TU WIEN Informatics

3D-Kopfverfolgung und Gestenerkennung mittels eines 8-mal-8 Infrarotsensor-Array

DIPLOMARBEIT

zur Erlangung des akademischen Grades

Diplom-Ingenieur

im Rahmen des Studiums

Visual Computing

eingereicht von

Omar Ismail, B.Sc.
Matrikelnummer 01327702

an der Fakultät für Informatik

der Technischen Universität Wien

Betreuung: Em.O.Univ.Prof. Dr. Walter G. Kropatsch
Mitwirkung: Darshan Batavia, Ph.D.
Dr. techn. Jiri Hladuvka

Wien, 20. Jänner 2024

Omar Ismail	Walter G. Kropatsch

TU WIEN Informatics

3D Head Tracking and Gesture Recognition using an 8-by-8 Array of Infrared Sensors

DIPLOMA THESIS

submitted in partial fulfillment of the requirements for the degree of

Diplom-Ingenieur

in

Visual Computing

by

Omar Ismail, B.Sc.
Registration Number 01327702

to the Faculty of Informatics

at the TU Wien

Advisor: Em.O.Univ.Prof. Dr. Walter G. Kropatsch
Assistance: Darshan Batavia, Ph.D.
Dr. techn. Jiri Hladuvka

Vienna, 20th January, 2024

Omar Ismail	Walter G. Kropatsch

Technische Universität Wien
A-1040 Wien ▪ Karlsplatz 13 ▪ Tel. +43-1-58801-0 ▪ www.tuwien.at

Erklärung zur Verfassung der Arbeit

Omar Ismail, B.Sc.

Hiermit erkläre ich, dass ich diese Arbeit selbständig verfasst habe, dass ich die verwendeten Quellen und Hilfsmittel vollständig angegeben habe und dass ich die Stellen der Arbeit – einschließlich Tabellen, Karten und Abbildungen –, die anderen Werken oder dem Internet im Wortlaut oder dem Sinn nach entnommen sind, auf jeden Fall unter Angabe der Quelle als Entlehnung kenntlich gemacht habe.

Wien, 20. Jänner 2024

Omar Ismail

Danksagung

Zu allererst möchte ich mich bei Em.O.Univ.Prof. Dr. Walter Kropatsch bedanken, welcher mich für Bildverarbeitung begeistern und motivieren konnte - seine Geduld hat letztendlich dazu geführt, dass ich den letzten Schritt meines Diplomingenieurs abschließe.

Diese Diplomarbeit ist ohne die Hilfe anderer nicht möglich gewesen und ich möchte mich hier bei ihnen bedanken. Vielen Dank an die Studienbeihilfe Wien, welche mir überhaupt die Möglichkeit gegeben hat, mein Studium zu verfolgen und abzuschließen. Ohne sie wäre es mir nicht möglich gewesen, mir ein Studium ohne Vollzeitarbeit zu leisten. Ich möchte mich auch bei LEWITT GmbH für ihre Unterstüzung während der Implementierungs- und Experimentenphase bedanken, insbesondere Dr. Christian Walter, Moritz Lochner und Pavol Puffler.

Nach dem Abschluss des praktischen Teils dieser Arbeit, hatte ich große Schwierigkeiten beim Schreiben dieser Arbeit. Hier kamen meine geliebten Personen mit ihrer scheinbar unendlichen Unterstützung ins Spiel: meine Mutter Wafaa Ahmed, meine Geschwister Sherin Ismail, Mohamed Ismail, Rayan Ismail, und Moaz Ismail, und meine Großmutter Aziza Ahmed waren immer da, als ich sie gebraucht habe. Meine unglaublichen Freunde Elias Marold, Ines Burgstaller, Lina Kröncke, Negín Sadeghi, Emilia Jäger, Cornelia Zimmel und Maja Zupančič, haben mich während meinem Studium begleitet, mir den Weg gezeigt und inspiriert, mehr zu erreichen. Gemeinsam haben wir Schwierigkeiten erlebt und gemeistert und ich weiß, dass ich ohne sie nicht dort wäre, wo ich jetzt bin.

Zum Schluss möchte ich mich bei Sophia Hannes bedanken, welche mir mit ihrer nicht endenden Unterstützung, Geduld und Liebe die nötige Kraft gegeben hat, diese Diplomarbeit neben einer Vollzeitstelle zu schreiben und abzuschließen.

Vielen Dank euch allen, die Welt ist ein besserer Ort mit euch in ihr.

Acknowledgements

First and foremost, I want to thank Em.O.Univ.Prof. Dr. Walter Kropatsch for inspiring and motivating me to keep pursuing academia - his patience is what ultimately led to me finishing the last step of my Master's Degree.

This thesis was not possible without the help of others, and I want to acknowledge them here. I want to thank the Austrian Study Grant Authority for making it possible for me to pursue my master's. Without their help, I probably would not have been able to afford to pursue a Masters Degree. I also want to thank LEWITT GmbH for supporting me during the implementation and experimentation phase of this thesis, specifically Christian Walter, Moritz Lochner, and Pavol Puffler.

After being done with the practical part, I have struggled heavily with writing. That is where my loved ones came in and lent me their unending support: my mother Wafaa Ahmed, my siblings Sherin Ismail, Mohamed Ismail, Rayan Ismail, and Moaz Ismail, and my grandmother Aziza Ahmed were always there when I needed them. My amazing friends Elias Marold, Ines Burgstaller, Lina Kröncke, Negín Sadeghi, Emilia Jäger, Cornelia Zimmel, and Maja Zupančič, were with me during my academic pursuit, guided and inspired me to do more. We have struggled and succeeded together and I know that without them, I would not have been where I am right now.

Finally, I want to thank Sophia Hannes, who, with her incredible support, patience, and love, gave me all the energy I needed to write and finish this thesis next to working a full-time job.

Thank you all, the world is a better place with you in it.

Kurzfassung

Kopfverfolgung und Gestenerkennung sind bekannte Problemstellungen im Bereich der Bildverarbeitung mit Lösungen anhand RGB-Kameras oder einer Kombination aus Infrarotsender und -empfänger. In dieser Diplomarbeit wird eine Methode für Kopfverfolgung und Gestenerkennung mit einem 8-mal-8 Infrarotsensor-Array vorgestellt. Dabei wird ein neuartiger Time-of-Flight Abstandssensor verwendet, welcher sowohl finanziell als auch rechentechnisch günstig ist. Zusätzlich werden dank der sehr niedrigen Auflösung des Feldes Privatsphärenbedenken reduziert.

Die Methode besteht aus zwei Teilen; zuerst wird ein Kopf mittels Kreiserkennung und Formeigenschaften in dem kombinierten Amplituden- und Tiefenbild gesucht. Wird kein Kreis gefunden, werden Annahmen über die Form getroffen, um die Position zu schätzen.

Anschließend wird die Distanz des ermittelten Kopfzentroiden verwendet, um einen Raum zwischen Sensoren und berechneter Kopfposition zu definieren (Gestenraum). Dieser Raum wird anschließend von der Gestenerkennung für die Verfolgung von Bewegungen über fünf Bildern verwendet. Falls die Richtung der größten Bewegung eine Geschwindigkeit von mindestens vier Pixel pro Sekunde hat, wird eine Geste erkannt.

Die Experimente werden sind in Feld- (Sensor auf einem Tisch in einem Wohnzimmer mit Tageslicht und Fenster hinter Person) und Laborexperimente (Sensor auf einem Drehtisch in einem Lichtzelt in einem Labor, mit künstlichem Licht von oben) aufgeteilt. Die Ergebnisse der Kopferkennung deuten auf eine durchschnittliche zweidimensionale Abweichung von 2.5 Pixel / 5.7 cm bei einer durchschnittlichen Distanz von 40.3cm (Laborexperimente), bzw. 1.9 Pixel / 4.2 cm bei einer durchschnittlichen Distanz von 42.5cm (Feldexperimente) zum Kopfmittelpunkt (= Nasenspitze).

Für die Gestenerkennung deuten die Ergebnisse auf eine durchschnittliche Erkennungsrate (Geste erkannt, unabhängig von der Richtung) von 33.54% bei Labor- bzw. 22.55% bei Feldexperimenten. Die durchschnittlichen Genauigkeit (Geste und Richtung korrekt erkannt) beträgt 42.33% bei Labor-, bzw. 47.82% bei Feldexperimenten, und die durchschnittlichen Falscherkennungsrate beträgt 28.73% bei Labor-, bzw. 21.54% bei Feldexperimenten.

Abstract

Human head tracking and gesture recognition are both known problems with solutions using RGB-cameras or an infrared emitter/receiver setup. In this thesis, we propose a method for head tracking and gesture detection using an 8-by-8 infrared sensor array. For this, a novel time-of-flight infrared sensor array is employed, which is both financially and computationally inexpensive, while also alleviating privacy concerns due to the very low resolution of the array.

The method is split into two parts: first, a human head is detected using circle detection on the filtered combination of depth and amplitude images. If no circle is detected, shape information is used to estimate the position of the head. To reduce false detection and outliers, the movement of the head is tracked over time.

Using the depth value of the detected centroid, gesture detection then looks for movement in the given space between the sensor and detected centroid depth (gesture space) and tracks it over five frames. If the major movement direction exceeds a speed of four pixels per second, a gesture is detected.

The experiments are split up into field (sensor on a desk in a living room with a window behind the person, daylight) and laboratory (sensor on a turntable in a photography light tent in a lab with artificial ceiling lighting). The results of head detection suggest an average centroid deviation of 2.5 pixels / 5.7 cm at an average depth value of 40.3 cm (laboratory experiments), or 1.9 pixels / 4.2 cm at an average depth value of 42.5 cm (field experiments) from the middle of the head (= tip of the nose).

For gesture recognition, the results suggest an average true detection rate (gesture detected, regardless of direction) of 33.54% (laboratory experiments), or 22.55% (laboratory experiments). The average accuracy (gesture and direction correct) is 42.33% for laboratory experiments or 47.82% for field experiments, and the average false positive rate is 28.73% for laboratory experiments or 21.54% for field experiments.

Contents

CHAPTER 1

Introduction

With the advance of technology, humanity continuously tries to improve and add to the ways we interact and communicate with our devices and by extension, each other. Human Interface Devices (HIDs) like keyboards, computer mice, joysticks, knobs, buttons, as well as touchscreens, work via physical interaction and direct contact. Infrared Head tracking and gesture recognition both are non-physical interaction methods that could enable an alternative or better use of existing and upcoming technology.

As of 2023, a large corpus of literature offers solutions for both gesture recognition and head tracking using RGB cameras, infrared (IR) transmitters and IR cameras, or a combination thereof. Devices like the Microsoft Kinect offer a ready-to-use combination of an RGB camera and IR transmitter/camera setup. The drawback of these methods is that they can be financially and computationally expensive, and - depending on the technology employed - do not work at close distances (e.g. the Kinect, which works best at distances between 1.2 and 3.5 meters, according to its manual). Another problem is the privacy concern: for motion recognition to work, the (high resolution) cameras need to be recording multiple frames, which resembles the behaviour of video cameras. This might hinder the acceptance of these technologies.

In this thesis, we propose a handcrafted method to track a human head and detect gestures using a financially inexpensive 8×8 Time-of-Flight-Infrared (ToF-IR) sensor array. The output of this algorithm for each frame will be the position of the head in (x, y)-coordinates, its distance in millimeters and the type of gesture executed, if one was detected.

Due to technical limitations of the sensor array, the algorithm was created with a distance of up to 73 cm in mind. The sensors used cannot capture the details of a human face due to the very low resolution, and can thus help alleviate privacy concerns. Combined with efficient pattern recognition, the computational costs are also held at a minimum.

With advances in this field, non-physical interaction methods could become more widely accepted and employed.

This thesis is segmented into seven chapters. First, the Introduction in Chapter 1 describes the problem this thesis aims to solve and defines the key terms used throughout. Chapter 2 then presents related work in the area of head detection and gesture recognition, and Chapter 3 introduces the theory of the methodology used in this thesis. Then, the algorithm is described in Chapter 4, and external and internal challenges are discussed in Chapter 5. Finally, the experimental results are presented in Chapter 6, with the conclusion being in Chapter 7.

1.1 Time-of-Flight-Infrared-Sensors

In this thesis, infrared (IR) sensors are categorised into passive and active IR sensors:

Definition 1.1.1 (Active IR Sensor)**.** An active IR sensor works by emitting infrared radiation and measuring the amount of photons (amplitude) reflected, returning a 1-dimensional output.

Definition 1.1.2 (Passive IR Sensor)**.** A passive IR sensor measures the infrared radiation emitted/reflected by objects in the field-of-view (FoV), returning a 1-dimensional output.

Another variable in IR sensors is the wavelength or range of wavelengths they are sensible to. For example, Landsat satellites [RWL+14] or infrared thermometers [LBJP12] operate using long-wavelength infrared (LWIR), which lies between 8 and 14 μm [UVG+14].

The sensor used is an STMicroelectronics VL53L1X attached to a breakout board with a gyro-sensor, as seen in Figure 1.1, which emits and measures photons with a wavelength of 0.94 μm, putting it in the category of near-infrared (NIR) sensors [UVG+14].

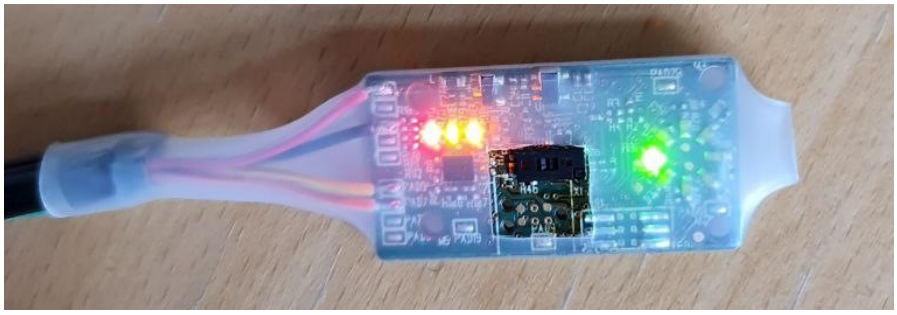


Figure 1.1: The sensor used, an STMicroelectronics VL53L1X (black element at the centre of the board), with the gyro sensor connected to the green LEDs to the right to visualise the pose of the sensor.

In detail, the VL53L1X has a size of $4.9 \times 2.5 \times 1.56$ mm, emits a 940 nm invisible laser (class 1), and receives using a single photon avalanche diode (SPAD) with an integrated lens. Its ranging can reach up to 4m distance (with degrading quality the farther away an object is), with a frequency of up to 50 Hz (not possible in 4×4 or 8×8 mode) and a full field of view (FoV) of 27°.

By combining its IR sensor with a Time-of-Flight sensor, it is possible to calculate the depth by measuring the time the photons need to reflect, thus returning an additional 1-dimensional output.

Definition 1.1.3 (Time-of-Flight Sensors)**.** Time-of-Flight sensors are able to calculate the distance of an object (its depth value) by measuring the time the emitted photons need to reflect, returning a 1-dimensional output.

The same principle has been used by Radar systems, where pulses of electromagnetic waves are emitted and their reflection is measured. Using signal processing and engineering, like with Synthetic Aperture Radars (SARs) [Cut90], the distance, velocity, angle and an image of the object reflecting the pulse can be determined.

Arranging multiple sensors in a rectangular array allows for two 2D outputs: An amplitude image and an infrared image, both in very low size (as of 2023, 8×8).

There are three ranging modes offered by the sensor: short (up to 136/135 cm in low/strong ambient light, respectively), medium (up to 290/76 cm in low/strong ambient light), and long (up to 360/73 cm in low/strong ambient light). The algorithm presented in this thesis uses the long-ranging mode, as it provides the lowest repeatability error (i.e. the highest measurement consistency), which is more important in scientific work.

1.2 Head Tracking

2D head tracking focuses on detecting and tracking a head in a two-dimensional space with two outputs: the vertical (usually the Y-coordinate) and horizontal position (usually the X-coordinate).

3D head tracking (which is where this thesis' objective lies) adds a third dimension to the detection and tracking: depth (Z-coordinate). This is not to be confused with 3D head *pose estimation* [KKL+21], which not only aims to detect and track the three-dimensional position of a head but also its *pose*, i.e. the three-dimensional rotation in the space, adding three further outputs (the rotation for each axis).

Working reliably, head tracking can have many usages in interaction and security: In the context of vehicle safety, a driver's head and distance could be tracked to ensure that the head is always at a safe distance from the steering wheel in case of activation of the airbag. 2D-holograms, visualisations, and 3D-modelling software could simulate three-dimensional behaviour using head movement to manipulate the virtual camera. 3D

holograms could use the distance of the head to control the zoom, making the object bigger when users move their heads closer and vice-versa.

In the context of entertainment, head tracking can be used to increase immersion by adapting camera movement in video games and movies to the head movement of players and viewers.

1.3 Gesture Recognition

In this thesis, a gesture is defined as a movement of the hands and arms to express intent. A big distinction made here is between static and dynamic gestures:

Definition 1.3.1 (Static Gestures)**.** Static gestures describe a certain pose (e.g., certain fingers are extended, while others are not, or the angle at which an arm is extended) and can be captured in one single frame. They have little to no barycentric movement, i.e. the centre of the hand does not move between frames.

Definition 1.3.2 (Dynamic Gestures)**.** Dynamic gestures describe a certain movement (e.g. a swipe from left to right, or "drawing" a shape in the air), necessitating the processing of multiple frames to discern. These gestures exhibit barycentric movement.

This thesis focuses on the latter, intending to detect gestures made by the movement of a flat hand with its palm facing the sensor. By adding the constraint of only parallel movement along the XYZ-axes, we arrive at the definition of the directional gesture recognition employed in this thesis.

Definition 1.3.3 (Directional Gestures)**.** Directional gestures are a subset of dynamic gestures, where the barycentric movement (i.e. the movement of the hand) follows a straight line. We further define directional gestures to move parallel to the X-(horizontal) and Y-(vertical)axes (left, right, up, and down gestures), or perpendicular to them (parallel to the Z-axis, front and back gestures).

Contact-less gesture recognition can be used to control screens and devices without needing to physically touch them, which is especially useful in environments where physical interaction is held to a minimum – like in medical institutions [MHWH17].

As with head detection, entertainment and future technologies like holograms could also benefit from contact-less gesture recognition. In the specific case of holograms, manipulation using the movement and rotation of the user's hand could provide a viable interaction method. If the technology is reliable enough, another possible application could be found in car entertainment systems, as an alternative to conventional touch screens.

CHAPTER 2

Related Work

Since our thesis is using a sensor with amplitude and depth data of small size (8×8) and narrow FoV (27°) – especially compared to systems like the Kinect (with an image size of 640×480 for the v1 and 1920×1080 for the v2 [WS17]) – we can only approximate and discuss the methodology separately. Of relevance are any papers that try to solve head detection/gesture recognition in low resolution and/or video.

Since our technique transforms the sensor data into very low-resolution grey-scale images, we find that methods using RGB cameras are also relevant. Thus, the methods for head detection discussed here will be using IR- (not to be confused with the simpler infrared sensor array discussed in this thesis) and/or RGB-cameras.

For gesture detection, a large corpus of scientific literature is dedicated to the recognition of static hand gestures (Definition 1.3.1). The task of dynamic or directional gesture (Definitions 1.3.2 and 1.3.3) recognition does not have the challenge of finer details such as finger pose, but adds the problem of tracking the hands or arms over multiple frames and ascertaining the flow of their movement.

2.1 State of the Art for Head Tracking

In this section, we consider Head Pose Estimation if it uses infrared imaging for its data acquisition. For performance evaluation and validation of face detection algorithms, publicly available data sets exist, like the Color FERET dataset [Mar00], the CMU Pose, Illumination and Expression (PIE) dataset [SBB01], the SCface dataset [GDG11], the XM2VTS dataset [RMG+99], the VGGFace2 dataset [CSX+18] or the novel IRHP database [LWZ+20]. However, because the sensor used in this thesis is not a camera, but a set of IR emitters and receivers, no public data set for this type of data is available (as of 2023), making direct comparison impossible.

The problem of low-resolution (LR) face recognition has garnered interest in areas like cost-efficient and/or long-distance surveillance. In their review, Wang et al. [WMJW+14] give an overview of the LR face recognition and outline four challenges:

Misalignment, where facial features are misaligned due to e.g. the angle of the camera and thus cannot be matched.

Noise, which gets amplified by a lower resolution due to the higher impact of the camera pose, lighting, environmental and technical issues.

Lack of effective features, making the extraction of features like Gabor or Local Binary Patterns difficult.

Dimensional mismatch, leading to difficulties with some subspace learning methods (further detailed by Choi et al. [CRP08]).

they [WMJW+14] have also named multiple papers that define a lower threshold for reliable face detection, where the required image size lies between 21×16 to 64×48, depending on the methodology used [LP02][BBSV06][FLCS12].

Following that, Wang et al. [WMJW+14] define problems below this size as low-resolution (LR) face recognition (FR), while Wilman WW Zou and Pong C Yuen. [ZY11] call them *very low-resolution* (VLR) face recognition. Since the output of our sensor array can be approximated to a 2D 8×8 image, our problem lies in the domain of VLR face recognition.

Definition 2.1.1 (Very Low-Resolution Face Recognition)**.** Very low-resolution face recognition is the task of recognising a human face with a size smaller than 21×16 pixels.

To make detection feasible and learn features, three possible approaches are defined in the review of Wang et al. [WMJW+14]:

Up-Scaling/Super-Resolution: Image is interpolated using methods like bi-cubic interpolation. With interpolation, no new features are added, but defects like noise are amplified. To combat this, *super-resolution* is employed, where an algorithm learns from a gallery of faces and takes advantage of a face's symmetry and self-similarity. This way, features are added and the effective resolution is increased.

Unified/Inter-resolution feature space: The low-resolution face is projected onto a common space with high-resolution faces and then compared. The issue with this technique is the possibility of noise being introduced with either of the bi-directional functions that project high and low-resolution images to the inter-resolution space.

Down-Scaling: The training data is down-scaled to the resolution of the task at hand, losing features and thus being generally the least ideal option.

Shinji Hayashi and Osamu Hasegawa [HH06] were one of the first to tackle the problem of LR FR and solved it by using an upper body detector and training face recognition on upper body images using AdaBoost. To enhance recognition, the image is up scaled via bi-cubic interpolation, and a new detector is defined using features with height (H) and width (W) bigger than 4. Finally, a support vector machine (SVM) is trained with the output of the detectors, leading to an experimental detection rate of 73%.

The solution of Wilman WW Zou and Pong C Yuen [ZY11] lies in a novel learning method for super-resolution algorithms, where the relationship between the high-resolution image space and the VLR image space is learned. Taking advantage of the self-similarity of the human face and adding two constraints, namely a new data constraint and a discriminative constraint, it is possible to learn a relationship operator which is used to interpolate features and to raise the effective resolution of the face.

In 2004, Krotosky et al.[KCT04] compared the performance of stereo IR cameras with that of Long-Wave Infrared (LWIR) cameras regarding head detection for airbag safety. They found that stereo IR works more reliably than LWIR due to the stability of reflectance regardless of head-wear, while the LWIR heat image would be altered by them. However, in the case where a hand is in the frame and is roughly the same size as the head, LWIR manages to discern a difference in temperature between the human head and hand, while stereo IR struggles to differentiate between the two.

To help with the lack of data for IR head pose recognition, Liu et al. [LWZ+20] created the IRHP database with 145 high-resolution low-light IR head pose images. Based on that, they propose a convolutional neural network (CNN) architecture that extracts and combines high and low-level features. The experimental results suggest a performance better than algorithms like DLDL [GXX+17], IndepCA(HOG) and CartCA/MvCA [CJH+19], and the CNN architecture proposed by Seungsu Lee and Takeshi Saitoh [LS18]. Khan et al. [KKL+21] did an extensive systematic review on that topic.

Opplinger et al. [OGG+22] use a combination of an LWIR camera and a 3D ToF camera to detect living beings. By fusing the two outputs of the cameras, they manage to achieve better results than either one of them separately, since LWIR cameras can detect body heat while 3D ToF cameras can create a three-dimensional image and return the reflectance amplitude of any given being/object.

2.2 State of the Art for Gesture Recognition

Gesture recognition using infrared is a topic which already finds its uses in entertainment and human-computer interaction (e.g. the Leap Motion Controller [WBRF13] or the Microsoft Kinect [HSXS13]). Note, that most of the work found and discussed here will focus on static gestures (Definition 1.3.1) while this thesis focuses on dynamic/directional gestures (Definitions 1.3.2 and 1.3.3).

In 2013, Wojtczuk et al. [WBA+13] used a setup similar to ours for gesture detection; their sensor had four passive LWIR sensors (i.e. non-emitting and measuring body heat)

instead of an 8×8 active (emitting) sensor array, which were aligned along the positive and negative X-/Y-axes for vertical and horizontal movement.

Due to the alignment of the sensors and the use of an aperture, movement/gestures parallel to the vertical or horizontal axes can be detected by tracking the amplitude response along a row or column of sensors [WBA+13]. For example, if the amplitude exceeds a certain threshold first at the left sensors and then at the right sensors, a move from left to right is registered. Conversely, if it first exceeds the threshold at the upper sensors and then at the lower ones, a move from top to bottom is registered.

In their review of IR gesture recognition using machine/deep learning, Rubén E Nogales and Marco E Benalcázar [NB21] discern between two types of hand gestures: static and dynamic gestures, which are defined the same as in this thesis (Definitions 1.3.1 and 1.3.2). To compare and analyse the papers discussed in their review, they [NB21] define five different modules that can be used in combination:

Data Acquisition: How data is acquired and in what modality. According to Rubén E Nogales and Marco E Benalcázar [NB21], there are only two ways for this: spatial position and depth data. Most papers in their review use the spatial position for IR gesture recognition. The setups used for data acquisition in these papers are Kinect, Leap Motion Controller (LMC), Intel RealSence, or an interactive gesture camera, with the Kinect and LMC being the most frequently used.

Pre-Processing: Signal pre-processing to enhance gesture detection, ranging from slight adjustments to the input data to a complete transformation of it. Methods used include dimensionality reduction, normalisation, segmentation, or filters, with normalisation being the most frequently used technique.

Feature Extraction: Extracting the relevant information that can be used to discern categories/classes. The techniques employed in the discussed papers include image segmentation, statistical operations, distance/spatial operations, convolution, Histogram of Gradients (HoG), and chronological-pattern indexing.

Classification: Classifying the input data using unprocessed, pre-processed, and/or extracted information.

Post-Processing: Post-processing of the output to filter false classifications or use the output as input for another module.

During their work, Rubén E Nogales and Marco E Benalcázar. [NB21] have found that the LMC is used more often for detailed gestures including the position of the fingers while the Kinect is used for broader movements of the whole arm, sacrificing accuracy of the exact hand pose. Moreover, all papers discussed employed supervised learning to solve the problem of gesture recognition using heuristics found through trial and error. Unfortunately, a predominant number of papers do not disclose their code, with only 10

of them reporting the processing speed, rendering a proper reproduction of the results impossible.

Tateno et al. [TZM19] proposed gesture recognition using a passive (thermal) 32×24 IR-array. Their technique employs barycentric movement detection and, depending on the movement, uses a CNN to detect static gestures (Definition 1.3.1) or a simple movement detection to detect moving gestures (Definition 1.3.2).

By using the body temperature for background subtraction and normalising the frames afterwards, the movement of the barycenter is tracked along the X- and Y-axis, enabling the detection of up, down, left, and right gestures. Experimental results suggest a detection rate of 97% for moving gesture detection and a total accuracy of 87.5% for static gesture detection.

CHAPTER 3

Theory of Head Detection and Gesture Recognition in Low-Resolution Infrared Amplitude Images

Human head detection is the task of detecting a human head in an image, while head tracking is detecting the same head over a set of frames. Due to a lack of information in the data, an insufficient algorithm, or simply hardware unfit for the task, errors can occur in the tasks of head detection and tracking.

Adding gesture recognition for directional gestures (Definition 1.3.3) to this task brings additional challenges. Due to the assumption that a human head is always in frame, the head needs to be ignored and only the movement of the hand should be tracked.

Conversely, the hand is a potential false positive candidate for head detection, meaning that for head detection, the hands need to be ignored. Thus, this algorithm alternates between detecting the head and the hand while trying to minimise computational complexity.

The functionality of the presented algorithm is limited to a frontal view, with the person facing the sensor at a distance of up to 73cm, using only one hand to execute a directional gesture.

Due to the image size of our sensor array (8×8, see Section 1.1), our data lacks information to accurately and reliably detect a human head in every frame (see Chapter 2). At this image size, a human head and hand are very similar in both amplitude and shape, as seen in Figure 5.1. Furthermore, the limitation of efficiency means that the detection of the head needs to sacrifice accuracy for efficiency.

According to the data-sheet of the sensor [STM18], a higher time budget (thus a longer range mode) will yield a lower repeatability error (Figure 3.1). There, repeatability is defined as "the standard deviation of the mean ranging value of 32 measurements" [STM18]. In other words, repeatability denotes the consistency of the distance measurements by the sensor. It does not denote accuracy – if the sensor is constantly off by exactly 5mm, it will have an accuracy of ±5mm, but a repeatability error of 0.0mm.

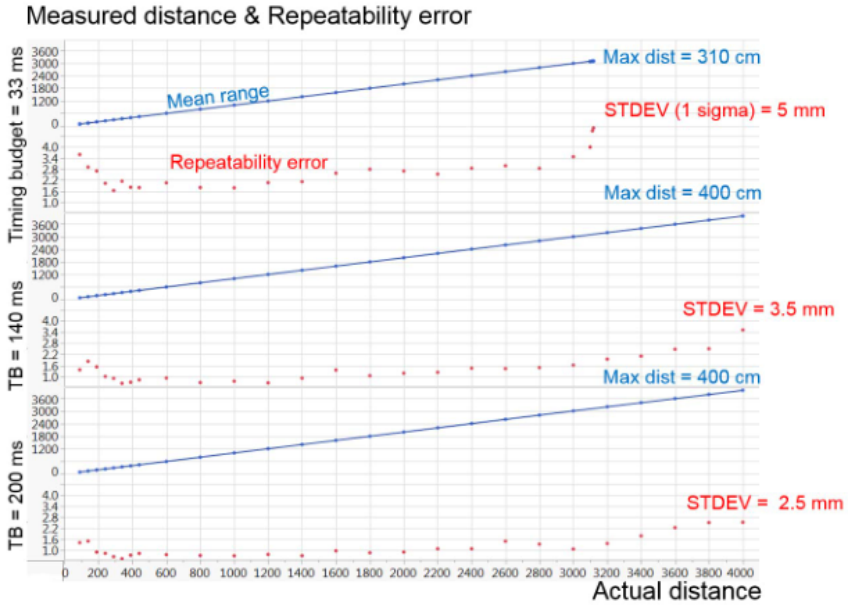
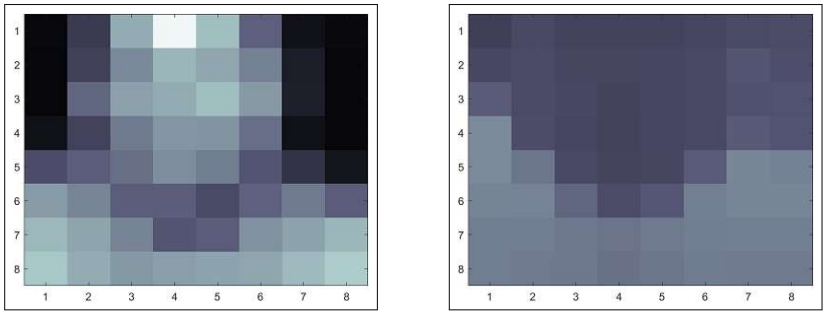


Figure 3.1: Maximum distance and repeatability error vs. timing budget of the sensor. Tested on a target with 54% reflectance and no ambient light, actual distance in mm. TB = timing budget in ms, STDEV = standard deviation. The blue line denotes the mean range, while the red dots are the repeatability error. From the VL53L1X data-sheet [STM18].

A lower repeatability error also reduces sudden erroneous distance measurements, which could be misread as movement by the gesture recognition algorithm presented in this thesis. Thus, we have opted for a long-ranging mode with a timing budget of 200 ms, sacrificing ambient light stability, compared to the short-ranging mode described by the data sheet [STM18]. This however lowers the recommended range to 73 cm [STM18], hence the maximum operating range set in this thesis.

3.1 Theory of Head Detection in Low-Resolution Infrared Amplitude Images

The sensor used in this thesis has two outputs: amplitude from the infrared emitter/receiver and distance from the ToF sensor. An example of the two outputs for the same frame is shown in Figure 3.2.



(a) Amplitude output of the sensor. The amplitude range for this frame is $[25 - 672]$.

(b) Distance output of the sensor. The distance range for this frame is $[202 - 391]$.

Figure 3.2: Examples of the two outputs of the sensor for the same frame. The range of the colour map is $[0 - 700]$.

To improve head detection with the sensor employed in this thesis, amplitude can be beneficial. As seen in Figure 3.3, the National Institute of Standards and Technology has mapped out the reflectance of human skin over various wavelengths. Our sensor emits light in the range of 940 nm, meaning that human skin has a reflectance factor of around 60%. By filtering for an amplitude range, objects with a reflectance different from human skin are excluded.

Without the limitation of maximising computational efficiency, many (or a combination) of the papers presented and discussed in Chapter 2 could be used after pre-processing the input "image" sent by the sensor array. The task then becomes a question of hardware: the better the hardware, the more elaborate/precise the head and gesture recognition can be. One could also use multiple sensors with separate algorithms to recognise heads and gestures.

Since the VL53L1X uses a lens to focus the photons on the sensing array, it is subject to perspective projection [SHB13]. This means that the field of view is akin to a pyramid, as seen in Figure 3.4. Thus, the measured depth and position values are not the same as the real-world distance and position of the object in frame from the sensor.

With the projection from three dimensions to a two-dimensional image, one can use shape abstraction to reach the core assumption of the head detection algorithm employed in this thesis:

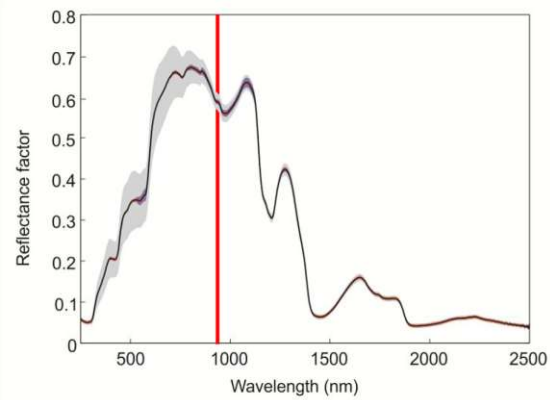


Figure 3.3: Reflectance of the human skin, according to the National Institute of Standards and Technology [CA13]. The thick red line marks the wavelength of the sensor used in this thesis, grey indicates the instrument's uncertainty. Reflectance factor denotes the relative amount of photons reflected ($[0.0, 1.0]$)

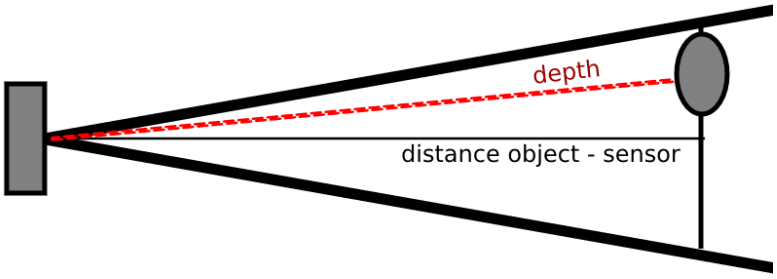


Figure 3.4: A 2D-visualisation of perspective projection exhibited by the sensor. The red line is the depth as measured by the ToF sensor, while the black line at the centre is the distance between the object and the sensor.

Assumption 1. Abstracted into simple shapes, a human head posed upright and facing the sensor is an ellipsoid on top of a rectangle of similar width (neck) on top of a trapezoid/rectangle (torso and shoulders).

This means that – using shape properties as a feature – to achieve the best possible results, the whole head needs to be in the frame at all times. Taking the limitation of the maximum recommended distance (73 cm) into account, the shoulders and torso could appear in frame, but not the lower body/legs. Under these conditions, multiple approaches can be taken:

For example, a centroid is the "balanced" centre of a given shape. One simple implementation of the centroid is calculating the arithmetic mean of all pixel positions in a given region. If only the head is in the frame, this approach would reliably find its centre.

However, if the neck and/or torso are also visible, that centroid would shift downwards, which could lead to incorrect results, especially if the head is tilted away from/towards the sensor.

Distance transforms are another possibility, which map a binary image into a distance matrix indicating the distance of each pixel of a region to its closest boundary. By only choosing the pixels with the highest distance, one can extract a morphological skeleton, which is a minimal representation of a shape [NGC92]. If only the head is in frame, the resulting skeleton would be a dot or line across the centre, with the centre of the line being the centre of the head (see Figure 3.5). The skeleton will contain multiple lines with branching points if the torso is also visible.

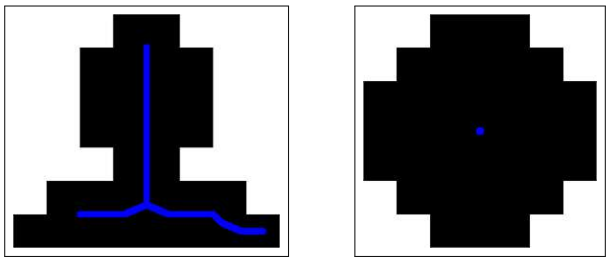


Figure 3.5: Skeletons of an abstracted torso and a circle. Skeletons have been thickened for better visibility and have an actual thickness of 1 px.

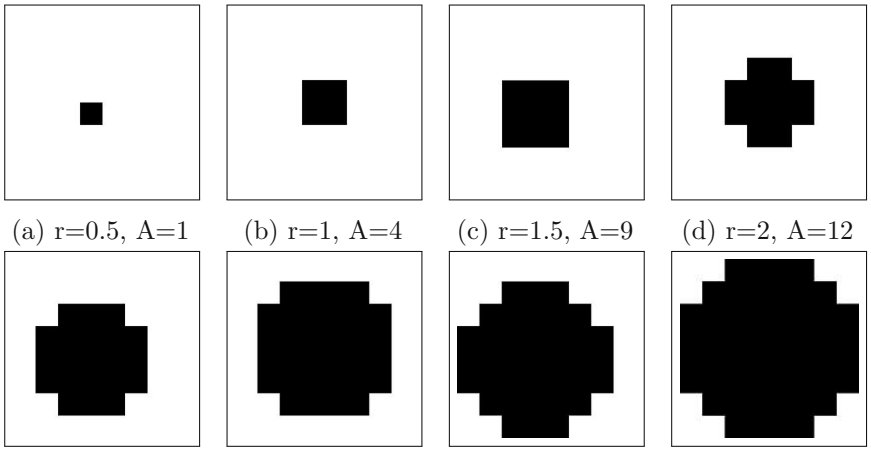
Both methods fail to accurately detect the head if the torso is in frame or the head is tilted. Recalling the core assumption described at the beginning of this chapter (Assumption 1), a head's elliptical shape will not be influenced by its pose or the visibility of the torso; thus, ellipse detection can be a viable method to find a human head reliably.

3.1.1 Circles and Spheres

If the ears are visible, they can alter the apparent shape in the sensors' output in such a way that it appears more circular by adding width to the ellipsoid, making circle detection another viable method. Furthermore, due to the concave shape of parts of the face, like the eye sockets and the space between the bottom lip and chin (a fact that is taken advantage of by Haar-like features), we gain details that help determine the centre of the head (as seen in Figures 3.13 and 4.2).

Remark. Haar-Like features were proposed by Paul Viola and Michael Jones [VJ01] as features for their Viola-Jones-Algorithm, taking advantage of variations in pixel intensity, like the shadows naturally cast by the human face to classify objects and faces.

However, since the native output of the sensor array used in this thesis is 8×8, finer details like the Haar-Like features cannot be used reliably and only eight "circles" are possible, with radii between 0.5 and 4 px, as seen in Figure 3.6 – provided that the head is precisely at the centre of the image.

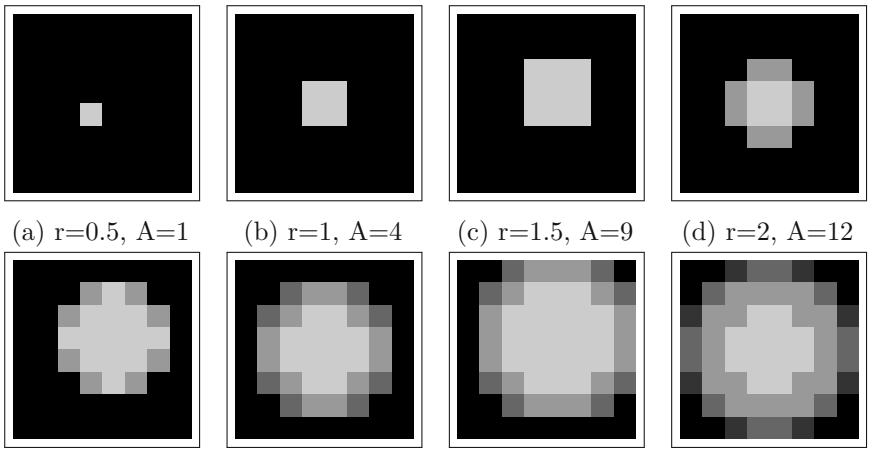


(a) r=0.5, A=1 (b) r=1, A=4 (c) r=1.5, A=9 (d) r=2, A=12

(e) r=2.5, A=21 (f) r=3, A=32 (g) r=3.5, A=37 (h) r=4, A=52

Figure 3.6: The possible discrete circles in an 8×8 image with their centre at the image centre.

Spheres projected into a two-dimensional image appear as grey-scale circles, as shown in Figure 3.7.



(a) r=0.5, A=1 (b) r=1, A=4 (c) r=1.5, A=9 (d) r=2, A=12

(e) r=2.5, A=21 (f) r=3, A=32 (g) r=3.5, A=37 (h) r=4, A=52

Figure 3.7: Examples for 2D-projected spheres in an 8×8 image. The grey-scale values depend on the surface angles and can differ from the ones shown here.

If the head is off-centre, the number of possible circles is reduced since parts of the head would be out of frame.

To be precise, the possible discrete pixel circles have a radius range of

$$[0.5, \lfloor d \rfloor], d \geq 1 \tag{3.1}$$

where d is the minimal distance between the centre of the head and the image boundaries along the X- or Y-axis. It is defined as

$$d = \min(\min(x, w - x), \min(y, h - y)) \tag{3.2}$$

where x, y are the X- and Y-coordinates of the head in the image space, and w, h is the image width and height, respectively. Of note is that circles with a radius below 2 are essentially squares, meaning that circle detection is impossible at this size range.

To facilitate circle detection, bi-cubic interpolation (with its side effect, the blob formation, see Subsection 3.1.3) is employed. Due to the bigger image size and blob formation, small "circles" with radii below 2 appear circular, while bigger circles remain circular.

Using circle detection to more accurately detect a human head makes the algorithm even more dependent on the positioning. Suppose the head is at the boundaries of the sensor's field of view, either due to being too close to the sensor or too far away from the image centre. In that case, two complications can occur:

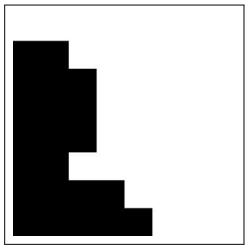
Clipping: Part of the head is outside the frame, thus altering the shape of the head visible to the sensor, see Figure 3.8a for a schematic example. This can be mitigated by taking advantage of the head's self-symmetry.

Edge loss: At least one side of the head is occluded by the image boundaries. Although the shape visually looks like a circle, the image boundaries are not recognised as edges. This leads to an ambiguity of shape, as it is not clear whether the visible shape represents the whole object or only a part of it. See Figures 3.8b and 3.8c for a schematic example.

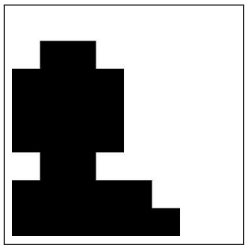
In cases where circle detection is impossible – due to clipping, edge loss, a distance that is too small or too big, or a pose that alters the apparent shape of the head from the point of view of the sensor – shape properties are used to estimate the position of the head.

Another assumption for this algorithm is that a hand has to be in front of the head to execute a dynamic gesture. In case two circles do get detected (hand in front and head in the back), one can ignore the circle with the lower depth value (i.e. the closer object).

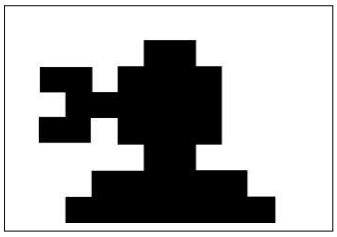
Of the methods presented, circle detection is the most computationally expensive (with a best-case time complexity of $O(n^2)$ for circle detection [KRG94], vs. a best-case time complexity of $O(\log n)$ for distance transform [BK23]), where n is the number of pixels in the image. Nevertheless, due to its stability regarding positioning and head pose, the algorithm discussed in this thesis uses circle detection as the primary and the computationally inexpensive shape centroid as the secondary/fallback method for head detection.



(a) Clipping due to being too far from the image centre or too close to the sensor.



(b) Edge loss due to alignment with the image boundaries or being too close to the sensor.



(c) Shape ambiguity due to edge loss. Is 3.8b truly the whole shape or just a clipped version of this?

Figure 3.8: Examples of sub-optimal head positioning.

3.1.2 Calculating the Angular Size

We can calculate the minimum distance for any given circular object to be entirely in frame: By visualising our FoV as a triangle, we can draw a line across the centre, resulting in two right triangles, as seen in Figure 3.9.

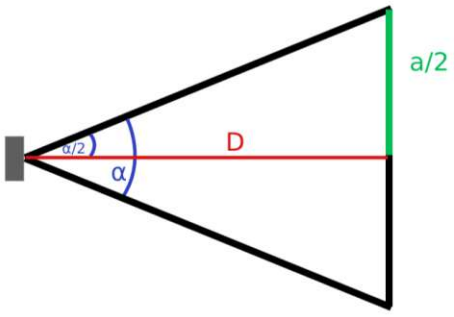


Figure 3.9: Side-way visualisation of the field of view. α denotes the angle of the field of view, D is the distance between an object and a is the image size of the object.

By applying the trigonometric equation

$$\tan(\alpha) = \frac{a}{b} \tag{3.3}$$

to one of the right triangles that make up half of the field of view, we get:

$$\tan(\frac{\alpha}{2}) = \frac{r}{D} \tag{3.4}$$

where $\frac{\alpha}{2}$ is half the angle of the field of view, D is the distance between the sensor and a given object and r is the radius of the object in the image. Now, by approximating a human head as a circle with a diameter of 18 cm, we can calculate the minimal distance needed for it not to take up the whole field of view:

$$\tan(\frac{27°}{2}) = \frac{9}{D} \Rightarrow D \cdot \tan(13.5°) = 9 \Rightarrow D = \frac{9}{\tan(13.5°)} \approx 37.5\ cm \tag{3.5}$$

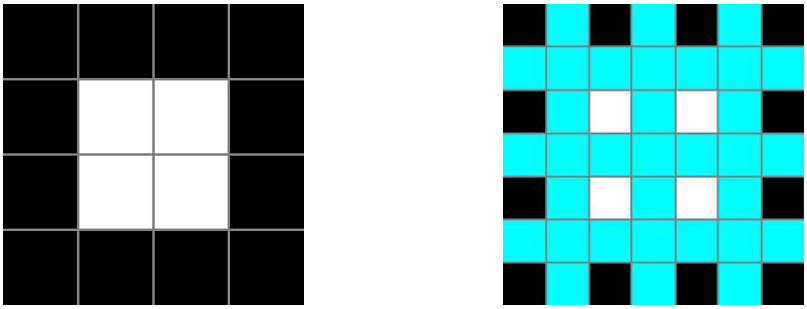
By changing α to be a quarter of its full size ($= 6.75°$), we can calculate the maximum distance needed for the approximated shape to be registered by at least four sensors (Figure 3.6.b):

$$\frac{9}{\tan(3.375°)} \approx 152.6\ cm \tag{3.6}$$

However, this is under the assumption that a completely flat, parallel circle is in front of the sensor. The concavity of the eye sockets, the area between the lower lip and chin, and the angle of the nasal ridge, will lead to non-orthogonal reflection of the photons. This means, that a part of the photons will be reflected in such a way that they don't return to the receivers of the sensor array, which alters the apparent shape and size.

3.1.3 Bi-cubic Interpolation

When up-sampling an image (i.e. resizing it to a bigger size), the values of newly created pixels in between are unknown (see Figure 3.10).



(a) 4×4 image of a white square on a black background.

(b) The same image after resizing it by a factor of 2, leading to an 8×8 image.

Figure 3.10: Schematic example of image resizing. The cyan pixels are newly created pixels with unknown values.

To interpolate (i.e. infer the value from the given data) the new pixel values, many methods exist, like the nearest neighbour interpolation, bi-linear interpolation, or bi-cubic interpolation.

Nearest-neighbour interpolation is a simple method, which assigns each new pixel the value of the closest original pixel. However, as can be seen in Figure 3.10b, a new pixel can have none or multiple closest original pixels. To consistently assign pixel values, the nearest-neighbour algorithm defines a fixed ”direction” to look for the closest original pixel.

The resulting image contains no new pixel values and can appear ”blocky” due to the abrupt changes in pixel values.

Given that the image size of our sensor array is 8×8, shapes like the head or the hand already appear blocky. Because we want to better differentiate between an organic shape like a body part and for example, rectangular furniture, the nearest neighbour interpolation is a poor choice for us, as shown in Figure 3.12.

Bi-linear interpolation creates a linear approximation of the new pixel values, creating a smoother image than Nearest-Neighbour interpolation, but the resulting image loses sharpness and edges appear less defined due to the linear approximation of pixel values (ramps).

As we need well-defined edges to reliably detect shapes efficiently, a function that creates ramps can be counterproductive. Thus, while bi-linear interpolation helps with rounding shapes (as seen in Figure 3.12), it is not favourable for shape detection.

Finally, bi-cubic interpolation interpolates the missing pixel values using the following model:

$$f(x,y) = \sum_{j=0}^{3}\sum_{i=0}^{3} a_{ij}x^i y^j \tag{3.7}$$

where a_{ij} is one of 16 coefficients and x, y is the position of the calculated pixel.

Simplified, bi-cubic interpolation approximates the values using splines, which not only depend on the values of the starting and ending pixel but also their tangents, thus also depending on the previous and the next pixels. A visualisation for a row of pixels is shown in Figure 3.11.

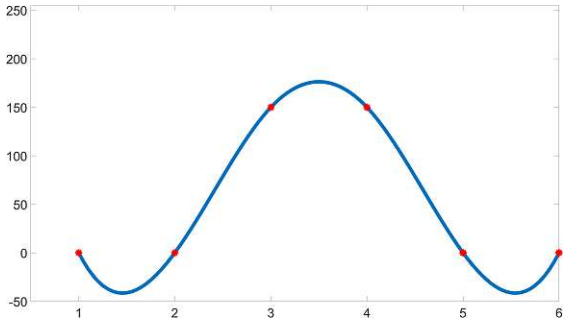
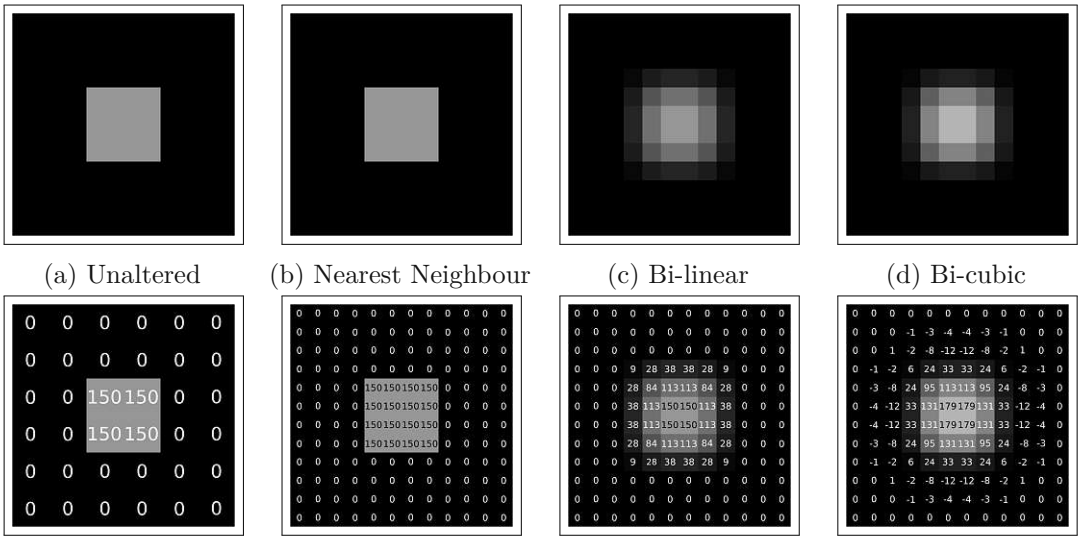


Figure 3.11: Visualisation of bi-cubic interpolation on a row of 6 pixels (red markers). The Y-axis denotes pixel value, and the X-axis is the pixel's position along the row.

While this approach leads to better-perceived image sharpness, it also tends to under- and overshoot, as seen in Figure 3.11: the values between pixels 1 and 2 are in the negative, while the values between 3 and 4 exceed the maximum value of the original pixel row (150).

To show the effect of the various interpolations, Figure 3.12 shows an image with a square having the same pixel values. Of note there is how the bi-cubic interpolation creates values outside the original value range and how both bi-linear and bi-cubic interpolation seem to make a "sphere" out of the square. Figure 3.13 compares the three interpolation methods applied on the sensor amplitude output.

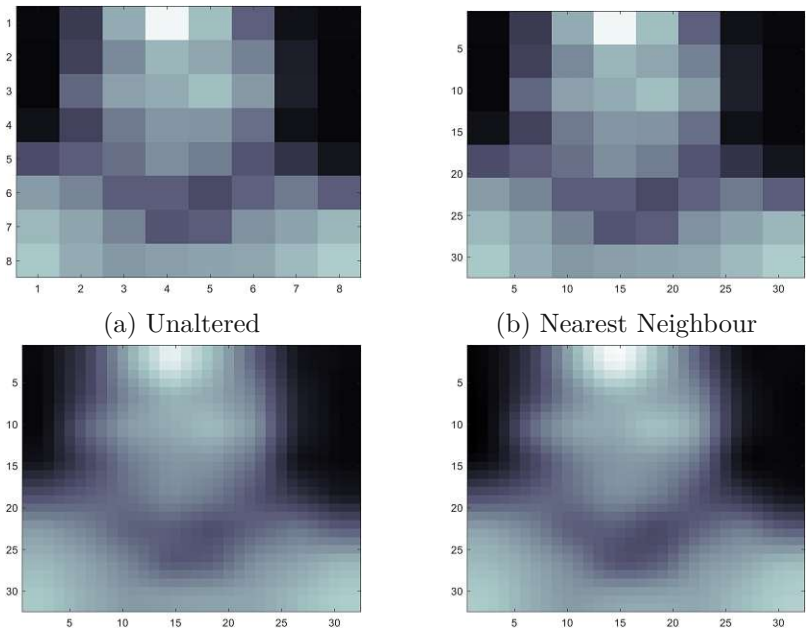
By approximating the model using a convolution kernel, Keys [Key81] proposed a bi-cubic interpolator which sacrifices accuracy for efficiency, and can be used for Image Pyramids [Kro90] or convolution layers in a deep learning network.



(a) Unaltered (b) Nearest Neighbour (c) Bi-linear (d) Bi-cubic

(e) Unaltered (f) Nearest Neighbour (g) Bi-linear (h) Bi-cubic

Figure 3.12: Comparison of various interpolation methods for a scale of 2 on a 6×6 image of a square.



(a) Unaltered (b) Nearest Neighbour

(c) Bi-linear (d) Bi-cubic

Figure 3.13: Comparison of various interpolation methods for the sensor output.

3.1.4 Edge detection

To analyse shapes present in an image, they first need to be extracted. Since a shape is defined by its edges (e.g. rectangles have four connected edges at 90° angles, squares are a special type of rectangle with all four edges having equal length, triangles have three connected edges with their three angles summing up to 180°, ...), one approach is extracting the edges present in an image.

According to Barrow and Tenenbaum [BT81], an edge in an image can be defined as a discontinuity in pixel values, be it colour or brightness/intensity, which implies

1. A discontinuity in depth
2. A discontinuity in surface orientation
3. A variation in reflective properties
4. A variation in illumination

One method of extracting edges is by filtering the image using convolution kernels (filtering). Edge detection kernels include the Prewitt [P+70] and Sobel [KVB88] filters, which are first-order kernels, and function by smoothing and normalising the image before approximating the first derivative of the image. The resulting horizontal and vertical edge images can be combined into a single edge image containing all horizontal and vertical edges [P+70] [KVB88].

However, in cases where edges are neither horizontal nor vertical, both edge detection methods will fail. Thus, second-order kernels like the Laplacian edge detector [MH80] can be used, which approximate the second-order derivative and discern between outward and inward edges. Due to using the second derivative of the image however, the Laplacian filter is more sensitive to noise, which can be mitigated by applying a noise-reducing filter like the Gaussian beforehand [MH80].

Finally, the Canny method [Can86] combines first-order kernels like Sobel and Prewitt with pre- and post-processing steps to achieve an edge detection that is less sensitive to noise. After calculating the gradient of the image, non-maximum suppression is used to determine the pixel with the highest intensity along the gradient direction of the edge and eliminate any other edge pixel that does not satisfy this condition – thus thinning the edges.

Next, a double threshold is employed as a first step to eliminate false edges; any edge pixel with an intensity value above the high threshold is considered a strong pixel and thus contributes to the final edge, while edge pixels with an intensity value above the weak threshold, but lower than the strong threshold, are considered weak pixels, which need further post-processing to determine their contribution to the final edge image. Edge pixels below the given weak threshold are considered non-relevant and are discarded.

The resulting edge image only contains three pixel values: 0 for non-edge pixels, 255 for strong edge pixels, and 128 for weak edge pixels.

Finally, weak edge pixels undergo Hysteresis, where they are either eliminated or transformed into strong edge pixels (i.e. raising their intensity value to 255). This is done by tracking every weak edge from one end to another; if the weak edge connects to a strong edge on any end, it is transformed into a strong edge. Otherwise, it is discarded by turning the intensity value of all its pixels to 0.

Due to the multiple passes, the Canny method is computationally more taxing. Nevertheless, the higher quality of the resulting edge image thanks to the noise stability and false positive elimination increases the odds of correctly identifying shapes, which is why it is chosen as the pre-processing for the next step, the Circular Hough Transform in Subsection 3.1.5.

3.1.5 Hough Transform

One of the known methods for finding geometric shapes like lines, circles, or other classes of (parametric) shapes is the Hough Transform [Hou62]. By extracting an edge image out of an input image and then transforming it to a parameter space (also known as Hough space) with polar coordinates, imperfect shapes can be still recognised, if there is an intersection in the Hough space (examples in Figure 3.14).

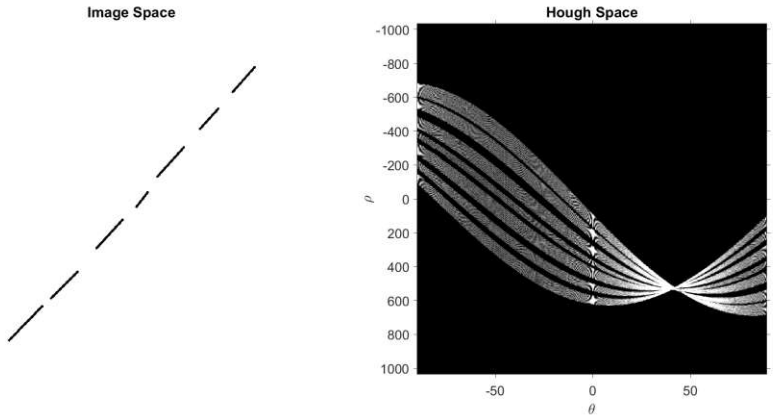
Instead of representing a line using the slope a and intercept b in the Cartesian system like $y = ax + b$, the polar representation uses ρ for the shortest distance between the origin and the line and θ for the angle between the X-axis and the distance line. Given ρ and θ, the following equation is true for any point along the line:

$$\rho = x \cos\theta + y \sin\theta \tag{3.8}$$

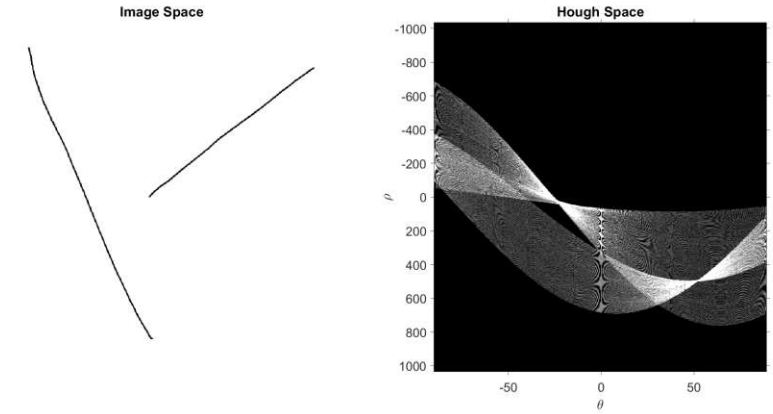
Thus, a line can be mapped to a single point in the parameter space of ρ and θ. Now, if there is a point in image space – meaning that x and y are fixed – its representation in the parameter space is a sinusoid.

Using these two properties of the parameter space, straight lines can be detected, even if they are imperfect or disconnected: Every point creates a sinusoid, which means that a drawn line in the image space will result in multiple sinusoids in the parameter space. These sinusoids will intersect at a single point: the point with the ρ and θ value of the line they are aligned with (see Figure 3.14 for examples).

The point of intersection is found by creating an accumulator space containing the sinusoids of the parameter space. Via voting, every point along the sinusoids increases the corresponding point in the accumulator space. If sinusoids intersect, their intersection point will have n votes, where n is the number of intersecting sinusoids. Finally, the local maxima (i.e. the points with the highest number of votes/intersections) are chosen as the parameters for the line candidates in the image space.



(a) Dashed, shaky line. The intersection at $\theta \approx 42.5$ and $\rho \approx 530$ shows, that the segments align along the line $530 = x \cos 42.5 + y \sin 42.5$.



(b) Two shaky lines. The two intersections mark the polar parameters of the two lines that align with them.

Figure 3.14: Examples of Hough transformation with lines.

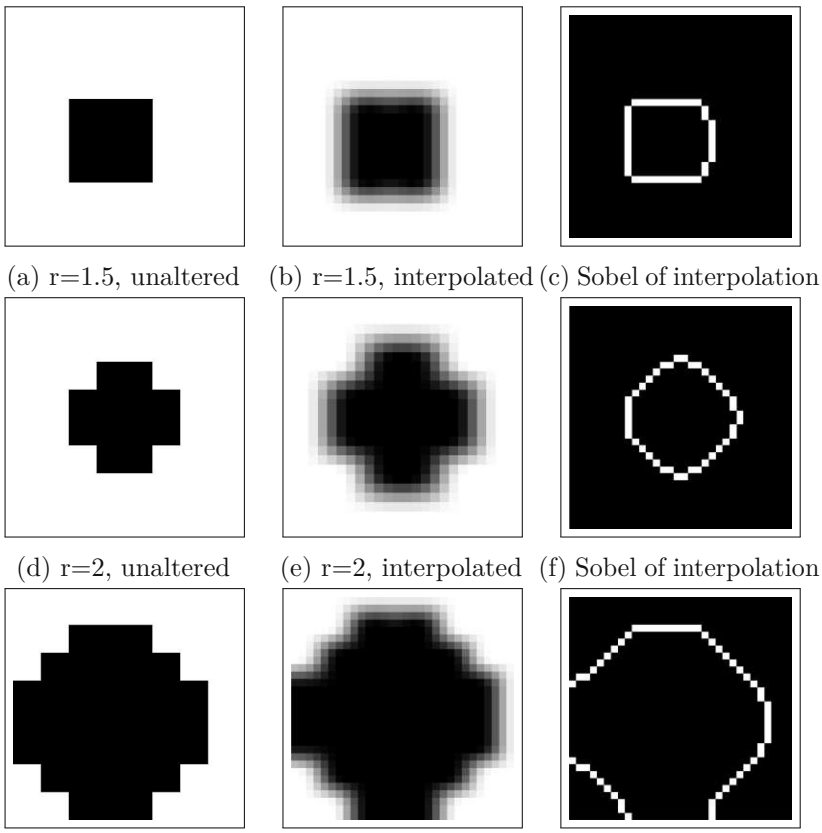
Of course, due to this property, lines can also be detected where there are none, e.g. when objects in an image happen to align coincidentally. Another drawback is that the end of a line cannot be determined using Hough Transform alone, since it only works with lines of infinite length.

In his work, Dana H Ballard [Bal81] present their Generalized Hough Transform, which is able to detect any arbitrary shape, but is computationally more taxing than the standard Hough Transform for lines.

Circular Hough Transform

The circular Hough Transform is a version of Hough Transform which uses a three-dimensional parameter space: the X-position of the centre, the Y-position of the centre and the radius.

Particularly at lower resolutions (and image sizes like 8×8 like in the case of this thesis), circle detection might provide a better detection rate than face detection techniques that rely on finer facial features. By taking advantage of the overshoot (or haloing) and blurring that occurs when using Bi-cubic Interpolation, circles with a radius ≥ 2 px appear more circular (see Figure 3.15).



(a) r=1.5, unaltered (b) r=1.5, interpolated (c) Sobel of interpolation

(d) r=2, unaltered (e) r=2, interpolated (f) Sobel of interpolation

(g) r=3.5, unaltered (h) r=3.5, interpolated (i) Sobel of interpolation

Figure 3.15: Four-time magnification of discrete circles using Bi-cubic Interpolation and their resulting Sobel edge images. Note how the "circle" with $r = 1.5$ merely turns into a square with rounded corners. Furthermore, observe how the Sobel edge detection behaves when the circle is at the image boundary. Even though the circle is symmetrical, the edge image appears to have two protrusions to the image boundaries.

One such circle detection algorithm is the Circular Hough Transform, which, instead of taking the polar coordinates used to describe a line like in Hough Transform, uses a three-dimensional space with the parameters of a circle: The two coordinates of the centre (a, b) and the radius r. If a circle of fixed radius is drawn on some/all points of an edge and all circles along the points intersect at one single point (found again using the accumulator matrix), then that point is the centre of the detected circle and its radius is equal to that of the circles in the parameter space, as seen in Figure 3.16.

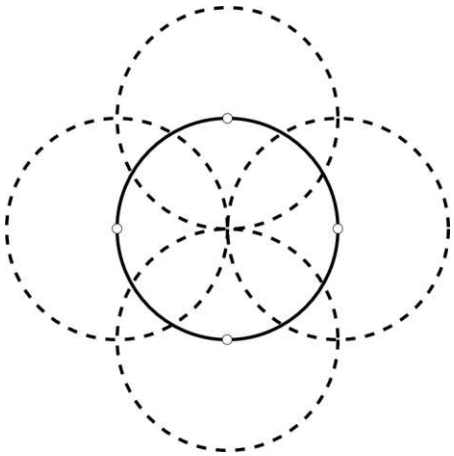


Figure 3.16: Example of Circular Hough Transform. The dashed circles are in the parameter space, and the solid is in the image space. As can be seen here, if their radius is equal to the image circle, they will intersect at the centre of it.

However, finding the intersections of the circles in the three-dimensional parameter space is more computationally taxing than in a two-dimensional space. Taking our image size of 8×8 alone, a search for just the position of a circle (meaning the x and y coordinates) means looking at 64 possible points. If we also need to know the radius of a circle, we not only have the 64 possible centre points but also the 16 possible radii (see Figure 3.6). Thus, we would not have 64, but $64 \cdot 16 = \mathbf{1024}$ possibilities.

One method to simplify this task is the Phase Code Hough Circular Transformation by Tim J Atherton and Darren J Kerbyson [AK99], which searches for circles in a range of radii, thus limiting the radius dimension and only needing to search in the two-dimensional x, y space of the circle centre. Unfortunately, even though the Phase Code Hough Circular Transformation is generally scale-invariant, a certain resolution is needed for a circle to be recognised as such (see Figure 3.6).

Thus, the circle detection works best when the head is at a depth where it reflects to at least 12 receivers (e.g. Figure 3.6d), but not more than 34, as it would then be at at least one image boundary, making shape detection impossible. Furthermore, circle detection in the application presented in this thesis assumes that the head is always in frame, upright, and facing the sensor. If the head is facing upwards, it might lose its apparent circularity.

3.1.6 Blob Detection and Connected-Component Labeling

Blob detection and Connected-Component Labeling (CCL) are two other possibilities – essentially finding groups of pixels based on similarity and shape (blob detection) or connectivity and an attribute (CCL).

Blob Detection

Methods like the Laplacian of Gaussian (LoG) [Lin93] or Difference of Gaussian (DoG) [Low04] are used to detect blobs of multiple sizes by convolving the image with their respective kernel using different scales. The resulting responses create a scale space, and by searching for extrema in the scale space, the position and characteristic scale of the blobs are determined. Figure 3.17 shows an example output for both methods.

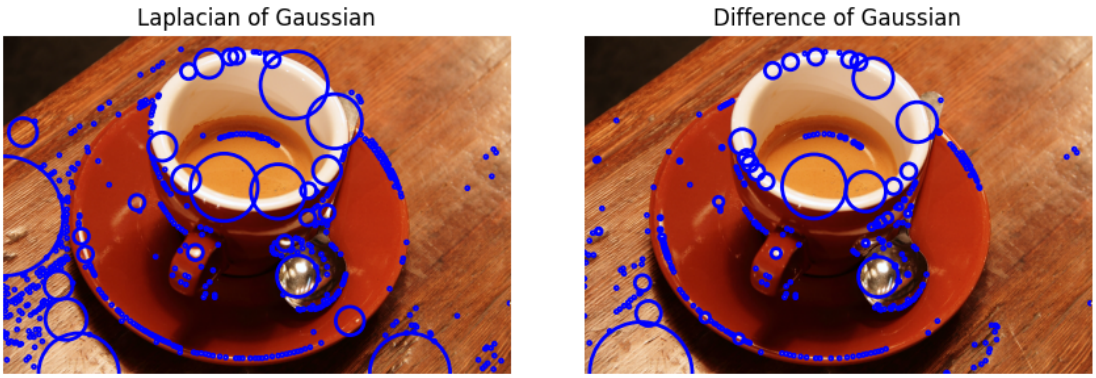
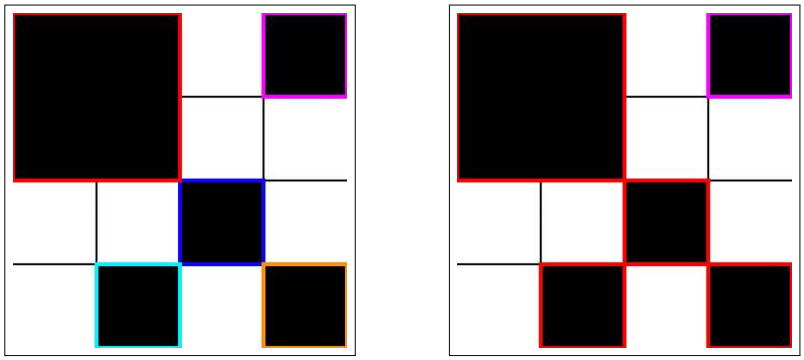


Figure 3.17: Example for Laplacian of Gaussian and Difference of Gaussian blob detection on an image of a coffee cup.

Connected Component Labeling

The attribute is what defines a region CCL – pixels that are similar in regards to the chosen attributes belonging to a region, provided they are in the chosen connectivity. In our application, we have two attributes from the get-go: depth and amplitude. Of course, further attributes can be created by combining or computing data, making more options available.

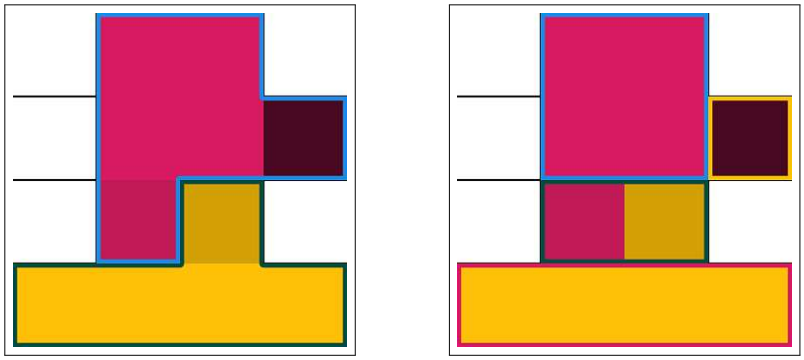
Connectivity defines the ”search window” and has multiple options, the common ones being the 4- and 8-connectivity. 4-connectivity compares the four neighbouring pixels $(x \pm 1, y)$ and $(x, y \pm 1)$, while 8-connectivity also compares the neighbouring corner pixels, i.e. looking at every pixel in $(x \pm 1, y \pm 1)$. Figure 3.18 shows an example of CCL using 4- and 8-connectivity given an arbitrary attribute.



(a) 4-connectivity, 5 regions (b) 8-connectivity, 2 regions

Figure 3.18: Examples for 4- and 8-connectivity and their impact on the number of detected regions.

Going back to depth and amplitude, if we, for example, encode amplitude with colour hue and depth with brightness, we can visualise how connected component labelling could work depending on the attribute, as seen in Figure 3.19.



(a) amplitude, 2 regions (b) depth, 4 regions

Figure 3.19: Examples for connected component labelling on amplitude and depth, using 8-connectivity. Amplitude is visualised by colour hue, while depth is denoted by the pixel intensity (brightness).

The issue with blob detection and CCL is their indiscriminate functionality; blob detection will detect any shape, as long as it is compact (i.e. if it roughly fits inside a circle), while CCL will return every region that is connected and similar regarding the given attribute. This means, that non-circular objects can be detected, increasing the false positive rate and thus reducing the accuracy of head tracking. Hence, we have decided to use circle detection for head detection.

3.1.7 Shape Properties

In computer vision, shape properties are multiple metrics and attributes that describe a shape (i.e. a coherent region of pixels), which can be used for decision-making and feature extraction. Shape properties are calculated mainly from binary images and include attributes like area, perimeter, orientation, centroid, and extrema, which are presented here.

Circularity

One shape property is the circularity/roundness of a region, which is calculated as

$$\frac{4\pi A}{P^2} \tag{3.9}$$

where A is the area and P is the perimeter of the region (i.e. the sum of the distance between boundary pixels in an 8-neighbourhood). Assuming a perfect circle, we know that the area is πr^2 and the perimeter is $2\pi r$. Inserting this into the circularity equation we get:

$$\frac{4\pi(\pi r^2)}{(2\pi r)^2} = \frac{4\pi^2 r^2}{4\pi^2 r^2} = 1 \tag{3.10}$$

However, as with the Circular Hough Transformation in Section 3.1.5, circularity is only reliable above a certain shape size. In the case of the sensors used in this thesis, the image size is too small, as seen in 3.11, where a discrete circle with a diameter of 7 has a higher circularity than a theoretical perfect circle; with an area of 37 and a perimeter of ≈ 18.4, we get a circularity of ≈ 1.4.

$$\frac{4 \cdot \pi \cdot 37}{18.4^2} \approx 1.4 \tag{3.11}$$

Note, that even with an up-sampling of 4 times, a circle with a diameter of 28 would still have a circularity above 1. Nevertheless, rectangular shapes will have a circularity below 1 and thus, circularity can eliminate some non-organic shapes.

The circularity of a rectangle is

$$\frac{4\pi ab}{(2a+2b)^2} = \frac{4\pi ab}{4a^2+8ab+4b^2} = \frac{4\pi ab}{4ab(\frac{a}{b}+2+\frac{b}{a})} = \frac{\pi}{\frac{a}{b}+2+\frac{b}{a}} \tag{3.12}$$

Since π is a constant and the denominator will always be bigger in $\mathbb{N}$ starting from $a = b = 1$, the circularity will always be < 0, getting smaller the bigger the difference between a and b gets. The rectangle with the highest circularity is a square, which will always have a circularity of ≈ 0.785 since its area is a^2 and its perimeter is $4a$. Thus, the circularity equation would be:

$$\frac{4\pi(a^2)}{(4a)^2} = \frac{\pi}{4} \approx 0.785 \tag{3.13}$$

Centroid

The centroid is the centre of mass of any given shape. It is calculated as the arithmetic mean of all pixel coordinates that belong to the shape:

$$(c_x, c_y) = \frac{\Sigma(p_x, p_y)}{n} \tag{3.14}$$

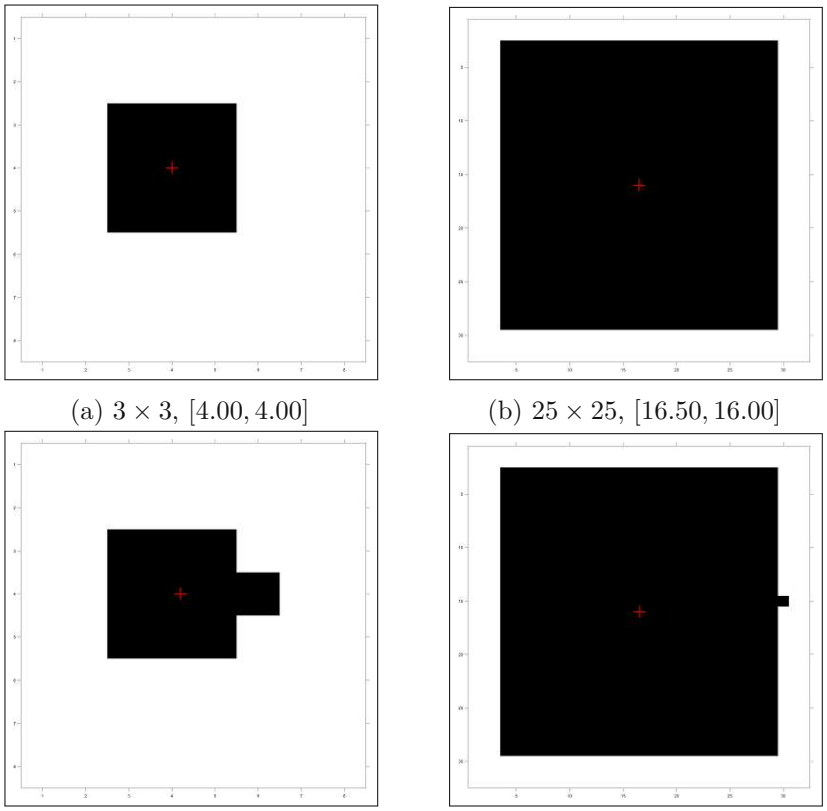
Where p_x, p_y are the (x, y)-coordinates of the pixel and n is the total amount of pixels in this region. Unlike pixels however, which have (x, y)-positions solely in $\mathbb{N}$ (e.g. $[1, 2, 5]$), the coordinates of a centroid are in $\mathbb{R}$ (e.g. $[1.4, 2.6, 5.8]$), meaning that they can be "inside" pixels.

Keep in mind that this method of calculation can be unstable, especially in small image sizes. Because every detected pixel of a region has an equal influence on the outcome, the relative amount of outliers and artefacts will be higher in small image sizes compared to larger ones; for example, 4 "faulty" pixels in an 8×8 image amount to 6.25%, while 4 pixels in a 32×32 image are 0.39%. Thus, outliers and inexact shape boundaries will shift the centroid more, the smaller the shape is (see Figure 3.20).

A way to mitigate this is calculating the *weighted* centroid, which also takes pixel values into account:

$$(c_x, c_y) = \frac{\Sigma p_v(p_x, p_y)}{n} \tag{3.15}$$

Where p_v is the pixel value in $[0, 1]$, (p_x, p_y) are the x and y coordinates of the pixel, and n is the total amount of pixels in this region. In the application presented in this thesis, this would be the depth or amplitude value. The higher the pixel value, the "heavier" it is, therefore shifting the centroid more towards it. Due to the spherical/ellipsoid shape of the human head, the centre should have a higher amplitude and/or lower depth, which is why the weighted centroid is employed in this thesis.



(a) 3×3, $[4.00, 4.00]$ (b) 25×25, $[16.50, 16.00]$

(c) $3 + 1 \times 3$, $[4.20, 4.00]$ (d) $25 + 1 \times 25$, $[16.52, 16.00]$

Figure 3.20: Comparison of impact of outlier pixels on two different sizes. One pixel shifts the centroid of the 3×3 square by 5% to the right, and that of the 25×25 square by 3%.

Remark. As mentioned at the beginning of this Chapter (Section 3.1), the centroid is not the only possible method to calculate the "center" of a given shape. Another way to calculate it can be done in two steps: first, calculate the distance transform by Azriel Rosenfeld and John L Pfaltz [RP68] or the eccentricity transform by Kropatsch et al. [KIHF06], which is more stable in regards to noise. Then, choose the pixel with the highest value for Distance Transform, or the lowest value for Eccentricity Transform. In the case where multiple points have the highest distance, a simple average between the centre candidates would suffice. Both methods could deliver more accurate and stable points of reference.

Shape Extrema

Definition 3.1.1. Shape extrema are the highest and/or lowest (x, y)-positions belonging to a region and are calculated using an $n \times 2$-sized matrix P_{xy} containing the pixel positions of all pixels belonging to a region.

By keeping one of the two coordinates fixed, eight extrema (shown in Figure 3.21) can be calculated: For example, if we choose right-bottom, we first look for the highest x-value of the region (the global maximum). Having found that extreme e_x, we now search for the lowest y-value belonging to the region *at that specific x-position* (y, e_x) (the local minimum). Hence, the following extrema can be calculated:

left-bottom: global minimum of x, local minimum of y.
$(x, y) = \min_{x=e_x}(P_{xy}),\ e_x = \min(P_x)$

left-top: global minimum of x, local maximum of y
$(x, y) = \max_{x=e_x}(P_{xy}),\ e_x = \min(P_x)$

right-bottom: global maximum of x, local minimum of y
$(x, y) = \min_{x=e_x}(P_{xy}),\ e_x = \max(P_x)$

right-top: global maximum of x, local maximum of y
$(x, y) = \max_{x=e_x}(P_{xy}),\ e_x = \max(P_x)$

bottom-left: global minimum of y, local minimum of x
$(x, y) = \min_{y=e_y}(P_{xy}),\ e_y = \min(P_y)$

bottom-right: global minimum of y, local maximum of x
$(x, y) = \max_{y=e_y}(P_{xy}),\ e_y = \min(P_y)$

top-left: global maximum of y, local minimum of x
$(x, y) = \min_{y=e_y}(P_{xy}),\ e_y = \max(P_y)$

top-right: global maximum of y, local maximum of x
$(x, y) = \max_{y=e_y}(P_{xy}),\ e_y = \max(P_y)$

If the whole region consists of just a single pixel, it would still have all eight extrema - two at each corner. This process also works with concave shapes, as the local extrema are searched in all pixel positions of P_{xy}.

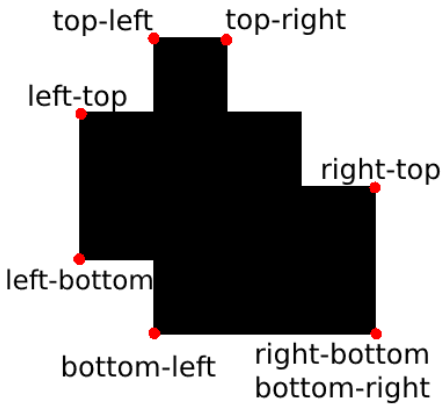


Figure 3.21: Example for extrema. It is possible for extrema to overlap, like with right-bottom and bottom-right.

Furthermore, one can combine two or more extrema to generate new ones, e.g. right-middle, which is the centre of right-top and right-bottom. Note, however, that those new extremes can sometimes lie outside of the shape, especially in the case of concavities. For example, calculating the average of top-left and left-top in the shape shown in Figure 3.21 would lead to such outside points.

3.2 Theory of Gesture Recognition in Low-Resolution Infrared Amplitude Images

Gesture recognition is a combination of movement tracking and pattern matching. The challenge lies in detecting the hand to track, and adjusting the sensibility to movement, since humans rarely stay perfectly still or recreate shapes perfectly.

Additionally, hands can be used as part of communication (e.g. for emphasis), which should not be recognised as deliberate gestures. Another point of consideration is the limits of the human anatomy in regards to hand and arm movement: While a hand may be able to ”draw” a circle with its fingertips, its range of movement is limited [HGMBJ90], especially without movement of the arm, which is also limited in its movement [GST+20] [STM11].

In this thesis, we define a gesture as a deliberate, consistent movement in one direction (directional gesture, Definition 1.3.3). Thus, we have four parallel (up, down, left, right) and two perpendicular (front, back) gestures, corresponding to the 6-connectivity in three dimensions $\mathbf{N}^3$. Due to the rectangular arrangement of the sensor field, diagonal or ”shape” (e.g. circle, square, triangle, L-shape) gestures could be implemented, but are not in the scope of this thesis.

Since ToF-IR-Sensors not only calculate the amplitude but also the depth, we can

already discern between parallel and perpendicular movement (see Figure 3.22). Parallel movements along the general axes can be detected by tracking the amplitude changes in every field of the array, while perpendicular gestures can be detected by tracking the depth changes.

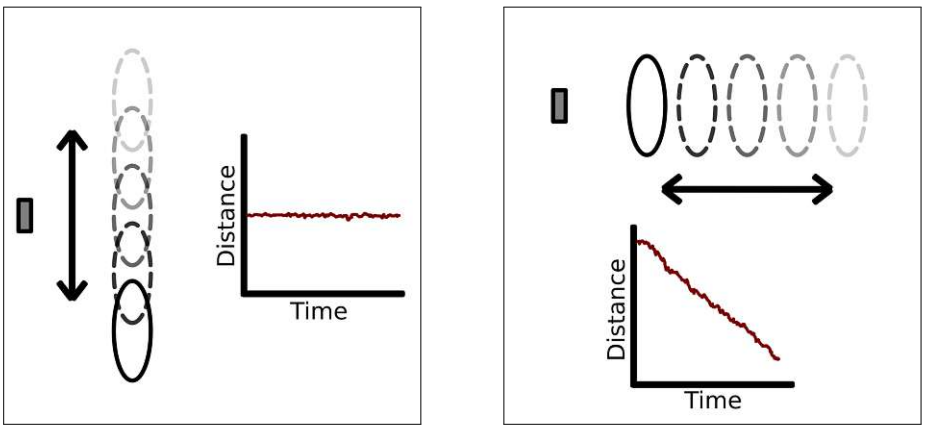


Figure 3.22: Parallel and perpendicular movement and its influence on the depth value. The sensor is the small grey rectangle on the left, while the ellipse is a moving object. With parallel movement, the depth measurements stay in a given range, while perpendicular movement can have any depth between 0 and ∞.

Since this thesis combines head detection with gesture recognition, the algorithm presented works under the assumption that a human head is in the field of view. Hence, just tracking the amplitude or depth changes without discerning between hand or head could lead to missing or spurious gesture recognition:

For example, suppose the hand is still for three frames and the algorithm correctly detects the hand in the first frame, then wrongly the head in the second, and finally the hand correctly again in the third frame. This would cause a difference in position (and thus "movement") to be registered equivalent to the hand's position in frame 1, to the head's position in frame 2, and back to the hand's position in frame 3.

If the centroid of the head is falsely recognised as the centroid of the hand over multiple frames, movement of the head, not the hand is tracked, leading to gestures being missed.

Additionally, if we manage to remove the head from the field of view, we still have to track the palm of the hands and not the arm. Hence, a method is needed to recognise the arm in the scene and track the movement of its hand.

Finally, we have to consider the aperture problem [Hil84]: If the arm is perfectly horizontal or vertical and the hand is out of view, we cannot determine horizontal or vertical movement, since the edge information will not change between frames. Thus, the algorithm presented in this thesis also requires the hand to always be in frame during the execution of a gesture.

3.2.1 Gesture Space

At an image size of 8×8, detecting a hand cannot rely on finer features like fingers, since the image size is not big enough to visualise fingers even at 1 px thickness; under the assumption, that the boundary or gap between fingers has a minimum width of 1 px, we have five fingers and four boundaries between them, meaning that we would need at least 9 px to visualise all fingers of the hand.

The algorithm in this thesis makes assumptions to simplify this task: Since the field of view of the sensors is narrow (27°) and our algorithm is limited to a close distance (up to 73 cm), we can assume that the hand must be in front of the head while gesturing and is the closest object to the sensor. This space between the sensor and head is what makes up the ”gesture space” in this thesis (seen in Figure 3.23 as an exemplary visualisation).

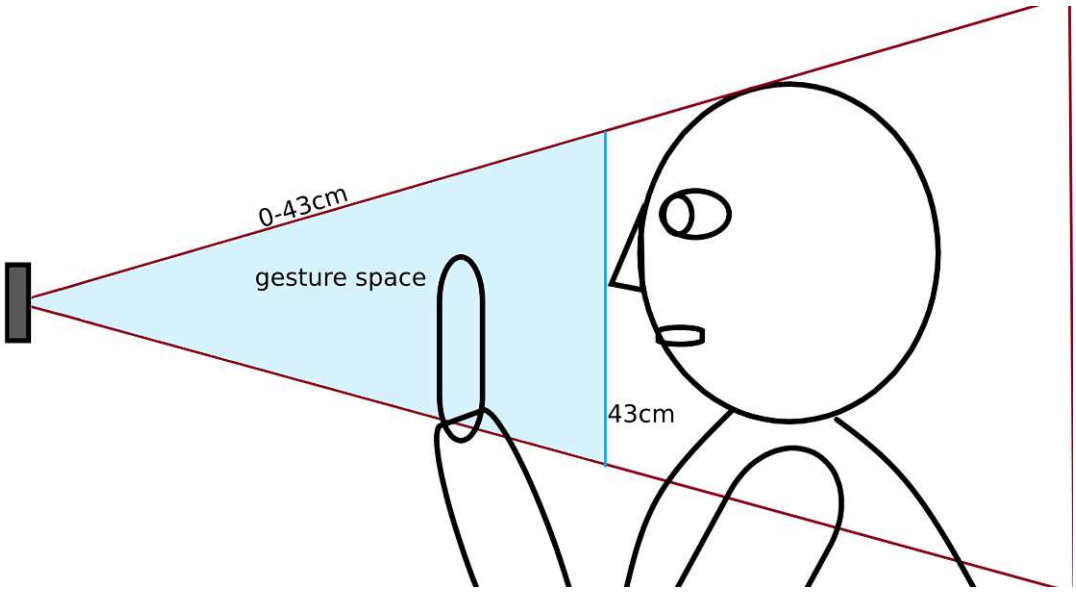


Figure 3.23: Exemplary visualisation of the created gesture space (blue). The sensor is the grey rectangle on the left.

Thus, all that needs to be done is to take the closest point to the sensor and use anything that is a distance ϵ away from it to mask the image. Of course, this also means that, in order to minimise false positives, no other objects should be in the field of view in front of the head.

3.2.2 Shape Properties for Hand Position Estimation

As discussed in Subsection 3.1.7, there are ways to estimate the position of the centroid of a shape. Assuming only the hand is in frame, a centroid would be a sufficient choice for tracking of the movement. However, since the arm will be visible at distances between 40 cm and 73 cm, calculating a simple centroid might yield a position on the forearm, which

can lead to a complication: the further down on the forearm the calculated centroid is, the smaller the perceived movement of the centroid when a gesture is performed without moving the elbow.

This is because, if the elbow is fixed, movement of the forearm can be considered as an arc, whose length can be calculated using

$$r \cdot \frac{\theta}{180°} \cdot \pi \tag{3.16}$$

where r is the length between the centre (in our case the elbow) and θ is the angle between start and finish point (see Figure 3.24). By converting degrees to radians $1° = \frac{\pi}{180}$ rad $\Rightarrow 180° = \pi$ rad, we can simplify the equation to:

$$r \cdot \theta,\ \theta \in [0 \text{ rad}, \pi \text{ rad}] \tag{3.17}$$

Thus, the farther down on the forearm the centroid is, the smaller r is and the less likely it is, for the movement to exceed the set threshold for gesture detection.

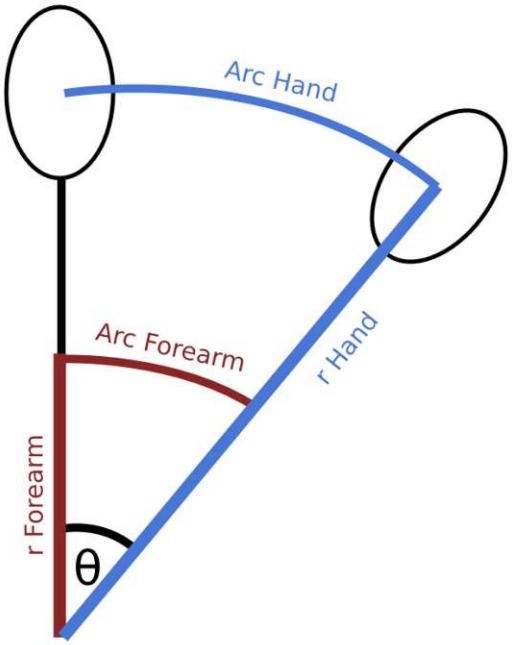


Figure 3.24: Visualisation of a left-to-right movement of an arm and the arcs created by it. θ is the angle between the start and finish point of the movement and r is the length between the elbow and another point of the arm (be it the middle of the forearm or the middle of the hand).

Because we want to make gesture recognition independent of arm pose, shape extrema cannot be considered, as they are highly pose-dependent in our application; since a hand could lie anywhere on a circle, it could be on any of the extremes, and calculating a centroid between top-left, top-right, right-top, and left-top can only work if the arm is facing upwards. Nevertheless, this approach could be combined with another shape property: Orientation.

Shape orientation is the angle between the major axis (the longest possible line) of the shape and the X-axis (in our case, the horizontal plane). Since the major axis of an arm is the line between joints, calculating its orientation and then combining a set of shape extrema depending on the orientation could lead to a hand (or end-of-forearm) detection with a tolerance of ≈ 10 cm, provided the hand and arm are the closest objects in the gesture space, Figure 3.25 visualises major axis and estimated centroid using shape extrema.

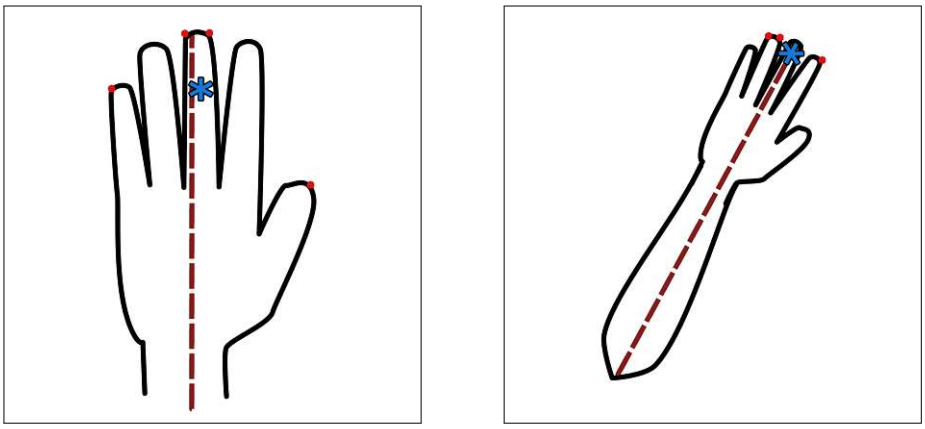


Figure 3.25: Schematic example for centroid calculation using Shape Properties and Orientation. The dashed red line marks the major axis of the shape, the red dots are the shape extrema used for the given orientation, and the blue asterisk is the centroid calculated from the given extrema.

Another option would be to calculate the convex hull of the hand/arm and apply the medial axis transform to build a skeleton [LKC94], as seen in Figure 3.26. Then, the branching point would be a good approximation for the hand centroid.

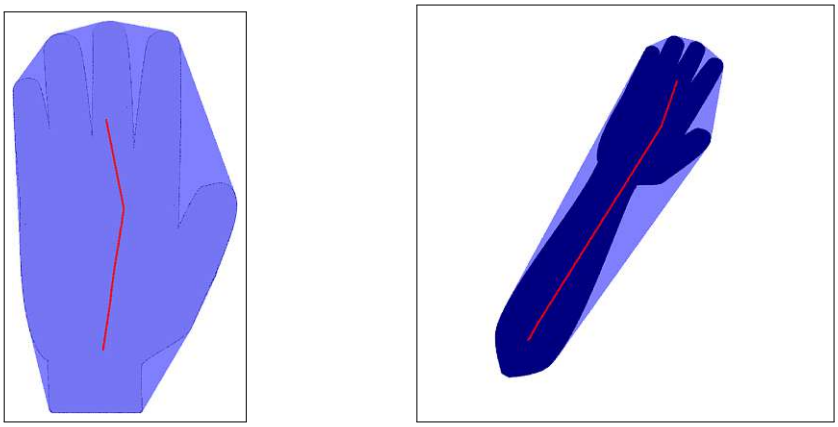


Figure 3.26: Schematic example for convex hull (blue overlay) creation and skeletonization (red line) using medial axis transform.

However, since a flat palm facing the sensors should have a higher amplitude than the arm (see Figure 3.3), the weighted centroid (see Subsection 3.1.7) can be used for pose independent hand detection, which is computationally efficient. By using the amplitude data for pixel weights, the centroid should be skewed towards the end of the arm and ideally, the hand.

CHAPTER 4

Algorithm of Head Detection and Gesture Recognition in Low-Resolution Infrared Amplitude Images

The algorithm described in this thesis combines the knowledge presented to track the position of a human head and detect gestures. This is done by first detecting the head using one of two approaches, depending on distance.

After estimating the position of the head, the space in front of it (i.e. the depth values between 0 and the detected head position) is defined as the space used for hand and gesture recognition (gesture space). If head detection is disabled, the whole depth range $[0 - 73]$ cm is considered as the gesture space. Then, a weighted centroid is used to estimate the position of the hand, and by tracking its movement to discern a direction, directional gestures (Definition 1.3.3) are recognised.

4.1 Algorithm of Head Tracking in Low-Resolution Infrared Amplitude Images

The head tracking algorithm consists of two parts: Detecting the head and tracking its position over time.

For head detection, a stream of frames is fed to the algorithm, containing amplitude and depth data. The first and most important step is to mask the image to improve the conditions for our head detection to work with. An ideal mask would only show the human skin (or better yet, the human head) and nothing else.

At close range ($0 - 37.5$ cm distance from the tip of the nose to the sensor), shape properties suffice, since the head takes up most of the visual field of the sensor (as shown in 3.6). Using the amplitude at these distances, we can even differentiate between the palm of a hand and a head. This is because a hand with its palm facing the sensor will always have a higher average amplitude than a head due to the roundness of a human head and the shadows cast by the eye sockets, nose and chin, which is corroborated in Figure 4.3.

We can apply this knowledge to every depth value by employing the inverse square law of photometry [Gla14]:

$$amplitude = \frac{amplitude_0}{depth^2} \tag{4.1}$$

where $amplitude_0$ is the amplitude at $depth = 0$, and $amplitude$ is the measured reflectance at a given point.

However, since this equation is an approximation, the resulting data still exhibits exponential behaviour, seen in Figure 4.1.

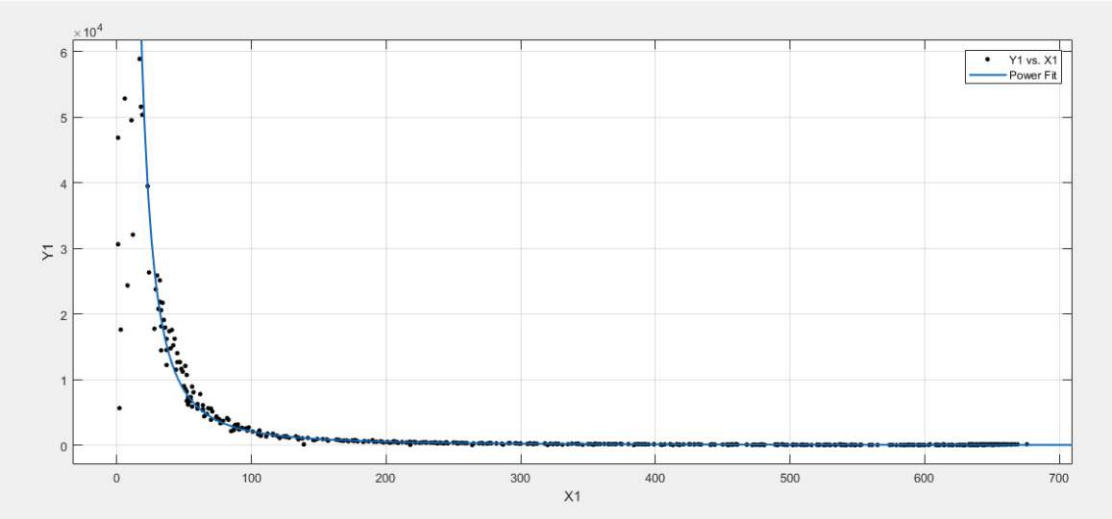


Figure 4.1: Amplitude (Y-axis, number of photons) over depth (X-axis, in mm) measurements from our experimental data. The measurements are taken from the tip of the nose.

Remark. The experiments in this thesis for the amplitude range seen in Figure 4.1 were done with only three subjects. Thus, we cannot make a generalised statement about the correct amplitude range.

By masking data outside this range and passing it on to the head detection algorithm, this approach virtually reduces the possible candidates to three: The head, the torso, and the hand. With enough data, infrared amplitude could be used to create a mask for human skin, as indicated by Mendenhall et al. [MNM15].

Due to the very low input image size, efficient real-time head detection using machine/deep learning can be achieved using both amplitude and depth data; ground truth can be created using a high-resolution sensor in tandem for annotation and after training, the weights can be uploaded to a microchip running inference.

Support Vector Machines [CV95] could also find a discriminating threshold for human skin. Both alternative methods, however, are outside the scope of this thesis, as we want to explore a comprehensible and reproducible method for solving this task.

Another idea could be depth masking, where one can take advantage of the fact that a human head is always attached to the body, which means that the lowest row of the IR array would always show part of the body: the torso, neck, or head. By taking the 50-percentile (median) of the depth values of the bottom row and creating a search window ϵ, a mask can be created, that only shows measurements at the depth of the torso/neck/head $\pm$ the search window ϵ.

However, if a person is positioned in such a way that the torso, neck, or head only takes up one sensor in the bottom row of the array, the other seven would sample the background, thus skewing the median towards the background depth values. The resulting 50-percentile would be a depth value closer to the background measurements and the search window ϵ could, depending on the depth values of the background, completely exclude the person in view since they would be significantly closer to the sensor.

4.1.1 Estimating the position of the head

Once the input is pre-processed (IR input masked on depth, reshaped to 8×8, and rotated based on the output of the gyro-sensor), head detection can begin. One of two modes is used, depending on the depth value of the closest measured point of the body (i.e. the measurement with the smallest depth value).

Body closer than 37.5cm

The body is so close to the sensor, that part of the head is clipped from the image (see 3.5). Thus, the resulting shape cannot be detected as a circle, but we can use the weighted centroid of the shape, which is within 10 cm of the true centroid of the head. This is under the assumption, that the object in frame indeed is a head and not another object of similar reflectance.

In this mode, a hand can and will be detected as a head, since there is no discrimination or filtering.

Body between 37.5cm and 73cm

The head is sufficiently far away to make circle detection possible, provided it is centred. It is also not so far away as to be registered by less than four sensors (see 3.6).

We assume that to make a gesture, a user's hand will always be closer to the sensor than their head. Thus, the difference between the closest and farthest depth measurement of the body (= depth range) will be larger with the hand in frame than without.

This assumption can be used to mitigate false positive detection of the hand. If the depth range exceeds 10 cm (the approximate depth difference between the tip of the nose and the neck), the midway distance is calculated:

$$d_{midway} = \frac{\min(d_{body})}{\max(d_{body})} \tag{4.2}$$

where d_{body} are the depth measurements of every sensor, filtered by the amplitude measurements as described at the beginning of Section 4.1.

Afterwards, a mask can be created to filter out any fields with depth values lower (= closer) than the midway distance, which is then applied to the amplitude data. This leaves only amplitude pixels that have a depth measurement in $[d_{midway}, \max(d_{body})]$.

This approach is not without fault however, as if the hand is closer to the body than the defined threshold (e.g. next to the face), false positive rate and reliability will not be improved, as the hand will still be in frame.

The resolution of the image is then up-scaled using Bi-cubic Interpolation, and to minimise the computational complexity of the following methods while still taking advantage of a higher fidelity and blob formation, a factor of 4 was chosen, as seen in Figure 4.2.

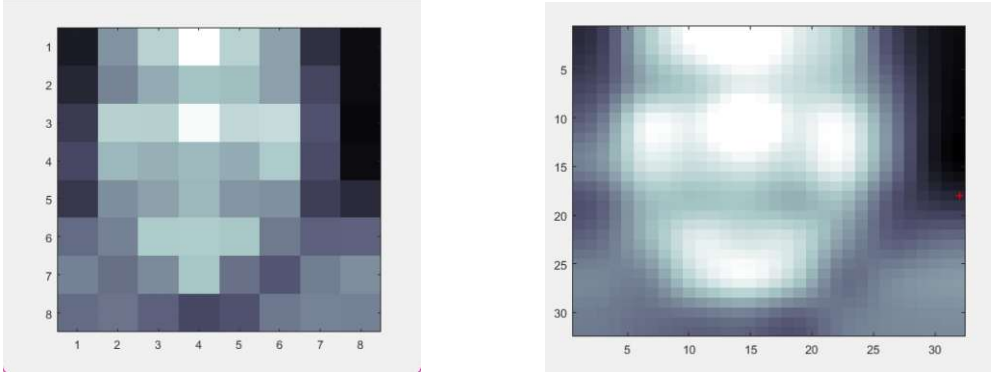


Figure 4.2: A human head at ≈ 30cm distance before and after up-scaling. Note the blob formation and higher amplitude values on the cheeks and chin after up-scaling.

Afterwards, Circular Hough Transform is applied, searching for circles with radii between 4 and 16 pixels (corresponding to circles with radii of 1-4 pixels in an 8×8 image, as seen in Figure 3.6).

This approach, however, has caveats:

1. If the head is at the edge of an image, its shape will not be circular (see Figure 3.15i) and will thus not be detected.
2. Circle detection does not differentiate between circular shapes. If there is another object in the field of view with a similar reflectance, circle detection will detect it, too.

If no circle can be detected, a centroid is calculated from the average of the top-left, top-right, left-top, and right-top shape extrema – under the assumption that the head is inside the field of view.

4.1.2 Selecting the correct head candidate

Since we have filtered for reflectance close to human skin, our head candidates should be at least one of three body parts: the head, the hand, and the torso.

As mentioned in Subsection 3.1.1, the shape of the head will lead to non-orthogonal reflection. In comparison, both the hand and torso are relatively flat and most of the time parallel to the sensor (if the user is facing the sensor), leading to a higher average amplitude compared to the head. This assumption is further corroborated by the findings of Mohamad et al. [MSJO14], visualised in Figure 2 of their publication (see Figure 4.3).

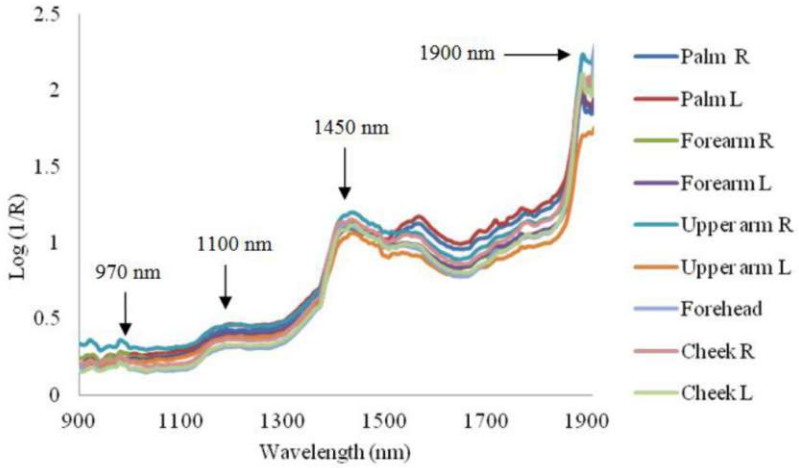


Figure 4.3: Near infrared reflectance spectra of nine different parts of face and hand. [MSJO14]. Note how the forehead and both cheeks always have a lower reflectance than the hands.

Hence, by calculating the average amplitude of each detected head candidate, the one with the lowest average amplitude is selected as the head.

4.1.3 Tracking the Head

Once a centroid has been selected and returned, it is stored in a list of positions and depths in memory with a size of $3 \times m$, for m frames. This array is updated every frame in the first-in-first-out (FIFO) principle. To smooth the positioning, one can average the positions like such:

$$P_{head}(x_n, y_n, z_n) = \frac{\sum_{i=(n-m)+1}^{n} P_{head}(x_i, y_i, z_i)}{m} \tag{4.3}$$

where n is the number of the n-th (=latest) frame in the image series, m is the memory size, $P_{head}(x_n, y_n, z_n)$ is the centroid position and measured depth of the head in the nth frame, and $P_{head}(x_i, y_i, z_i)$ is the centroid position and depth of the head in the i-th frame.

Then, the gradients $\frac{\partial P_{head}}{\partial x}, \frac{\partial P_{head}}{\partial y}, \frac{\partial P_{head}}{\partial z}$ of the centroid positions and depth values in memory are computed for each frame to get the change in position/depth, essentially applying Optical Flow [HS81] to a single three-dimensional point.

4.2 Algorithm of Gesture Recognition

If head tracking is enabled, gesture recognition waits for the head detection to return the depth of the head position. The head position is then used to create a gesture space (see Figure 3.23), and the movement of objects with depth measurements in $[0, depth_{head}]$ is tracked.

If head tracking is disabled, the gesture space is set to the whole depth range of $[0, 73]$ cm.

4.2.1 Shape Properties for Gesture Recognition

After creating the gesture space, the weighted centroid is then calculated on the closest shape within to estimate the position of the hand. As described in Subsection 3.2.2, the weighted centroid using amplitude data is an efficient and pose-independent approach to estimating the position of the hand.

4.2.2 Gesture recognition

As with tracking the head, the centroid positions and depth measurements of the hand are saved and averaged in a list of size $3 \times k$, for k frames:

$$P_{hand}(x_n, y_n, z_n) = \frac{\sum_{i=(n-k)+1}^{n} P_{hand}(x_i, y_i, z_i)}{k} \tag{4.4}$$

and the gradients $\frac{\partial P_{hand}}{\partial x}, \frac{\partial P_{hand}}{\partial y}, \frac{\partial P_{hand}}{\partial z}$ of the centroid positions of the hand are calculated.

If the sum of the absolute changes exceeds a given threshold in at least one of the three axes,

$$\begin{pmatrix} \sum_{i=n-k}^{n} \left| \frac{\partial P_{hand}}{\partial x} \right| \\ \sum_{i=n-k}^{n} \left| \frac{\partial P_{hand}}{\partial y} \right| \\ \sum_{i=n-k}^{n} \left| \frac{\partial P_{hand}}{\partial z} \right| \end{pmatrix} > \begin{pmatrix} t_x \\ t_y \\ t_z \end{pmatrix} \tag{4.5}$$

the maximum absolute change is detected as a gesture in that direction (the X-axis determines left-right gestures, the Y-axis determines up-down gestures, and the Z-axis determines forward-backward gestures). The threshold is highly dependent on the chosen array length k, which essentially defines the gesture speed.

For example, if the array length is 15, it would be equal to a memory of 1 second (at a frame rate of 15 frames per second). Setting the threshold for the (x, y)-movement to 1 would mean that a movement needs to have a speed of at least 1 pixel (or sensor) per second to be recognised as a gesture. Furthermore, due to the sum of absolute changes being used for gesture detection, setting the array length to 15 could potentially cause a delay of 1 second, if the oldest frame position $P_{hand}(x_{n-k}, y_{n-k}, z_{n-k})$ alone exceeds the set threshold.

Thus, a balance between array length and threshold is needed. Since the gesture recognition in this thesis is solely concerned with the direction and does not recognise ”shape” gestures (circle, square, triangle, etc.), an array length of 5, and an (x, y) threshold of 1.33 pixels was chosen. A shorter array length increases the detection sensitivity (thus also increasing the false positive rate), but also reduces erroneous, delayed detection described above.

Of note are the different dimensions used by the axes. While the X- and Y-axes are in the image space/pixel dimension, the Z-axis uses depth data, which is in millimetres. Applying the same threshold (e.g. moving at least 1.3 units in one direction for 15 frames), would make the algorithm overly sensitive to small movements along the Z-axis (moving at a speed of at least 0.266 mm per frame in this example). Thus, a separate (higher) threshold must be set for the depth values to be detected as a forward/backward gesture. In this thesis, the threshold set for Z-axis movement is 150 mm per second, to account for unconscious moving/shaking of the hand.

CHAPTER 5

Challenges of Head Detection and Gesture Recognition in Low-Resolution Infrared Amplitude Images

The challenges are divided into two categories: internal (adjustable) and external (non-adjustable) challenges. Internal challenges refer to the challenges of our implementation, while external challenges are environmental.

5.1 Internal Challenges

The biggest internal challenges of our head tracking are efficiency and reliability in combination with low image resolution. Detecting the head at low image sizes on the (non-thermal) infrared spectrum alone requires various compensations to mask objects that do not belong to the human body (like the amplitude range or the reflectance).

At the original image size (8×8), circle detection does not work, as shape variety is limited, leading to shape similarities (like the hand and head in Figure 5.1).

Shape variety and image fidelity can be increased by artificially increasing the resolution using Bi-cubic Interpolation and taking advantage of its blob formation, which helps with circle detection but increases the number of pixels in the image and thus, the length of computation.

Additionally, circle detection will fail at certain distances and positions; if the head is too close and connects with the image borders, no circle can be detected, and if the head is too far, less than 4 of the 64 sensors can detect it and the resulting shape is thus neither

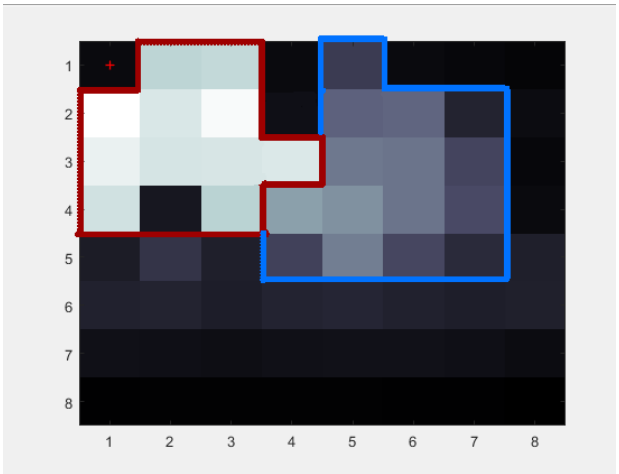


Figure 5.1: Sensor output showing the similarity of the shapes of a human hand (red) and head (blue). The hand is closer to the sensor, thus the higher amplitude/brightness.

square nor elliptical/circular. In these cases, shape properties are used to estimate the position of the head, which are not as reliable.

Choosing the correct head candidate, while keeping computational costs at a minimum is another considerable challenge. Our algorithm relies on the fact that a human head has a lower average amplitude than the palm of a hand or a torso. However, in the case where no hand is in frame, and only the head and another circular object with a similar reflectance like human skin is detected, this assumption might lead to false positive detection.

For gesture recognition, the problem of efficiency is further exasperated by the computational needs of head detection, thus forcing us to sacrifice accuracy to ensure efficient computation. Our algorithm has to rely on shape properties to detect the palm of a hand. Using shape extremes would not work, as depending on the arm pose, the palm of a hand could be at any of the left, top, and right extremes.

One possible idea to alleviate this problem is to take advantage of the circle detection done in the head detection step; after selecting a head candidate, one of the discarded circles could be in front of the head, and in this case, it is likely to be a flat hand. However, the palm of a hand might not always be detected as a circle and we would have to fall back to shape properties again, increasing the computational cost with indeterminate benefit.

Once the hand is detected, recognising a gesture reliably brings further challenges. For better usability, gesture recognition should allow for varying execution speeds, and have a threshold for angle and line imperfections (for example due to anatomy, as seen in Figure 3.24).

Relying on speed or movement length alone will result in random hand movement, gesticulation for speech emphasis, or simply the adjustment of the hand position before executing a gesture to be falsely detected as a gesture. Thus, a combination of both speed and distance needs to be defined to define a gesture, which would rely heavily on heuristics.

5.2 External Challenges

By far the biggest external challenge is the very low resolution of the sensor array, which in the case of our algorithm is "misused" to work similarly to an IR camera. Since human heads are rarely rectangular, circle detection at such low resolutions is almost impossible.

One idea to mitigate this, other than increasing the resolution, would be to place a fish-eye lens in front of the sensor, thus "warping" the input to be more circular. However, we could not test this theory.

Our 3D ToF-sensor faced the same problem as the stereo IR camera used by Krotosky et al.[KCT04] for their comparison: In low resolutions, the shape, size, and reflectance of a human hand are so close to that of a human head, that it is difficult to differentiate between the two due to the similarity in reflectance and apparent shape at low resolution (see Figure 5.1).

However, the image size is not the only problem we face with an 8×8 sensor array: the field of view is very narrow at 27°. This in turn makes detecting larger movements difficult and leads to clipping of shapes at closer distances.

Adding to that, the maximum frame rate in 8×8-mode is 15 frames per second, leading to choppy movement, potentially losing track of fast hand gestures. For example, by solving 3.4 for a at a distance of $D = 30$ cm, we can calculate the width of the field of view, which is ≈ 14.4 cm. Knowing this, a horizontal gesture at the speed of 1 m/s will at best have two frames, where the hand is moving, while 13 frames will have the hand outside of the field of view. While the sensor does allow for a faster frame rate (up to 30 Hz), it only offers this in 4×4 mode, which halves the (already small) sampling resolution.

As with every infrared sensor, one of the challenges is other infrared light sources such as candles and sunlight which falsify the amplitude and depth measurements. Although the sensor used offers a short-range mode to increase stability in regards to ambient light, we are operating the sensor at long-range mode to reduce the repeatability error. Thus, we have found that direct sunlight can still interfere with its amplitude and depth readings (see Section 6.4).

Another challenge is the reflectance, as some materials/surfaces of clothing and accessories can change the measured reflectance of the head, hand, and torso. Furthermore, it is theoretically possible that a material has the same or a very similar reflectance of human skin, leading to false positives.

For people using prostheses, the material of the prosthesis might not have the same reflectance, thus leading to different accuracy with gesture recognition. Further experimentation with a wide range of materials and prostheses is required to evaluate the performance in these cases.

Finally, simply the position and angle of the sensors can pose a challenge, since not only the imaged shapes could get distorted, but also the movement could not be defined as strictly perpendicular or parallel to the sensor. For this, arm pose estimation could be used to correctly determine the gesture being performed.

Regardless of the pose of the camera, distance is an important factor for gesture recognition: A person performing the same horizontal or vertical gesture at different distances will have varying calculations of gradients in the pixel dimension.

Say, for example, a gesture is done once at a distance of 40 cm and once at a distance of 70 cm. Solving 3.4 for r, one can see that $\tan(13.75°) \cdot 40 \approx 9.6$ and $\tan(13.75°) \cdot 70 \approx 16.8$. Thus, the field of view covers a width/height of ≈ 19.2 cm with a horizontal/vertical resolution of ≈ 2.4 cm per pixel at a distance of 40 cm, while 70 cm covers a width/height of ≈ 33.6 cm with a horizontal/vertical resolution of ≈ 4.2 cm per pixel. To overcome this challenge, distance compensation is needed, which is outside the scope of this thesis.

CHAPTER 6

Experimental Results

For evaluation, two types of experiments were performed: laboratory and field experiments.

Laboratory experiments were done under controlled conditions, and are done in a laboratory using artificial, indirect lighting, a turntable for accurate control of the rotation, a filled glove on a pendulum for gesture simulation, and accurate distance measuring.

Field experiments were done in an ”organic” setting, i.e. a living room with an open window facing the sensor at the back of the room to introduce ambient lighting. Two persons conducted the field experiments during the late morning, afternoon, and evening with artificial lighting from above.

6.1 Evaluation Goals

The goal of the laboratory experiments was to determine the functional range of angles and distances of the sensor to the user and to provide a fixed set of parameters (distance, lighting, movement speed) that enable the quantifiable reproduction of the experiments performed.

Field experiments focus on usability and ”real-life”-performance by having human subjects perform a set of movements in an environment with direct and indirect sunlight. With field experiments, the parameters of lighting, distance, gesture speed, and silhouette vary to determine edge cases and shortcomings of the algorithm and/or the sensor used.

6.2 Evaluation Method

Evaluation was done using three metrics described in this section. The average deviation over the number of frames is used for evaluating the performance of head and hand

detection, while detection rate and accuracy evaluate the performance of the gesture recognition.

The ground truth is created by annotating every frame, depending on the application:

Head Detection / Tracking:

Center of the head: (x, y)-coordinates on the image

Gesture Recognition:

Center of the hand: (x, y)-coordinates on the image

Direction of gesture: The direction of the gesture, if one was executed.

In total, 10,685 frames from 20 experiments were annotated for the evaluation of the algorithm presented in this thesis.

6.2.1 Deviation and Outliers

The average deviation is the distance between the ground truth and the centroid computed by the algorithm over every frame. The deviation for each frame is calculated using the Euclidean distance:

$$||[x_1, y_1] - [x_2, y_2]||_2 = \sqrt{|x_1 - x_2|^2 + |y_1 - y_2|^2} \tag{6.1}$$

where $[x_1, y_1]$ is the annotated center of the head or hand in the ground truth, and $[x_2, y_2]$ is the position of the centroid calculated by the algorithm.

To approximate the deviation in cm, further calculations are done: after calculating the Euclidean distance, we solve 3.4 for $\frac{a}{2}$ using the distance of the head in the ground truth as D to get half the approximate FoV in cm:

$$D_{GT} \cdot \tan(13.5°) = \frac{a}{2} \tag{6.2}$$

Dividing the approximate FoV by 8 (number of sensors along horizontal/vertical dimension) yields the approximate resolution at distance D_{GT}, which can then be multiplied by the Euclidean distance:

$$\frac{D_{GT} \cdot \tan(13.5°) \cdot 2}{8} \cdot ||[x_1, y_1] - [x_2, y_2]||_2 \tag{6.3}$$

Deviations are considered outliers if they are bigger than the upper fence using the Interquartile Range:

$$||[x_1, y_1] - [x_2, y_2]||_2 > Q3 + (1.5 \cdot IQR) \tag{6.4}$$

where $Q3$ is the 75-quartile of all deviations and $IQR = Q3 - Q1$, with $Q1$ being the 25-quartile of all deviations.

6.2.2 Gesture Recognition Evaluation

The evaluation for gesture recognition is done for each frame by comparing the ground truth to the algorithm output in regards to four factors:

Accuracy: Do the detected gestures match in direction? This metric checks the equivalency of ground truth annotation and algorithm output. Any frame, where the output differs from the ground truth, is counted as an error.

True Positive Rate: Was a gesture recognised, regardless of direction? This metric is true for every frame where *both* the ground truth and output are non-zero (gesture). If the ground truth has annotated 0 (no gesture) for a given frame, it is ignored for this metric.

False Positive Rate: Was a gesture recognised, when it was not annotated in the ground truth? This metric is true for every frame where the ground truth is 0 (no gesture) and the algorithm output is non-zero (gesture) for any frame. If the ground truth is annotated as non-zero for a given frame, it is ignored for this metric.

Miss Rate: Was no gesture recognised, when one was annotated in the ground truth? This metric is true for every frame where the ground truth is non-zero (gesture) and the algorithm output is 0 (no gesture) for any frame. If the ground truth is annotated as 0 for a given frame, it is ignored for this metric.

6.3 Laboratory Experiments

These experiments were done in a laboratory with indirect artificial ceiling light. The sensor is placed on the centre of a turntable with a diameter of 60 cm inside a photography light tent with dimensions $50 \times 50 \times 50$ cm (see Figure 6.1b). For additional experiments regarding gesture speed, a pendulum with a filled glove as an approximation for a human hand (Figure 6.1a) was used.

The turntable was fixed at four different angles: -20°, 0°, 20, and 45°. Experiments for head and hand detection of a human subject (hand length from palm to tip of the middle finger = 18 cm; length from chin to top of head = 21 cm) were performed at a fixed distance of 30 cm (edge of the turntable).

To help with the annotation, a mobile camera with a video resolution of 1920×1080 px and a frame rate of 30 frames per second was recording at the same time.

(a) Turntable with the "Hand"-Pendulum.

(b) Turntable with a ruler to mark the distance.

Figure 6.1: The turntable setup used for laboratory experiments.

6.3.1 Head Detection

Head detection evaluation is done by calculating the Euclidean distance for two-dimensional pixels, as described in Section 6.2, and then averaging the distance over all frames.

1,780 frames from 4 experiments were evaluated for simulated head detection. The results are shown in Tables 6.1 and 6.2:

Table 6.1: Head detection mean deviations in 2D for laboratory experiments (px).

	Average Deviation (px)	Min / Max Deviation (px)	Number of Outliers	Number of Frames
Head (-20°)	2.4	0.1 / 4.4	0	347
Head (0°)	1.5	0.1 / 3.3	0	537
Head (20°)	2.3	0.1 / 5.7	0	555
Head (45°)	3.6	2.4 / 4.6	0	341
Total Average	**2.5**	**0.7 / 4.5**	**0**	**445**
Median	**2.4**	**0.1 / 4.5**	**0**	**442**

Of note is that the performance is consistent in the range of ±20°. Starting from 20°, part of the head is out of frame, which means clipping occurs.

Table 6.2: Approximate head detection mean deviations in 2D for laboratory experiments (cm).

	Average Deviation (cm)	Min / Max Deviation (cm)	Average Depth Value (cm)	Number of Frames
Head (-20°)	6.2	0.1 / 15.6	45.1	347
Head (0°)	3.6	0.1 / 8.2	43.1	537
Head (20°)	6.1	0.2 / 20.0	41.0	555
Head (45°)	6.8	4.9 / 9.8	32.0	341
Total Average	**5.7**	**1.3 / 13.4**	**40.3**	**445**
Median	**6.2**	**0.2 / 12.7**	**42.1**	**442**

While the average deviation of Head (45°) in Tables 6.1 and 6.2 appears to be similar to that of Head (±20°), the minimum deviation is more telling. Because the head is completely out of the frame for most of the time, the resulting average deviation is skewed due to the small amount of frames to compare.

Analysing the input after amplitude filtering shows another problem with the chosen approach: filtering the amplitude for a range based on the approximation 4.1 does not work as intended, as the plywood section of the turntable is still visible after filtering. Hence, when the head is entirely out of frame, the only object in frame is the turntable, which is at a distance of $\leq$ 30cm. As such, the shape centroid is calculated and returned as the centre of the head, explaining the average maximum deviation of $\geq$ 4.5 px (i.e. more than half the image width) at an angle.

These results suggests that either, amplitude filtering is not the right approach for preprocessing, or that the inverse law of photometry is not accurate enough for amplitude filtering.

Furthermore, looking at the average maximum deviation of 4.5 px / 13.4 cm, we can see that the chosen approach fails to reliably detect the head even under the best possible controlled conditions. Such a distance means that the detected centroid was most likely found away from the head, as 4.5 px is more than half the image size, and 13.4 cm away from the tip of the nose could already be the background. However, remember that the laboratory experiments were done to discern reproducible weaknesses in the methodology presented in this thesis. Placing the sensor at an angle to the user was done deliberately to test the field of view of the sensor and as such, it seems that the sensor performs best between -20° and 20°.

6.3.2 Gesture Recognition

For the evaluation of gesture recognition, head detection was turned off, and a fixed gesture space of 73 cm was used.

Additionally, since the sensor is placed on a turntable in a photography light tent (see Figure 6.1), and the gesture recognition algorithm takes the closest object without

discriminating, only the photo tent or the turntable would be visible after masking, rendering the results useless. Thus, for these experiments only, a minimum depth value of 30 cm (radius of the turntable) was set to the gesture space.

Each frame output is compared to the annotations and set to true or false depending on one of the four metrics described in Section 6.2. Then, the total amount of "true" values is averaged over the amount of frames to achieve the average accuracy.

Evaluation with a human subject (length between the palm of hand and tip of middle finger = 18 cm) was done by executing 24 gestures per hand, four per direction (up, down, left, right, front, back), resulting in a total of 48 gestures per instance.

Pendulum evaluation let the pendulum swing thrice from left to right and thrice from back to bottom for each instance, letting it swing until it stood still.

Since gesture detection relies on correct hand detection, the output of the hand detection step is evaluated as well; by annotating the position of the hand in the ground truth and calculating the distance between the hand detection output and the ground truth like in Section 6.2, we can evaluate the performance of the hand detection.

In total, 3,114 frames from six experiments were evaluated for laboratory gesture recognition. The results for exact gesture recognition are seen in Table 6.3:

Table 6.3: Gesture recognition accuracy for laboratory experiments).

	True Detection Rate	False Detection Rate	Miss Rate	Average Accuracy
Hand (-20°)	46.07%	50.53%	53.93%	19.78%
Hand (0°)	26.03%	14.49%	73.97%	44.98%
Pendulum (0°)	40.00%	17.59%	60.00%	56.74%
Hand (20°)	47.60%	68.62%	52.40%	23.89%
Pendulum (20°)	29.52%	21.13%	70.48%	44.40%
Hand (45°)	11.97%	0.00%	88.03%	64.19%
Total Average	**33.54%**	**28.73%**	**66.47%**	**42.33%**
Median	**34.80%**	**19.36%**	**65.24%**	**44.69%**

Unfortunately, gesture recognition does not perform as well as head detection. Although a minimum depth value was set up to mitigate false positive detection of the light tent and turntable, it could not be eliminated, leading to a bad overall performance. Considering the average miss rate of 66.47%, it seems that the chosen method of hand detection (weighted centroid) is still too imprecise and results in the centroid being found on the forearm (see Figure 3.24).

Looking at the results of the hand detection evaluation in Tables 6.4 and 6.5 provides further information.

Table 6.4: Gesture recognition mean distances in 2D for laboratory experiments (px).

	Average Deviation (px)	Min / Max Deviation (px)	Number of Outliers	Number of Frames
Hand (-20°)	3.7	2.1 / 7.6	32	384
Hand (0°)	3.0	0.3 / 4.5	1	841
Pendulum(0°)	1.9	0.3 / 5.3	0	251
Hand (20°)	2.0	0.3 / 7.4	17	759
Pendulum (20°)	1.9	1.0 / 3.4	0	327
Hand (45°)	2.7	1.8 / 3.5	0	552
Total Average	**2.5**	**1.0 / 5.3**	**8.3**	**519**
Median	**2.4**	**0.7 / 4.9**	**0.5**	**468**

Table 6.5: Approximate gesture recognition mean distances in 2D for laboratory experiments (cm).

	Average Deviation (cm)	Min / Max Deviation (cm)	Average Depth Value (cm)	Number of Frames
Hand (-20°)	7.6	1.2 / 18.1	36.7	384
Hand (0°)	6.9	0.6 / 12.6	36.6	841
Pendulum(0°)	4.9	0.7 / 14.0	40.6	251
Hand (20°)	4.6	0.2 / 20.8	39.2	759
Pendulum (20°)	4.2	0.6 / 9.8	34.8	327
Hand (45°)	6.7	1.3 / 10.3	40.5	552
Total Average	**5.8**	**0.8 / 14.3**	**38.1**	**519**
Median	**5.8**	**0.7 / 13.3**	**38.0**	**468**

As can be seen in Tables 6.4 and 6.5, the average two-dimensional deviation is 2.5 px / 5.8 cm, which means a deviation by more than a quarter of the image width/height, or more than half the width of the hand of the person performing the laboratory experiments. Additionally of note are the high number of outliers of Hand (± 20), which explain the particularly high false positive rate of both experiments. Outliers in hand position estimation will lead to movement being detected where there is none.

If the sensor is at an angle, and the depth value of the hand from the sensor exceeds 40 cm, the calculated centroid would stay on the turntable or light tent, since they are the closest objects in view. Furthermore, if the hand/arm moves closer to the sensor than the set minimum depth value of 30 cm, it will not be considered for hand detection, again making the turntable or light tent the "closest" object to the sensor. Frames, where the algorithm "jumps" from the hand/arm to the turntable or the light tent might lead to false positive recognition of movement. If the hand stays out of the depth range of $30 - 40$ cm, the centroid will not move, which might explain the miss rate of 66.47%.

6.4 Field Experiments

The purpose of field experiments is to ascertain reliability in less controlled environments and more "realistic" situations. The sensor is placed in a living room at the edge of a table with a height of 74 cm, 4.75 metres across an unobstructed window facing east. Figure 6.2 shows a schematic illustration of the room.

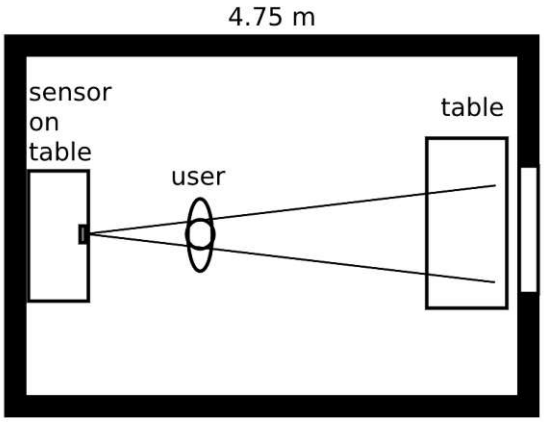


Figure 6.2: Schematic overview of the field experiment setup. Both tables are below the window on the right.

Head detection was evaluated during the late morning, when ambient light is strong from the window, and in the evening with artificial lighting. Gesture recognition was evaluated during the afternoon when ambient light was weaker.

As with the simulated experiments, a mobile camera with a resolution of 1920×1080 px and a frame rate of 30 frames per second was recording simultaneously to help annotation of the ground truth.

6.4.1 Head Detection

To evaluate the performance of head detection for qualitative experiments, two human subjects with differing skin tones executed a set of head movements (Person 1: length from chin to top of head = 21 cm, Person 2: length from chin to top of head = 18 cm). To test the impact of different silhouettes, over-ear headphones and glasses were used. Person 1 was evaluating during the late morning, while Person 2 was evaluating during the evening.

As with the head detection in the simulated experiments (Section 6.3.1), the average deviation is calculated over every frame. 4,449 frames from 7 experiments were evaluated for head detection under these conditions. The results can be seen in Tables 6.6 and 6.7.

Table 6.6: Head detection mean distances in 2D for field experiments (px).

	Average Deviation (px)	Min / Max Deviation (px)	Number of Outliers	Number of Frames
Person 1 @ 20cm	3.8	0.6 / 6.2	8	855
Person 1 @ 35cm	1.9	0.0 / 6.8	9	904
Person 1 @ 40cm	1.9	0.1 / 6.6	9	660
Person 1 @ 50cm	2.3	0.4 / 5.4	3	941
Person 2 @ 45cm	0.8	0.1 / 3.3	21	283
Person 2 (Glasses) @ 45cm	0.6	0.0 / 4.3	18	363
Person 1 (Headphones) @ 45cm	2.2	0.1 / 7.0	10	443
Total Average	**1.9**	**0.2 / 5.7**	**11.1**	**636**
Median	**1.9**	**0.1 / 6.2**	**9**	**660**

Table 6.7: Approximate head detection mean distances in 2D for field experiments (cm).

	Average Deviation (cm)	Min / Max Deviation (cm)	Average Depth Value (cm)	Number of Frames
Person 1 @ 20cm	4.9	0.3 / 7.4	21.6	855
Person 1 @ 35cm	4.6	0.0 / 16.7	41.1	904
Person 1 @ 40cm	4.9	0.2 / 15.4	45.1	660
Person 1 @ 50cm	6.0	0.8 / 18.2	48.3	941
Person 2 @ 45cm	2.1	0.2 / 7.5	49.0	283
Person 2 (Glasses) @ 45cm	1.7	0.1 / 9.0	49.1	363
Person 1 (Headphones) @ 45cm	5.2	0.2 / 17.6	43.0	443
Total Average	**4.2**	**0.3 / 17.6**	**42.5**	**636**
Median	**4.9**	**0.2 / 15.4**	**45.1**	**660**

The results indicate an average two-dimensional deviation of 1.9 px / 4.2 cm, with exceptional performance of the algorithm in the experiments with Person 2. Considering that the evaluation of Person 1 was performed during the late morning with strong ambient light from the window in the background, it seems that the sensor is indeed susceptible to ambient infrared radiation. This is further indicated by the higher maximum deviation in comparison to the laboratory experiments in Tables 6.1 and 6.2. Thus, it seems that the algorithm cannot perform head detection reliably under strong ambient light, leading to the centroid jumping between head and window.

In contrast to the laboratory experiments, amplitude filtering successfully masks anything other than the head and torso of the participants, as the change of the silhouette due to over-ear headphones and the use of glasses did not negatively impact the performance. This further points toward amplitude filtering being dependent on ambient lighting.

Of note is that the higher number of outliers of Person 2 is due to the low average deviation, leading to a smaller interquartile range and thus a higher sensitivity to variance.

Thus, the head detection algorithm described in this thesis can be used in the context of (home-)entertainment. For medical and safety applications like in a vehicle, a significantly higher reliability (i.e. a lower amount and smaller extent of outliers) is needed. Especially in vehicles, where ambient lighting changes rapidly, a lighting-independent method is necessary.

6.4.2 Gesture recognition

For gesture recognition, a series of consecutive gestures were performed: two up-gestures, then two down-gestures, an alteration of left and right gestures twice, a back gesture, and finally a front gesture. The gestures were performed during the afternoon, with the length between the palm and tip of the middle finger being 18 cm.

As with the laboratory experiments of gesture recognition, the four metrics and the calculation of the deviation described in Section 6.2 were used. This time, head detection was turned on to evaluate the performance of the combined algorithms. As there is no turntable in frame, no compensation was needed and thus the minimum depth value was removed.

In total, 1,342 frames from 3 experiments were evaluated for qualitative gesture recognition. The results for gesture recognition are shown in Table 6.8.

Table 6.8: Gesture recognition accuracy for field experiments (with head detection).

	True Detection Rate	False Detection Rate	Miss Rate	Average Accuracy
Person 1 @ 20cm	37.90%	9.16%	62.10%	66.47%
Person 1 @ 35cm	20.13%	39.02%	79.87%	32.39%
Person 1 @ 40cm	9.63%	16.45%	90.37%	44.60%
Total Average	**22.55%**	**21.54%**	**77.45%**	**47.82%**
Median	**20.13%**	**16.45%**	**79.87%**	**44.60%**

What is immediately apparent is the worse performance compared to the laboratory experiments in Table 6.3, indicating that the creation of the gesture space might not be the correct approach for reliable gesture recognition with the head in frame. This can be verified by turning off head detection and using a fixed gesture space of 73 cm, as seen in Table 6.9.

As can be seen in Table 6.9, both the true positive and the false positive rate is increased when head detection is turned off. This might be an indicator of the memory length being too short with 5 frames or the threshold being too low at 1.33 px per 5 frames since both regulate the sensitivity to movement. Another possibility is that the gesture

Table 6.9: Gesture recognition accuracy for field experiments (without head detection).

	True Detection Rate	False Detection Rate	Miss Rate	Average Accuracy
Person 1 @ 20cm	58.95%	25.10%	41.05%	54.91%
Person 1 @ 35cm	35.71%	63.42%	64.29%	19.81%
Person 1 @ 40cm	51.11%	71.71%	48.89%	16.38%
Total Average	**48.59%**	**53.41%**	**51.41%**	**30.37%**
Median	**35.71%**	**63.42%**	**48.89%**	**19.81%**

space created by the head detection is too small and could achieve better results by being set bigger.

The two-dimensional deviations shown in Tables 6.10 and 6.11 offer further insight.

Table 6.10: Gesture recognition mean distances in 2D for field experiments.

	Average Deviation (px)	Min / Max Deviation (px)	Number of Outliers	Number of Frames
Person 1 @ 20cm	4.9	0.0 / 7.8	0	489
Person 1 @ 35cm	3.0	0.2 / 8.7	0	449
Person 1 @ 40cm	3.4	0.2 / 7.9	0	404
Total Average	**3.8**	**0.1 / 8.1**	**0**	**447**
Median	**3.0**	**0.2 / 7.8**	**0**	**449**

Table 6.11: Approximate gesture recognition mean distances in 2D for field experiments.

	Average Deviation (cm)	Min / Max Deviation (cm)	Average Depth Value (cm)	Number of Frames
Person 1 @ 20cm	7.5	0.0 / 15.6	24.6	489
Person 1 @ 35cm	8.3	0.2 / 22.0	41.7	449
Person 1 @ 40cm	8.7	0.7 / 24.5	42.6	404
Total Average	**8.2**	**0.3 / 20.7**	**36.3**	**447**
Median	**8.3**	**0.2 / 22.0**	**41.7**	**449**

As can be seen in Tables 6.10 and 6.11, the average deviation of 3.8 px / 8.2 cm correlates with the worse performance, and the maximum deviation is constantly above 7.8 px / 15.6 cm – almost an entire image length/width or the length of the hand of the person doing the evaluations.

This can be explained by ambient light creating false measurements with the sensor, but could also just further indicate that a weighted centroid on the closest object alone is not the right approach for accurate and reliable gesture recognition.

Another point of consideration is the importance of positional stability, compared to head tracking. With head tracking, as long as the centroid is in the region of the head,

an average deviation of 3.8 px / 8.2 cm does not necessarily result in poor performance.

However, with gesture recognition, if the centroid abruptly changes position from frame to frame, it will inadvertently lead to false directional changes, even if the average of every frame is taken for gesture recognition. Therefore, robust and reliable hand detection is needed first to make usable gesture recognition possible.

CHAPTER 7

Conclusion

In this thesis we have shown an efficient hand-crafted method for tracking a human head and recognising gestures using an 8×8 infrared sensor array. The sensor used is a novel 3D-ToF IR-sensor array (ST VL53L1X) with the capability to return both amplitude and depth data, resulting in a new form of input data not yet used.

We have discussed related work and have given a broad overview of the theory and methodologies employed in the tasks of head detection and gesture recognition. Challenges faced during implementation and experiments, as well as possible future complications, have been documented including ways to mitigate them. The experimental results are discussed and analysed in depth, with theories explaining particular results.

Head detection has shown promising results (total average deviation of 1.9 px/4.2 cm in field experiments, and an average deviation of 2.5 px/5.7 cm in laboratory experiments) and if efficiency is less of a concern, could be further improved upon, with possible starting points described in this thesis.

Gesture recognition performed poorly (total average true positive rate of 22.55% and false positive rate of 21.54% in field experiments, and a true positive rate of 33.54% and false positive rate of 28.73% in laboratory experiments), alternatives have been discussed that could make simultaneous head and gesture detection a reliable possibility.

The technology of IR-ToF sensors in combination with pattern recognition algorithms could be a viable interface for machines in fields like (home-)entertainment, medicine, and even automobiles. Especially at very low resolutions, privacy concerns can be kept at a minimum and the acceptance of private usage of such technology can be favourable.

List of Figures

List of Tables

Bibliography

[AK99] Tim J Atherton and Darren J Kerbyson. Size invariant circle detection. *Image and Vision computing*, 17(11):795–803, 1999.

[Bal81] Dana H Ballard. Generalizing the Hough transform to detect arbitrary shapes. *Pattern recognition*, 13(2):111–122, 1981.

[BBSV06] Bastiaan J Boom, GM Beumer, Luuk J Spreeuwers, and Raymond NJ Veldhuis. The effect of image resolution on the performance of a face recognition system. In *2006 9Th international conference on control, automation, robotics and vision*, pages 1–6. IEEE, 2006.

[BK23] Majid Banaeyan and Walter G. Kropatsch. Distance Transform in Parallel Logarithmic Complexity. In *12th International Conference on Pattern Recognition Applications and Methods*. SCITEPRESS, 2023.

[BT81] Harry G Barrow and Jay M Tenenbaum. Interpreting line drawings as three-dimensional surfaces. *Artificial intelligence*, 17(1-3):75–116, 1981.

[CA13] Catherine C Cooksey and David W Allen. Reflectance measurements of human skin from the ultraviolet to the shortwave infrared (250 nm to 2500 nm). In *Active and Passive Signatures IV*, volume 8734, pages 152–160. SPIE, 2013.

[Can86] John Canny. A Computational Approach to Edge Detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PAMI-8(6):679–698, 1986.

[CJH+19] Ke Chen, Kui Jia, Heikki Huttunen, Jiri Matas, and Joni-Kristian Kämäräinen. Cumulative attribute space regression for head pose estimation and color constancy. *Pattern Recognition*, 87:29–37, 2019.

[CRP08] Jae Young Choi, Yong Man Ro, and Konstantinos N. Plataniotis. Feature subspace determination in video-based mismatched face recognition. In *2008 8th IEEE International Conference on Automatic Face & Gesture Recognition*, pages 1–6, 2008.

[CSX+18] Qiong Cao, Li Shen, Weidi Xie, Omkar M Parkhi, and Andrew Zisserman. Vggface2: A dataset for recognising faces across pose and age. In *2018 13th IEEE international conference on automatic face & gesture recognition (FG 2018)*, pages 67–74. IEEE, 2018.

[Cut90] LJ Cutrona. Synthetic aperture radar. *Radar handbook*, 2:2333–2346, 1990.

[CV95] Corinna Cortes and Vladimir Vapnik. Support-vector networks. *Machine learning*, 20:273–297, 1995.

[FLCS12] Clinton Fookes, Frank Lin, Vinod Chandran, and Sridha Sridharan. Evaluation of image resolution and super-resolution on face recognition performance. *Journal of Visual Communication and Image Representation*, 23(1):75–93, 2012.

[GDG11] Mislav Grgic, Kresimir Delac, and Sonja Grgic. SCface–surveillance cameras face database. *Multimedia tools and applications*, 51:863–879, 2011.

[Gla14] Andrew S Glassner. *Principles of digital image synthesis*. Elsevier, 2014.

[GST+20] Tiffany K Gill, E Michael Shanahan, Graeme R Tucker, Rachelle Buchbinder, and Catherine L Hill. Shoulder range of movement in the general population: age and gender stratified normative data using a community-based cohort. *BMC musculoskeletal disorders*, 21:1–9, 2020.

[GXX+17] Bin-Bin Gao, Chao Xing, Chen-Wei Xie, Jianxin Wu, and Xin Geng. Deep label distribution learning with label ambiguity. *IEEE Transactions on Image Processing*, 26(6):2825–2838, 2017.

[HGMBJ90] Mary C Hume, Harris Gellman, Harry McKellop, and Robert H Brumfield Jr. Functional range of motion of the joints of the hand. *The Journal of hand surgery*, 15(2):240–243, 1990.

[HH06] Shinji Hayashi and Osamu Hasegawa. A detection technique for degraded face images. In *2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'06)*, volume 2, pages 1506–1512. IEEE, 2006.

[Hil84] Ellen C Hildreth. The computation of the velocity field. *Proceedings of the Royal society of London. Series B. Biological sciences*, 221(1223):189–220, 1984.

[Hou62] Paul VC Hough. Method and means for recognizing complex patterns, December 18 1962. US Patent 3,069,654.

[HS81] Berthold KP Horn and Brian G Schunck. Determining optical flow. *Artificial intelligence*, 17(1-3):185–203, 1981.

[HSXS13] Jungong Han, Ling Shao, Dong Xu, and Jamie Shotton. Enhanced computer vision with microsoft kinect sensor: A review. *IEEE transactions on cybernetics*, 43(5):1318–1334, 2013.

[KCT04] Stephen J Krotosky, Shinko Y Cheng, and Mohan M Trivedi. Face detection and head tracking using stereo and thermal infrared cameras for" smart" airbags: a comparative analysis. In *Proceedings. The 7th International IEEE Conference on Intelligent Transportation Systems (IEEE Cat. No. 04TH8749)*, pages 17–22. IEEE, 2004.

[Key81] Robert Keys. Cubic convolution interpolation for digital image processing. *IEEE transactions on acoustics, speech, and signal processing*, 29(6):1153–1160, 1981.

[KIHF06] Walter G Kropatsch, Adrian Ion, Yll Haxhimusa, and Thomas Flanitzer. The eccentricity transform (of a digital shape). In *International Conference on Discrete Geometry for Computer Imagery*, pages 437–448. Springer, 2006.

[KKL+21] Khalil Khan, Rehan Ullah Khan, Riccardo Leonardi, Pierangelo Migliorati, and Sergio Benini. Head pose estimation: A survey of the last ten years. *Signal Processing: Image Communication*, 99:116479, 2021.

[KRG94] Senthil Kumar, Nathan Ranganathan, and Dmitry Goldgof. Parallel algorithms for circle detection in images. *Pattern Recognition*, 27(8):1019–1028, 1994.

[Kro90] Walter G. Kropatsch. Image pyramids and curves, 1990. Universität Innsbruck.

[KVB88] N. Kanopoulos, N. Vasanthavada, and R.L. Baker. Design of an image edge detection filter using the sobel operator. *IEEE Journal of Solid-State Circuits*, 23(2):358–367, 1988.

[LBJP12] Barid Baran Lahiri, Subramaniam Bagavathiappan, T Jayakumar, and John Philip. Medical applications of infrared thermography: a review. *Infrared physics & technology*, 55(4):221–235, 2012.

[Lin93] Tony Lindeberg. Detecting salient blob-like image structures and their scales with a scale-space primal sketch: A method for focus-of-attention. *International Journal of Computer Vision*, 11(3):283–318, 1993.

[LKC94] Ta-Chih Lee, Rangasami L Kashyap, and Chong-Nam Chu. Building skeleton models via 3-d medial surface axis thinning algorithms. *CVGIP: Graphical Models and Image Processing*, 56(6):462–478, 1994.

[Low04] David G Lowe. Distinctive image features from scale-invariant keypoints. *International journal of computer vision*, 60:91–110, 2004.

[LP02] A. Lemieux and M. Parizeau. Experiments on eigenfaces robustness. In *2002 International Conference on Pattern Recognition*, volume 1, pages 421–424 vol.1, 2002.

[LS18] Seungsu Lee and Takeshi Saitoh. Head pose estimation using convolutional neural network. In *IT Convergence and Security 2017: Volume 1*, pages 164–171. Springer, 2018.

[LWZ+20] Hai Liu, Xiang Wang, Wei Zhang, Zhaoli Zhang, and You-Fu Li. Infrared head pose estimation with multi-scales feature fusion on the IRHP database for human attention recognition. *Neurocomputing*, 411:510–520, 2020.

[Mar00] Karen Marshall. Color FERET database. `https://doi.org/10.18434/M31475` (Accessed 2024-01-18), 2000.

[MH80] David Marr and Ellen Hildreth. Theory of edge detection. *Proceedings of the Royal Society of London. Series B. Biological Sciences*, 207(1167):187–217, 1980.

[MHWH17] Andre Mewes, Bennet Hensen, Frank Wacker, and Christian Hansen. Touchless interaction with software in interventional radiology and surgery: a systematic literature review. *International journal of computer assisted radiology and surgery*, 12:291–305, 2017.

[MNM15] Michael J Mendenhall, Abel S Nunez, and Richard K Martin. Human skin detection in the visible and near infrared. *Applied optics*, 54(35):10559–10570, 2015.

[MSJO14] M Mohamad, ARM Sabbri, MZ Mat Jafri, and AF Omar. Correlation between near infrared spectroscopy and electrical techniques in measuring skin moisture content. In *Journal of Physics: Conference Series*, volume 546, pages 12–21. IOP Publishing, 2014.

[NB21] Rubén E Nogales and Marco E Benalcázar. Hand gesture recognition using machine learning and infrared information: a systematic literature review. *International Journal of Machine Learning and Cybernetics*, 12(10):2859–2886, 2021.

[NGC92] C Wayne Niblack, Phillip B Gibbons, and David W Capson. Generating skeletons and centerlines from the distance transform. *CVGIP: Graphical Models and image processing*, 54(5):420–437, 1992.

[OGG+22] Moritz Oppliger, Jonas Gutknecht, Roman Gubler, Matthias Ludwig, and Teddy Loeliger. Sensor Fusion of 3D Time-of-Flight and Thermal Infrared Camera for Presence Detection of Living Beings. In *2022 IEEE Sensors*, pages 1–4. IEEE, 2022.

[P+70] Judith MS Prewitt et al. Object enhancement and extraction. *Picture processing and Psychopictorics*, 10(1):15–19, 1970.

[RMG+99] Gaël Richard, Y Mengay, I Guis, N Suaudeau, Jérôme Boudy, Philip Lockwood, C Fernandez, F Fernández, Constantine Kotropoulos, Anastasios Tefas, et al. Multi modal verification for teleservices and security applications (M2VTS). In *Proceedings IEEE International Conference on Multimedia Computing and Systems*, volume 2, pages 1061–1064. IEEE, 1999.

[RP68] Azriel Rosenfeld and John L Pfaltz. Distance functions on digital pictures. *Pattern recognition*, 1(1):33–61, 1968.

[RWL+14] David P Roy, Michael A Wulder, Thomas R Loveland, Curtis E Woodcock, Richard G Allen, Martha C Anderson, Dennis Helder, James R Irons, David M Johnson, Robert Kennedy, et al. Landsat-8: Science and product vision for terrestrial global change research. *Remote sensing of Environment*, 145:154–172, 2014.

[SBB01] Terence Sim, Simon Baker, and Maan Bsat. The CMU Pose, Illumination and Expression database of human faces. *Carnegie Mellon University Technical Report CMU-RI-TR-OI-02*, 2001.

[SHB13] Milan Sonka, Vaclav Hlavac, and Roger Boyle. *Image processing, analysis and machine vision*. Springer, 2013.

[STM11] Matthew Sardelli, Robert Z Tashjian, and Bruce A MacWilliams. Functional elbow range of motion for contemporary tasks. *JBJS*, 93(5):471–477, 2011.

[STM18] STMicroelectronics. *A new generation, long distance ranging Time-of-Flight sensor based on ST's FlightSense™ technology*, 2018. Rev. 3.

[TZM19] Shigeyuki Tateno, Yiwei Zhu, and Fanxing Meng. Hand gesture recognition system for in-car device control based on infrared array sensor. In *2019 58th Annual conference of the society of instrument and control engineers of Japan (SICE)*, pages 701–706. IEEE, 2019.

[UVG+14] Rubén Usamentiaga, Pablo Venegas, Jon Guerediaga, Laura Vega, Julio Molleda, and Francisco G Bulnes. Infrared thermography for temperature measurement and non-destructive testing. *Sensors*, 14(7):12305–12348, 2014.

[VJ01] Paul Viola and Michael Jones. Rapid object detection using a boosted cascade of simple features. In *Proceedings of the 2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition. CVPR 2001*, volume 1, pages I–I. Ieee, 2001.

[WBA+13] Piotr Wojtczuk, David Binnie, Alistair Armitage, Tim Chamberlain, and Carsten Giebeler. A touchless passive infrared gesture sensor. In *Adjunct Proceedings of the 26th Annual ACM Symposium on User Interface Software and Technology*, pages 67–68, 2013.

[WBRF13] Frank Weichert, Daniel Bachmann, Bartholomäus Rudak, and Denis Fisseler. Analysis of the accuracy and robustness of the leap motion controller. *Sensors*, 13(5):6380–6393, 2013.

[WMJW+14] Zhifei Wang, Zhenjiang Miao, QM Jonathan Wu, Yanli Wan, and Zhen Tang. Low-resolution face recognition: a review. *The Visual Computer*, 30:359–386, 2014.

[WS17] Oliver Wasenmüller and Didier Stricker. Comparison of Kinect v1 and v2 depth images in terms of accuracy and precision. In *Computer Vision–ACCV 2016 Workshops: ACCV 2016 International Workshops, Taipei, Taiwan, November 20-24, 2016, Revised Selected Papers, Part II 13*, pages 34–45. Springer, 2017.

[ZY11] Wilman WW Zou and Pong C Yuen. Very low resolution face recognition problem. *IEEE Transactions on image processing*, 21(1):327–340, 2011.