

Verwendung einer Thermografiekamera im Human-computer interaction Kontext

Florian Kletz, Jorin Kleimann

Hochschule für Angewandte Wissenschaften Hamburg
Fakultät Technik und Informatik, Department Informatik
Berliner Tor 7, 20099 Hamburg, Germany
{florian.kletz,jorin.kleimann}@haw-hamburg.de
<http://www.haw-hamburg.de>

Zusammenfassung. Diese Arbeit beschäftigt sich mit dem Einsatz von Thermografiekameras in Human-computer interaction Umgebungen. Neben konkreten Software-Implementierungen zur Persistierung von Temperaturrohdaten und zur Kalibrierung und Erkennung von Gesichtsmerkmalen in Thermalbildern wird zudem auf die dafür verwendeten Techniken und Methodiken eingegangen.

Schlüsselwörter: Thermografiekamera, Thermografie, HCI, mentale Belastung, Infrarot, Kamerakalibrierung, Bilderkennung

Abstract. The following work deals with the usage of thermography cameras in human-computer interaction environments. In addition to specific software implementations for the persistence of temperature raw data and for the calibration and recognition of facial features in thermal images, the techniques and methodologies used for this are discussed.

Keywords: Thermographic camera, thermography, HCI, mental workload, infrared, camera calibration, feature extraction

Inhaltsverzeichnis

1	Einleitung	1
1.1	Zielsetzung	1
1.2	Gliederung	1
2	Projekte	3
2.1	Thermal Bike Recording	3
	Projektaufbau:	3
	Architektur:	4
	Integration / Kommunikation:	4
	Laufzeitsicht:	6
2.2	IR-Vis	7
	Projektaufbau:	7
	Architektur:	8
	Laufzeitsicht:	9
	Zusätzliche Komponenten:	9
3	Kamerakalibrierung	13
3.1	Grundlagen Kamerakalibrierung mit OPENCV	13
	Extrinsische Kameraparameter:	13
	Intrinsische Kameraparameter:	14
3.2	Vorgehensweise	15
3.3	Fazit	17
4	Detektion im Thermografiebild	18
4.1	Versuche mit OPENCV / EMGU CV	18
5	Zusammenfassung	21
5.1	Ausblick	21
A	Anlagen	24
A.1	Anlage I - (Laufzeitansicht Thermal Bike Recording)	24
A.2	Anlage II - (Laufzeitansicht IR-Vis)	25

Abbildungsverzeichnis

1	Schematische Darstellung des Projektaufbaus	3
2	Setup des Projekts	3
3	Klassendiagramm	5
4	Verteilungsansicht	5
5	Schematische Darstellung des Projektaufbaus	7
6	Setup des Projekts	7
7	Klassendiagramm	8
8	Verteilungsansicht	9

9	Beispiele dynamischer Farbcodierungen	11
10	Umwandlung von Weltkoordinaten in Kamerakordinaten und Pixelkoordinaten [2]	14
11	Versuchsaufbau der Kalibrierung	15
12	Unkalibriertes und übereinander gelegtes VIS/Thermalbild	16
13	Thermografiebild des Schachbretts vor der Optimierung.....	16
14	Thermografiebild des Schachbretts nach der Optimierung inkl. erkanntem Muster.....	16
15	Thermalbild Tracking mit „haarcascade_frontalface_default.xml“	19

Glossar

- Thermografiekamera Eine Kameraart die im nicht sichtbaren Spektrum des Lichts arbeitet. Diese bildgebende Kamera zeichnet mittels Infrarotstrahlung die Oberflächentemperaturen auf. Alternative Bezeichnungen lauten: Thermalkamera, Infrarotkamera oder Wärmebildkamera. Ausführliche Erklärungen finden sich in [1]
- Visuelles Bild Mit dem Visuellen Bild oder kurz *Vis-Bild*, wird in dieser Ausarbeitung ein Bild verstanden, das von einer Kamera (Videokamera, Webcam, etc.) aufgezeichnet bzw. erzeugt wird.
- Thermografiebild Mit dem Thermografiebild wird in dieser Ausarbeitung ein Bild verstanden, das von einer Thermografiekamera mittels Falschfarbendarstellung aus Temperaturwerten generiert wird.

1 Einleitung

In einer hochtechnisierten Welt wird die Bedienung von technischen Geräten immer komplexer und anspruchsvoller für den Nutzer.[10]

Um die Bedienung von technischen Geräten zu vereinfachen, verändern und zu revolutionieren befasst sich der Forschungsbereich *Human-computer interaction* (HCI) mit neuen Technologien und Verfahren. Ein Ansatz hiervon ist es sog. *Companion Systeme* zu entwickeln, welche individuell auf den Benutzer eingehen und sich hierzu beispielsweise auf die emotionalen Befindlichkeiten des jeweiligen Nutzers einstellen.

Ein *Companion System* kann dazu Informationen des Nutzer aus verschiedenen Quellen benutzen. Hierzu zählen Analysen von bereits gesammelten Daten des Nutzers, aber auch Informationen die während der Bedienung eines Systems aufgezeichnet und in Echtzeit verarbeitet werden. Diese Informationen können sowohl aktiv durch den Nutzer (beispielsweise mittels Gestik, Mimik oder Sprache), als auch durch die reine Anwesenheit vorgenommen werden, ohne dass wissentlich mit dem System interagiert wird.

Diese Ausarbeitung befasst sich mit dem zuletzt genannten Aspekt – einem System das kontaktlos und nicht-invasiv Informationen über den Nutzer aufzeichnet und verarbeitet. Kern der Informationen die hier analysiert werden sind Temperaturdaten, die mittels dem bildgebenden Verfahren der Thermografie von einem Nutzer aufgezeichnet werden, um dadurch ein technisches System zu beeinflussen bzw. zu steuern.

1.1 Zielsetzung

Im Rahmen des Hauptprojekts sollen die im Grundprojekt erarbeiteten theoretischen Konzepte umgesetzt werden. Hierzu sollen konkrete Softwarelösungen implementiert werden und der dazugehörige Projektaufbau entwickelt sowie dokumentiert werden. Für die Softwarelösungen soll sich weiter in das SDK des Herstellers INFRA TEC eingearbeitet werden und dies genutzt werden, um sowohl Temperaturdaten zu persistieren, als auch eine Live-Analyse zu ermöglichen. Darüber hinaus soll sich in die Möglichkeiten der Detektion auf Thermografiebildern eingearbeitet werden.

1.2 Gliederung

Dieser Projektbericht gliedert sich in fünf Kapitel.

Das erste Kapitel befasst sich mit der Einleitung für diese Ausarbeitung. Ebenso wird die Zielsetzung beschrieben.

Das zweite Kapitel ([2 Projekte](#)) beschreibt die während dieses Projekts selbst entwickelten Softwarelösungen. Für diese Softwarelösungen werden neben dem Projektaufbau die technischen Anforderungen und Eigenschaften, wie auch die Realisierung erläutert.

Des Weiteren wurden Komponenten entwickelt, die Teil der beiden genannten Softwarelösungen sind. Diese werden in dem Kapitel ebenso beschrieben.

Im dritten Kapitel ([3 Kamerakalibrierung](#)) wird auf die Methodiken der Kamerakalibrierung eingegangen. Dabei werden technische und optische Grundlagen geschaffen und die dabei angewandte Methodik definiert.

Im vierten Kapitel ([4 Detektion im Thermografiebild](#)) wird die Notwendigkeit einer zuverlässigen Detektierung von Objekten (hier von Gesichtern) erläutert. Ebenso werden in diesem Kapitel Verfahren aufgezeigt, mit denen dies ermöglicht wird.

Das letzte Kapitel ([5 Zusammenfassung](#)) fasst die Ausarbeitung und die im Projekt gewonnenen Erkenntnisse nochmals zusammen. Zudem gibt es in diesem Kapitel einen Ausblick auf mögliche zukünftige Arbeiten.

Angefügt findet sich ein [Glossar](#), das jene Wörter beschreibt, die über den normalen Informatik-Kontext hinausgehen, sowie die obligatorischen [Abbildungs-](#) und [Literaturverzeichnisse](#).

2 Projekte

Im Rahmen dieses Projekts wurden zwei voneinander unabhängige Softwarelösungen entwickelt. Die Software „THERMAL BIKE RECORDING“ dient zum Persistieren von Thermografiedaten und zur Auswertung im pre-Processing, die Software „IR-VIS“ hingegen zur Live-Temperaturanalyse. Die beide entwickelten Softwarelösungen werden in diesem Kapitel beschrieben und erläutert.

2.1 Thermal Bike Recording

Die erste entwickelte Softwarelösung dient als zusätzliche Komponente des EMOTIONBIKE¹ (EMOBIKE)-Projekts der HAW Hamburg. Eine Dokumentation von Sinn und Zweck des EMOBIKE sowie der technischen Implementation und Realisierung findet sich in [6]. Daher wird in dieser Ausarbeitung nicht näher auf diese Aspekte eingegangen.

Diese Komponente zeichnet synchronisiert mit den anderen Komponenten des EMOBIKES Temperaturinformationen der Probanden auf, sobald sie an Versuchen teilnehmen. In der aktuellen Ausprägung beschränkt sich die Software auf das Kommunizieren mit dem EMOBIKE-Controller (über dedizierte Kanäle) sowie auf das Aufzeichnen der sog. IRB Files – den Temperaturdaten pro Frame. Eine Auswertung der Temperaturinformationen erfolgt ausschließlich im pre-Processing.

Neben der eigentlichen Software zum Aufzeichnen der Informationen musste ein Wrapper implementiert werden, um mit dem EMOBIKE-Ökosystem zu kommunizieren. Dieser wird am Ende dieses Absatzes erläutert wird.

Projektaufbau: An den physikalischen Aufbau wurden besondere Ansprüche hinsichtlich der Position der Kamera gestellt. Zum einen durfte die VARIOCAM nicht das Sichtfeld des Probanden einschränken (der Proband schaut während

¹ <http://www.emotionbike.org/>

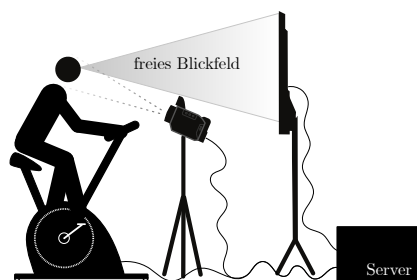


Abb. 1: Schematische Darstellung des Projektaufbaus (exklusive anderer Anbauteile)



Abb. 2: Setup des Projekts

der Versuche auf einen Fernseher), zum anderen musste sich die Kamera neben den anderen Messinstrumenten und Anbauteilen des Ergometers (Kinect, Eyetracker, Fahrradlenker, ...) in den Aufbau eingliedern. Dabei durfte ihre Position weder verändert noch eingeschränkt werden. Als weitere Anforderung wurde identifiziert, dass die Kamera aus zwei Gründen nicht „fest“ ein-/angebaut werden kann. Aufgrund variierender Körpergröße der Probanden muss sich die Kamera in der Y-Achse bewegen/neigen lassen. Des Weiteren muss es möglich sein, die Kamera nach den Versuchsreihen wieder zu entfernen, um sie sicher zu verwahren.

Nach der Analyse der Anforderungen wurden zwei potentielle Aufstellorte identifiziert. Als erste Option wird das Anbringen oberhalb des Fernsehers anvisiert. Hierdurch werden die anderen Anbauteile nicht behindert und durch die verhältnismäßig große Entfernung zum Probanden kann ein großer Bereich abgedeckt werden. Da der Proband (besonders dann, wenn er den Fahrradlenker festhält) automatisch den Kopf nach unten neigt, entfällt diese Position der Kamera, da hierdurch keine frontale Aufnahme des Gesichts ermöglicht wird. Als zweite – und später realisierte – Option wird die Kamera auf einem separaten Stativ unterhalb des Probanden (etwa auf Höhe des Lenkers) montiert (siehe Abbildung 1). Hierdurch wird u. a. der vorher genannten Neigung des Kopfes perfekt begegnet. Problematisch ist hier der begrenzte Platzbereich. In den ersten Iterationen wurde die Kamera etwas aus der X-Achse des Blickfelds verschoben. Grund hierfür waren die baulichen Begebenheiten der anderen Anbauteile. In späteren Iterationen wurde dieser Versatz durch eine neue Halterung für die vorhandenen Anbauteile beseitigt. Dadurch wird ebenfalls eine „verschattung“ von Gesichtsbereichen durch die Nase beseitigt.

Architektur: In diesem Abschnitt wird kurz auf die Architektur der Software und der dazugehörigen Komponenten eingegangen. Die Kommunikation mit der Kamera erfolgt über ein SDK, das bereits in [1] beschrieben wurde. Visualisiert wird die Architektur in Abbildung 3. Bei dem SDK handelt es sich um eine „nicht verwaltete WIN32 API“, weshalb ein dynamisches Laden dieser nötig ist, um sie im .NET-Ökosystem verwenden zu können. Das dynamische Laden der benötigten Funktion wird mittels des Interface IRBGRABINTERFACE vorgenommen und der Hauptklasse zur Verfügung gestellt. Damit sich die Software immer in einem fest definierten Zustand befindet, wird ein Enumerator `internalConnectionStates` genutzt, dessen Status sich mit den entsprechenden Zustandsübergängen verändert. Die Kommunikation mit dem restlichen EMOBIKE-Ökosystem erfolgt über den Message Broker ACTIVE MQ². Hierfür ist in das Projekt der EMOBIKE ACTIVE MQ WRAPPER eingebunden, der im Abschnitt [Integration / Kommunikation](#) beschrieben ist.

Integration / Kommunikation: Neben der physikalischen Integration in das EMOBIKE-Ökosystem – welche bereits im Abschnitt 2.1 beschrieben ist – ist eine softwareseitige Integration nötig, um mit diesem zu kommunizieren.

² <http://activemq.apache.org/>

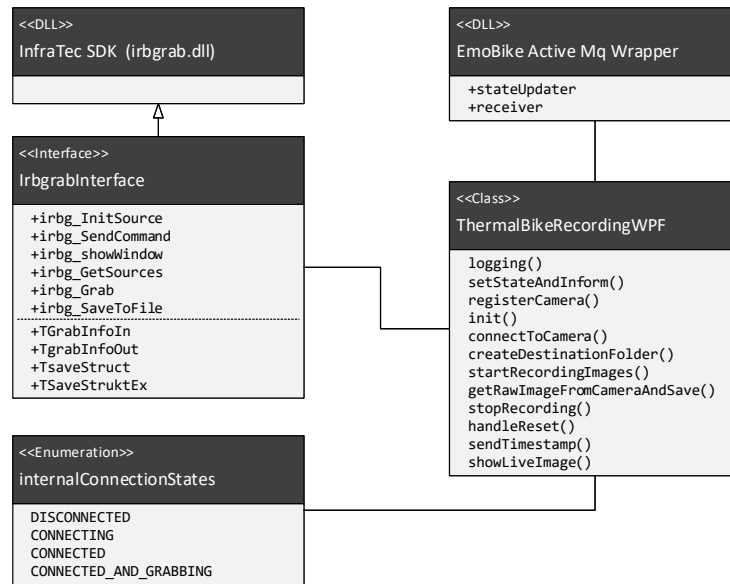


Abb. 3: Klassendiagramm

Die Kommunikation im EMOBIKE-Projekt ist mittels dem Message Broker ACTIVE MQ via JSON-Nachrichten realisiert. Die bisherigen Komponenten sind in verschiedensten Programmiersprachen implementiert, bei dem das .NET-Ökosystem ein Novum darstellt. Um auch zukünftige Komponenten, die im .NET Framework realisiert werden, einfach in das EMOBIKE-Projekt zu integrieren, wurde ein Wrapper als Dynamic Link Library (DLL) implementiert.

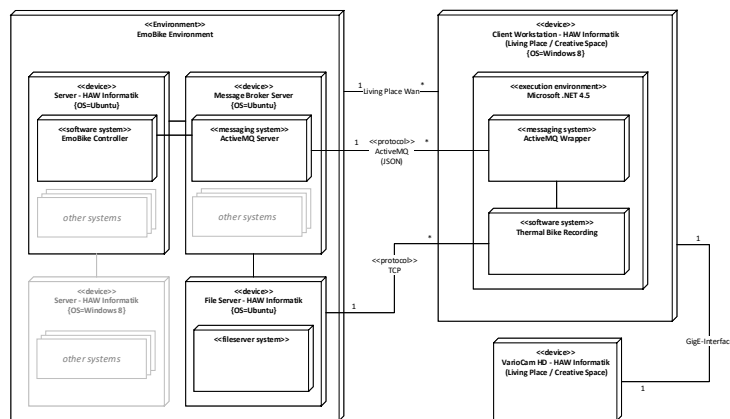


Abb. 4: Verteilungsansicht

Dieser Wrapper – der die Integration zum EMOBIKE bildet – wird nachfolgend kurz erläutert:

EmobikeActiveMq.dll Die DLL erfüllt zwei Funktionen. Mit ihr ist es möglich, dass sowohl Sender als auch Empfänger für ACTIVE MQ Topics fungieren. Die ACTIVE MQ *Servernamen*, *Ports* und *Topics* können hier individuell gewählt werden. Darüber hinaus werden die zu sendenden/empfangenden Nachrichten serialisiert bzw. de-serialisiert, um als JSON-Nachrichten übertragen zu werden. Durch die lose Kopplung können mehrere Instanzen erstellt werden (hier: Status Updater, Heartbeat, sowie Empfänger), die asynchron die Nachrichten verarbeiten.

Die Empfängerinstanz kategorisiert dabei automatisch nach den folgenden Schlüsselwörtern, für die Event Handler angeboten werden:

– initializeMeasure	– getTime
– startMeasure	– unknownMessage
– stopMeasure	– plainMessage
– doReset	

Für den zu übermittelnden Status wird ein Enumerator mit den EMOBIKE-Standardbefehlen angeboten, der nachfolgende Werte zur Verfügung stellt:

– ONLINE = 0	– FINISHED = 5
– READY = 1	– RESET = 6
– MEASURE_PREPARED = 2	– ERROR = 7
– WORKING = 3	– TIME = 8
– WRITING = 4	

Laufzeitsicht: Zum besseren Verständnis der Kommunikation und dem Funktionsablauf der Software stellt [Anlage I - \(Laufzeitansicht Thermal Bike Recording\)](#) die Laufzeitsicht des Kommunikationsprozesses visuell dar. In ihr enthalten sind sämtliche Komponenten, die bei der Kommunikation beteiligt sind. Dargestellt wird sowohl die Initialisierung- und Aufzeichnungsphase als auch das Verhalten, sobald Fehler/Resets von anderen Komponenten gemeldet werden.

2.2 IR-Vis

Eine weitere entwickelte Softwarelösung dient der Echtzeit-Auswertung von Temperaturinformationen. Hierbei ist das Ziel, pixelgenau Temperaturwerte dynamisch abzufragen. Erweitert wird die Software mit dem Bild einer Videokamera, durch das es möglich ist – nach einem Mapping der Informationen der verschiedenen Kameras, welches in Kapitel 3 beschrieben wird – Temperaturwerte auf einem Videobild abzufragen und darzustellen.

Die Software ist so designed, dass sie autark von anderen Softwarekomponenten betrieben werden kann.

Projektaufbau: Bei diesem Projekt wird die Thermoagrafiekamera frontal auf das zu analysierende Objekt ausgerichtet (beispielsweise auf eine Person). Wie eingangs erwähnt, benötigt dieses Projekt eine zusätzliche Videokamera zum Aufzeichnen bzw. Übertragen des Bildes (nachfolgend zur besseren Abgrenzung als „Visuelles Bild“, kurz Vis-Bild genannt). Für diese Kamera (bzw. dessen Bildsensor) ist eine bauliche Nähe zur Thermoagrafiekamera (bzw. des Bildsensor) nötig. Grund hierfür ist, dass der Stereoskopieeffekt möglichst gering gehalten werden soll. In dem konkreten Projekt wird dies mit einem Klemmstativ realisiert, welches an dem Stativ der Thermografiekamera befestigt ist (vgl. Abbildung 5 und 6). Die Videokamera wird lotrecht über dem Objektiv der Thermografiekamera befestigt, d. h. X- und Z-Achse sind an der Front der Objektive identisch. Die Kameras sind lediglich in der Y-Achse verschoben.

Für dieses Projekt wird eine MANTA G-235C Kamera, mit einem 1" 16mm/F1.4 Objektiv der Firma KOWA, als Videokamera verwendet die – wie die Thermografiekamera – über den GigE Vision Interface-Standard ange-

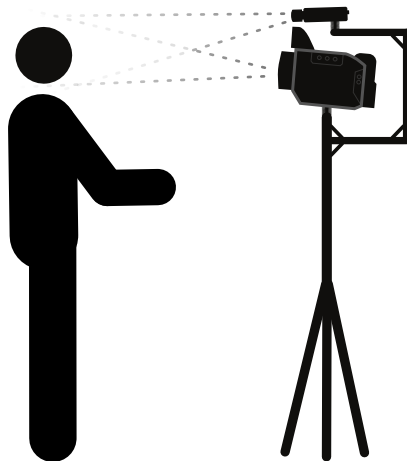


Abb. 5: Schematische Darstellung des Projektaufbaus



Abb. 6: Setup des Projekts

sprochen wird. Das ausgewählte Modell wird mittels *Power over Ethernet* (PoE) mit Strom versorgt, weshalb in dem konkreten Projektaufbau ein separater PoE-Splitter verwendet wird (vgl. Abbildung 8).

Architektur: In diesem Abschnitt wird kurz auf die Architektur der Software und der dazugehörigen Komponenten eingegangen.

Auch dieses Projekt nutzt für die Kommunikation mit der Thermografiekamera das bereits in Abschnitt 2.1 erwähnte SDK. Im Rahmen dieses Projekts wurde für das SDK ein komplett eigenständiger CLR-Wrapper implementiert, auf dessen Eigenschaften und den Aufbau im Abschnitt **Zusätzliche Komponenten** eingegangen wird. Dieser Wrapper (**ThermographyWrapper**) stellt dem eigentlichen Programm (**Ir-Vis**) u. a. die (Thermal)Bild-³ und Temperaturdaten sowie weitere Methoden bereit.

Zum Ansprechen des Videokamera SDKs wird ein Wrapper (**VimbaHelper**) verwendet. Dieser Wrapper ermöglicht es, die Videobilder sowie weitere Methoden (zum Ansprechen der Vis-Kamera) dem Hauptprogramm zur Verfügung zu stellen. Ebenso werden noch zwei weitere Klassen (**CameraInfo**, **RingBitmap**) benötigt, die jedoch nur interne Bedeutung haben. Eine vollständige Übersicht der genutzten Klassen und Methoden findet sich in Abbildung 7, die Verteilungsansicht in Abbildung 8.

³ Das Thermalbild wird mittels Falschfarbendarstellung (vgl. [1]) aus den Temperaturwerten erstellt. Die Farbcodierung sowie technische Details finden sich im Abschnitt **Farbverteilung/Farbschemen**

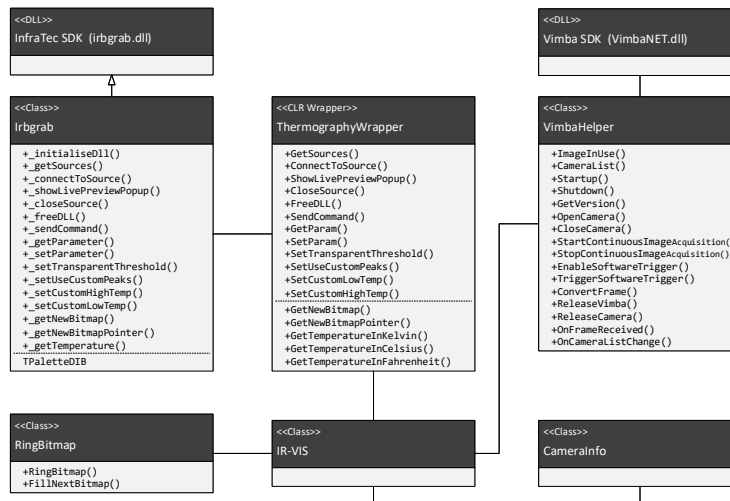


Abb. 7: Klassendiagramm

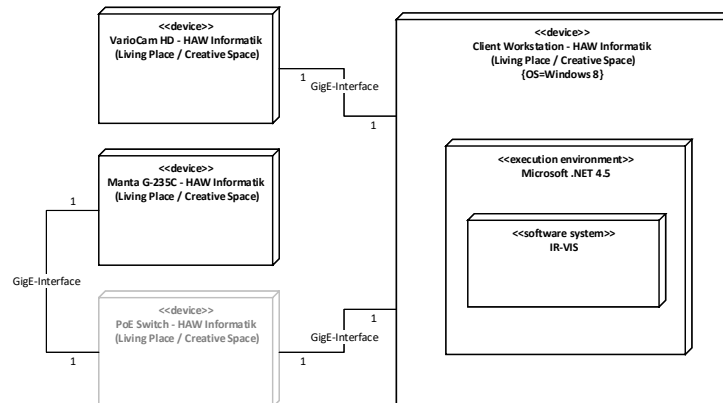


Abb. 8: Verteilungsansicht

Laufzeitsicht: Zum besseren Verständnis der Kommunikation und des Funktionsablaufs der Software stellt [Anlage II - \(Laufzeitansicht IR-Vis\)](#) die Laufzeitsicht des Programms visuell dar. Besonderes Augenmerk wird auf die Initialisierungsphase gelegt. Die Grafik stellt hierzu einen Überblick bereit, welche Komponenten miteinander interagieren müssen.

Zusätzliche Komponenten: Um effizient mit der Hardware der Thermografiekamera zu kommunizieren, wurde ein *Common Language Runtime (CLR)*-Wrapper geschrieben, der wiederum anderen Anwendungen als DLL zur Verfügung gestellt wird. Die Funktionsweise dieses Wrappers wird im nachfolgenden Abschnitt erläutert.

Diese Komponente bildet das Back-End des Thermografie-Anteils der IR-Vis Software.

ThermografieWrapper.dll

Der Wrapper greift mittels C++ auf das SDK des Herstellers INFRATEC zu. Die vom Hersteller bereitgestellte DLL wird dazu in der Initialisierungsphase dynamisch in den Speicherbereich geladen. Hierdurch ist es möglich, die Kamera softwareseitig anzusprechen. Das SDK bietet eine Vielzahl von Methoden an, die genutzt werden können. Eine vollständige Übersicht der Methoden findet sich in [14]. Von diesen ist eine Teilmenge in dem Programm implementiert.

Kern der DLL ist es, die Live-Temperaturrohdaten der Kamera abzurufen und diese weiter zu verarbeiten, mit dem Ziel diese anderen Programmen zur Verfügung zu stellen. Die Kamera liefert die Temperaturrohdaten als ein eindimensionales Array pro Frame, bei dem jedes Element einen Pixel repräsentiert. Bei der Standardauflösung der Kamera beinhaltet dieses Array 786.432 Elemente (1024 * 768 Pixel). Um diese effizient zu verarbeiten, eignet sich die hardwarenahe Programmiersprache C++, die es erlaubt mit Pointern auf die Speicherbereiche zuzugreifen.

Um die Temperaturrohdaten für das menschliche Auge aufzubereiten, wird aus dem eindimensionalen Array durch den Einsatz von mehreren verschachtelten *For-Schleifen* ein zweidimensionales Bild erzeugt. Um die Temperaturwerte sichtbar zu machen, müssen den einzelnen Temperaturen Farben zugewiesen werden. Dieses Verfahren wird im Abschnitt [Farbverteilung/Farbschemen](#) beschrieben.

Ein weiterer Aspekt dieses Programms ist es, zu einer gegebenen Pixel-Position (X;Y basierend) den gemessenen Temperaturwert abzufragen. Hierzu wird mit arithmetischen Operationen der Speicherbereich ermittelt und der entsprechende Temperaturwert in Grad Celsius, Fahrenheit und/oder Kelvin zurückgegeben.

Neben diesen Funktionen werden auch grundlegende Funktionen der Kamera bereitgestellt, wie Setzen der Fokussierung und das Ausführen des Autofokus.

Um die Effizienzvorteile von C++ im .NET-Ökosystem nutzen zu können, wurde ein CLR-Wrapper implementiert, der Anfragen aus Anwendungen beispielsweise in C# an die entsprechenden Methoden in der C++ Klasse weiterleitet. So werden im konkreten Anwendungsfall innerhalb der C++ Klasse die Temperaturinformationen von der Kamera gelesen und aus diesen Bilder generiert, welche über den Wrapper anderen Anwendungen bereit gestellt werden (welche nicht zwangsläufig in C++ implementiert sein müssen). Es genügt ein einfacher Methodenaufruf. Das Zusammenspiel lässt sich sowohl in der [Abbildung 7](#) als auch in der [Anlage I](#) nachvollziehen.

Farbverteilung/Farbschemen Um aus den Temperaturdaten ein für den Menschen interpretierbares Bild zu erzeugen, muss jedem einzelnen Temperaturwert eine Farbe zugeordnet werden. Dieses Verfahren nennt sich Falschfarbendarstellung und wurde bereits in [\[1\]](#) erläutert.

Zugewiesen werden können die Farben nach zweierlei Gesichtspunkten. Eine Option ist es, Temperaturwerten eine explizite Farbe zuzuweisen. So kann eine exemplarische Farbzuzuweisung bei einer zu erwartenden Temperaturspanne zwischen $0,0^{\circ}\text{C}$ und $30,0^{\circ}\text{C}$ die folgende sein: $30,0^{\circ}\text{C} = \text{rot}$; $0,0^{\circ}\text{C} = \text{blau}$ (und dazwischen die Abstufungen zwischen diesen beiden Farben). Bei diesem Beispiel entsteht allerdings für das menschliche Auge das Problem, dass aufgrund der vielen Nuancen eine Zuordnung zwischen Temperatur und Farbe schwierig bis unmöglich ist. Die Temperaturen $23,0^{\circ}\text{C}$, $24,0^{\circ}\text{C}$, $24,3^{\circ}\text{C}$ oder $26,5^{\circ}\text{C}$ würden für das menschliche Auge nahezu identisch aussehen. Somit kann lediglich eine Aussage über eine ungefähre Temperatur (eine Spanne zwischen mehreren Grad) getroffen werden.

Wählt man eine kleinere Spanne, beispielsweise: $30,0^{\circ}\text{C} = \text{rot}$; $25,0^{\circ}\text{C} = \text{blau}$; $20,0^{\circ}\text{C} = \text{grün}$; [...]; $0,0^{\circ}\text{C} = \text{weiß}$; (und dazwischen die Abstufungen zwischen diesen Farben), kann nach dem Betrachten einer Farbe eine sehr viel detaillierter Aussage über die (ungefähre) Temperatur getroffen werden. Hierbei entsteht das Problem, dass der Farbraum endlich ist und sich daher vorab genau überlegt werden muss, welche Farben und welche Temperaturbereiche dargestellt werden sollen.

Hierbei treten folgende Probleme auf. Zum einen muss vorher die zu erwartende Temperaturspanne festgelegt werden und dazu vorab bereits bekannt sein. Zum anderen muss diese Temperaturspanne im besten Fall jederzeit voll ausgeschöpft werden. D. h. in den konkreten Beispielen: Temperaturen dürfen nur in dem Bereich zwischen $0,0^{\circ}\text{C}$ und $30,0^{\circ}\text{C}$ vorkommen und möglichst auch jederzeit sowohl die maximale als auch die minimale Temperatur vorkommen.

Weitaus flexibler – und hier genutzte Variante – ist eine dynamische Zuweisung der Farbe basierend auf den minimalen (T_{min}) und maximalen Temperaturwerten (T_{max}) des Gesamtbildes. Hierzu wird vorab ein Farbspektrum (S) festgelegt (vergleiche die Beispiele in Abbildung 9). Mit folgender Formel lässt sich nun dynamisch ein Farbwert für einen Pixel im Bild (B_i) basierend auf der Temperatur von dessen Pixel (T_i) setzen:

$$x = ((T_i - T_{min}) * (\frac{|S|}{T_{max} - T_{min}})) \quad (1)$$

$$B_i = S_x \quad (2)$$

Hierdurch können alle gemessenen Temperaturen dargestellt werden. Sollte nur noch ein kleiner Temperaturunterschied im Gesamtbild gemessen werden, werden selbst minimale Temperaturunterschiede visualisiert. Sobald etwas Wärmeres oder Kälteres auf dem Bild gemessen wird, ändert sich auch hier dynamisch die Zuordnung der Farbe zur Temperatur. Hierdurch wird es möglich, das Verhältnis der Temperaturen zueinander zu visualisieren.

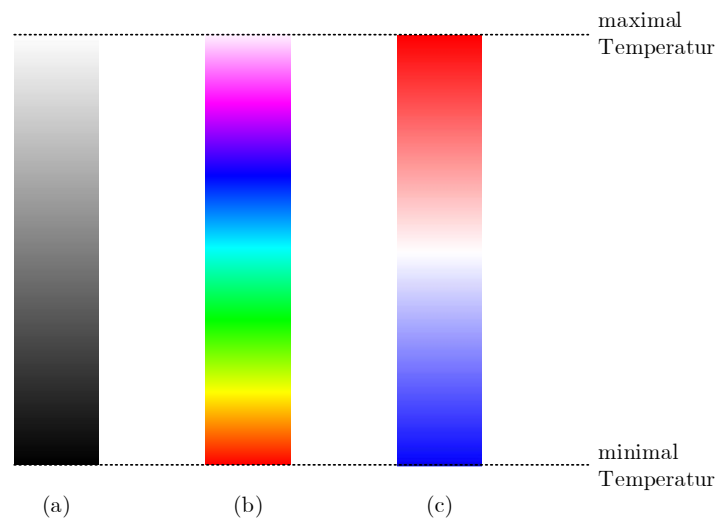


Abb. 9: Beispiele dynamischer Farbcodierungen

In den konkreten Beispielen in Abbildung 9 bedeutet dies für die Spektren (a) + (b), dass jeweils die maximalste/wärmste Temperatur in der Farbe weiß

dargestellt wird. Analog dazu wird die minimalste/kälteste Temperatur in den Farben schwarz bzw. rot dargestellt. Bei dem Farbspektrum (c) sind die Extreme des Bildes hervorgehoben. D. h. es werden besonders warme Bereiche als rot, besonders kalte als blau, dargestellt. Dies ist für den Menschen eine sehr intuitive Farbgebung, da rot und blau in der Farbpsychologie i. d. R. mit Wärme bzw. Kälte assoziiert werden.[15]

3 Kamerakalibrierung

In diesem Kapitel wird die Methodik zur Kalibrierung der Thermografiekamera in Kombination mit der visuellen Kamera beschrieben. Neben den technisch-optischen Grundlagen der von uns verwendeten Methodik wird in diesem Kapitel außerdem die Vorgehensweise und die dabei entwickelte Verfeinerung der Kalibrierung aufgezeigt.

3.1 Grundlagen Kamerakalibrierung mit OpenCV

Um die Kalibrierung zweier Kameras zu erläutern, müssen vorab einige Begriffe erklärt werden, unter anderem die sog. intrinsischen und extrinsischen Parameter. Diese Kameraparameter dienen dazu, das sog. Weltkoordinatensystem auf das Koordinatensystem der Kamera abzubilden.

Wir gehen in unserer Betrachtung davon aus, dass die Kamera bzw. in unserem Beispiel die Thermografiekamera und die visuelle Kamera innerhalb dieses Weltkoordinatensystems positioniert sind. Nimmt man nun ein Objekt mit den beiden Kameras auf, so müssen die realen Koordinaten des Objektes in die Koordinaten der beiden Kameras umgewandelt werden und in einem weiteren Schritt dann wiederum in Bild-Pixel-Koordinaten. Dieses Mapping der einzelnen Koordinatensysteme ist in der Abbildung 10 aufgezeigt.

Extrinsische Kameraparameter: Die extrinsischen Kameraparameter beschreiben die Position und Orientierung der Kamera im Raum. Mit Hilfe dieser Parameter lässt sich der Zusammenhang zwischen Welt- und Kamerakoordinatensystem herstellen. Gemäß unserer Abbildung gibt es 6 extrinsische Parameter:

- Translation in x-Richtung: T_x
- Translation in y-Richtung: T_y
- Translation in z-Richtung: T_z
- Rotation um x-Achse in bestimmtem Winkel (R_x)
- Rotation um y-Achse in bestimmtem Winkel (R_y)
- Rotation um z-Achse in bestimmtem Winkel (R_z)

Diese Parameter beschreiben demnach sowohl die Verschiebung als auch die Drehung der Kamera zum Weltkoordinatensystem. So ergibt sich für die Bildkoordinaten $(u,v,1)$ der Zusammenhang zwischen der Rotationsmatrix (R) multipliziert mit dem jeweiligen Punkt (hier X,Y,Z) addiert mit der Verschiebung um den Translationsvektor vom Mittelpunkt t .

$$\begin{pmatrix} u \\ v \\ 1 \end{pmatrix} = R \cdot \begin{pmatrix} X \\ Y \\ Z \end{pmatrix} + t \quad (1)$$

$$\begin{pmatrix} u \\ v \\ 1 \end{pmatrix} = \begin{bmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{32} & r_{33} \end{bmatrix} \cdot \begin{pmatrix} X \\ Y \\ Z \end{pmatrix} + \begin{pmatrix} t_1 \\ t_2 \\ t_3 \end{pmatrix} \quad (2)$$

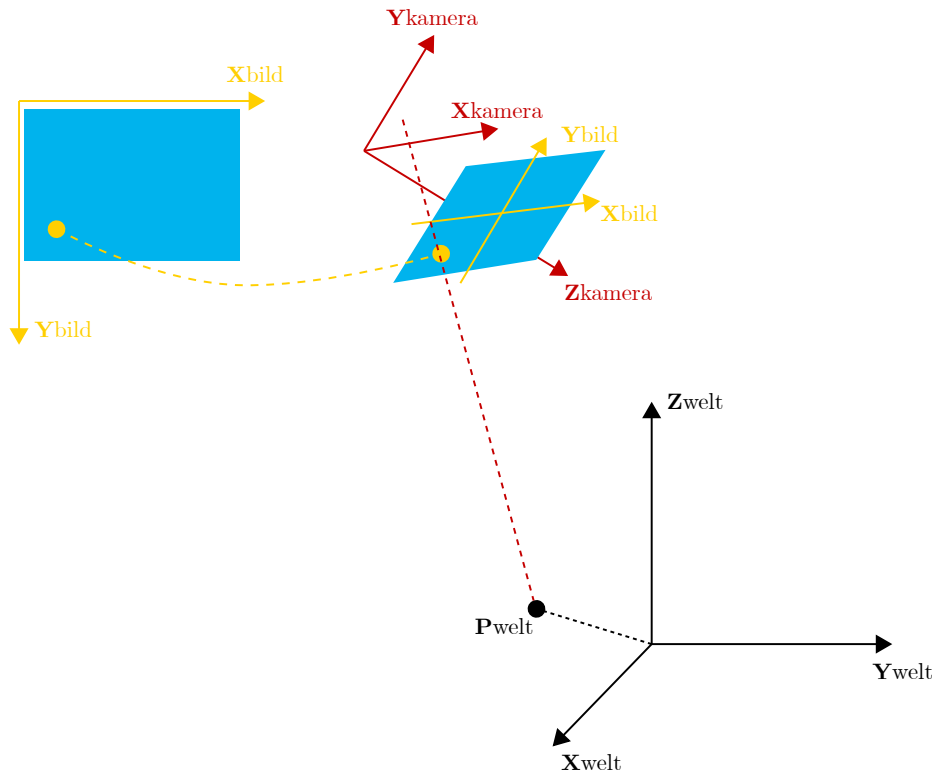


Abb. 10: Umwandlung von Weltkoordinaten in Kamerakoordinaten und Pixelkoordinaten [2]

Intrinsische Kameraparameter: Sind nun die extrinsischen Kameraparameter definiert und ist so eine Beziehung zwischen Welt- und Kamerakoordinatensystem hergestellt, müssen die einzelnen mit der Kamera aufgenommenen Kamerapunkte (Kamerakoordinaten) in einen in einem Bild darstellbaren Pixelwert umgewandelt werden. Die intrinsischen Kameraparameter sind für jede Kamera mit dem dazugehörigen Objektiv feste Werte und beschreiben die „inneren Werte“ der Kamera. Man spricht im allgemeinen von 5 intrinsischen Parametern:

- Brennweite des Objektivs (f)
- x-Wert des Bildmittelpunktes
- y-Wert des Bildmittelpunktes
- Pixelskalierung in x-Richtung
- Pixelskalierung in y-Richtung

In der von uns benutzten Implementierung 2.2 ist es möglich, mit Hilfe von OPENCV⁴ und einem sogenannten Kalibrierungsmuster, welches als Weltkoordinatensystem dient, die extrinsischen und intrinsischen Kameraparameter zu

⁴ <http://opencv.org>

ermitteln. Nach der erfolgreichen Ermittlung werden diese Parameter beider Kameras in einen Zusammenhang gestellt [5]. Wir haben zur Kalibrierung ein Schachbrett mit insgesamt 9x6 Feldern benutzt. OPENCV nutzt den von Zhenyou Zhang entwickelten Algorithmus zur Kamerakalibrierung [16]. Durch die festgelegte Größe und Anzahl der einzelnen Felder des Schachbrettes lassen sich die Weltkoordinaten auf die Koordinaten der Kamera abbilden. Dazu benötigt man Aufnahmen, in denen das Schachbrett sich in möglichst vielen verschiedenen Winkeln des Bildrahmens befindet. Durch die hohe Verbreitung von OPENCV war eine genaue Kalibrierung der visuellen Kamera ohne weiteres möglich. Um jedoch die Thermografiekamera mit demselben Schachbrettmuster sauber kalibrieren zu können, war einige Arbeit nötig. Die dabei verwendete Vorgehensweise unserer Methodik ist im folgenden Abschnitt zu beschreiben.

3.2 Vorgehensweise

Bei einem ersten Proof-of-Concept-Versuch, in dem 100 Aufnahmen des Schachbretts mit der Thermografiekamera gemacht wurden, wurde schnell klar, dass eine saubere Kalibrierung nicht ohne eine Verfeinerung der Methodik möglich ist. Nach einer eingehenden Literatur-Recherche, wurden Ansätze einer möglichen Lösung dieses Problems im Paper mit Titel „Improving calibration of thermal stereo cameras using heated calibration board“ [9] gefunden. Für den Versuch wurde das Schachbrett in DIN-A4-Größe auf ein Brett geklebt. Nach kurzer Bestrahlung (wenige Sekunden) des Schachbretts mit Hilfe eines Wärmestrahlers (ähnlich Baustrahler) besitzen so die schwarzen Kästchen (durch die höhere Lichtaufnahme der schwarzen Farbe) eine höhere Temperatur als die weißen Kästchen (ca. 2,4 Grad Kelvin Differenz). Dieser Effekt bringt zutage, dass das Schachbrett auch im Thermografiebild als solches zu erkennen ist.

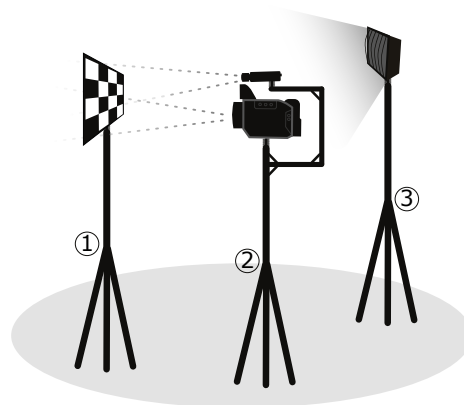


Abb. 11: Versuchsaufbau der Kalibrierung (1) Schachbrettmuster; (2) Vis-/Thermografiekamera; (3) Wärmestrahler

Jedoch ist der Algorithmus nicht in der Lage, die Kanten des Schachbretts zu erkennen. Dies liegt an den deutlich zu unscharfen bzw. zu wenig randscharfen Kanten des Schachbretts im Thermografiebild, da durch die Strahlungswärme nach außen keine klaren Übergänge erkennbar sind. Dieser Effekt ist in Abbildung 13 zu erkennen. In der unteren rechten Ecke des Bildes ist ein Teil dieser Unschärfe zu erkennen.

Um diesem Effekt der Strahlungswärme entgegenzuwirken und den Temperaturunterschied zwischen weißen und schwarzen Kästchen zu erhöhen wurden jeweils die weißen Kästchen ausgeschnitten. Dies hat den Effekt, dass sich die weißen Holzkästchen deutlich langsamer erwärmen als das zuvor vorhandene weiße Papier, was die Temperaturdifferenz erhöht und so genauere Kanten liefert.

Softwareseitig ist speziell für den Fall der Kamerakalibrierung in der ThermografieWrapper.dll (siehe 2.2) die Variable `useCustomPeaks` implementiert. Ist diese aktiv lässt sich ein Schwellwert für die maximale und minimale Temperatur definieren. Setzt man diesen Schwellwert auf die Temperatur der weißen Kästchen (unterster Wert) und die Temperatur der schwarzen Kästchen (als maximalen Wert) und färbt das Bild ein, erhält man ein ausreichend scharfes Bild. Auf diesem Bild lässt sich die Kalibrierung ohne weiteres vollziehen.

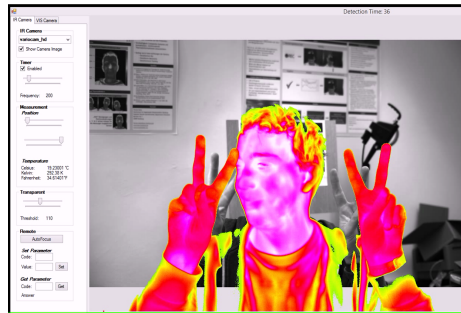


Abb. 12: Unkalibriertes und übereinander gelegtes VIS/Thermalbild

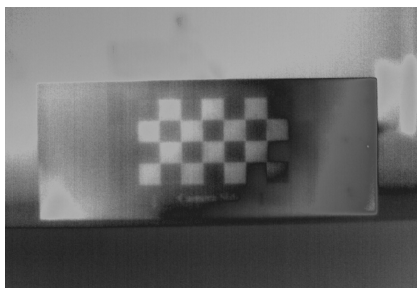


Abb. 13: Thermografiebild des Schachbretts vor der Optimierung

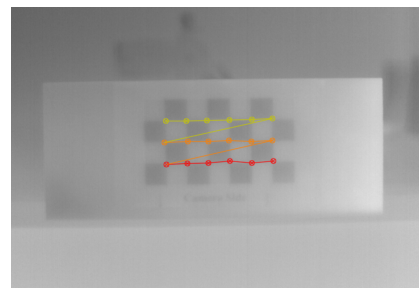


Abb. 14: Thermografiebild des Schachbretts nach der Optimierung inkl. erkanntem Muster

3.3 Fazit

Sind extrinsische und intrinsische Kameraparameter der zu kalibrierenden Kameras ermittelt wäre dennoch ohne die von uns getätigte Überarbeitung des Bildes eine Kalibrierung nicht ohne weiteres Möglich. Durch den definierten Schwellwert bekommt man jederzeit unabhängig von Raumtemperatur oder Farbechtheit des Schachbretts scharfe Kanten. Das oben beschriebene Verfahren kann jederzeit auch auf Thermografiebilder anderer Kameras angewendet werden. Durch die Möglichkeit des Bereitstellens des kalibrierten Thermografiebildes werden der Verwendung weiterer Kameras mit der INFRA TEC VARIOCAM die Tore geöffnet. Es wäre zum Beispiel denkbar, zur Verwendung von detaillierteren Gitternetzen im Gesicht auch eine MICROSOFT KINECT oder ähnliche Tiefenkameras mit der Thermografiekamera zu kalibrieren, um dann auf dem Thermografiebild ein Tracking bestimmter Körperteile zu ermöglichen. Diese Aspekte werden im folgenden Kapitel erläutert.

4 Detektion im Thermografiebild

Eine Erkennung ist für viele Anwendungszwecke in der Thermografie – besonders für jene im HCI Kontext – obligatorisch. Oftmals werden die Temperaturwerte für einen bestimmten Bildausschnitt, einen Bereich oder ein Objekt innerhalb des Bildes benötigt. In den bisher betrachteten Anwendungsfällen wurden meist Temperaturwerte von Personen bzw. von bestimmten *Regions of Interests* (RoIs), beispielsweise des Gesichts, verwendet. Da sich Personen i. d. R. (wenn auch nur minimal) bewegen, ist es sinnvoll und zum Teil unerlässlich, diese RoIs dynamisch zu ermitteln.

Für den Prozess namens „Feature extraction“ gibt es auf regulären Kamerabildern (Vis-Bildern) bereits eine Vielzahl von Algorithmen, Methoden und Frameworks, um RoIs zu extrahieren bzw. zu erkennen. Diese finden sich beispielsweise in [12], [8], [7]. Für Thermografiebilder hingegen ist dieser Bereich (vermutlich aufgrund geringerer Verbreitung) bisher nicht so weit entwickelt und fortgeschritten[4]. Konkret bedeutet dies, dass es kaum Frameworks am Markt gibt, die beispielsweise eine souveräne Gesichtserkennung auf Bildern mit Falschfarbendarstellung ermöglichen. Dennoch ist es möglich, bestehende Algorithmen auch in diesem Bereich einzusetzen.

Nachfolgende Abschnitte listen jene Aspekte auf, die im Rahmen dieses Projekts für das Tracking auf Thermografiebildern erarbeitet wurden.

4.1 Versuche mit OpenCV / Emgu CV

In den ersten Versuchen wurde Tracking (bzw. konkret Detektion) von Gesichtern mittels der Programmbibliothek OPENCV erprobt. Hierzu wurde der EMGU CV⁵ Wrapper verwendet um die Gesichtserkennungsmöglichkeiten mit, zuvor durch die Thermografiekamera generierten, Thermografiebildern bestückt.

OPENCV arbeitet im Gesichtserkennungsbereich mit der *Viola-Jones-Methode* zur Mustererkennung in digitalen Bildern, welche ausführlich in [13] beschreiben ist. In dieser Methode werden sog. *Haar-like features* verwendet, welche mit vorab trainierten Bilderdatenbanken arbeiten. In diesen Datenbanken sind – vereinfacht gesagt – Informationen, die typischerweise bei dem gesuchten Objekt vorkommt, gespeichert. Bei Fotografien von menschlichen Gesichtern beispielsweise die Information, dass der Bereich um die Augen typischerweise dunkler ist als die darunterliegenden Wangenpartien. Dieses Verfahren wurde bereits in [3] erläutert und wird hier ebenfalls verwendet.

Für dieses System gibt es bereits eine Vielzahl angelernter und frei verfügbarer Datenbanken, die im XML Format vorliegenden, sog. *haarcascade*-Dateien. Die wahrscheinlich am häufigsten genutzte Variante hiervon ist die „*haarcascade_frontalface_default.xml*“, die speziell für frontale Aufnahmen von menschlichen Gesichtern konzipiert wurde, die die oben genannten Eigenschaften abprüfen. Im Rahmen dieses Experimentes wurde die Datenbank auf

⁵ <http://www.emgu.com/>



Abb. 15: Thermalbild Tracking mit „haarcascade_frontalface_default.xml“

mehrere durch die Thermografiekamera generierte Bilder angewandt. Im konkreten Fall wurden diese mittels Falschfarbendarstellung generiert. Bei der Generierung wurde das Schema verwendet, bei der die wärmste Temperatur des Bildes weiß, die kälteste Temperatur schwarz eingezeichnet wird (Farbcodierung (a) aus Abbildung 9). Vergleiche hierzu Abschnitt [Farbverteilung/Farbschemen](#).

Erstaunlicherweise war bereits auf diesen Bildern eine relativ zuverlässige Erkennung des Kopfes möglich, wie exemplarisch in Abbildung 15 dargestellt.

Neben der *haarcascade*-Datenbankdatei für frontale Portraitaufnahmen von Personen gibt es noch weitere für Körperteile und Sinnesorgane. Besonders hilfreich in diesem Kontext ist dabei die ebenfalls häufig genutzte Variante „*haarcascade_eye.xml*“, die für die Erkennung von Augen in einem zuvor erkannten Gesicht ausgelegt ist. Mit dieser können zwar sehr gut Augen in Vis-Bildern herkömmlicher Kameras erkannt werden, leider jedoch keine Augen in Thermografiebildern. Grund hierfür ist, dass die Datenbankdateien nicht auf diese besondere Anforderung getrimmt sind. Generell ist jedoch auch ein Erkennen von Augen in Thermalbildern möglich wie [11] zeigt.

Hauptziel, welches in diesem Vorhaben mit dem Erkennen von Augen verfolgt wird, ist die Festlegung von RoI auf dem Kopf, um dort Messungen bestimmter Kopfregionen vornehmen zu können. Hilfreich ist hierzu neben den Kopfproportionen ebenfalls die Position der Augen, um von dieser Position aus beispielsweise die Lage der Stirn zu ermitteln.

Prinzipiell ist es möglich, eigene *haarcascade*-Datenbanken anzulernen und somit eigene Datenbanken für Gesichter, Augen, Nasen etc. zu erzeugen. Das Verfahren hierfür wurde ebenfalls in [3] erläutert. Für das Anlernen wird eine hohe Zahl von Trainingsbildern benötigt, die eine möglichst große Variation aufweisen. D. h. es werden Aufnahmen von Personen unterschiedlichen Alters, Geschlechts, Ethnie, Statur, mit unterschiedlichen Haarlängen, Bartlängen

usw. benötigt. Diese müssen wiederum mit verschiedenen Farbcodierungen aufgezeichnet werden, um ein möglichst robustes Erkennen zu ermöglichen.⁶

Um ein valides und präzises Detektieren zu ermöglichen, wurde eine andere Herangehensweise gewählt. Der Projektaufbau wurde mit einer Vis-Kamera erweitert (Vergleiche den Projektaufbau in Kapitel 2.2). Die oben genannten Datenbanken sind mit Vis-Bilder aus mehreren 10.000 Bildern trainiert worden und bieten somit eine große Variation an Gesichtern, ergo ein relativ zuverlässiges Erkennen von verschiedensten Gesichtern auf Vis-Bildern. Aufgrund des in Kapitel 3 beschriebenen Mappings von Thermografiebildern zu Vis-Bildern wird die Information der Gesichtserkennung auch für das Thermografiebild ermöglicht.

⁶ Findet das Anlernen mit nur einer Farbcodierung statt, kann ein detektieren ebenfalls nur mit exakt der selben Farbcodierung stattfinden.

5 Zusammenfassung

Im Rahmen des Hauptprojekts wurden die theoretischen Erfahrungen des Grundprojekts in praktische Anwendungen umgesetzt. Hierzu wurden vornehmlich mehrere Softwarelösungen implementiert sowie weitere theoretische Konzepte betrachtet und umgesetzt.

Konkret wurden zwei größere Softwarelösungen entwickelt, um Temperaturrohdaten der Thermografiekamera zu verarbeiten. Diese beiden Softwarelösungen sind im Kapitel 2 beschrieben. Die Software THERMAL BIKE RECORDING dient dabei hauptsächlich zum Persistieren von Temperaturrohdaten, die im pre-Processing ausgewertet werden können. Diese Software ist in das EMOBIKE Projekt eingebunden, weshalb eine technische und physikalische Integration nötig war, die ebenfalls erläutert wurde. Die Software IR-VIS dient als Gegenstück zur ersten Software und ermöglicht eine Echtzeit-Analyse der Temperaturdaten. Bei dieser Software wurde der Aufbau mit einer regulären Netzwerkkamera erweitert, sodass Temperaturinformationen mit einem Kamerabild zusammengeführt werden.

Für das Zusammenführen der Informationen mussten die Kameras zueinander kalibriert werden. Die theoretische und praktische Vorgehensweise sowie die entstandenen Probleme wurden im Kapitel 3 betrachtet. Ebenso wurde dort das hervorgebrachte Ergebnis diskutiert.

Als ein weiterer Aspekt wurde im Kapitel 4 die Notwendigkeit und die Möglichkeiten der Detektion in Thermografiebildern aufgezeigt. Dies ist ein wichtiger Aspekt, um dynamisch Temperaturinformationen eines sich bewegenden Objekts zu ermöglichen. Dazu wurden im selbigen Kapitel die Erkenntnisse hierzu aufbereitet.

5.1 Ausblick

Im nächsten Schritt werden die entwickelten Softwarelösungen unter Realbedingungen genutzt. Dazu wird die Software THERMAL BIKE RECORDING erstmalig bei Versuchsreihen des EMOBIKE-Projekts aktiv Temperaturdaten aufzeichnen, die danach ausgewertet werden, um u. a. die bereits in [1] aufgezeigte Eignung für den HCI Kontext zu ermitteln.

Ebenso soll die Software IR-VIS mit all seinen Komponenten in ein reales Projekt integriert werden, um Temperaturdaten auf Vis-Bildern abzufragen. Mögliche Szenarien sind hier der Aufbau eines *Smart Mirrors* der Temperaturinformationen für festgelegte RoIs bereitstellt. Auch Temperaturüberwachungs-Einrichtungen sind denkbar, die beim überschreiten eines Schwellwerts den Nutzer informieren und warnen.

Großes Potential zur Weiterentwicklung liegt im Bereich der Detektion auf Thermografiebildern. Dieser Aspekt bietet eine Vielzahl an Möglichkeiten, um das Detektieren auf Thermografiebildern zu verbessern, es zu erweitern, es robuster zu machen oder zu beschleunigen.

Ein Aufbau eines sog. *Network Video Recorder* der die Thermografiebilder generiert und Nutzern/Softwareentwicklern als Stream zur Verfügung stellt bringt

den Vorteil, dass die Berechnungslast an einem Server (ggf. sogar mittels CUDA) stattfindet und mehrere Softwaresystemen gleichzeitig auf eine Thermografiekamera zugreifen können. Hier wäre eine Implementierung einer eigenen *Thermal-API* für den Zugriff auf den Bildstream und Bilddaten denkbar.

Die Software IR-VIS ist hinsichtlich Performance und Zuverlässigkeit zu optimieren, wenn eine Verwendung über den Einsatz in Testumgebungen hinausgehen soll.

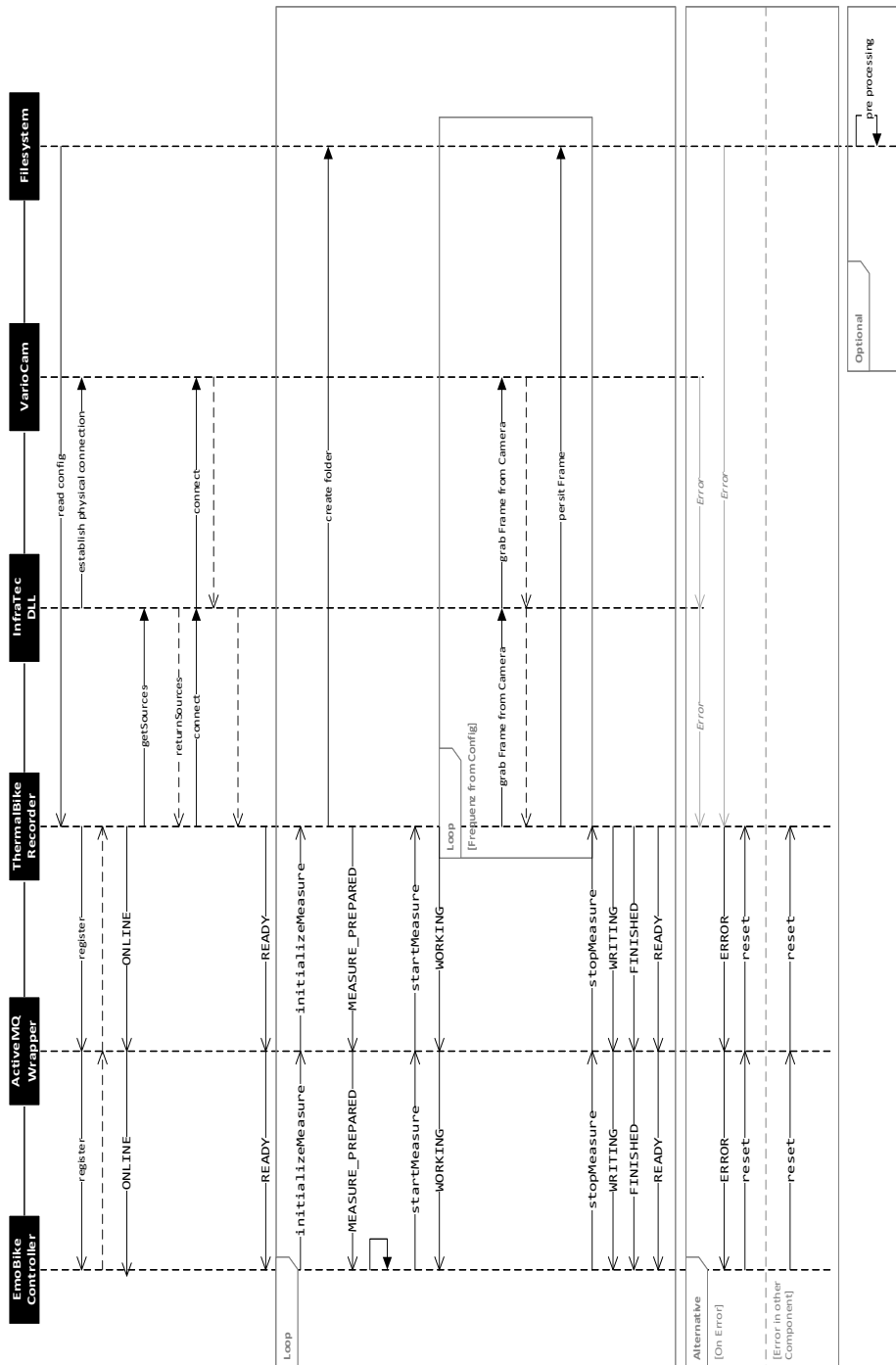
Ebenso steht die weitere Integration in den CREATIVE SPACE⁷ an. Auch hier sind etliche Kombinationen und Verschmelzungen mit Bereits im CREATIVE SPACE befindlichen Technologien ein denkbarer Schritt.

⁷ <http://creative-space.haw-hamburg.de/>

Literatur

1. Jorin Kleimann Florian Kletz. *Evaluierung von Lichtfeld- und Thermografiekameras im Human-computer interaction Kontext*. 2016.
2. Tim Hofmann. *Geometrische Kamerakalibrierung*. Fachhochschule Wiesbaden, 2008/2009.
3. Florian Kletz. *Thermalbilder im Kontext von Companion Systemen*. 2016.
4. Mariusz Marzec, Robert Koprowski, Zygmunt Wróbel, Agnieszka Kleszcz, and Sławomir Wilczyński. Automatic method for detection of characteristic areas in thermal face images. *Multimedia Tools and Applications*, 74(12):4351–4368, 2015.
5. Birgit Möller. *Multi-Mosaikbilder - Ein Ansatz zur ikonischen Repräsentation von Bilddaten aktiver Kameras*. 2015.
6. Larissa Müller, Sebastian Zagaria, Arne Bernin, Abbes Amira, Naeem Ramzan, Christos Grecos, and Florian Vogt. *EmotionBike: A Study of Provoking Emotions in Cycling Exergames*, pages 155–168. Springer International Publishing, Cham, 2015.
7. Erik Murphy-Chutorian and Mohan Manubhai Trivedi. *Head pose estimation in computer vision: A survey*, volume 31, pages 607–626. IEEE, 2009.
8. Dong ping Tian et al. *A review on image feature extraction and representation techniques*, volume 8, pages 385–396. Citeseer, 2013.
9. P. Saponaro, S. Sorensen, S. Rhein, and C. Kambhamettu. Improving calibration of thermal stereo cameras using heated calibration board. In *2015 IEEE International Conference on Image Processing (ICIP)*, pages 4718–4722, Sept 2015.
10. Geschäftsführerin des Scherbaum. *Beyond Computing-Forschungszentren, Gruppen und Seminare*.
11. Andrea Selinger and Diego A Socolinsky. Face recognition in the dark. In *Computer Vision and Pattern Recognition Workshop, 2004. CVPRW'04. Conference on*, pages 129–129. IEEE, 2004.
12. Richard Szeliski. *Computer vision: algorithms and applications*. Springer Science & Business Media, 2010.
13. OpenCV Developers Team. Face detection using haar cascades. 2015.
14. InfraTec GmbH Infrarotsensorik und Messtechnik. *Access to real time digital data with InfraTec SDK*. 2015.
15. Norbert Welsch and Claus Chr. Liebmann. *Farbpsychologie und Symbolik*, pages 53–113. Spektrum Akademischer Verlag, Heidelberg, 2012.
16. Zhengyou Zhang. A flexible new technique for camera calibration. *IEEE Trans. Pattern Anal. Mach. Intell.*, 22(11):1330–1334, November 2000.

A.1 Anlage I - (Laufzeitansicht Thermal Bike Recording)



A.2 Anlage II - (Laufzeitansicht IR-Vis)

