HSR
HOCHSCHULE FÜR TECHNIK
RAPPERSWIL

FHO Fachhochschule Ostschweiz

emitec
industrial

Street View mit Infrarotkamera

Studienarbeit

Abteilung Informatik
Hochschule für Technik Rapperswil

Herbstsemester 2011/2012

Autoren	Philipp Eichmann und Roman Giger
Betreuer	Prof. Oliver Augenstein
Projektpartner	emitec industrial Rotkreuz
Gegenleser	Prof. Dr. Josef M. Joller

Inhaltsverzeichnis

Kapitel 1

Allgemein

1.1 Aufgabenstellung

Studiengang:	Informatik (I)
Semester:	HS 2011/2012 (19.09.2011-19.02.2012)
Durchführung:	Studienarbeit
Fachrichtung:	Software
Institut:	Institut für Software
Verantwortlicher:	Augenstein, Oliver
Betreuer:	Augenstein, Oliver
Gegenleser:	Josef Joller
Ausschreibung:	Ziel der Arbeit ist die Entwicklung eines Algorithmus zum unkenntlich machen von Personen mit einer möglichst 100%-igen Erfolgsrate. Der Algorithmus, der im Rahmen dieser Arbeit implementiert werden soll, basiert dabei auf der Annahme, dass sich durch Änderungen der Aufnahmetechniken das unkenntlich machen von Personen deutlich vereinfachen und zuverlässiger gestalten lässt. Die Arbeit ist experimentell und es kann nicht garantiert werden, dass der Algorithmus die erhoffte Erfolgsrate tatsächlich erreicht. In diesem Fall ist es Teil der Aufgabe, die Grenzen des Algorithmus zu verstehen.
Voraussetzungen:	Die Themen sind komplex und verlangen mathematische Kenntnisse (lineare Algebra) und C Programmierung (VC oder GNU C). Die vertieften Kenntniss in der Grafikprogrammierung/Bildverarbeitung können erworben werden.

1.2 Erklärung über die eigenständige Arbeit

Wir erklären hiermit, dass wir die vorliegende Arbeit selber und ohne fremde Hilfe durchgeführt haben, ausser derjenigen, welche explizit in der Aufgabenstellung erwähnt ist oder mit dem Betreuer schriftlich vereinbart wurde, dass wir sämtliche verwendeten Quellen erwähnt und gemäss gängigen wissenschaftlichen Zitierregeln korrekt angegeben haben.

Rapperswil, 22.12.2011

Philipp Eichmann

Roman Giger

1.3 Abstract

Abteilung	Informatik
Name der Studierenden	Philipp Eichmann, Roman Giger
Studienjahr	HS 2011
Titel der Studienarbeit	Street View mit Infrarotkamera
Examinator	Prof. Oliver Augenstein
Themengebiet	Software
Projektpartner	emitec industrial Rotkreuz
Institut	Institut für Software

In dieser Arbeit wird ein alternativer Ansatz zur Aufnahme und Auswertung von Strassenbildern untersucht, so dass Gesichter mit einer möglichst hohen Wahrscheinlichkeit detektiert werden können. Dazu dient ein Wärmebild, das zu einer Farbbildaufnahme die Temperaturen der fotografierten Objekte liefert. Aufgrund dieser Temperaturinformationen können einzelne Bildbereiche detailliert untersucht und auf die Vorkommnisse von Gesichtern beurteilt werden. In einem weiteren Schritt wird mittels verschiedener Filter die Anzahl und Auswirkung von Störfaktoren minimiert.

Als Resultat wurde ein Algorithmus entwickelt, der beliebig viele Farbbilder mit dazugehöriger Wärmeinformation in mehreren Arbeitsschritten nach Gesichtern absucht und diese mit einer Verpixelung anonymisiert.

Kapitel 2

Management Summary

2.1 Ausgangslage

Mit Diensten wie Street View von Google oder Streetside von Microsoft bieten die Unternehmen, als Ergänzung zu den frei zugänglichen Satellitenbildern, 360°-Panoramabilder von Strassen in Städten an. Kurz nach deren Einführung gerieten sie wegen nicht-anonymsierter Gesichter und Nummernschilder von Autos in den Fokus von Datenschützern, die ein sofortiges Unkenntlichmachen dieser sensiblen Bildauschnitte forderten. Die Anbieter reagierten mit einer automatischen Gesichtserkennungs- und Anonymisierungsfunktion, wobei es immer wieder vorkommt, dass Gesichter nicht erkannt werden.
Anbieter von Strassen-Panoramabilder sind rechtlich verpflichtet, sensibles Bildmaterial zu anonymisieren, inbesondere die Gesichter von Passanten und Verkehrsteilsnehmern. Dies bedeutet für grosse Datenmengen einen enormen Zeit- und Rechenaufwand.

Mit einer effizienteren Aufnahmetechnik und einer zuverlässigeren Gesichtserkennungsmethode lässt sich die Anzahl möglicher Gesichtsbereiche auf ein Minimum reduzieren und gleichzeitig die Trefferrate erhöhen. Dies verkürzt die Berechnungszeit pro Bild und vermindert die Anzahl manueller Nachbearbeitungen im Falle von nicht erkannten Gesichtern. Dabei bleiben die notwendigen Modifkationen an der Aufnahmevorrichtung beschränkt.

2.2 Vorgehen, Technologien

Bisherige Anbieter setzen üblicherweise Vorichtungen mit speziellen Kameras und 3D-Sensoren auf fahrenden Vehikeln ein, um Panoramaansichten von Strassen zu generieren. Diese müssen für den in dieser Arbeit vorgeschlagenen Ansatz mit einer zusätzlichen Wärmebildkamera ausgerüstet werden. Eine solche Kombination ermöglicht es, zu jedem aufgenommenen Bild die Wärmebildinformationen zu nutzen, um später mittels Software Menschen besser zu erkennen und deren Gesichtern zu verwischen.

2.3 Ergebnisse

Diese Arbeit zeigt, wie die Auswertung von Temperaturinformation für die Anonymisierung von Gesichtern genutzt werden kann und wo Potenzial für die Verbesserung bestehender Lösungen vorhanden ist. Sie präsentiert dazu einen einsatzfähigen, konfigurierbaren Protoypen, mit dem sich beliebig viele Bilder verarbeiten lassen.

Kapitel 3

Technischer Bericht

3.1 Einleitung und Übersicht

Jedes Objekt mit einer Temperatur über dem absoluten Nullpunkt strahlt Wärme in Form von elektromagnetischen Wellen im Infrarotbereich ab. [FLIRINF11]. Dieser Umstand kann dazu benutzt werden, die Temperaturen von Objekten exakt zu bestimmen. Dazu werden spezielle Wärmebildkameras eingesetzt, die im Gegensatz zu herkömmlichen Fotokameras, elektromagnetische Wellen mit Frequenzen von 8-14μm aufzeichnen. Die sonst für Menschen unsichtbaren Infrarotstrahlen lassen sich so mittels Thermografien veranschaulichen, indem sie jedem Bildpixel einen Temperaturwert zuordnen. Auf diese Weise lassen sich Menschen, Tiere und andere Objekte anhand ihres typischen Temperaturbereichs erkennen.

Die Firma emitec stellte uns im Rahmen dieser Arbeit zwei FLIR Infrarotkameras, FLIR T640 [FLIRT640] und FLIR SC655 [FLIRSC655] zur Verfügung.

Abbildung 3.1: Beispiel einer Thermografie

3.1.1 FLIR Quickreport 1.2 SP2

FLIR Quickreport ist eine Software, die zur Auswertung von thermografischen Bildern, welche mit einer FLIR Wärmebildkamera aufgenommen wurden, dient. Sie bietet wichtige Funktionen zur Anpassung des Temperaturbereichs, so dass Temperaturnuancen besser erkannt und uninteressante Temperaturen herausgefiltert werden können. Bildpunkte unter dem festgelegten Bereich werden Schwarz gefärbt, solche darüber sind als Weiss zu erkennen. Dies ermöglicht es die Charakteristiken von Wärmebildern zu analysieren. Des weiteren sind Anpassungen der Parameter für die Temperaturberechnung wie Emissionsgrad[1], Atmosphärentemperatur, Relative Luftfeuchtigkeit und Abstand zum Objekt möglich. Die entsprechenden Temperaturen pro Pixel können mittels einer Export-Funktion als Textdatei verfügbar gemacht werden.

[1]Menschliche Haut hat einen Emissionsgrad von 0.98. Was heisst, dass 98% der absorbierten elektromagnetischen Strahlung als Wärmestrahlung an die Umgebung zurückgegeben wird. Dies ist ein wichtiger Parameter um Objekte exakter zu messen.

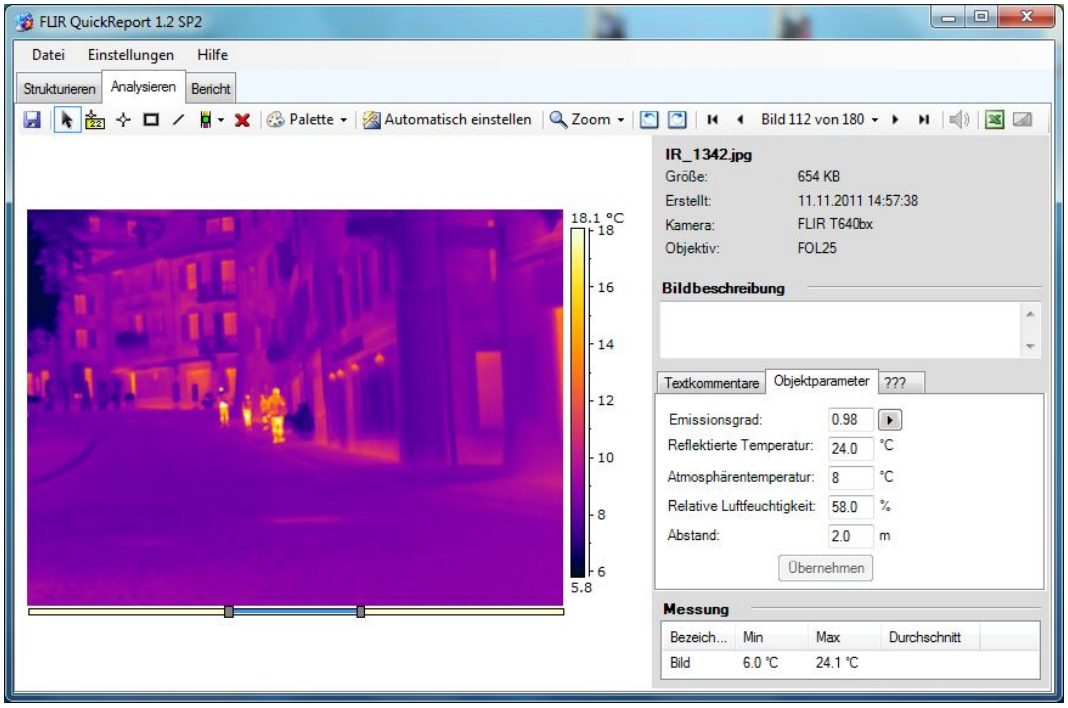

Abbildung 3.2: Flir Quickreport 1.2 Screenshot

3.1.2 OpenCV

Um Bilder automatisiert analysieren zu können, bedarfs es einer performanten Programm-Bibliothek, die eine gut dokumentierte Schnittstelle zu Bildmanipulationsalgorithmen bietet.

Abbildung 3.3: OpenCV Logo

OpenCV [LOCV08] [AOCV11] stellt zahlreiche bewährte und gut getestete Bildbearbeitungsalgorithmen bereit. Sie ist unter der BSD-Lizenz frei vefügbar. Ursprünglich wurde OpenCV in C geschrieben. Um die Speicherverwaltung zu vereinfachen, existiert seit OpenCV 2.0 eine darauf aufbauende Schnittstelle für C++. Neue Funktionen werden seither tendenziell nur noch für die C++ API zur Verfügung gestellt. Es gibt jedoch einige ältere Funktionen wie z.B. `cvConvexityDefects` welche nicht über die C++-Schnittstelle aufrufbar sind, da unterschiedliche Datenstrukturen verwendet werden.

3.2 Analyse

3.2.1 Image Registration

3.2.1.1 Parallax

Unter Parallax[2] versteht man die Positionsdifferenz die ensteht, wenn ein Objekt aus zwei Blickwinkeln betrachtet wird. Dies kann insbesondere bei Menschen und Tieren gut beobachtet werden. Bei Kameras mit zwei unterschiedlich positionierten Linsen tritt eine Verschiebung auf, die problematisch wird, wenn mit jeder Linse gleichzeitig ein Bild aufgenommen wird und die Aufnahmen übereinander gelegt werden müssen.

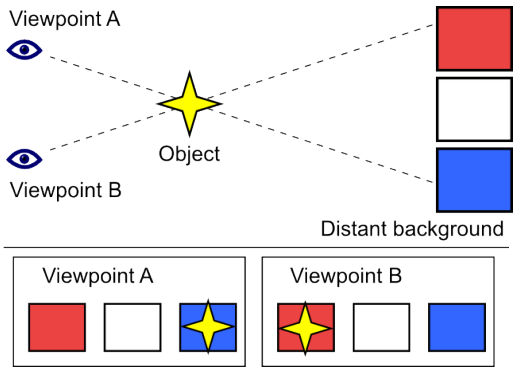

Abbildung 3.4: Ein Beispiel für Parallax, Quelle: [BOOY06]

3.2.1.2 Zeitliche Verzögerung

In der Praxis kommt zusätzlich oft eine zeitliche Verzögerung hinzu, was sich bei stationären Aufnahmen kaum bemerkbar macht, bei bewegten Objekten jedoch die Verschiebung verstärkt. Dieses Problem tritt auch bei Wärmebildkameras auf und muss, damit die zwei entstandenen Bilder exakt übereinander gelegt werden können, korrigiert werden.

3.2.1.3 Korrektur mittels Point-Mapping

Die durch Parallax und Zeitverzögerung entstandene Verschiebung kann näherungsweise korrigiert werden. Dieser Arbeitsschritt wird Image Registration genannt. Dazu bieten sich verschiedene Techniken an, wobei das Matching von einem Wärmebild auf ein Farbbild einem Sonderfall entspricht [BROWN92]. Im einfachsten Fall reicht dafür eine affine Tranformation bereits aus, um bestimmte Punkte (Kontrollpunkte) auf beiden Bildern zu matchen und so das Wärmebild global anzupassen. Dabei spielt es keine Rolle, wie stark das Wärmebild deformiert wird, zumal es nur dazu dient, warme Regionen auf einem Farbbild wiederzuerkennen.

Eine affine Transformation umfasst Operationen wie Skalierung, Translation, Rotation, Scherung und Verzerrung. Es ist eine globale Transformation, die sich auf das gesamte Bild auswirkt, da sich die Beziehungen zwischen den einzelnen Punkten nicht ändern. Sie ist im zwei-dimensionalen Raum definiert als:

[2]Parallax wird u.a. auch für Distanzmessungen in der Astronomie benutzt.

$$\begin{pmatrix} x' \\ y' \\ 1 \end{pmatrix} = \begin{pmatrix} a & b & t_x \\ c & d & t_y \\ 0 & 0 & 1 \end{pmatrix} \cdot \begin{pmatrix} x \\ y \\ 1 \end{pmatrix} \tag{3.1}$$

Gemäss [BROWN92] braucht es für die Berechnung der Transformationsmatrix mit n Unbekannten genau n Kontrollpunkte. Diese können jedoch wegen lokalen Verzerrungen nicht exakt gematcht werden, weshalb die Koeffizienten mit einer Approximationsmethode wie Least-Squares[3] bestmöglich angenähert werden.

Für das Point-Mapping müssen geeignete Kontrollpunkte auf beiden Bildern gewählt werden. Beim Matching von Wärmebilder auf Farbbilder kann dies unter Umständen schwieriger sein als die Punktfindung auf zwei Farbbildern. In unserer Arbeit gehen wir nicht näher auf diese Problematik ein, da dies als Hardwareproblem angesehen werden kann und der Fehler sich bei fixer Positionierung relativiert.

3.2.2 Face-Detection

Um herauszufinden, wie Gesichter auf Wärmebilder erkannt werden können, lohnt es sich, zuerst einen Blick auf die Face-Detection auf Farbbildern zu werfen. Auf diesem Gebiet wird seit längerem geforscht und es existieren verschiedene Implementationsansätze.

Hintergrund Eliminierung Im einfachsten Fall befindet sich ausschliesslich ein Gesicht auf einfarbigem Hintergrund, so dass dieser einfach herausgefiltert werden kann und als Resultat nur noch das Gesicht zu sehen ist.

Hautfarbe Menschliche Haut hat die Eigenschaft, dass sich deren Farbton in einem bestimmten Bereich befindet (Siehe auch 3.3.6.5). Diese Tatsache kann dazu genutzt werden, um Gesichter auf einem Farbbild zu erkennen. Sobald jedoch andere Bereiche im Bild dieselben Farbtöne aufweisen oder die Belichtung die Echtheit des Farbtons verfälscht, ist dieser Algorithmus äusserst anfällig auf False-Positives [FTOZ11].

Geometrische Formen Gesichtscharakteristiken wie die Ausprägung des Mundes, der Augen, Augenbrauen, Nase etc. können auf geometrische Formen abgebildet werden, nach denen mit Hilfe von Kantendetektion und Filtern gesucht werden kann. Auch diese Implementation ist stark anfällig auf Störfaktoren aus der Umgebung und erzielt oft nur bei Portraitaufnahmen befriedigende Resultate.

Bewegungen Face-Detection auf bewegten Bildern ermöglicht, die Bilder auf Veränderungen über die Zeit zu untersuchen. Typerischerweise bewegen sich Menschen bzw. Köpfe und Gesichter oder es kann über das Blinzeln der Augen auf ein potentielles Gesicht geschlossen werden.

Binäre Klassifikation mit Hilfe neuronaler Netze Der erfolgreichste und zugleich komplexeste Ansatz basiert auf Neuronalen Netzen.[4] Sie können mit Testdaten trainiert werden, so dass sie für weitere Inputs aufgrund der bisherigen Erfahrung entscheiden können, ob in einem bestimmten Bereich ein Gesicht vorkommt oder nicht (binäre Klassifikation).

[3] Least-Squares ist ein Verfahren, um für ein überbestimmtes Gleichungssystem eine optimale Lösung zu finden.

[4] Neuronale Netze sind mathematische Modelle, die den neuronalen Netzen aus der Biologie nachempfunden sind. Sie werden u.a. gebraucht um Muster in Daten zu erkennen.

3.2.2.1 Face-Detection in OpenCV

Basierend auf der Arbeit von Paul Viola und Michael M. Jones [VIOLA01] liefert OpenCV ein Modul zur Objektdetektion, welches binäre Klassifizierer einsetzt. Bei der Beurteilung eines Bildes wird innerhalb eines bestimmten Auschnittes nach sogenannten Haar-like Features[5] gesucht [LOCV08]. Dabei handelt es sich um Kontrastunterschiede in einem bestimmten rechteckigen Bereich. Beispielsweise macht man sich den Umstand zu Nutze, dass die Augenpartie eines Menschen typischerweise dunkler ist als die Wangen [WHIOM06]. Für die Berechnung dieser Unterschiede werden zwei adjazente Rechtecke über den Bildausschnitt gelegt und die Differenz der aufsummierten Pixelwerte beider Flächen kalkuliert. Die Anzahl der Rechtecke und deren Grösse kann nach Bedarf variiert werden.

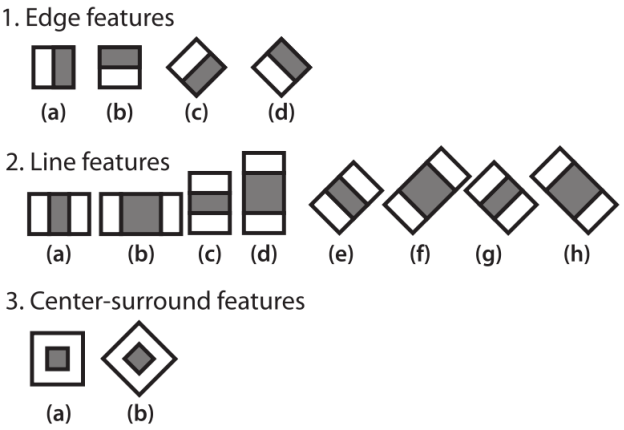

Abbildung 3.5: Haarlike Features, Quelle: [LOCV08]

Die Differenzbildung lässt sich mit Hilfe von Intergal Images[6] sehr effizient umsetzen, so dass sie nur noch $O(1)$ Zeit benötigt.

OpenCV verwendet pro Feature einen Klassifizierer, ein Weak Classifier. Sie werden weak genannt, da die Klassifizierungen oftmals falsch sind bzw. nur etwas bessere Resultate liefern als zufälliges Raten. Gemäss der Arbeit von Michael Kearns [SCHAP01] können jedoch viele Weak Classifiers zu einem Strong Classifier gebündelt werden, was als Boosting bezeichnet wird (Boosted Classifier). Die Aufgabe eines Boosting-Algorithmus ist es, Weak Classifiers, die oft falsch liegen, schwach und gut performende stark zu gewichten. OpenCV stellt verschiedene Boosting-Algorithmen zur Verfügung, wobei üblicherweise AdaBoost eingesetzt wird.

Die Schwellwerte, die die Klassifizierer benutzen, um eine Entscheidung zu treffen, werden durch maschinelles Lernen mit neuronalen Netzen ermittelt. Im Fall einer Gesichtserkennung werden tausende Gesichtsportraits, als positive Beispiele und noch viel mehr negative Beispiele wie Umgebungsfotos verarbeitet, um das Neuronale Netz zu trainieren und so

[5] Haar-like Features erhielten ihren Namen, weil sie ähnlich berechnet werden wie die Koeffizienten von Haar Wavelets.

[6] Ein Integral Image eines Bildes speichert für jeden Pixel die Summe aller Pixelwerte im Rechteck vom Bildursprung bis zu diesem Pixel. Damit kann mit wenigen Operationen die Pixelsumme beliebiger Rechtecke im Bild berechnet werden.

einen geeigneten Schwellwert zu bestimmen.

Die Boosted Classifiers werden dann in einem Entscheidungsbaum angeordnet, der bei der Analyse eines Bildauschnittes durchlaufen wird. Abb. 3.6 zeigt ein Bespiel für die kaskadierten Boosted Classfier $F_1 ... F_N$

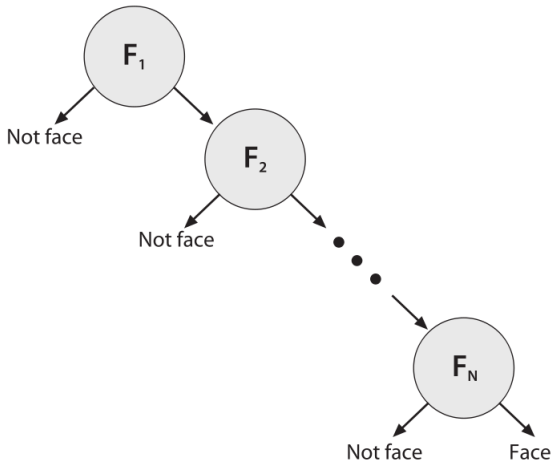

Abbildung 3.6: Cascade Boosted Classifier, Quelle: [LOCV08]

OpenCV liefert verschiedene Kaskaden (Haar Cascades) in Form von XML-Dateien mit, die sich durch die berücksichtigten Features unterscheiden. Zum Beispiel `FrontalFace`, `Fullbody` und `UpperBody`, die je nach Einsatz dynamisch geladen werden können. Die Entscheidungsbäume sind so geordnet, dass, je weiter unten im Baum sich ein Boosted Classfier befindet, umso deterministischer die Features sind. Die Härte des Auswahlverfahrens kann also auch über die Höhe des Entscheidungsbaums definiert werden. Will man möglichst viele Gesichter erkennen und damit auch eine grössere Anzahl False-Positives in Kauf nehmen, so können weit unten liegende Klassifizierer aus dem Baum entfernt werden.

3.2.3 Wärmebildanalyse

3.2.3.1 Schwächen der Thermografie

Da Wärmebilder anhand von Infrarotstrahlung generiert werden (siehe 3.1), muss beachtet werden, dass elektromagentische Wellen mit Wellenlänge von 8-10μm kein Glas durchdringen können. Die Wärmestrahlen in diesem Bereich werden hauptsächlich reflektiert (siehe Abb. 3.7). Gesichter z.B. hinter Schaufenstern oder Autoscheiben können daher nicht erkannt werden. Da nicht alle Wellen reflektiert werden, verliert die Strahlung an Intensität was zu einer deutlichen Abkühlung von gespiegelten Objekten führt. Z.B. ein reflektiertes Gesicht mit 30°C liegt dann noch bei etwa 16°C und macht so das Detektieren schwieriger [FLIRMAN10].

Abbildung 3.7: Infrarotstrahlung reflektiert an Schaufenster

3.2.3.2 Eigenschaften von Gesichtern

Gesichter weisen auf Thermografien typischerweise eine Temperatur von rund 30°C auf. Dies ist jedoch nicht immer der Fall und hängt von der Distanz zur Kamera und der Umgebeungstemperatur ab. Je grösser die Entfernung, umso tiefer die Gesichtstemperatur. Grund dafür ist die begrenzte Anzahl Pixel, auf die eine Temperatur abgebildet wird. Es kann also vorkommen, dass bei kleineren Objekten die direkt angrenzenden Temperaturen Einfluss auf deren Werte nehmen. Dabei können Unterschiede von bis zu 5-10°C entstehen (siehe Abb. 3.8). Daher sollte eine typische Gesichtstemperatur in einem verhältnismässig grosszügigen Bereich definiert werden.
Eine weitere charakteristische Eigenschaft von Gesichtern ist ihre elliptische Form. Sie heben sich dadurch besonders bei kalter Umgebungstemperatur klar von der Umgebung ab. Zu beachten ist dabei, dass Brillen diese unter Umständen in zwei Teile zerlegen und dass Haare, die deutlich kälter sind als menschliche Haut, die Geometrie der Ellipse beinflussen können (siehe Abb. 3.9).

Abbildung 3.8: Gesichtsform und Temperaturverhalten bei Distanz

Kalte Umgebungstemperaturen führen zu Temperaturnuancen im Gesicht, die vorallem bei Nahaufnahmen gut sichtbar werden. Zusätzlich entstehen bei kalter Witterung, auf Asphalt oder Hausfassaden, viel weniger Störfaktoren als bei warmem Wetter (siehe Abb. 3.9).

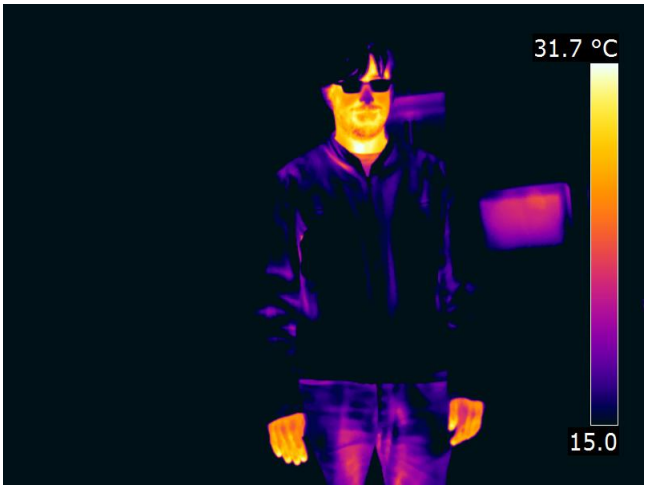

Abbildung 3.9: Gesichtstemperatur nach etwa einer Stunde bei einer Aussentemperatur von 8°C

3.2.3.3 Eingrenzung des Temperaturbereichs

Mit der Erkenntnis, in welchem Temperaturbereich Gesichter liegen (Abb. 3.10), können nun alle Pixel auf dem Farbbild, die in den Körpertemperaturbereich fallen, verpixelt werden. Bei kühler Aussentemperatur, mit kleiner Anzahl Störfaktoren, funktioniert dieser simple Ansatz recht gut (siehe Abb. 3.11). Es fällt kaum auf, dass auch Bereiche verpixelt werden, auf denen kein Gesicht zu sehen ist. Unterstützend wirkt dabei auch die schlechte Qualität des Bildes.

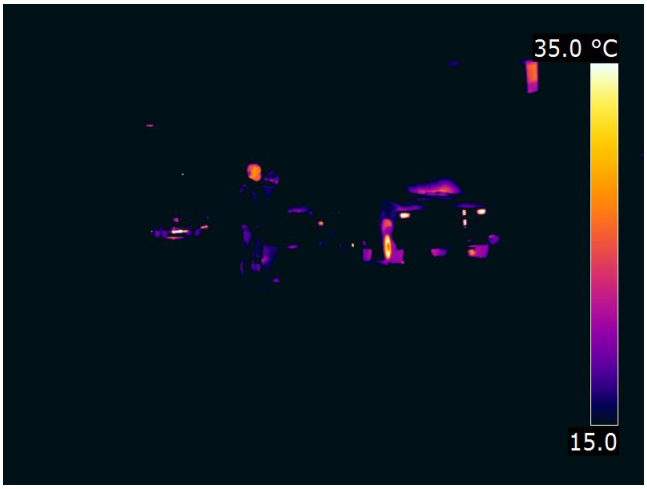

Abbildung 3.10: Wärmebild mit eingeschränktem Temperaturbereich bei kalter Aussentemperatur

Abbildung 3.11: Verpixelung bei kalter Aussentemperatur und mittlerer Bildqualität

Hingegen bei warmer Aussentemperatur und Sonneneinstrahlung werden viele Bildbereiche fälschlicherweise verwischt, wodurch die Qualität des Bildes leidet. Alles was nicht zu den zu warmen (im rechten Bild weiss) oder zu kalten Bereichen (schwarz) gehört, wird mit dieser Technik weichgezeichnet. (Abb. 3.12).

Abbildung 3.12: Wärmebild mit eingeschränktem Temperaturbereich bei warmer Aussentemperatur

Abbildung 3.13: Verpixelung bei warmer Aussentemperatur und guter Bildqualität

Wie in Abb. 3.13 ersichtlich ist, wird mehr als die Hälfte der Bildfläche verwischt. Dies obwohl kein einziges Gesicht darauf zu sehen ist. An Stellen mit geringer Farbvarianz ist die Verpixelung kaum wahrnehmbar, an anderen Stellen ist sie sehr deutlich zu sehen. Bei guter Bildqualität fällt die fälschliche Verpixelung viel mehr ins Gewicht.

3.2.3.4 Umgebungstemperatur

Die Umgebungstemperatur ist mitbestimmend was für eine Strategie gewählt wird. Eine Klassifizierung der Umgebungstemperatur ist also sinnvoll. Um Rückschlüsse auf die Umgebungstemperatur zu gewinnen, eignen sich Histogramme. Werden alle Temperaturen eines Bildes auf 1 Grad genau gerundet und davon eine Statistik erstellt, lässt sich daraus ermitteln, ob es sich um ein Bild an einem kühlen oder einem warmen Ort handelt.

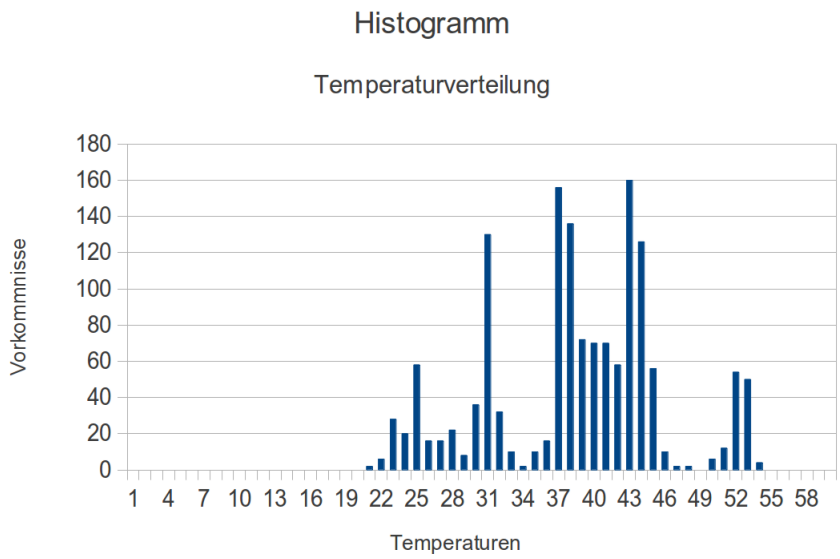

Abbildung 3.14: Temperaturverteilung als Histogramm veranschaulicht

In Abb. 3.14 handelt es sich um einen Ausschnitt eines Bildes mit Sonneneinstrahlung.

3.2.3.5 Fazit

Bei kalter Umgebungstemperatur funktioniert die blosse Verpixelung aller Körpertemperaturbereiche gut. Der Grossteil der Artefakte befindet sich auf dem menschlichem Körper, was vernachlässigt werden kann, zumal ja eine Anonymisierung erwünscht ist. Problematisch sind allerdings warme Umgebungstemperaturen und die Tatsache, dass Gesichter hinter Glas mit einer Wärmebildkamera nicht erkannt werden können (siehe 3.2.3.1).

Solche Probleme können mit folgenden Techniken angegangen werden:

Kombination aus Face Detection und Wärmebildern Durch eine Kombination der Face-Detection (3.2.2) und den Wärmedaten kann auf allen Stellen, an denen die Temperaturdaten ein Gesicht vermuten lassen, eine tolerantere Face-Detection durchgeführt werden. Dies verspricht eine Verbesserung der Performance und Trefferquote. Dieser Ansatz ist ohne grossen Aufwand zu implementieren und wird in dieser Arbeit nicht näher behandelt.

Binäre Klassifizierer für Wärmebilder Ein Mensch auf einem Wärmebild weist analog zum Farbbild charakteristische Merkmale auf. Somit kann wie unter 3.2.2 beschrieben, ein Neuronales Netz darauf trainiert werden, Gesichter oder Körper auf Wärmebildern zu erkennen. Dies setzt jedoch einen imensen Aufwand für die Bereitstellung genügender Trainingsdaten voraus, weshalb im Rahmen dieses Projekts nicht genauer darauf eingegangen wird.

Konturerkennung Da sich auf Wärmebildern Gesichter im Normalfall gut von der Umgebung unterscheiden lassen, können mit Hilfe einer Konturerkennung alle Umrisse erkannt und

aufgrund deren Eigenschaften und den dazugehörigen Metadaten[7] entschieden werden, ob es sich dabei um ein potentielles Gesicht handeln kann. Damit wird die Anzahl der False-Positives eingeschränkt. Diese Arbeit widmet sich im Folgenden diesem Lösungsansatz.

3.2.4 Verpixelung - Faltungsfilter[DIP08]

Punkt-Operationen zeichnen sich dadurch aus, dass sie einen Pixel-Wert nur aufgrund des Original-Werts an der selben Position bestimmen und können deshalb nicht für komplexere Bildmanipulationen wie Schärfung und Verwischung eingesetzt werden. Diese Effekte werden mit Hilfe von Filtern umgesetzt.

3.2.4.1 Grundlagen

Filter verwenden, im Gegensatz zu Punkt-Operationen, die Nachbarn eines Pixels um dessen Wert zu bestimmen. Die zu Grunde liegende Technik basiert auf der mathematische Faltung, welche definiert ist als:

$$(I * G)(x) = \int I(\tau)G(x-\tau)d\tau \tag{3.2}$$

Um den Wert eines Pixels im Bild I mit einem 3×3 Faltungskern G zu ermitteln, wird eine zweidimensionale, diskrete Faltung eingesetzt:

$$I'(x,y) = \sum_{i=-1}^{1}\sum_{j=-1}^{1} I(x+i, j+i) \cdot G(i,j) \tag{3.3}$$

Der Faltungskern G dient also als Gewichtungsfunktion der umliegenden Pixel. Die Glättung an den Rändern kann mit Pseudo-Werten implementiert werden.

3.2.4.2 Lineare Glättungs-Filter

Um Bilder zu glätten, bieten sich verschiedene lineare Filter an.

Boxfilter Ein Box-Filter, zum Beispiel, besteht aus einer Matrix, in der aller Koeffizienten denselben Wert haben und sich zu einer Summe von 1 aufaddieren. Er bildet folglich das arithmetische Mittel aller umliegenden Werte.

$$G_{Box}(i,j) = \begin{pmatrix} \frac{1}{9} & \frac{1}{9} & \frac{1}{9} \\ \frac{1}{9} & \frac{1}{9} & \frac{1}{9} \\ \frac{1}{9} & \frac{1}{9} & \frac{1}{9} \end{pmatrix} \tag{3.4}$$

Alle benachbarten Pixel werden gleich gewichtet. Dies ermöglicht eine besonders einfache und effiziente Implementierung.

Gaussfilter Einem Gaussfilter liegt eine zweidimensionale Gaussfunktion zu Grunde, die die umliegenden Pixel mit wachsender Nähe zum Mittelpunkt stärker gewichtet. Auch hier muss die Summe der Koeffizienten 1 sein.

$$G_{Gauss}(i,j) = \begin{pmatrix} 0.075 & 0.125 & 0.075 \\ 0.125 & 0.2 & 0.125 \\ 0.075 & 0.125 & 0.075 \end{pmatrix} \tag{3.5}$$

[7] In diesem Fall stehen Temperaturdaten und das Originalbild als Metadaten zur Verfügung

Laufzeit und Effizienz In ihrer einfachsten Implementation bestehen lineare Filter aus vier in sich verschachtelten Schleifen. Die beiden äusseren iterieren über jeden Pixel eines Input-Bildes, die beiden innern iterieren über den Faltungskern. Somit ergibt sich für die Laufzeit das Produkt aus Bild und Kernelgrösse.

$$I'(x,y) = O(W_{Bild} \cdot H_{Bild} \cdot W_{Kernel} \cdot H_{Kernel}) \tag{3.6}$$

Laufzeit im Frequenzraum Je nach Grösse des Kernels und des Bildes sind lineare Filter sehr teuer. Die Laufzeit kann jedoch optimiert werden, indem man sich das Faltungstheorem zu Nutze macht:

Eine lineare Faltung im Bildraum entspricht einer punktweisen Multiplikation im Frequenzraum.

$$\mathcal{F}\{f * g\} = \mathcal{F}\{f\} \cdot \mathcal{F}\{g\} \tag{3.7}$$
$$\mathcal{F}\{f \cdot g\} = \mathcal{F}\{f\} * \mathcal{F}\{g\} \tag{3.8}$$
$$\tag{3.9}$$

und somit gilt für die Faltung zweier Funktionen:

$$f * g = \mathcal{F}^{-1}\{\mathcal{F}\{f\} \cdot \mathcal{F}\{g\}\} \tag{3.10}$$

$\mathcal{F}$ entspricht einer Diskreten Fourier Transformation.

Der Faltungsalgorithmus kann somit wie folgt optimiert werden:
Die Bildfunktion I und der Faltungskern G werden mit einer DFT in ihren Frequenzraum $\mathcal{F}\{I\}$ und $\mathcal{F}\{G\}$ überführt. Miteinander multipliziert und wieder in den Bildraum rücktransformiert entsteht $\mathcal{F}^{-1}\{\mathcal{F}\{I'\}\} = I * G = I'$, bzw. der gesuchte neue Pixelwert.

Mit einer Fast-Fourier-Transformation lässt sich die Umwandlung ins Frequenzspektrum und zurück mit folgender Laufzeit berechnen

$$O(W_{Bild} \cdot H_{Bild}\ log(W_{Bild} \cdot H_{Bild})) \tag{3.11}$$

Die Multiplikation im Frequenzraum muss insgesamt $W_{Bild} \cdot H_{Bild}$ Mal ausgeführt werden. Für ein Filter angewendet auf ein quadratisches Bild mit einer Seitenlänge M ergibt sich dadurch eine Gesamtlaufzeit von

$$O(M\ log(M) + M^2) \tag{3.12}$$

3.2.4.3 Invertierbarkeit linearer Filter

Unter bestimmten Bedingung ermöglicht die Filterung im Frequenzraum das Rückgängigmachen von angewendeten Filtern. Das Originalbild I kann wie folgt zurückgewonnen werden:

$$\mathcal{F}\{I * G\} = \mathcal{F}\{I\} \cdot \mathcal{F}\{G\} \tag{3.13}$$
$$\mathcal{F}\{I\} = \frac{\mathcal{F}\{I * G\}}{\mathcal{F}\{G\}} \tag{3.14}$$
$$I = \mathcal{F}^{-1}\{\mathcal{F}\{I\}\} \tag{3.15}$$

3.2.4.4 Nicht invertierbare Filter

In einigen Anwendungen, wie zum Beispiel bei der Anonymisierung von Gesichtern ist eine Invertierung bzw. die Rekonstruktion des Originalbilds unerwünscht. Dies kann verhindert werden, indem Informationen vernichtet, anstatt wie das bei linearen Filtern der Fall ist, mittels Faltung verschmiert werden.

Median Der Median-Filter sucht den Median aus allen umliegenden Werten.

Max-Filter Der Max-Filter sucht den höchsten Wert aller umliegenden Pixel.

Min-Filter Der Min-Filter sucht den niedrigsten Wert aller umliegenden Pixel.

3.3 Realisierung

3.3.1 Technologien

Um eine Lösung für die in der Analyse beschriebene Problematik zu implementieren, müssen vorab einige Technologieentscheide gefällt werden.

Als Programmiersprache wählen wir C++ nach dem C++11 Standard. Einerseits kann so von den neusten Sprachfeatures und der Performance von nativem Code profitiert werden, andererseits ist die C/C++-Schnittstelle für OpenCV bestens dokumentiert.

Für das Parsing der Parameter die dem Command-Line-Tool übergeben werden, eignet sich CLI von Code Synthesis (`http://www.codesynthesis.com`). CLI untersützt eine eigens definierte DSL[8] die es einem erlaubt, vom Programm untersützte Optionen, deren Typen, Default-Werte und Dokumentationen anzugeben und auszuwerten. Darüber hinaus kann der CLI-Compiler eine Command-Line-Interface Dokumentation in HTML oder Man-Page-Format generieren.

3.3.2 Übersicht

Im folgenden Flussdiagramm wird eine kurze Übersicht über die einzelnen Schritte des Algorithmus gezeigt.

[8] Domain Specific Language

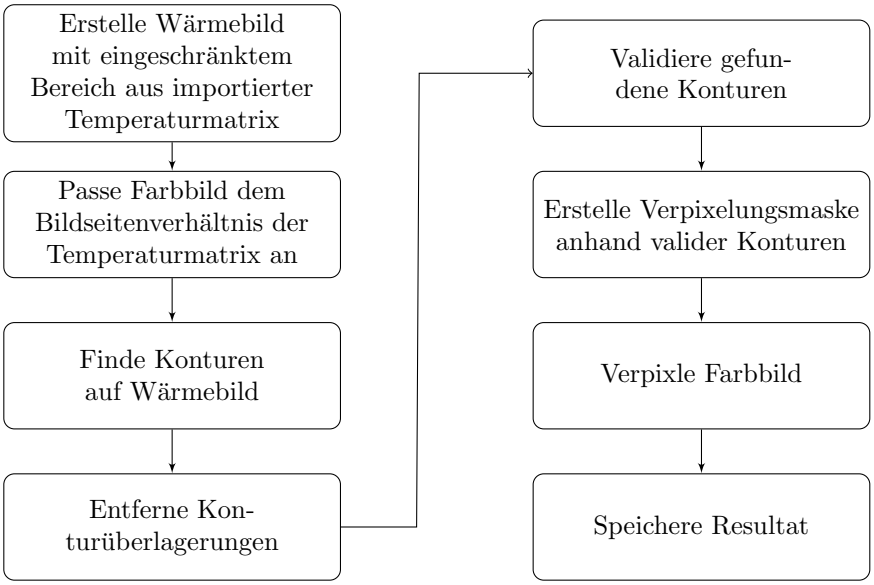

Abbildung 3.15: Ablauf des Anonymisierungsalgorithmus

3.3.3 Import der Temperaturdaten

Abbildung 3.16: Relevante und Irrelevante Bereiche

Als manuelle Vorbereitung für das Programm müssen die Temperaturdaten einer Wärmebildaufnahme mit dem FLIR Quickreport in eine Textdatei exportiert werden.
Bei Programmstart werden die Daten als Temperaturmatrix importiert und das dazugehörige Farbbild geladen. Jeder Wert in der Matrix kann einem Pixel im Farbbild zugeordnet werden, wobei zu beachten ist, dass das Farbbild im Normalfall höher aufgelöst ist und einen grösseren Bereich abdeckt als ein Wärmebild. Deshalb wird das Farbbild zuerst auf den Wärmebildbereich zugeschnitten.

Anmerkung
Verschiedene FLIR Infrarotkameras unterstützen das GenICam-Protokol welches eine allgemeine Programmierschnittstelle bereitstellt. So wäre es zukünftig möglich, ohne proprietäre

Software auf die Temperaturdaten zuzugreifen. [GENICAM]

3.3.3.1 Wärmebild Generierung

In einem nächsten Schritt werden alle Werte in der Temperaturmatrix, die keine potentielle Gesichtstemperatur (z.B. 25°-37°C) aufweisen, ausgenullt. Danach dient der höchste und tiefste Wert in der Matrix als untere bzw. obere Grenze aller relevanten Temperaturen. Dann werden die relevanten Temperaturen linear auf die 254 Graustufen verteilt, um darauf ein Wärmebild zu generieren (Abb. 3.18 links). Alle ausgenullten und somit irrelevanten Bereiche sind nun schwarz.
Gemäss empirischen Messungen weist ein Gesicht durchschnittlich eine Temperatur von ca. 31°C auf. Nun liegt es nahe, die relevanten Temperaturen so auf die 254 Graustufen zu verteilen, dass der Kontrast eines typischen Gesichts am höchsten ist. Damit wird einem 31°-Wert die Farbe Weiss zugeordnet und je weiter ein Wert von 31°C abweicht, umso tiefer ist seine Graustufe (Abb. 3.18 rechts). Dies führt zu merklich besseren Resultaten bei den Konturensuche.

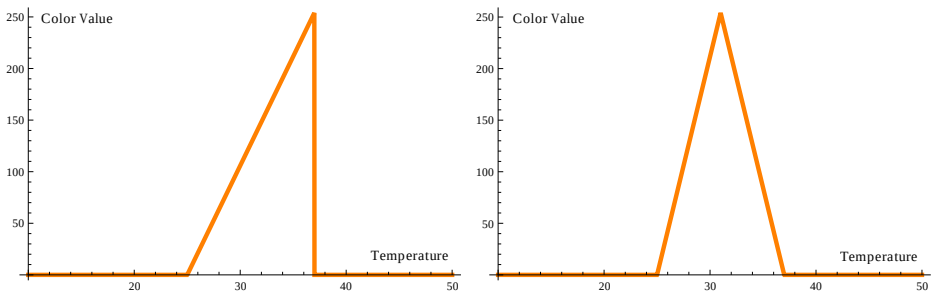

Abbildung 3.17: Links eine lineare Farbverteilung, rechts eine gemittete Verteilung

Abbildung 3.18: Links lineare, rechts gemittet Verteilung

Abb. 3.19 zeigt den Unterschied der Kontursuche der beiden Bilder aus Abb. 3.18. Die Fenster im linken unteren Bereich zeigen, dass die gemittete Version durch den erhöhten Kontrast Konturen besser aufsplitten kann.

3.3.4 Konturen suchen

Mit den OpenCV Funktionen `cvFindContours` und `cvDrawContours` [LOCV08] lassen sich Konturen suchen und zeichnen.

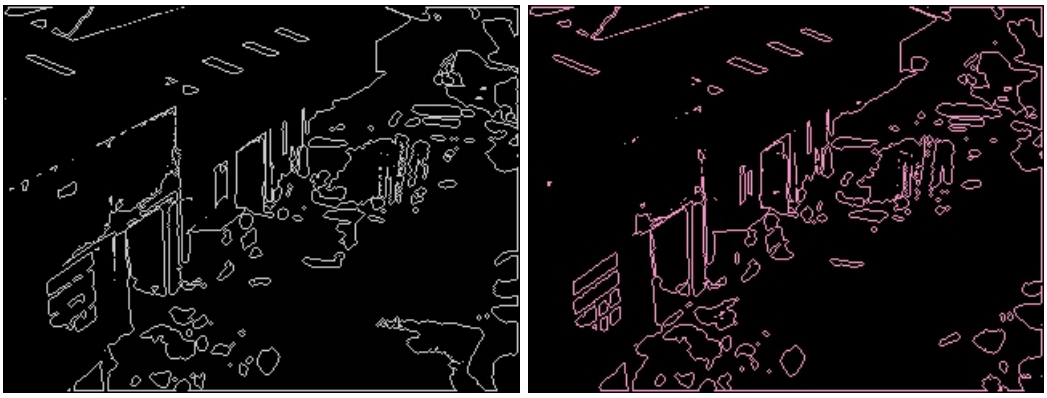

Abbildung 3.19: Konturen zeichnen mit cvDrawContours.

Als Resultat erscheinen viele kleinere Konturen und eine grosse Kontur. In Abb. 3.19 links wurde der Temperaturbereich linear auf die Farben aufgeteilt, rechts wurden die Temperaturen gemittet.

3.3.5 Konturen validieren

Zur Beurteilung der verschiedenen Konturen wird eine Reihe von Filtern eingesetzt, die die Eigenschaften einer Kontur analysieren und sie je nach Beschaffenheit als potentielles Gesicht akzeptieren oder ablehnen. Bei zu grossen Konturen kann nicht bestimmt werden, ob sich ein Gesicht darin befindet oder nicht. Daher muss sie in kleinere Konturen aufgesplittet werden, welche anschliessend genauer analysiert werden können.

Prinzip der Aufsplittung Für die Aufsplittung wird mit `cvBoundingRect` ein Rechteck um die Kontur gelegt. Aus der Fläche dieses Rechtecks wird anhand der Temperaturmatrix ein weiteres Wärmebild erstellt, wobei diesmal der Temperaturbereich weiter eingeschränkt wird[9]. Auf dem neuen Wärmebild wird wieder nach Konturen gesucht, die alle wieder eine Prüfung durchlaufen und aufgesplittet werden, solange ihre Grösse ein Maximum überschreitet. Alle resultierenden, aufgesplitteten Konturen sind potenzielle Gesichter. Abb. 3.20 zeigt ein solches Aufsplittungsverfahren. Dabei wird jeweils nur der Bereich der resultierenden, zu grossen Kontur angezeigt.

[9]Die Einschränkung des Temperaturbereichs kann auf verschiedene Weisen stattfinden. Entweder beim Minimum und beim Maximum oder nur beim Minimum. Die Maximaltemperatur eines Gesichts ist oft fix vorgegeben, wohingegen die Minimaltemperatur recht tief angesetzt werden kann. Daher macht es vorallem bei warmen Bildern, bei denen die Gesichtstemperatur nicht stark variiert, Sinn, den Temperaturbereich nur von unten her schrittweise einzuschränken. Ansonsten läuft man Gefahr, die wärmsten Gesichtsbereiche zu verlieren 3.3.3.1

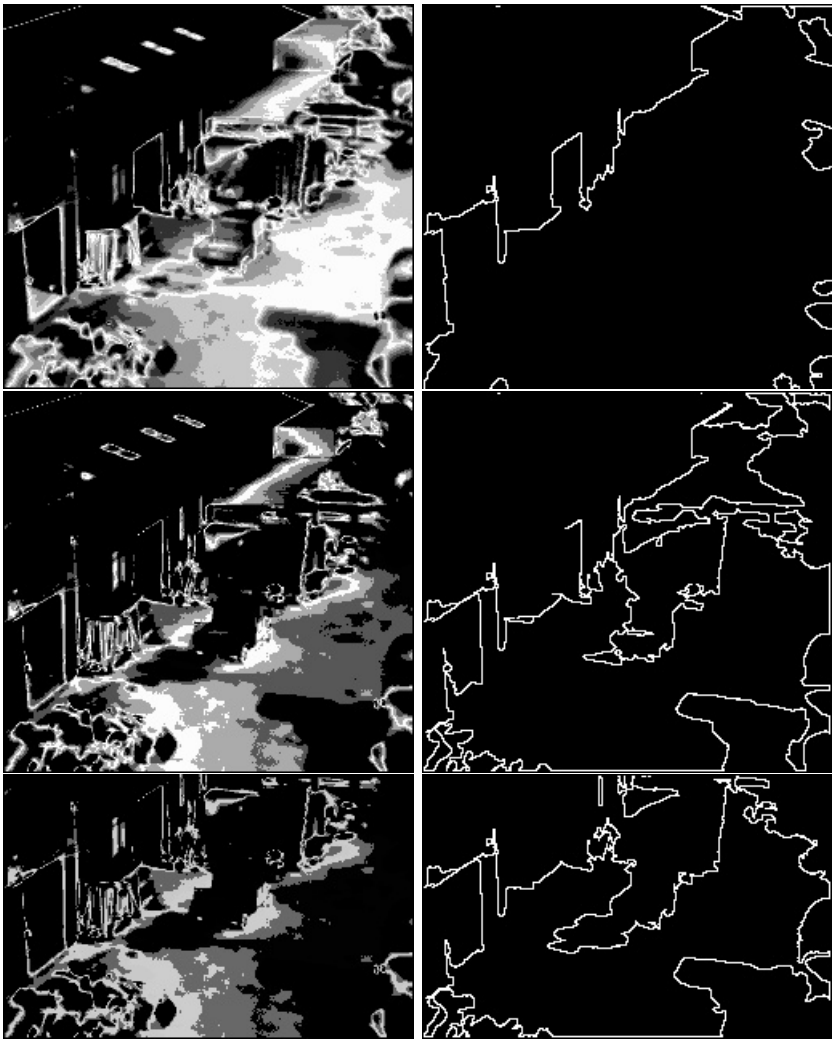

Abbildung 3.20: Aufsplittungsprozess

Hier ist ersichtlich, wie sich die Kontur Schritt für Schritt verkleinert und immer mehr Details zum Vorschein kommen.

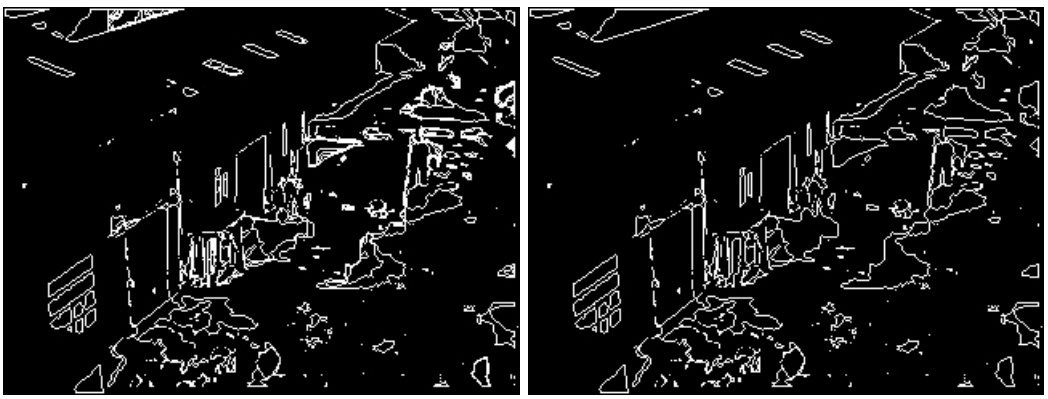

Abbildung 3.21: Aufgesplittete Konturen

In Abb. 3.21 links sind Überlagerungen verschiedener Konturen zu sehen. Diese entstehen durch das Aufsplitten, da gewisse Konturen mehrmals in verschiedenen Grössen gefunden werden. Wenn alle Konturen ausgemalt werden und ein weiteres Mal nach Konturen gesucht wird, verschwinden diese Überlagerungen in Abb. 3.21 rechts.

3.3.6 Filter

Es gibt zwei Arten von Filter. Einerseits jene, die für die Aufsplittung der zu grossen Konturen sorgen und andererseits solche, die Konturen als Gesichter ausschliessen. Für die nachfolgende Effizienzprüfung der einzelnen Filter wird eine Bilderkollektion, und zusätzlich zur Veranschaulichung die zwei folgenden Beispielbilder verwendet. Die Schwellwerte, die dazu verwendet wurden, wurden empirisch evaluiert.

Abbildung 3.22: Beispielbilder - Filter Effizienz

In der Abbildung ist ein Bild mit vielen Personen in einem Raum mit etwa 20°C Umgebungstemperatur und ein Bild von einem Fabrikareal zu sehen, welches durch Sonneneinstrahlung erwärmt wurde. Die Bilder zeigen jeweils die Temperaturen zwischen 25°C und 37°C. Die Weitere gesteste Bilder entsprechen demselben Stil.

3.3.6.1 Maximale Fläche

Ein Kriterium für eine zu grosse Kontur ist ihre Fläche. Um einen guten Schwellwert zu ermitteln, muss die Distanz zwischen Kamera und Gesicht berücksichtigt werden.

Effizienz Um die Aufsplittung anhand der Fläche zu testen, werden alle Konturen die eine Fläche grösser als 1000 Pixel haben, weiter aufgesplittet.

Bild	**False-Positives entfernt**
Bilder Kollektion	60-80%
Abb. 3.22 Leute	71%
Abb. 3.22 Areal	68%

Tabelle 3.1: Maximal-Fläche Auswertung

Abbildung 3.23: Konturen aufsplitten - vorher, nachher

Durch die Aufsplittung grosser Flächen kristallisieren sich Details heraus. Damit können viele False-Positives eliminiert werden. **Vorsicht** ist bei der Wahl des Flächenkriteriums geboten. Ein Gesicht welches die Maximalfläche überschreitet, wird Stück für Stück verkleinert, bis nur noch Fragmente des Gesichts übrigbleiben.

3.3.6.2 Kompaktheit

Die Kompaktheit in Kombination mit der Grösse der Kontur ist ein weiterer Indikator, ob ein weiteres Aufsplitten möglich ist. Ein Aufsplitten einer zu kleinen Kontur macht keinen Sinn und kann sogar gefährlich sein, wenn Gesichter durch Störfaktoren wie Brillen unförmig werden. Daher muss ein Flächenkriterium eingeführt werden. Die Kompaktheit lässt sich mit folgender Formel berechnen.

$$Compactness = \frac{Area}{Perimeter^2} \tag{3.16}$$

Abbildung 3.24: Kompaktheit 0.0041 - Entferne Menschenmenge

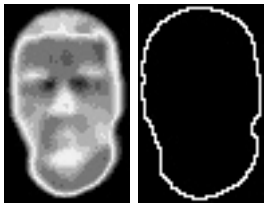

Abbildung 3.25: Kompaktheit 0.062 - Gesicht

Da bei solchen Konturen wie in Abb. 3.24 ein Gesicht nicht mit Sicherheit ausgeschlossen werden kann, darf hier analog zum „Maximale Fläche“-Filter, nur eine Aufsplittung stattfinden anstatt die Kontur zu verwerfen.
Gut sichtbare Gesichter mit einem gewissem Abstand haben einen eher hohen Kompaktheitswert um 0.06. Die meisten Konturen liegen etwa bei 0.01 - 0.02.

Effizienz Der „Maximale Fläche“-Filter ist viel effektiver als der Kompaktheitsfilter. Deshalb wird versucht, eine Effizienzsteigerung mit dem Kompaktheitsfilter zu erzielen, indem beide in Serie geschaltet werden. Als Kompaktheitswert wird 0.01 gewählt und um eine Mindestfläche zu garantieren wird ein Schwellwert von 500 Pixel verwendet.

Bild	**False-Positives entfernt**	**Effizienzsteigerung**
Bilder Kollektion	50%	1-2%
Abb. 3.22 Leute	46%	1%
Abb. 3.22 Areal	63%	4%

Tabelle 3.2: Kompaktheit Verbesserung der Aufsplittung Auswertung

Gemäss Tests wurde keine merkliche Effizienzsteigerung erreicht. Da es unwahrscheinlich aber trotzdem möglich ist, dass ein Teil eines Gesichts eine unförmige Kontur darstellt, beispielsweise durch das Tragen einer Brille oder einer speziellen Frisur, kann der Kompaktheitsfilter zu mehr False-Negatives führen.

3.3.6.3 Konvexe Defekte

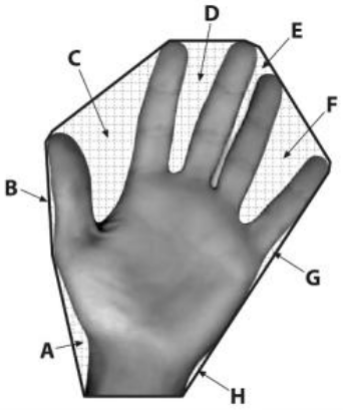

Abbildung 3.26: Konvexe Defekte [LOCV08]

Eine weitere Möglichkeit, die Kompaktheit einer Kontur zu beurteilen, ist die Messung konvexer Defekte. Dies geschieht folgendermassen: Um die Kontur wird eine Konvexe Hülle gelegt. Das heisst, die Kontur wird mit Geraden so umgeben, dass jeder Punkt der Kontur innerhalb dieser Hülle liegt. Die Flächen die dadurch neu entstehen, aber nicht zur Kontur gehören werden konvexe Defekte genannt (Abb. 3.26 A, ..., H). Die Tiefe eines Defekts ist die Länge der längsten Geraden entlang des Lots[10] von der Hülle bis zur Kontur. Ähnlich wie bei der Kompaktheit, zielt dieser Filter darauf ab, unförmige Konturen weiter aufzuteilen, da ein typisches Gesicht von elliptischer Form ist und kaum konvexe Defekte hat. Da das bisherige Aufsplittungsverfahren gut funktioniert und die Auswertung der Defekte False-Negatives birgt, wurde auf die Implementation dieses Filters schliesslich verzichtet.

3.3.6.4 Minimale Fläche

Je nach Qualität der Aufnahmeserie variiert die Mindestgrösse für ein Gesicht. Ab einer gewissen Entfernung werden Gesichter automatisch unscharf und können nicht mehr erkannt werden. Daher ist es sinnvoll, eine Mindestfläche für ein Gesicht zu definieren. So können alle kleineren Konturen entfernt werden.

Effizienz Es wird eine Mindestfläche von 40 Pixel gewählt

Bild	**Anzahl Entfernte Konturen**	**False-Positives entfernt**
Bilder Kollektion	500-1000	0.5-2%
Abb. 3.22 Leute	931	1.7%
Abb. 3.22 Areal	463	1.7%

Tabelle 3.3: Minimale-Fläche Auswertung

Mit dem Filter kann Rauschen entfernt werden. Es werden hunderte kleine Konturen entfernt, die von den anschliessenden Filtern nicht mehr bearbeitet werden müssen, und sich

[10] Senkrecht auf einer Linie stehend.

somit positiv auf die Performance auswirkt. Dieser Filter macht daher direkt nach der Aufsplittung am meisten Sinn. Die minimale Fläche hängt hier von der Qualität der Aufnahme ab und sollte individuell evaluiert werden.

Abbildung 3.27: Kleine Konturen entfernen - vorher, nachher

3.3.6.5 Gesichtsfarbe

Die menschliche Haut besitzt eine farbliche Charakteristik, welche für die Face-Detection genutzt werden kann. Unabhängig von der Hautfarbe enthält die Haut viele Rottöne.

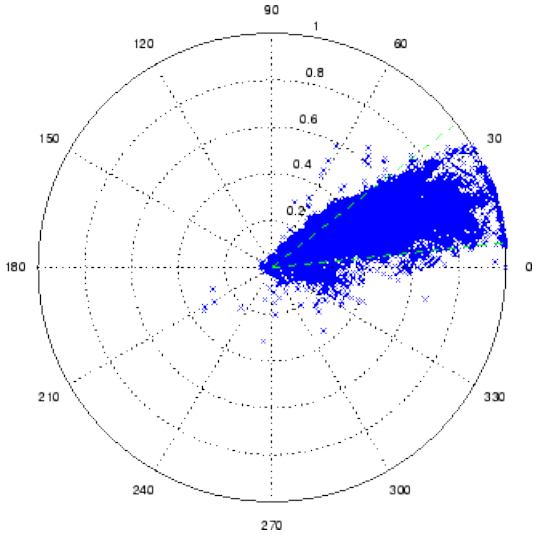

Abbildung 3.28: Analyse der Gesichtsfarbe auf der Hue-Saturation-Ebene [JSSG01]

Die Grafik zeigt, dass alle Farbtöne[11] im Bereich von 60°-300° ausgeschlossen werden können. [JSSG01]

[11] Teil des HSV-Raums: Hue-Saturation-Value Farbton-Farbsättigung-Dunkelstufe

Abbildung 3.29: Hue-Skala 360° [KAL07]

Bei der Implemenation wird mit `cvBoundingRect` ein Rechteck um die Kontur gelegt und der entsprechende Inhalt auf dem Originalbild nach ihrer Farbinformation abgesucht. Ist kein Farbton vorhanden, welcher auf die Existenz eines Gesichtes hinweist, kann ein Gesicht theoretisch ausgeschlossen werden. Diese Implementierungsvariante kann zu Fehlern führen, da die Farbtöne um die Kontur herum einen Einfluss auf das Ergebnis nehmen können. Weiter ist der Farbton von der Qualität der Aufnahme abhängig, wobei farbiges Licht ebenso ein Störfaktor sein kann.

Um diesen Filter zu testen, werden alle Farbtöne zwischen 80 und 260 ausgeschlossen. Sind 20% der Fläche durch die erlaubten Farbtöne vorhanden, wird die Kontur als potentielles Gesicht angesehen.

Effizienz Mit diesem Filter und dieser Parametrisierung lassen sich nur kleinste Konturen ausschliessen. Es kann aus folgenden Gründen nicht ausgeschlossen werden, dass das Verfahren noch verbessert werden kann:

- Implementation mit cvBoundingRect ist fehleranfällig
- Qualität der Aufnahmen ist nicht optimal
- Das Ausschlusskriterium kann evtl. optimiert werden

3.3.6.6 Temperatur Variation

Gesichter haben normalerweise eine Temperatur die 25°C - 37°C. Grosse Abweichungen gibt es selten. Hat jemand eine Zigarette im Mund, entsteht ein Loch welches sich auf der Thermografie als schwarzer Fleck bemerkbar macht. Hat jemand eine Brille auf, können ebenfalls Löcher entstehen. Diese Löcher sind im Vergleich zum gesamten Gesicht eher klein oder teilen es entzwei. Ist jedoch eine zu grosse Variation an Temperaturen vorhanden ist dies ein Indiz dafür, dass es sich nicht um ein Gesicht handelt. Dies könnte z.B. ein Autoreifen mit zu heisser Bremsscheibe, oder eine Lampe sein.

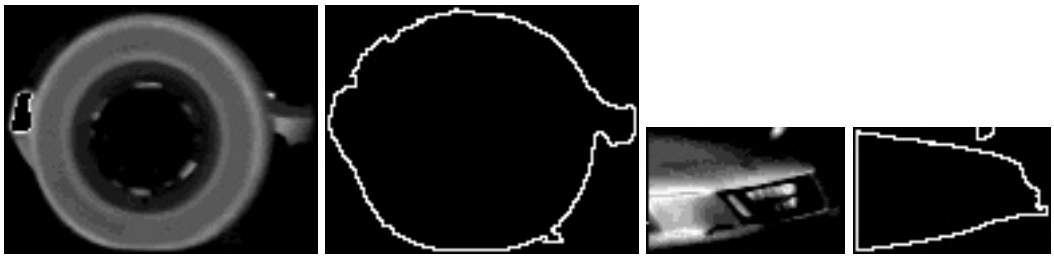

Abbildung 3.30: Unnatürliche Temperaturunterschiede

Bei der Implementation wird ein Rechteck auf die Kontur gelegt und alle vorhandenen Temperaturen innerhalb des Rechtecks werden auf ihre Gültigkeit bezüglich des am Anfang gewählten Temperaturbereichs überprüft. Alle zu kalten und zu warmen Temperaturen werden separat gezählt. Überschreitet eine dieser Werte einen gewissen Prozentsatz, so wird die Kontur ausgeschlossen. Dadurch beeinflusst die Region um die Kontur die Auswertung so, dass Fehler entstehen können.

Effizienz Es werden 20% zu kalte oder zu warme Pixel als Ausschlusskriterium gewählt.

Bild	**False-Positives entfernt**
Bilder Kollektion	0.5-5 %
Abb. 3.22 Leute	0.3%
Abb. 3.22 Areal	0.8%

Tabelle 3.4: Temperatur-Variation Auswertung

Um Löcher nicht fälschlicherweise als Gesichter zu erkennen, kann ein solcher Filter verwendet werden.

3.3.6.7 Ellipse-Filter

Mit der OpenCV Funktion `cvFitEllipse2` wird eine möglichst gut auf die Kontur passende Ellipse berechnet. Einerseits lässt sich so das Höhen-Breiten-Verhältnis auslesen, was wieder auf die Kompaktheit hinweist, andererseits lässt sich aber auch die Ausrichtung der Ellipse auslesen, was als zusätzliches Ausschlusskriterium genutzt werden kann. Ellipsen die Gesichter repräsentieren sind vertikal ausgerichtet, da der Körper oft Teil der Kontur ist wie in Abb. 3.31 zu sehen ist. Markant horizontal ausgerichtete Ellipsen wie in Abb. 3.32 hingegen stellen andere Wärmequellen dar und können ausgeschlossen werden. Es wird davon ausgegangen, dasss keine liegenden Personen vorkommen.

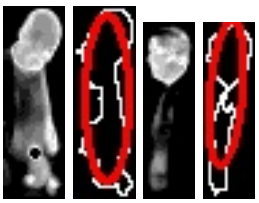

Abbildung 3.31: Vertikal ausgerichtete Konturen wie Personen

Abbildung 3.32: Horizontal ausgerichtete Konturen wie Wärmeabstrahlungen von Automotoren

Die erlaubte Ausrichtung der Ellipsen wird mit zwei Winkeln $\alpha - \beta$ wie in Abb. 3.33 spezifiziert.

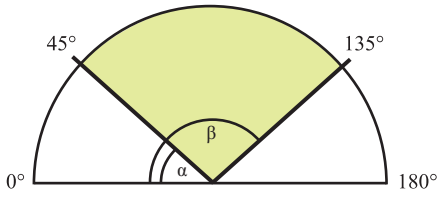

Abbildung 3.33: Winkelbereich von Ellipsen

Das Höhen-Breiten Verhältnis ist in Abb. 3.34 ersichtlich.

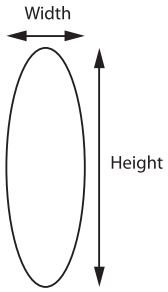

Abbildung 3.34: Höhen-Breiten-Verhältnis

Effizienz Alle Ellipsen mit einem Breiten-Höhen Verhältnis bis 0.4 und einer Ausrichtung zwischen 45° - 135° werden als potenzielle Gesichter angesehen.

Bild	**False-Positives entfernt**
Bilder Kollektion	5-15 %
Abb. 3.22 Leute	5.5%
Abb. 3.22 Areal	5.3%

Tabelle 3.5: Ellipse - Auswertung

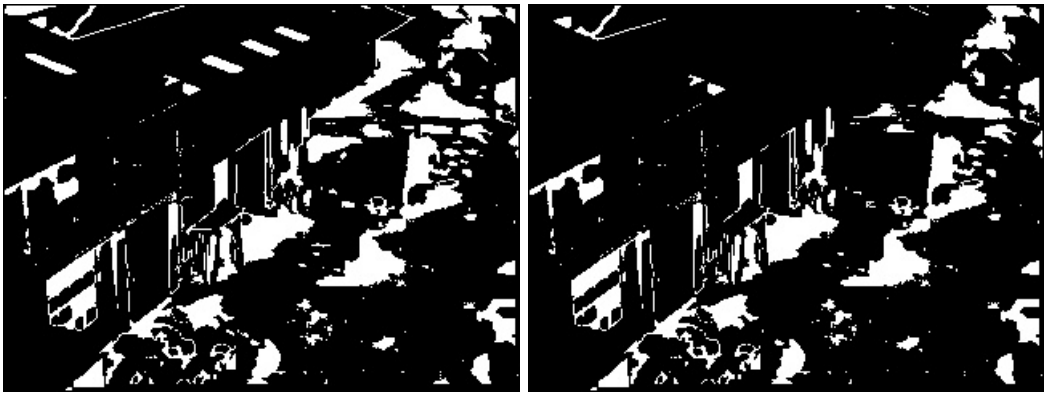

Abbildung 3.35: Ellipse - vorher, nachher

3.3.6.8 Auswertung

Um die herausgefilterten Bereiche pro Filter hervorzuheben werden folgende Farbzuordnungen verwendet.

Blau	Maximale Fläche
Pink	Minimale Fläche
Rot	Temperatur Variation
Grün	Ellipse
Weiss	Zu verpixelnde Konturen

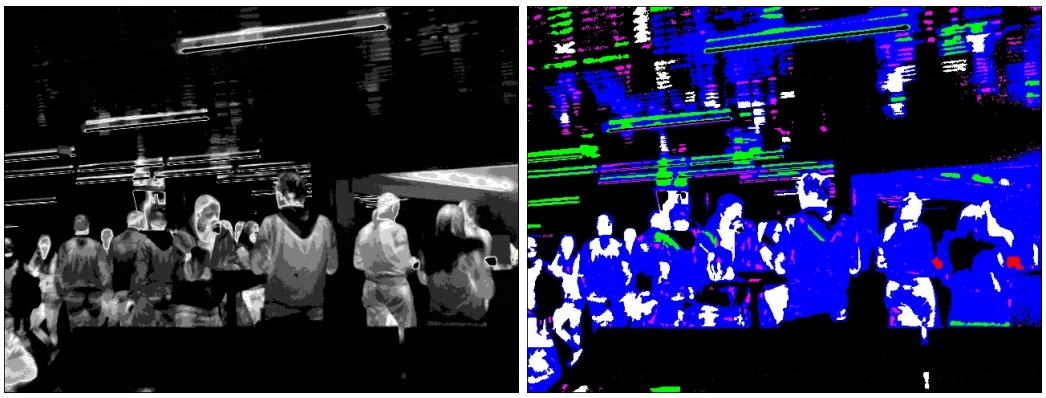

Abbildung 3.36: Filterung Leute

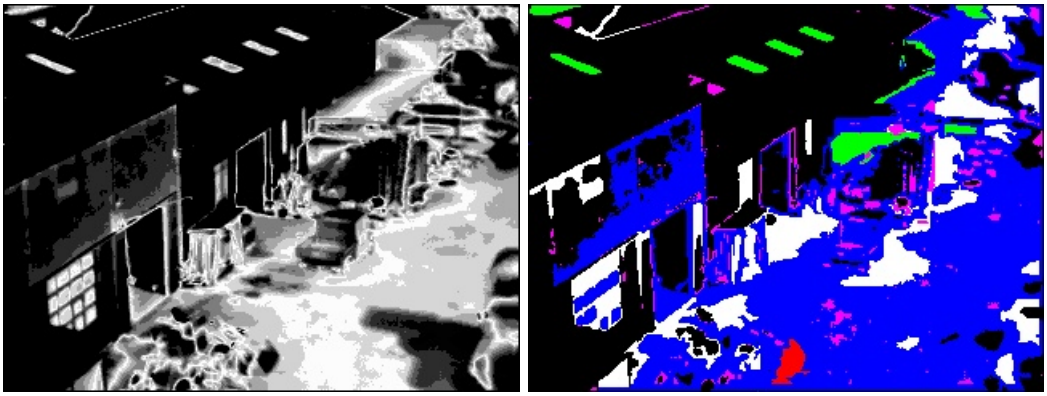

Abbildung 3.37: Filterung Areal

In Abb. 3.37 ist ersichtlich, dass anfänglich schwarze Bereiche im Resultat geschrumpft sind. Dies liegt an der Aufsplittung und der dazugehörigen erneuten Konturensuche. Dem kann entgegengewirkt werden, indem die anfängliche Maske nochmals darüber gelegt wird. Eine der Konturen im rechten oberen Bereich ist gar weiss geworden. Dies liegt an der bereits erwähnten fehleranfälligen Implementierung der „Temperatur Variation“. Einige verbleibende Konturen sind ziemlich unförmig und können anhand weiterer Kriterien herausgefiltert oder verkleinert werden. Ein weiteres denkbares Kriterium zur weiteren Aufsplittung wäre der Maximale Abstand zweier Punkte einer Kontur.

Abbildung 3.38: Bereinigte Verpixelung bei warmer Aussentemperatur

Im Gegensatz zum anfänglich stark verpixelten Bild Abb. 3.13, wurde hier mit dem Einsatz der Filter ein wesentlich besseres Resultat erzielt. Zum Wärmebild mit den Leuten existiert leider keine Fotografie, daher ist diese hier nicht aufgeführt.

3.4 Architektur

Die Architektur des Programms wurde so ausgelegt, dass einerseits die Vorteile einer objektorientierten Programmiersprache wie C++ ausgenutzt werden können, andererseits der Ablauf des Algorithmus nachvollziehbar bleibt und nicht durch ein komplexes Design verschleiert wird. Der Programmcode umfasst Rund 1500 logische Zeilen Code.

3.4.1 Klassendiagramm

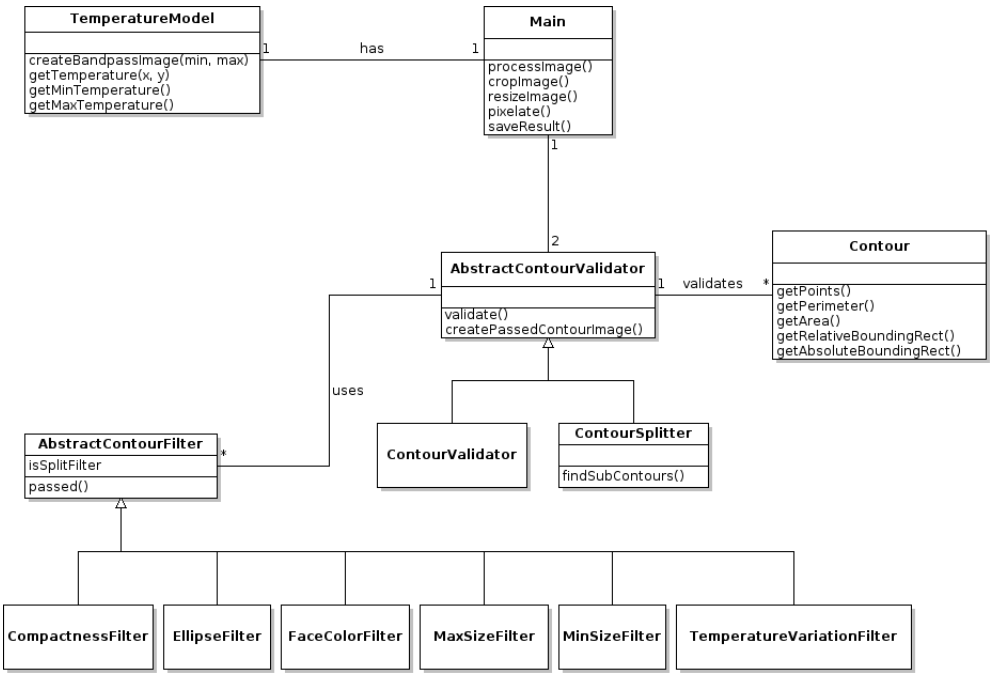

Abbildung 3.39: Klassendiagramm

Main `Main` dient zur Steuerung des Ablaufs und zum Einstieg ins Programm. Diese Klasse parst die ans Programme übergebene Parameter und leitet sie an die von ihr instanziierten Objekte weiter. Die Memberfunktion `processImage()` steuert den gesamten Ablauf, vom Laden des Originalbildes bis zur Speicherung des Resultats. `processImage()` wird pro zu verarbeitendes Bild einmal aufgerufen.

TemperatureModel Das `TemperatureModel` ist verantwortlich für den Import einer Temperaturmatrix aus ihr wird danach ein Wärmebild erstellt. Es besitzt zusätzlich die Fähigkeit, den Temperaturbereich auf dem Wärmebild beliebig einzuschränken, Temperaturdaten auszulesen oder Ausschnitte eines Wärmbildes zu kreieren.

ContourValidator Einem `ContourValidator` können beliebig viele Filter mittels `addFilter()` hinzugefügt werden, mit denen die gefundenen Konturen beim Aufruf von `validate()` validiert werden.

ContourSplitter Der `ContourSplitter` verwendet Filter dazu, zu grosse bzw. zu wenige charakteristische Konturen rekursiv aufzusplitten. Er verhält sich gleich, wie ein `ContourValidator`, ruft aber im Fall eines nicht bestandenen Filters `findSubContours()` auf, was zu einem Validieren aller Unterkonturen führt.

Contour Mit der `Contour`-Klasse wird die Kontur gekapselt. Sie bietet verschiedene Funktionen, welche die Analyse erleichtern.

Filter-Klassen Filter, die von `AbstractContourValidator` erben, werden zur Validierung der Konturen eingesetzt. Dazu überschreiben sie die abstrakte Methode `passed()` (Template-Method Pattern).

3.4.2 Sequenzdiagramm

Das folgende Sequenzdiagramm Abb. 3.40 zeigt die Verarbeitung eines Bildes.

Aufbereitung Zuerst werden die Temperaturdaten und die originale Bildaufnahme aufbereitet.

Filter Initialisierung Anschliessend werden die benötigten Filter mit ihren Einstellungen geladen. Die Validierungsabläufe werden durch diese Filter spezifiziert.

Aufsplittung Jede gefundenen `CvContour` durchläuft diesen Schritt. Die Eigenschaften welche zur jeweiligen Beurteilung berechnet werden, werden in `Contour` gespeichert. Die `CvContour` wird, wenn zu gross, solange in weitere `CvContours` aufgesplittet bis nur noch kleine Konturen vorhanden sind (Rekursiver Aufruf). Aus diesen Konturen entsteht eine Maske in Form eines binären Bildes. Auf diesem werden alle verbleibenden `CvContours` gesucht. Dies ist nötig um entstandene Überlagerungen zu entfernen.

Ausschluss Jede verbleibende `CvContour` durchläuft diesen Schritt. Die `CvContour` wird wieder als `Contour` gespeichert, welche ihre jeweiligen Beurteilungsgrundlagen zwischenspeichert. So muss bespielsweise die Fläche der Kontur trotz mehrmaligem Einsatz nur einmal berechnet werden. Sobald eine `CvContour` ein Kriterieum nicht besteht, wird diese verworfen. Alle anderen werden auf die endgültige Maske gezeichnet.

Ergebnis Die Maske wird passend zur Fotoaufnahme skaliert und als Verpixelungsmaske verwendet. Das verpixelte Bild wird abgespeichert.

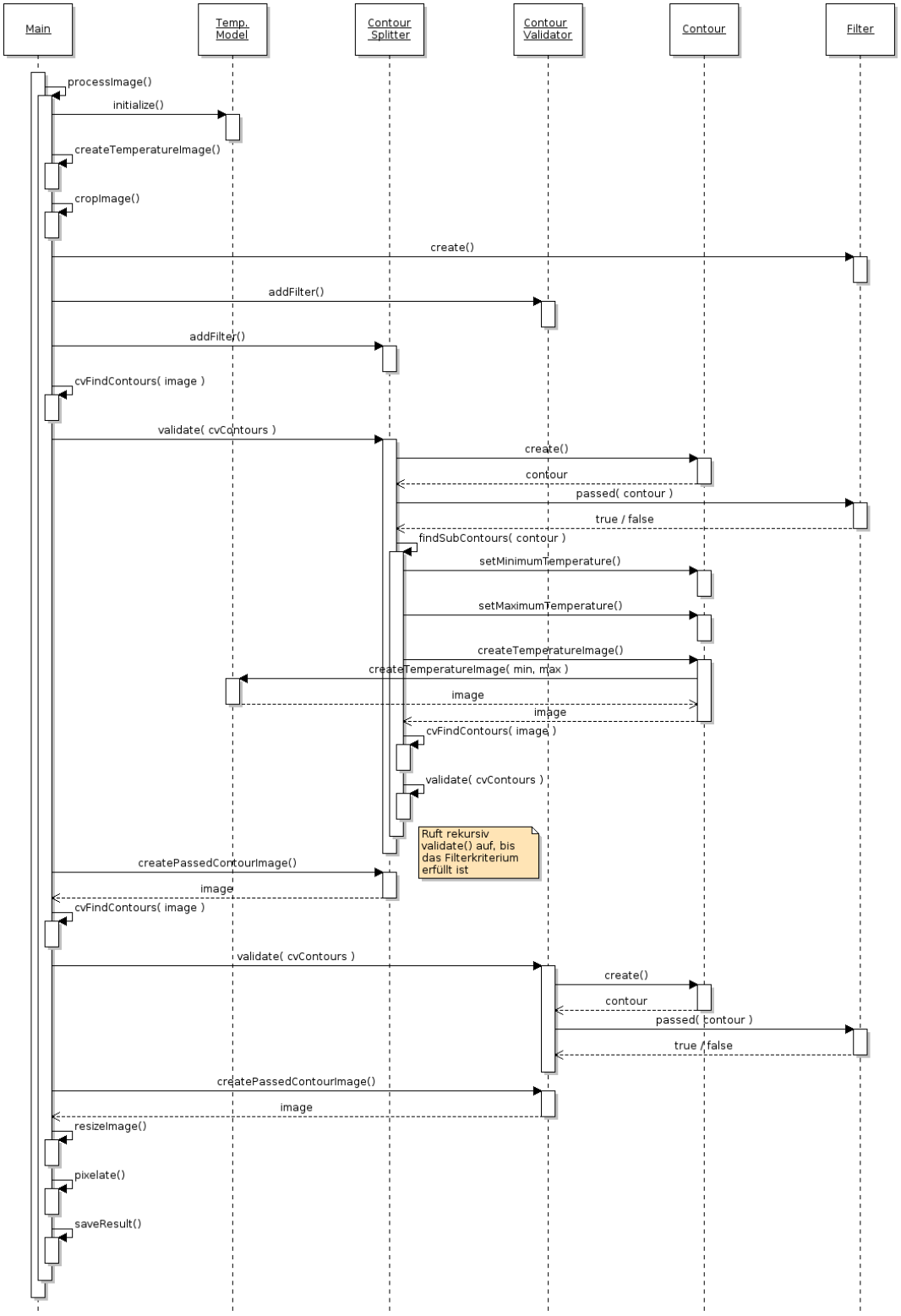

Abbildung 3.40: Sequenzdiagramm

3.5 Ergebnisse

Das Ziel dieser Arbeit war, Personen mit möglichst 100%-iger Erfolgsrate unkenntlich zu machen. Da es die Eigenheiten der Wärmebilder nicht zulassen, können Personen hinter Scheiben und Spiegelbilder nicht erkannt werden (3.2.3.1). Jedoch werden unechte Gesichter, wie solche auf Werbeplakten, nicht verpixelt.

Bei kalter Umgebungstemperatur lassen sich alle warmen Bereiche verpixeln, ohne dass das Bild grossen Schaden nimmt (3.11). Die False-Positives fallen oft nicht stark auf, können aber, falls gewünscht, mit den vorgestellten Filtern entfernt bzw. minimiert werden. Bei warmer Umgebungstemperatur nimmt das Bild grösseren Schaden, wobei die Filter enorm wichtig werden (3.3.6.8). Ist die Qualität der Aufnahme nicht allzu hoch, fallen False-Positives gar nicht gross auf. Die Filter sind je nach Parametrisierung anfälliger auf False-Negatives.

3.6 Schlussfolgerung

Die Verwendung von Wärmeinformation ist von der Witterung abhängig. Werden die Wärmebilder bei optimalen Temperaturen erfasst, ist mit guten Ergebnissen zu rechnen. Je nach Witterungsänderung muss die Parametrisierung angepasst werden, um weiterhin optimale Ergebnisse zu erzielen. Denkbar wäre der Einsatz von Sensoren zur Erfassung der aktuellen Witterungsverhältnisse.

Die implementierten Filter entfernen oder verkleinern False-Positives. Es ist noch Potenzial vorhanden False-Positives mit der Konturenanalyse besser auszuschliessen. Einerseits durch neue Filter andererseits durch exaktere bzw. flexible Parametrisierung.

Denkbar ist auch eine binäre Klassifizierung der Wärmebilder, um Personen auf Wärmebildern zu identifizieren (3.2.3.5).

Es ist Potenzial vorhanden die herkömmliche Gesichtserkennung mit Hilfe der Wärmebildinformation bezüglich der Performance zu optimieren. An Stellen die die Wärmebildinformation ein Gesicht vermuten lässt, wird der Grenzwert der Gesichtserkennung heruntergesetzt. Dabei muss die Aufbereitung der Wärmebildinformation auf ihren Performanceverbrauch analysiert und optimiert werden. Dabei ist eine Parallelisierung des Aufbereitungsvorgangs vorstellbar.

Kapitel 4

Anhang

4.1 Projektplan

4.1.1 Zielsetzung

Streetview mit IR-Cam ist ein rein experimentelles Projekt, dessen Ausgang zu Projektbeginn nur schwer abzuschätzen ist. Ziel der Arbeit ist es, ein funktionsfähiges Programm zu entwickeln, dass im besten Fall eine Lösung zum Problem darstellt oder aber aufzeigt, wieso das Problem nicht bzw. nicht im Rahmen dieses Projekts zu lösen ist.

4.1.2 Planung

Das Projekt basiert auf keiner vorherigen Arbeit. Als Ausgangslage dient lediglich die Problemstellung mit Literatur- und Recherchehinweisen.
Die Bearbeitungszeit kann in folgende drei Phasen aufgeteilt werden.

1. Analyse
2. Implementation
3. Transition

Zu beachten ist, dass die Analyse- und Recherchenarbeit einen hohen Anteil am Gesamtaufwand ausmacht. Zudem bedarf es bei der Planung der einzelnen Arbeitspakete ein hohes Mass an Flexiblität, da neue Erkenntnisse aus der Analyse den Projektverlauf stark beinflussen bzw. die Definition Projektziele entscheidend verändern könnnen.

4.1.3 Zeitplan

Woche	**Fokus**	**Arbeitspakete**
1	Einarbeitung ins Thema, Analyse erster IR-Bilder	1.1, 1.4, 1.5, 1.6
2	Projektplanung, Einarbeitung in OpenCV	1.4, 1.5, 1.6
3	Einarbeitung in Face Dectection, Neurale Netze	1.3, 1.4, 1.5
4	Wärmebildanalyse mit Filter, Thresholds, Template-Matching, Objekterkennung	1.6, 1.7, 1.8, 1.9, 1.11
5	Image Matching, Parallax-Problematik, affine Transformationen	1.13, 1.14
6	Prototyp, Verpixelungsalgorithmen, Faltungsfilter	1.10, 2.1
7	Verbesserung der Kantendetektionen	1.7, 1.9, 2.1
8	Aufnahme und Analyse neuer IR-Bilder, Prototyping	1.4, 1.15, 2.1
9	Architektur und Modellierung	1.15
10	Implementierung der Filter	2.2, 2.3
11	Implementierung der Filter, Option-Parsing	2.2, 2.3, 2.4
12	Testing und Dokumentation	3.1, 3.3
13	Testing und Dokumentation	3.1, 3.2 , 3.3
14	Korrekturen und Abschlussarbeiten	alle

Tabelle 4.1: Zeitplan

4.1.3.1 Arbeitspakete

Analyse		
	1.1	Konkretisierung der Ziele
	1.2	Setup Infrastruktur
	1.3	Face-Detection mit neuronalen Netzen
	1.4	Analyse der Wärmebilder, Einarbeitung in Thermografie
	1.5	Einarbeitung OpenCV
	1.6	Einarbeitung Image Processing
	1.7	Kantendetektion-Analyse
	1.8	Farbraum-Analyse
	1.9	Erkennung geometrischer Objekte
	1.10	Faltungskern-Analyse
	1.11	Template Matching
	1.12	Artefakt-Filter
	1.13	Parallax-Problematik
	1.14	Affine Transformationen
	1.15	Architektur und Modellierung
Implementation		
	2.1	Prototyping
	2.2	Filterdesign
	2.3	CLI-Options
	2.4	Refactoring
Transition		
	3.1	Testing
	3.2	Parameter-Justierung
	3.3	Dokumenation

Tabelle 4.2: Übersicht der Arbeitspakete

4.1.3.2 Arbeitspaket-Details

1.1	Die Projektziele müssen nach neustem Erkenntnisstand fortlaufend justiert werden, um zu vermeiden, dass der Projekterfolg gefährdet wird
1.2	Aufsetzen, kompilieren, konfigurieren von Eclipse CDT, GNU GCC/G++, OpenCV, Boost Library, C++11
1.3	Einarbeitung in die Funktionsweise der Funktion `CascadeClassifier::detectMultiScale`, Aufarbeitung der Viola-Jones-Algorithmus
1.4	Erörterung der Infrarot-Eigenschaften, Studium der FLIR IR-Cam Manuals, Einarbeitung in die FLIR Quick Report Software
1.5	Einarbeitung in OpenCV mittels „Learning OpenCV“ und „OpenCV 2“
1.6	Einarbeitung Image Processing mit diversen Büchern
1.7	Einarbeitung in die Funktionsweise von `cvFindContours`
1.8	Farbraum-Analyse, Konvertierung von RGB in HSV-Farbraum
1.9	Erkennung geometrischer Objekte mittels Hough-Transforms, Kantenanalyse
1.10	Beurteilung verschiedener lineare Faltungsfilter, Aufarbeitung reversibler Faltungsfilter
1.11	Einarbeitung in Funktionsweise von `cvMatchTemplate`
1.12	Suchen von Kontureigenschaften uninteressanter Kanten, Entwicklung verschiedener Konturfilter die uninteressante Bereiche ausschliessen
1.13	Untersuchunng der Parallax-Problematik
1.14	Affine Transformationen
1.15	Architektur und Modellierung
2.1	Erstellung eines funktionalen Prototyps in C/C++
2.2	Entwicklung verschiedener Konturfilter die uninteressante Bereich ausschliessen
2.3	Einbindung des CLI-Options Frameworks ≪CLI≫ von CodeSynthesis.com
2.4	Refactoring des vorhandenen Codes, Verbesserung des Memory-Managements, Optimierung der Performance
3.1	Validierung der Parameter, Stabilitätstests, Testing der Filter-Qualität
3.2	Justierung und Optimierung der Filter-Parameter, um möglichst viele Gesichter zu erkennen
3.3	Dokumenation der Arbeitsergebnisse, Beschreibung des Algorithmus, Korrekturlesen, LaTeX-Konfiguration

Tabelle 4.3: Arbeitspaketdetails

4.2 Benutzerhandbuch

Als Vorgabe bestehen zwei Ordner mit einigen Fotoaufnahmen und den dazugehörigen Temperaturmatrizen[1] sowie die Ausführbare Datei `streetview`. Der Ordner `input320` enthält Temperaturmatrizen mit der Auflösung 320x240 der Ordner `input640` enthält Temperaturmatrizen mit der Auflösung 640x480.

[1] Die Temperaturmatrix muss jeweils wie das dazugehörige Bild benannt sein. Aus ihr wird das Wärmebild generiert. Als Encoding wird ANSI verwendet.

4.2.1 Daten laden und speichern

Minimal muss der zu verarbeitende Ordner angegeben werden. Für alle weiteren Parameter werden die Standard Werte für `input640` geladen.

```
./streetview -d input640/
```

Standardmässig werden die zu verpixelnden Teile rot markiert. Um in den **Verpixelungsmodus** zu schalten wird `-p` verwendet.

```
./streetview -d input640/ -p
```

Um das **Logging** anzuzeigen wird `-v` verwendet.

```
./streetview -d input640/ -v
```

Standardmässig wird der **Zielordner** `out/` gewählt. Dieser kann mit `-o` selber definiert werden.

```
./streetview -d input640/ -o output/
```

Um den Ordner `input320` zu laden werden folgende Parameter benötigt.

```
./streetview -d input320/ -w 320 -h 240 -s 2.79 -x 582 -y 384
```

Dies ist nötig um die Wärmebildinformationen mit dem normalen Bild in Übereinstimmung zu bringen. Dazu gehört die Breite und die Höhe des Wärmebildes `-w -h`, um wieviel das Wärmebild aufskaliert werden muss `-s` und die Kooridinaten, wo die beiden Bilder übereinstimmen `-x -y`. Diese Informationen werden auch verwendet um den entsprechenden Bereich des Originalbildes auszuschneiden.
Um den relevanten **Temperaturbereich** selber zu definieren wird `--mintemp --maxtemp` verwendet.

```
./streetview -d input640/ --mintemp 5 --maxtemp 40
```

Die **Verkleinerung des Temperaturbereichs** während der Aufsplittung erfolgt standardmässig von beiden Seiten um 0.5°C `--range-dec-mode both`. Falls eine Verkleinerung des Bereichs nur vom unteren Grenzwert gewünscht ist wird die Option `lower` verwendet.

```
./streetview -d input640/ --range-dec-mode lower
```

Wie der Temperaturbereich auf die Pixelfarben aufgeteilt wird, kann mit `--temperature-distribution` bestimmt werden. Standardmässig wird `triangle` verwendet, es steht aber auch die Option `linear` zur Verfügung. Siehe dazu Abb. 4.1

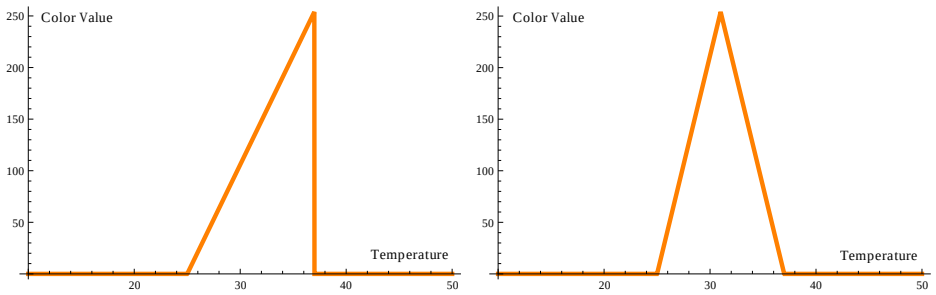

Abbildung 4.1: Temperaturbereich Verteilung: Links `triangle`, rechts `linear`

4.2.2 Filter

Bis jetzt wurden die vordefinierten Filter geladen. Die Auswahl kann mit `--filters` selber definiert. Mögliche Optionen sind:

- `s` - Maximalfläche (Aufsplittung)
- `c` - Kompaktheit (Aufsplittung)
- `m` - Minimalfläche
- `t` - Temperatur Variation
- `f` - Gesichtsfarbe
- `e` - Ellipse

Die vordefinierte Option ist `--filters smte`. Zuerst werden alle Aufsplittungsfilter, anschliessend alle Ausschlussfilter in der angegebenen Reihenfolge ausgeführt. Das nächste Beispiel splittet zuerst alle Konturen mit **zu grosser Fläche** auf und entfernt anschliessend alle Konturen mit **zu kleiner Fläche**.

```
./streetview -d input640/ --filters sm
```

Um nun selber das Kriterium für die zu grosse Fläche zu wählen wird `--max-area` gewählt, entsprechend für die zu kleinen Konturen `--min-area`. Alle Pixel-Werte beziehen sich hierbei auf die Grösse des Wärmebildes, müssen also auf der Fotoaufnahme um den Skalierungsfaktor `-s` dividiert werden.

```
./streetview -d input640/ --filters sm --max-area 500 --min-area 100
```

Nachfolgend Beispiele aller übrigen Filter mit ihren Standardwerten.

Kompaktheit Jede Kontur die den `--compactness-threshold` unterschreitet und eine Fläche grösser als 500 Pixel hat, wird aufgesplittet.

```
./streetview -d input640/ --filters c --compactness-min-area 500 --compactness-
    threshold 0.01
```

Temperatur Variation Jede Kontur welche den `--temperature-variation-threshold` überschreitet wird ausgeschlossen.

```
./streetview -d input640/ --filters t --temperature-variation-threshold 0.2
```

Gesichtsfarbe Jede Kontur die den `--facecolor-threshold` unterschreitet wird ausgeschlossen.

```
./streetview -d input640/ --filters f --facecolor-threshold 0.2
```

Ellipse Jede Kontur welche eine Ausrichtung ausserhalb von `--ellipse-min-angle` bis `--ellipse-max-angle` hat und `--ellipse-width-height-threshold` unterschreitet wird ausgeschlossen.

```
./streetview -d input640/ --filters e --ellipse-width-height-threshold 0.2 --
    ellipse-min-angle 60 --ellipse-max-angle 120
```

4.2.3 Manual Page

Das erstellte Kommandozeilentool verfügt des weiteren über eine Manual Page welche im Ordner `32bit` oder `64bit` über `man ./man/streetview` aufgerufen werden kann.

4.3 Persönliche Berichte

4.3.1 Philipp Eichmann

Die Wahl des Studienarbeitthemas viel mir leicht. Noch im vergangenen Frühlingssemester habe ich Prof. Oliver Augenstein auf eine Studienarbeit zum Thema Computer Vision angesprochen, worauf er eine Aufgabenstellung für diese Projekt ausgearbeitet und ausgeschrieben hat. Roman und ich waren sofort begeistert.

Zu Beginn des Projekts war zunächst nicht klar, wieviel wir während der Umsetzungsphase erreichen würden, zumal wir uns zuerst in eine Thematik einarbeiten mussten, die uns beiden bis dahin noch fremd war.
Die Analysephase war äusserst spannend. Begleitet von viel Literatur, Wärmebildern und den wertvollen Inputs von Prof. Oliver Augenstein erarbeiteten wir uns eine gründliche Einsicht in die Problemstellung. Das Lesen diverser technischer Artikel und Papers verhalf uns zu einem besseren Verständnis. Die Image Processing Themen waren teilweise sehr komplex. Ich profitierte jedoch enorm davon, die theoretischen Konzepte zu untersuchen und anschliessend praktisch anzuwenden.

Unser Betreuer liess uns jeweils viel Freiheit bei der Bearbeitung der verschiedenen Themen und wir konnten den Ablauf des Projekt grösstenteils selbst steuern. So entschieden wir uns beispielsweise, dass die Image Registration den Rahmen dieser Arbeit sprengen würde und konzentrierten uns deshalb auf die Reduktion von Artefakten.

Erfreulich war, dass uns die Firma emitec aus Rotkreuz mehrmals verschiedene Wärmebildkameras erklärt und ausgeliehen hatte. Herr Dettling und Herr Wüthrich waren stets interessiert an unserer Arbeit und bemüht, uns ihr Material zur Verfügung zu stellen.

Besonders Spass gemacht hat mir zudem die Implementierung mit C/C++. Es war das erste Mal, dass wir diese Technologie produktiv für ein Projekt einsetzen konnten. Obwohl das Setup mit Eclipse CDT, OpenCV, Boost aufwendig und etwas mühsam zu konfigurieren war, traten während des Projekts kaum nennenswerte Technologieprobleme auf. Einzig die teils exotischen Compilerfehlermeldungen waren etwas gewöhnungsbedürftig. Oft entstanden aber daraus gute Diskussion, wie z.B. wo besser Referenzen als Pointer verwendet werden sollten oder wo ein `const` Sinn macht, wo nicht.
Auch in allen anderen Belangen war die Zusammenarbeit mit Roman ausgesprochen angenehm und konstruktiv. Wir ergänzten uns prima und konnten uns in den vielen Gesprächen immer einigen. Kritische Teile der Software programmierten wir zu zweit, so dass beiden die Architektur während des gesamten Projektverlaufs immer klar war.

Das Projekt stellte sich als zeitaufwendiger heraus als wir anfänglich dachten. Hauptgrund war vorallem die intensive Analysephase und die qualitativen und quantitiven Beweise der verschiedenen Filter. Um diese gut zu beschreiben, war viel unproduktiver Code zur Evaluation nötig. Zudem war es nicht ganz einfach, die Funktionsweise der Filter und die Ergebnisse aus der Analysephase kompakt und dennoch verständlich in Worten zu fassen. Obwohl als Endergebnis nur ein Protoyp geplant war, waren wir bemüht, eine saubere Architektur zu entwerfen und den Gebrauch des Tools mit ManPage und Benutzerhandbuch gut zu dokumentieren. Dies führte dazu, dass wir mehr Zeit als geplant in unser Projekt investierten, sich aber der Mehraufwand zu unserer Zufriedenheit sehr gelohnt hat.

Insgesamt war es für mich ein enorm spannende Arbeit, von der ich aus diversen Bereichen viel mitnehmen kann.

4.3.2 Roman Giger

Bei der Auswahl der Studienarbeit entschieden wir uns für ein sehr interessantes Themengebiet, welches viel Neuland beinhaltete. Dementsprechend war der Profit, den wir daraus schlagen konnten sehr hoch. Die Unterstützung von Prof. Oliver Augenstein ergänzte unser Team sehr gut, indem er uns neue Denkanstösse und mathematische Konzepte aufzeigte.

Am Anfang las ich mich ins Themengebiet ein und installierte die OpenCV Library. Dies klappte ziemlich gut, wahrscheinlich nicht zuletzt, weil wir in unserem Team mit Debian und Ubuntu arbeiteten.

Die Analysephase dauerte länger als bei anderen Projekten, die Mittel und die Möglichkeiten kristallisierten sich Woche für Woche mehr heraus. Es gab Wochen, da kam ich nicht vom Fleck, dafür gelang Philipp ein Durchbruch und umgekehrt. Wir arbeiten in der Analyse oft zusammen. Sobald es aber möglich war teilten wir die Aufgaben auf. Im Nachhinein wären mehr Aufteilungen möglich gewesen. Uns war immer wichtig, dass wir auf dem selben Stand sind und hatten daher viele Gespräche.

Die OpenCV Library ist in C/C++ geschrieben. Für mich war es das erste Projekt mit C/C++. Im Gegensatz zu Java oder C# musste hier vermehrt aufs Memory Management geachtet werden. Für mich war es sehr spannend zu beobachten, wie sich das direkt auswirkt. Anfangs unerklärliche Effekte, da beispielsweise Memory zu früh freigegeben wurde, konnten danach genauestens nachvollzogen werden.

Der Prototyp war ein sequentieller Ablauf von Funktionen. Den ersten Architekturansatz realisierten ich mit Philipp im Pair-Programming damit wir uns eine gemeinsamen Programmier-Stil aneignen konnten. So konnten wir auch unser Wissen über C/C++ zusammenlegen. Ich konnte viel von Philipp profitieren, da er sich im Vorfeld mehr für C++ interessierte als ich.

Der Kontakt mit der Firma emitec aus Rotkreuz war von grossem Wert für unser Projekt. So konnten wir mit ihren Wärmebildkameras Daten für unser Projekt generieren. Es war spannend einen Einblick in den Einsatzbereich und die Funktionsweise von Wärmebildkameras zu erhalten.

Da die Analysephase relativ lange gedauert hat, war die Zeit für die Implementierung und die Dokumentation etwas knapp bemessen. Um aber nicht Gefahr zu laufen, die Dokumentation in den letzten Nächten zu schreiben, einigte ich mich mit Philipp auf Wochenendarbeit. Dies machte sich im Endeffekt bezahlt, da noch einiges an Code geschrieben werden musste, um die einzelnen Filter zu dokumentieren. Hier machte sich auch das Versionsmanagement mit `git` bezahlt. So konnte auf dem Doku-Branch gehackt werden, während das Refactoring auf Hochtouren lief.

Ich blicke auf intensive, lehrreiche und auch spassige 14 Wochen zurück.

Abbildungsverzeichnis

Literaturverzeichnis

[AOCV11] Robert Lagani Re. *OpenCV 2 Computer Vision Application Programming Cookbook* -. Packt Publishing, Limited, Birmingham, 2011.

[BOOY06] Booyabazooka. Parallax example. `http://en.wikipedia.org/wiki/File:Parallax_Example.svg`, 05 2006. [Online; accessed 11-December-2011].

[BROWN92] Lisa Gottesfeld Brown. A survey of image registration techniques. Technical report, Dept. of Computer Science, Columbia University New York, 1992.

[DIP08] Wilhelm Burger and Mark J. Burge. *Digital image processing - an algorithmic introduction using Java.* Springer, Wiesbaden, 1st edition. edition, 2008.

[FLIRINF11] FLIR. Thermografie und Infrarotlicht. `http://www.flir.com/cs/emea/de/view/?id=41536`, 2011. [Online; accessed 12-December-2011].

[FLIRMAN10] FLIR. *User's Manual - FLIR B series, FLIR T series.* FLIR, a460 edition, 7 2010.

[FLIRSC655] FLIR SC655. FLIR SC655 Wärmebildkamera. `http://www.flir.com/cs/emea/de/view/?id=41415`, 2011. [Online; accessed 06-December-2011].

[FLIRT640] FLIR T640. FLIR T640 Wärmebildkamera. `http://www.flir.com/cs/emea/de/view/?id=41437`, 2011. [Online; accessed 06-December-2011].

[FTOZ11] Filipe António Gonçalves Tomaz. Skin Segmentation. `http://w3.ualg.pt/~ftomaz/fr/fr.php`. [Online; accessed 07-December-2011].

[GENICAM] EMVA. Genicam standard. `http://www.emva.org/cms/index.php?idcat=27`. [Online; accessed 18-December-2011].

[JSSG01] Jamie Sherrah and Shaogang Gong. Skin Colour Analysis. `http://homepages.inf.ed.ac.uk/rbf/CVonline/LOCAL_COPIES/GONG1/cvOnline-skinColourAnalysis.html`, 2001. [Online; accessed 06-December-2011].

[KAL07] Kalan. Hue scale 0—360°. `http://de.wikipedia.org/w/index.php?title=Datei:HueScale.svg&filetimestamp=20070128094006`, 01 2007. [Online; accessed 11-December-2011].

[LOCV08] Gary Bradski and Adrian Kaehler. *Learning OpenCV - computer vision with the OpenCV library.* O'Reilly Media, Inc., Sebastopol, CA, 1. aufl. edition, 2008.

[SCHAP01] Robert E. Schapire. The boosting approach to machine learning - an overview. Technical report, AT&T Labs - Research, Shannon Labratory, 2001.

[STOOP11] Ruedi Stoop. Skript - wissensbasierte systeme. Technical report, Institute of Neuroinformatics, University of Zurich, 2011.

[VIOLA01] Paul Viola and Michael J. Jones. Robust real-time face detection. Technical report, Microsoft Research, Redmond, WA 98952, USA, 2001.

[WHIOM06] Jacob Whitehill and Chrisitan W. Omlin. Haar features for facs au recognition. Technical report, Dept. of Computer Science, Western Cape University, 2006.

Glossar

Domain Specific Language Eine DSL ist eine für eine bestimmte Problemlösung entworfene Programmiersprache. Wie z.B. SQL für Datenbankabfragen.

Emissionsgrad Ein Emissionsgrad von 1 bedeutet 100%-iges absorbieren von elektromagnetischer Strahlung, welche zu 100% wieder als Wärmestrahlung an die Umgebung zurückgegeben wird. Ein Emissionsgrad von 0 bedeutet 0%-iges absorbieren also 100%ige Reflektierung und entsprechend keine Wärmeentwicklung. Dies ist ein wichtiger Parameter zur Temperaturberechnung mittels Infrarotstrahlung.

False-Negative Etwas das fälschlich als falsch angesehen wird.

False-Positive Etwas das fälschlich als richtig angesehen wird.

Faltungskern In der Mathematik und der Bildbearbeitung wird mit Faltungskern die Matrix bezeichnet, welche zur Faltung verwendet wird.

Kernel Siehe Faltungskern

Neuronale Netze Ein sehr stark vereinfachtes Modell eines menschlichen Gehirns. Mittels Computerhilfe werden Vorgänge des Gehirns abgebildet. Der grosse Vorteil solcher Netze ist ihre Lernfähigkeit. Mit genügend Testdaten lässt sich so eine Entscheidungsfähigkeit trainieren.[STOOP11]

Punkt-Operationen Im Gegensatz zu einem Filter(Siehe auch Faltungskern) wird bei Punkt-Operationen jeder Punkt unabhängig von den anderen berechnet.

Thermografie Infrarotstrahlen werden mittels Wärmebildkamera in elektronische Signale umgewandelt, welche anschliessend mit einem bildgebenden Verfahren fürs menschliche Auge sichtbar gemacht werden. Diese resultierenden Bilder lassen kleinste Temperaturunterschieden erkennen.