.
- *mirror:* Mit dieser Option könnte eine Verschiebung auf einer Seite des Modells automatisch auch entsprechend gespiegelt auf der anderen Seite des Modells angewandt. Dafür müssten jedoch gesichert sein, dass das Gesicht exakt symmetrisch ist. In der Regel ist dies jedoch nicht der Fall, weshalb diese Option nicht verwendet wird.

- *proportional:* Dieses Parameter entscheidet darüber, ob das proportionale Verschieben der benachbarten Punkte eines Merkmalspunktes genutzt wird oder nicht. Entsprechend der Zielsetzung dieser Arbeit wird sie genutzt, *ENABLED*. Einzige Ausnahme bildet die Mundlinie, die Innenkante der Lippen, weil diese Punkte jeweils alleine stehen, ohne Nachbarn.

- *proportional_edit_falloff:* Mit diesem Parameter wird spezifiziert, welcher Modus für das proportionale Verschieben verwendet wird. Die Verschiebung der Nachbarn erfolgt in einem bestimmten Radius um den Merkmalspunkt und der Modus beeinflusst, wie stark sich die Verschiebung des Merkmalspunktes auf die Nachbarn in zunehmendem Abstand auswirkt. Die verschiedenen Modi können der Abbildung 4.8 unten entnommen werden. Hier wird *SMOOTH* verwendet.

- *proportional_size:* Dieses Parameter gibt den Radius der proportionalen Verschiebung an. Als Standardwert wird hierfür der zweifache Abstand vom Merkmalspunkt zu dem am weitesten entfernten Nachbarn gewählt. Ausnahmen bilden der Punkt zwischen den Augenbrauen mit anderthalbfachem und die äußeren Brauenpunkte mit einfachem Abstand. Hier hat sich der zweifache Abstand als zu hoch erwiesen.

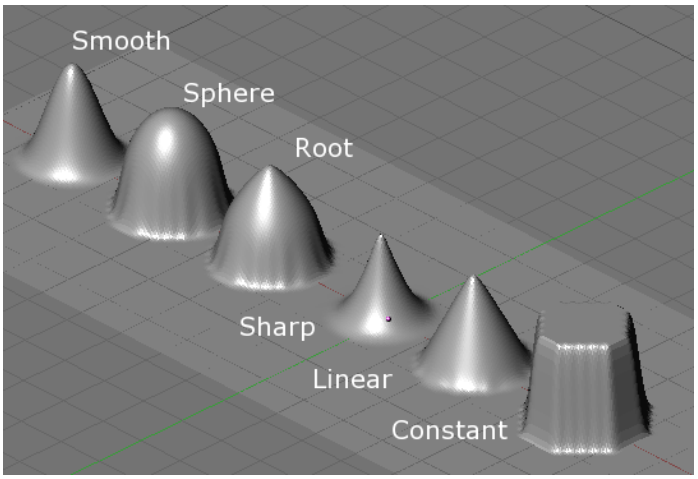

Abbildung 4.8: Mögliche Modi für proportioanl_edit_falloff, entnommen Soylent (2005)

Die proportionale Verschiebung erfolgt dabei Korrespondenzpunkt für Korrespondenzpunkt. Die Funktion berücksichtigt dabei von sich aus keine Vertexgruppen, sondern lediglich den ausgewählten Punkt und alle Punkte, die im Radius liegen. Damit keine Punkte mehrfach verschoben werden, werden jeweils alle Vertexgruppen, die nicht zur aktiven Gruppe gehören, ausgeblendet.

5 Ergebnisse

In diesem Kapitel wird die vorstehend beschriebene Anwendung in Bezug auf das Modellierungsergebnis getestet, um eine Grundlage für die Beantwortung der Fragestellung dieser Arbeit zu schaffen. Dazu wird ein Versuch durchgeführt, mit dessen Ergebnissen die Modellierung bewertet wird. Zunächst erfolgt die Beschreibung des Versuchs und anschließend die Aufführung der Ergebnisse.

5.1 Beschreibung des Versuchs

Um die Qualität der Modellierung beziehungsweise Modifikation aussagekräftig zu bewerten, ist es erforderlich, die Anwendung mehrfach mit Bildern von verschiedenen Personen durchlaufen zu lassen. Wie bereits geschrieben, erfolgt die Modifikation auf Basis der Ergebnisse der Bildanalyse. Es wird deshalb davon ausgegangen, dass die Modifikation nur zu einem zufriedenstellenden Ergebnis führt, wenn das Ausgangsbild von Pose und Mimik dem generischen 3D-Gesichtsmodell ähnelt und das Ergebnis der Bildanalyse ebenfalls zufriedenstellend ausfällt. *Zufriedenstellend* bedeutet dabei für diese Arbeit, dass das Gesicht der Person, deren Bild als Vorlage dient, wiedererkannt werden kann und keine offensichtlich falschen Deformierungen entstehen.

Der Versuch simuliert die Bedienung durch „normale“ Benutzer. In Kreutz (2016) erfolgt die Bewertung der Ergebnisse der Bildanalyse unter zwei Gesichtspunkten, der Eignung des Eingabematerials (A) und dem Einfluss der Benutzerauswahl (B). Die Eignung wird durch die Variation des Bildmaterials bei gleichartiger Vormarkierung getestet. Der Einfluss der Benutzerauswahl wird durch verschiedene Auswahlen im gleichen Bild untersucht. Für die Bewertung von (A) wird berücksichtigt, wie gut die gefundenen Punkte das entsprechende Merkmal und die durch das Modell vorgegebenen Punkte be Bei der Bewertung wird berücksichtigt, wie gut die gefundenen Punkte das entsprechende Merkmal und die durch das Modell vorgegebenen beschreiben. Für die Bewertung von (B) wird berücksichtigt, wie groß der Einfluss einer nicht vorgabengetreuen Vormarkierung durch den Benutzer, im Vergleich zu einer vorgabengetreuen Vormarkierung, auf das Ergebnis ist und, ob ein vorher schlechtes Ergebnis durch Abweichungen von den Vorgaben verbessert werden kann. *Vorgabengetreu* bezieht sich dabei auf die Auswahl, die dem Benutzer durch das Hilfebild gezeigt wird.

Für Versuch (A) wurden im Vorfeld zwei Gruppen gebildet, eine, bei der davon ausgegangen wird, dass die Bildanalyse zu einem zufriedenstellenden Ergebnis führt (zwölf

Bilder) und eine, bei der davon ausgegangen wird, dass dies, aus verschiedenen Gründen, nicht der Fall ist (zehn Bilder). Für die ausführliche Versuchsbeschreibung und die genauen Ergebnisse, siehe Kreutz (2016), Kapitel 3.

Für den Versuch in dieser Arbeit werden alle Bilder der als geeignet eingestuften Gruppe aus Versuch (A) betrachtet, die eine positive Bewertung erhalten haben. Der Anteil dieser an der Gruppe ist mit neun von zwölf groß genug, dass für diesen Versuch eine Testgruppe gebildet werden kann, die ausreichend Kandidaten beinhaltet, um den gewählten Ansatz der Modellierung im Rahmen der eingangs formulierten Frage zu bewerten.

Es ist grundsätzlich möglich, dass die Bildanalyse auch bei Bildern zu einem zufriedenstellenden Ergebnis führt, die für die Modellierung im Grunde als ungeeignet einzustufen sind, weil sie nicht den Vorgaben des Modells entsprechen, hauptsächlich aufgrund von Mimik oder Pose. Die Testgruppen von Kreutz (2016) enthalten allerdings kein derartiges Bild.

Außer der Reihe wird ein Bild untersucht, bei dem bereits die Bildanalyse zu keinem zufriedenstellenden Ergebnis geführt hat, weil das Gesicht stark zur Seite gedreht (fast im Profil) und dadurch nur teilweise sichtbar ist, die Pose also nicht zum Modell passt. Dieses Bild wird in 3.2 der Arbeit von Kreutz (2016) als *Frau gedreht* in der Kontrollgruppe aufgeführt.

5.2 Ergebnisse des Versuchs

Die Bewertung der Modifikation erfolgt unter zwei Gesichtspunkten, dem Vergleich mit dem Analyseergebnis und dem Vergleich mit dem Ausgangsbild. In beiden Fällen wird eine visuelle und zugleich in Teilen auch subjektive Bewertung vorgenommen. Zu beachten ist, dass das Modell als Ergebnis der Modellierung nicht texturiert ist, was gerade den Vergleich mit dem Ausgangsbild erschwert. Für eine bessere Bewertung und zur Veranschaulichung dieser wurde deshalb im Nachhinein von Hand das Ausgangsbild auf das Modell projiziert. Dies führt gleichzeitig jedoch auch zu einer leichten Ungenauigkeit, weil davon auszugehen ist, dass die Texturierung nicht exakt erfolgt ist. Eine automatische Texturierung ist im Übrigen ein Punkt, der als Erweiterungsmöglichkeit der Anwendung im Fazit aufgeführt wird.

Zunächst erfolgt der Vergleich der Modifikation mit dem Analyseergebnis. Hierfür wird die Position der Korrespondenzpunkte des Modells nach der Modifikation mit den entsprechenden Punkten des Ergebnisses der Bildanalyse verglichen und bewertet, wie hoch die Übereinstimmung ist. Dies ist prinzipiell unabhängig davon, ob das Gesicht gut getroffen wurde oder nicht.
Der Grad der Übereinstimmung wird bewertet mit *keinerlei* (-2), *geringe* (-1), *teilweise* (0), *überwiegende* (1) und *exakte* (2). Die einzelnen Merkmale (Brauen, Augen,

Nase, Mund, Kinn) werden dabei einmal getrennt bewertet und einmal als Summe der Merkmale. Die schlechteste Gesamtbewertung ist entsprechend eine *-10*, die beste eine *10*.

Das Ergebnis dieses Versuchs ist der folgenden Tabelle 5.1 zu entnehmen. Die Bezeichnung der Bilder entspricht dabei denen aus der Arbeit von Kreutz (2016).

Bild	Analysebewertung	Augenbrauen	Augen	Nase	Mund	Kinn	Summe
Barbie	1	1	1	1	1	0	4
Frau1	2	1	1	1	1	0	4
Frau2_1	2	1	1	0	-1	0	1
Frau2_2	2	1	1	-1	1	0	2
Frau3	1	1	1	0	1	-1	2
Frau4	2	1	1	1	1	0	4
Mädchen1	2	1	1	0	1	-1	2
Mann3	1	1	1	1	1	0	4
Mann4	1	1	1	0	1	1	4
Durchschnitt		1	1	0,33	0,78	-0,11	3

Positive Bewertungen	31	69%
Neutrale Bewertungen	10	22%
Negative Bewertungen	4	9%

Tabelle 5.1: Ergebnisse des Vergleichs von *Analyseergbnis* und *Modell*

Grundsätzlich ist zu erkennen, dass bei keinem Bild im Vergleich von Bildanalyse und Modellierung eine exakte Übereinstimmung bei einem oder mehreren Merkmalen besteht, allerdings auch keine von (-2). Hierbei ist jedoch die Texturierung von Hand zu berücksichtigen, die den Eindruck leicht beeinflusst. Eine (1) als Bewertung kann also als sehr positiv betrachtet werden. Des Weiteren lassen sich folgende Erkenntnisse gewinnen:

- Die Augenbrauen, die Augen und der Mund haben, mit einer Ausnahme bei jedem Testbild für den Vergleich eine positive Bewertung (1) erhalten.
- Jedes Bild, bei dem ein Merkmal eine negative Bewertung (-1) erhalten hat, hat auch bei mindestens einem anderen Merkmal maximal eine neutrale Bewertung (0) erhalten.
- Das Kinn hat im Vergleich am schlechtesten abgeschnitten und hat als einziges Merkmal einen negativen Mittelwert über alle Bilder.
- Insgesamt sind 69% der Einzelbewertungen positiv und nur 9% negativ.
- Die durchschnittliche Bewertung für die Bilder als Ganzes liegt bei 3.

Im Folgenden wird das Modell mit dem Ausgangsbild verglichen und die Ähnlichkeit bewertet. Die Bewertungsskala entspricht der Skala aus dem ersten Versuch. Das Ergebnis kann der Tabelle 5.2 unten entnommen werden.

Ausgangsbild	Übereinstimmung
Barbie	-1
Frau1	1
Frau2_1	1
Frau2_2	-1
Frau3	-2
Frau4	1
Mädchen1	-2
Mann3	0
Mann4	-1
Durchschnitt	-0,44

Tabelle 5.2: Ergebnisse des Vergleichs von *Ausgangsbild* und *Modell*

Zu erkennen ist, dass die Ähnlichkeit der Modelle mit dem Ausgangsbild insgesamt eher mäßig ausgeprägt ist. Die durchschnittliche Bewertung liegt etwa bei -0,4 und es gibt Bewertungen von (-2), aber keine von (2). Auch hier hat die Textur einen Einfluss auf das Empfinden, dieser ist jedoch nicht ausschlaggebend. Dafür sind andere gravierendere Schwächen auffällig; so ist beispielsweise festzustellen, dass die Modellierung des Kinns und der Augenbrauen in bestimmten Fällen entweder nicht gut oder überhaupt nicht funktioniert. Beim Kinn etwa ist die Modellierung von eher kantigen Formen nicht zufriedenstellend und die Augenbrauen stehen häufig stark nach außen über beziehungsweise die umliegenden Partien nicht in adäquatem Maße. Dadurch entstehen übermäßig wulstige Augenbrauen. Auf Gründe hierfür wird in 6.2 eingegangen.
Die folgende Abbildung 5.1 zeigt einige der Bilder (*Frau1*, *Frau4*, *Mann4*).

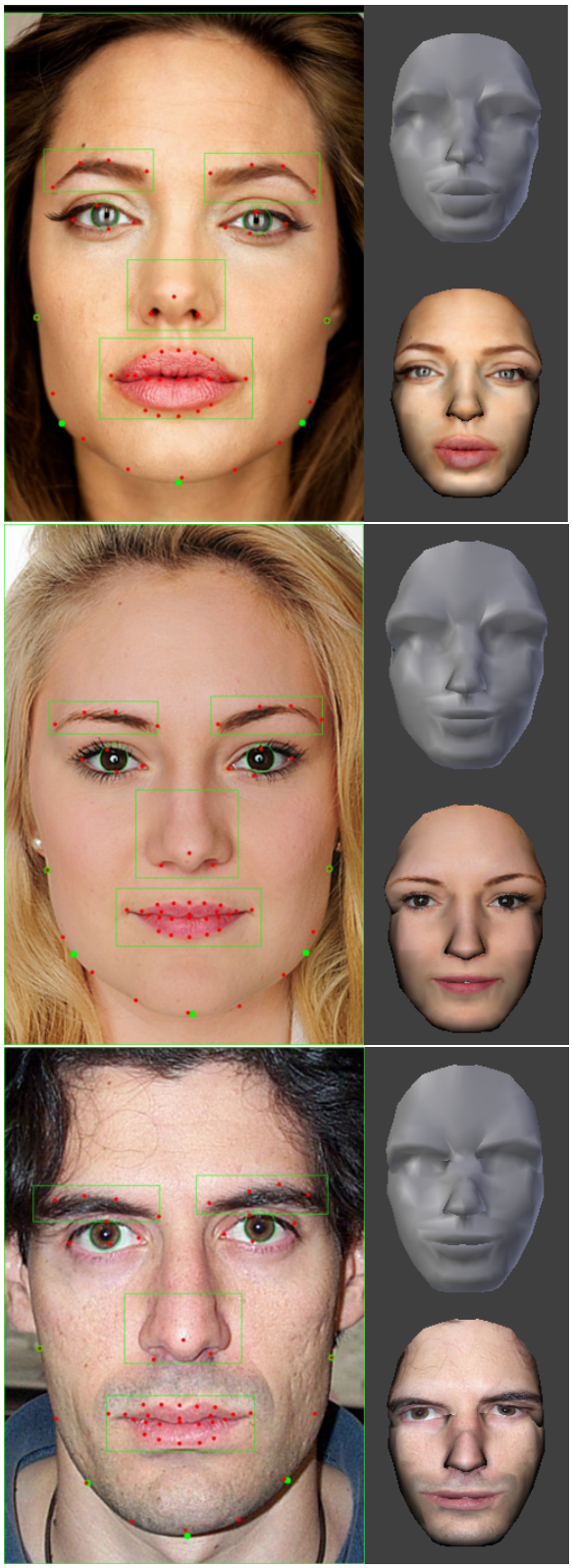

Abbildung 5.1: Ergebnisse der Anwendung: jeweils Ergebnis der Analyse (links), Modell ohne Textur (oben), Modell mit Textur (unten)

Nun folgt das in 5.1 angekündigte Beispiel *Frau gedreht*, das kein repräsentatives Ergebnis dieser Anwendung darstellt, weil es nicht der Vorgabe entspricht, ein Frontalbild zu sein.

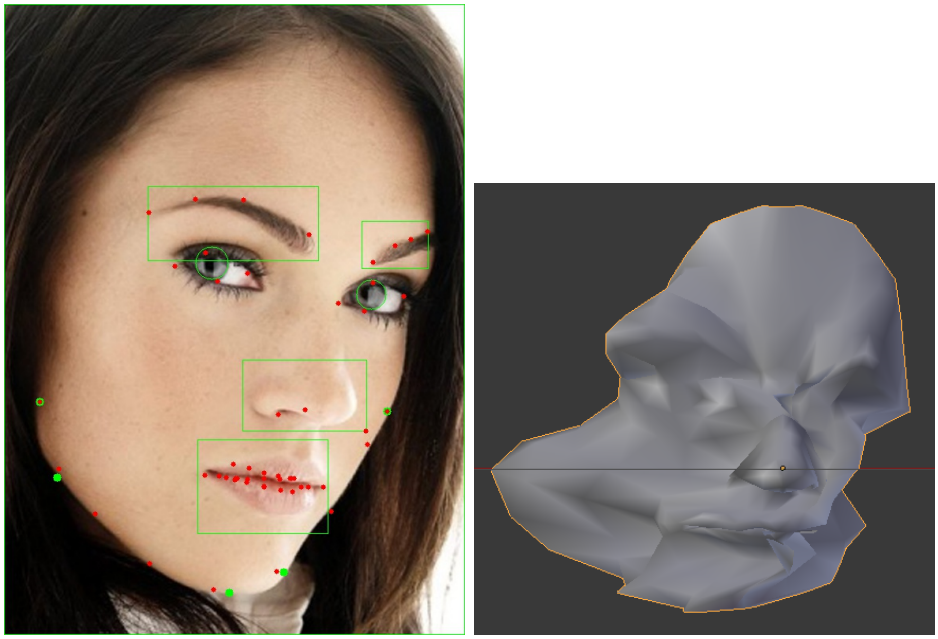

Abbildung 5.2: Ergebnis der Bildanalyse (links), Ergebnis der Modellierung (rechts)

Wie zu erkennen ist, wurde im Rahmen der Bildanalyse kaum eines der Merkmale korrekt erkannt, weil sie, aufgrund der Kopfhaltung, überwiegend nur teilweise auf dem Bild zu sehen sind. Auch die Vormarkierung ist nur bedingt möglich gewesen, gerade die der linken Kinnkontur. Die gut sichtbare Kinnkante entspricht hier nicht der Kante, die in der Frontansicht zu sehen wäre und die eigentlich markiert werden muss. Diese Kontur ist prinzipiell auch in diesem Bild, ist hier jedoch Kontrast zum Hintergrund nicht gut sichtbar. Diese falsche, aber für die Funktionsweise der Bildanalyse notwendige Auswahl führt zu dem oben gezeigten, komplett deformiertem Modell. Ohne diese Auswahl kann die Anwendung nicht durchlaufen, weil für das Kinn überhaupt keine Daten gefunden werden können.

6 Fazit

Die Entwicklung und die Verwendung der Anwendung beziehungsweise der in dieser Arbeit behandelten Teilbereiche GUI und Modifikation des generischen 3D-Gesichtsmodells haben gezeigt, dass der gewählte Verfahrensansatz, die verwendeten Ressourcen und deren Implementierung sowohl Stärken als auch Schwächen aufweisen.

In Bezug auf die in dieser Arbeit behandelte Frage, inwiefern die vollautomatische Nutzung von komplexen Funktionen einer schon bestehenden Grafiksuite sowohl den Modellierungsprozess vereinfachen als auch ein zufriedenstellendes Ergebnis liefern kann, lässt sich sagen, dass der Modellierungsprozess deutlich vereinfacht ist, aber, unter Berücksichtigung der in 5.2 aufgeführten Beobachtungen, das Ergebnis mit dem gewählten Ansatz *nicht zufriedenstellend* ist. Dafür gibt es verschiedene Gründe, die in 6.2 erläutert werden, ebenso Verbesserungsmöglichkeiten.

Zunächst folgen nun je ein Fazit zum GUI und zur Modellierung. Anschließend werden Erweiterungsmöglichkeiten für die Anwendung sowie einige Schlussbemerkungen.

6.1 Grafische Benutzeroberfläche

Die Verwendung von Qt für die Programmierung des GUI hat sich als gute Wahl erwiesen. Die Bibliothek bietet viele Möglichkeiten, gewünschte Funktionalitäten zu realisieren. Das resultierende GUI hat sich als funktional erwiesen, bietet aber zugleich verschiedenen Stellen Verbesserungsmöglichkeiten, sowohl technischer als auch praktischer beziehungsweise ergonomischer Natur.

- *Bildauswahl:* Die Bildauswahl per Drag and Drop ist, so wie gewünscht, einfach zu handhaben. Die Implementierung einer weiteren Bildauswahl für ein Profilbild lässt sich ohne größere Probleme umsetzen, wenn ein solches in den Analyse- und Modifikationsprozess aufgenommen wird, dazu mehr in 6.2.

- *Auswahlmaske:* Die Auswahlmaske ist funktional, ist jedoch noch etwas umständlich, weil für jeden Wechsel auf ein anderes Merkmal die entsprechende Checkbox angeklickt werden muss. Dies erhöht den Zeitaufwand zwar nicht deutlich, die Funktionsweise könnte jedoch angenehmer gestaltet werden. Bei der Implementierung eines Profilbildes muss dieser Bereich in jedem Fall überdacht werden, weil die Anforderungen, wie sie in 3.1 formuliert sind, sich, zumindest in gewissem Maße, ändern. Eine technische Schwäche hat sich während

der Durchführung des Versuchs bei besonders großen Bildern gezeigt, hier ist die gewählte Implementierung suboptimal mit der Folge, dass das Verschieben des Bildes sehr ruckelig abläuft. Dies ist keine Schwäche von Qt, sondern der Implementierung, und muss bei einer Weiterentwicklung bedacht werden, ebenso eine dynamische Anpassung der Linienstärke der Auswahlbereiche und der Griffpunkte für die Skalierung, abhängig vom Zoomfaktor. Auch hat der Zoomfaktor einen Einfluss auf die Qualität der Bilddarstellung. Diese leidet, wenn hineingezoomt wird, stärker als dies beispielsweise in der Bildergalerie von Windows der Fall ist. Hier ist eine bessere Berechnung bei der Zoomfunktion sinnvoll und notwendig.

- *Design:* Das Design der Anwendung ist rein funktional. Es wurde kein spezielles Augenmerk auf ästhetische Aspekte wie die Farbgestaltung gelegt. Im Zuge einer Weiterentwicklung ist dieser Aspekt zu beachten.

6.2 Modifikation des 3D-Modells

Der Schritt der Modellierung des Modells ist durch die Vollautomatisierung deutlich vereinfacht, die Ergebnisse, die so erzielt werden können, sind jedoch nicht zufriedenstellend. Dies ist zum Teil durch die gewählte Translate-Funktion selbst begründet, zum Teil aber auch durch die Kombination von Translate-Funktion, generischem 3D-Gesichtsmodell und Bildanalyse.

Mit der Translate-Funktion von Blender müssen die Vertices einzeln beziehungsweise in Gruppen angesprochen werden, damit kein Vertex mehrfach und damit zu viel bewegt wird. So lassen sich die Ergebnisse der Bildanalyse auf das Modell anwenden, die jeweiligen Verschiebungen stehen jedoch in keinerlei Verhältnis zu einander. Bei der verhältnismäßig geringen Zahl an Korrespondenzpunkten und den deshalb großen Vertexgruppen entstehen leicht „Bruchstellen“ im Modell. Auch bereiten stärkere Abweichungen des Gesichts vom generischen Modell Probleme. Dies ist besonders beim Kinn festzustellen, eher kantige oder schmale Kinnlinien werden schlecht nachgebildet.

Hier gibt es verschiedene Verbesserungsmöglichkeiten:

- *Zahl der Korrespondenzpunkte/Vertexgruppen:* Die Zahl der Korrespondenzpunkte kann erhöht werden, wodurch die Zahl der Vertexgruppen kleiner wird. So wird die „Reichweite“ der einzelnen Korrespondenzpunkte reduziert.

- *Unterschiedliche generische Modelle:* Der Grund für die Verwendung eines generischen Modells ist, die Abweichung zu verschiedenen Gesichtsformen zu reduzieren, einen Mittelwert zu bilden. Dies führt jedoch zu Problemen, wenn die Varianz der Gesichtsformen zu groß ist. Die Gesichter von Kindern und

Erwachsenen unterscheiden sich beispielsweise sehr deutlich. Hier besteht die Möglichkeit, Gesichtsformen in Gruppen einzuteilen und für jede ein eigenes generisches Modell zu entwickeln und der Benutzer gibt an, welches verwendet werden soll.

- *Radius und proportional_edit_falloff:* Die Berechnung des Radius' kann für die einzelnen Korrespondenzpunkte stärker individualisiert werden und der proportional_edit_falloff-Modus kann stärker variiert werden (siehe 4.3.3). Der Kreis als Form kann sich allerdings als problematisch erweisen. In einigen Fällen würde bei Punkten, die etwa gleich weit vom Korrespondenzpunkt der Vertexgruppe entfernt liegen, ein unterschiedlicher Verschiebungswert zu einem besseren Ergebnis führen, mit einem Radius ist dies aber nicht möglich.

- *Verwendung einer Matrix:* Eine grundsätzliche Alternative zur Translate-Funktion ist das Verwenden einer Matrix. Hierbei fließen die Ergebnisse der Bildanalyse in einer Matrix zusammen, wodurch die Interpolation entsteht, die bei der Translate-Funktion fehlt. Eine solche Matrix macht Vertexgruppen überflüssig und benötigt lediglich Korrespondenzpunkte, weil sie in einer Berechnung auf das gesamte Modell angewendet wird, die aktuelle Zahl der Korrespondenzpunkte wäre dafür in etwa ausreichend. Dies ist mit Blender möglich, aber nur programmatisch, also nicht in einer Art und Weise, die auch ein Benutzer von Hand in Blender durchführen kann.

Es ist also eindeutig festzustellen, dass beim aktuellen Ansatz entweder das generische Modell verändert werden muss und damit einhergehend auch die Bildanalyse, oder aber das Modellierungsverfahren selbst.

Natürlich muss berücksichtigt werden, dass in der aktuellen Version nicht auf Tiefeninformationen zugegriffen werden kann und die Implementierung einer zusätzlichen Profilansicht der nächste Erweiterungsschritt ist. Man kann aber davon ausgehen, dass dadurch das aktuelle Problem nicht grundsätzlich behoben wird.

6.3 Erweiterungsmöglichkeiten

Neben den bisher genannten Erweiterungs- und Verbesserungsmöglichkeiten gibt es auch noch einige weitere, wobei diese funktionaler und/oder ergonomischer Natur sind:

- *Texturierung:* Aktuell wird das Modell nicht automatisch texturiert. Dies muss der Benutzer im Nachhinein von Hand durchführen. Eine vollautomatische Texturierung des Modells durch Projizierung der Bilder auf das Modell ist möglich, macht jedoch auch erst in Verbindung mit einem Profilbild Sinn, um dunkle

Bereiche an Stellen, die nicht auf dem Frontalbild zu sehen sind, zu vermeiden, beispielsweise die Rückseite der Nasenflügel.

- *Speichern/Laden:* Aktuell kann der Benutzer ein Bild durch Drag und Drop auswählen und eine Vormarkierung speichern oder laden, wobei dies an einem festgelegten Ort und mit festgelegtem Dateinamen geschieht. Dies kann um Menüpunkte erweitert werden, mit denen der Benutzer ein Bild auch per Dateidialog auswählen und eine Vormarkierung per Dateidialog an beziehungsweise von einer beliebigen Stelle und mit beliebigen Namen speichern oder laden kann.

- *Individualisierung:* Aktuell bekommt der Benutzer bei jedem Durchlauf Hinweise angezeigt. Hier kann eine „Nicht erneut anzeigen“-Funktion implementiert werden.

- *Nachkorrektur des Analyseergebnisses:* Aktuell werden die Ergebnisse der Bildanalyse direkt weiterverwendet. Stattdessen besteht die Möglichkeit, diese dem Benutzer zu zeigen, damit dieser wenigstens klar erkennbare Ausreißer korrigieren kann.

- *Besonderheiten im Gesicht:* Derzeit wird davon ausgegangen, dass das Gesicht zumindest nahezu symmetrisch ist und keine „Anomalien“ aufweist, zum Beispiel Narben. Um solche bei der Bildanalyse und Modellierung berücksichtigen zu können, kann man dem Benutzer eine Möglichkeit geben, diese gesondert zu markieren.

6.4 Schlussbemerkung

Abschließend bleibt zu sagen, dass die Tücke des Themas im Detail liegt. Die Frage, ob es möglich ist, einen rein manuellen Modellierungsprozess zu vereinfachen, hat sich gar nicht erst gestellt, vielmehr hat sich das „*Wie*“ als der entscheidende Punkt herausgestellt. Anwendungen wie das in der Einleitung erwähnte Faceworx gehen einen Weg, beim dem eine Bildanalyse kaum bis gar nicht für den Modellierungsprozess erforderlich ist, dafür ist der Bedienungsaufwand für den Benutzer jedoch relativ hoch.

Bei der im Rahmen dieser Arbeit und der Arbeit von Kreutz (2016) entwickelten Anwendung bleibt der Bedienungsaufwand für den Benutzer augenscheinlich gering. Um ein Ergebnis zu erzielen, das zumindest annehmbar ist, muss der Benutzer unter Umständen einige verschiedene Vormarkierungen ausprobieren.

Der gewählte Ansatz in der aktuellen Form führt eindeutig noch nicht zu einem Verfahren, welches sich als verlässlich genug bezeichnen lässt. Es sollte jedoch möglich sein, das Verfahren zu einem Punkt weiterzuentwickeln, an dem zuverlässig ein gutes

Ergebnis erreicht wird, welches dem „normalen“ Benutzer weiterhin einen relativ geringen Bedienungsaufwand abverlangt.

Abbildungsverzeichnis

Tabellenverzeichnis

Literaturverzeichnis

Altmann, Markus: *About Nonuniform Rational B-Splines - NURBS* http://web.cs.wpi.edu/~matt/courses/cs563/talks/nurbs.html, Letzter Zugriff: 11. 08. 2016

Blender Foundation: *Python API Overview*, https://www.blender.org/api/blender_python_api_2_77_1/info_overview.html, Letzter Zugriff: 11. 08. 2016

Blender Foundation: *Transform Operators*, https://www.blender.org/api/blender_python_api_2_57_release/bpy.ops.transform.html, Letzter Zugriff: 11. 08. 2016

Engelfried, Lars: *The Function Pointer Tutorials* http://www.newty.de/fpt/callback.html, 2011, Letzter Zugriff: 07. 07. 2016

Facebook: *DeepFace: Closing the Gap to Human-Level Performance in Face Verification*, https://research.facebook.com/publications/deepface-closing-the-gap-to-human-level-performance-in-face-verification/, 2014, Letzter Zugriff: 11. 08. 2016

Fisher, Perkins, Walker, Wolfart: *Image Synthesis*, http://homepages.inf.ed.ac.uk/rbf/HIPR2/snthops.htm, 2003, Letzter Zugriff: 11. 08. 2016

Fraß, Christian: *Grundlagen der 3D-Modellierung*, https://www.inf.tu-dresden.de/content/institutes/smt/cg/teaching/seminars/ProseminarSS09/09%20Arbeiten/Christian%20Frasz/pscg_less.pdf, 2009, Letzter Zugriff: 11. 08. 2016

Geis, Thomas: *text*, http://www.procontext.com/aktuelles/2006/08/die-neue-din-en-iso-9241-110.html, 2006, Letzter Zugriff: 07. 07. 2016

Gonzales, Dave: *How Terminator: Genisys made a „synthespian“ of young Arnold Schwarzenegger*, http://www.geek.com/news/how-terminator-genisys-made-a-synthespian-of-young-arnold-schwarzenegger-1626902/, 2015, Letzter Zugriff: 11. 08. 2016

Hähnel, Michael: *Modellbasierte posen- und mimikinvariante Gesichtserkennung*, 1. Aufl., 2008

International Organization for Standardization: *Ergonomics of human-system interaction*, http://www.iso.org/iso/home/store/catalogue_tc/catalogue_detail.htm?csnumber=38009, 1996, Letzter Zugriff: 11. 08. 2016

ITWissen: *GUI (Graphical User Interface)*, http://www.itwissen.info/definition/lexikon/graphical-user-interface-GUI-Grafische-Benutzeroberflaeche.html, Letzter Zugriff: 11. 08. 2016

Kreutz, Evelyn: *3D-Gesichtsmodellierung mit bilddatenbasierter Modifikation eines generischen Gesichtsmodells - Gesichtsbildanalyse und Bereitstellung der Daten für die Modifikation*, 1. Aufl., BA 2016

PhiPoint Solutions: *The Human Face and the Golden Ratio*, http://www.goldennumber.net/face/, Letzter Zugriff: 11. 08. 2016

Rosver: *Blender 3D: Noob to Pro/Box Modeling*, https://en.wikibooks.org/wiki/Blender_3D:_Noob_to_Pro/Box_Modeling, 2009, Letzter Zugriff: 11. 08. 2016

Schallehn, Eike: *Geometrische Modellierung*, http://wwwiti.cs.uni-magdeburg.de/iti_db/lehre/gif/gif_18.pdf, 2008, Letzter Zugriff: 11. 08. 2016

Shene, Dr. Ching-Kuang: *Mesh Basics*, http://www.cs.mtu.edu/~shene/COURSES/cs3621/SLIDES/Mesh.pdf, 2010, Letzter Zugriff: 11. 08. 2016

Shene, Dr. Chin-Kuang: *Constructive Solid Geometry*, http://www.cs.mtu.edu/~shene/COURSES/cs3621/NOTES/model/csg.html, Letzter Zugriff: 11. 08. 2016

Shepard, Mike: *Interactive Storytelling - Narrative Techniques and Methods in Video Games*, http://scalar.usc.edu/works/interactive-storytelling-narrative-techniques-and-methods-in-video-games/, 2014, Letzter Zugriff: 11. 08. 2016

Soylent Green: *Blender Dokumentation: Proportional Editing-Tool*, https://de.wikibooks.org/wiki/Datei:Blender3D_PETFallOffModes.png, Letzter Zugriff: 11. 08. 2016

Stelzner, Dr. Dr. Ruben: *Der goldene Schnitt - Das Mysterium der Schönheit*, http://www.golden-section.eu/kapitel5.html, 2003, Letzter Zugriff: 11. 08. 2016

The Qt Company: *The future is written in Qt*, https://www.qt.io/, Letzter Zugriff: 07. 06. 2016

The Qt Company: *User Interfaces*, http://doc.qt.io/qt-5/topics-ui.html, Letzter Zugriff: 11. 08. 2016

The Qt Company: *About Qt*, http://wiki.qt.io/About_Qt, Letzter Zugriff: 07. 06. 2016

The Qt Company: *Graphics View Framework*, http://doc.qt.io/qt-5/graphicsview.html, Letzter Zugriff: 16. 06. 2016

The Qt Company: *Model/View Programmming*, http://doc.qt.io/qt-4.8/model-view-programming.html, Letzter Zugriff: 07. 07. 2016

Von Rolbeck, Hanns: http://integralmed.eu/pages/basis-diagnostik/kiefergelenk---cmd---gpd.php, Letzter Zugriff: 11. 08. 2016

Waters, Keith: „A muscle model for animating three-dimensional facial expression.“, *ACM Computer Graphics*, 21(4):17?24, 1987

Wikipedia: https://de.wikipedia.org/wiki/Wikipedia:Hauptseite, Letzter Aufruf: 11. 08. 2016

Wikipedia: *Constructive Solid Geometry*, https://de.wikipedia.org/wiki/Constructive_Solid_Geometry, 2012, Letzter Zugriff: 11. 08. 2016

Wirth, Dr. Thomas: *Die EN ISO 9241 - 10*, http://kommdesign.de/texte/din.htm, Letzter Zugriff: 07. 07. 2016

3dmax-Tutorials: *CV Surface*, http://www.3dmax-tutorials.com/CV_Surface.html, Letzter Zugriff: 11. 08. 2016

Für den Versuch verwendete Bilder:

Barbie: http://www.salubia.de/bilder/puppen/bianca01g.jpg, Letzter Zugriff: 19. 07. 2016

Frau1: https://www.whitezine.com/fr/photography/portraits-by-martin-schoeller.html/attachment/martin-schoeller-angelina-jolie-portrait, Letzter Zugriff: 14. 07. 2016

Frau2_1: *Caltech Human face (Front) dataset* http://www.robots.ox.ac.uk/~vgg/data3.html, Letzter Zugriff: 14. 07. 2016

Frau2_2: *Caltech Human face (Front) dataset* http://www.robots.ox.ac.uk/~vgg/data3.html, Letzter Zugriff: 14. 07. 2016

Frau3: http://www.polodigital.net/images/ErikaH/ErikaH010f.jpg, Letzter Zugriff: 19. 07. 2016

Frau4: http://www.fotostyle-in.de/file/2015/02/passfoto-frau.jpg, Letzter Zugriff: 19. 07. 2016

Mädchen1: http://www.fotostudio-mohl.de/blog/passfoto/passfoto-passbilder-bei-fotostudio-mohl-in-reutlingen-tel-07121-55294.html, Letzter Zugriff: 19. 07. 2016

Mann3: *Caltech Human face (Front) dataset* http://www.robots.ox.ac.uk/~vgg/data3.html, Letzter Zugriff: 14. 07. 2016

Mann4: *Caltech Human face (Front) dataset* http://www.robots.ox.ac.uk/~vgg/data3.html, Letzter Zugriff: 14. 07. 2016

Frau gedreht: http://www.wallpapersimages.co.uk/master/NEW/1024x768/6883940-portrait.jpg, Letzter Zugriff: 19. 07. 2016