Trinity College Dublin
Coláiste na Tríonóide, Baile Átha Cliath
The University of Dublin

School of Computer Science and Statistics

CPR Assistant: Chest Compression Depth from depth images

Sadhbh Daly

[redacted]

May 5, 2020

A Final Year Project submitted in partial fulfilment

of the requirements for the degree of

MAI (Electronic and Computer Engineering)

Declaration

I hereby declare that this project is entirely my own work and that it has not been submitted as an exercise for a degree at this or any other university.

I have read and I understand the plagiarism provisions in the General Regulations of the University Calendar for the current year, found at `http://www.tcd.ie/calendar`.

I have also completed the Online Tutorial on avoiding plagiarism 'Ready Steady Write', located at `http://tcd-ie.libguides.com/plagiarism/ready-steady-write`.

Signed [redacted] Date: 30/4/2020

Abstract

CPR is an emergency technique used to maintain blood circulation in a person suffering from a cardiac arrest. CPR involves the use of chest compressions and artificial ventilation, thirty chest compressions followed by two rescue breaths. CPR is used both during out-of-hospital cardiac arrest (OHCA) and in hospital emergency situations, before a defibrillator can be administered.

There are three quality parameters, chest compression rate (CCR), chest compression depth (CCD), and chest compression fraction (CCF) that govern the effectiveness of the CPR performed. This project focuses on CCD.

CCD is the depth the chest is pressed down by the CPR performer. A study conducted by Duval *et al.*[1] to find the optimal combination of CCR and CCD found 107 chest compressions per minute and a compression depth of 4.7cm increased the chance of survival. Within 20.0% of the CCR and CCD increased survival by 6.0%. This combination of CCR and CCD was effective regardless of the patient's age, gender, presenting cardiac rhythm or cardiac adjunct used.

The proposed algorithm uses optical flow to locate the CPR performer, locates the shoulders to create a template and uses chamfer matching to find the shoulders in each frame.

8 videos of a CPR performer performing CPR were tested. Two CPR performers were filmed for the test videos. The proposed algorithm was not successful in calculating the CCD as the MAE was $\pm$ 16.6mm. This is too high to yield an accurate result. The algorithm detected over 90% compressions in 60% of the videos tested.

Acknowledgements

I would like to thank my supervisor Dr Kenneth Dawson-Howe for his continued time, support and guidance during this project. I would also like to thank my supervisor, Mr. Glenn Strong and Trinity College Dublin for their support and understanding during this difficult and disruptive time of the Covid-19 pandemic. Thank you to my family and friends for their endless encouragement and assistance.

Contents

List of Figures

List of Tables

1 Introduction

1.1 Objective

The objective of this project is to evaluate proof of concept that the chest compression depth (CCD) of a cardiopulmonary resuscitation (CPR) performer can be calculated using a depth camera. A similar project was undertaken by Whittam[2]. A depth camera was not used and as a result there were concerns the CCD calculated was not accurate. The project outlined in this report attempts to improve the accuracy of the calculated CCD by using a depth camera and tracking a different part of the CPR performers body to establish the vertical movement.

1.2 Motivation

CPR is a technique used to continue blood circulation in a person who has experienced a cardiac arrest. Performing CPR on a person suffering from a cardiac arrest, before medical professionals arrive, can increase the person's chance of survival[1]. The quality parameters that quantify the quality of the CPR are chest compression fraction (CCF), chest compression rate (CCR), and chest compression depth (CCD). Performing CPR within the CPR quality parameters range increases the patient's chance of survival more than simply performing CPR without adhering to the CPR quality parameters.

There are devices that can be used for training purposes that accurately monitor the CPR quality parameter, such as the Resusci Anne mannequin. However, these training devices are costly making them not readily available to people learning CPR. These devices also cannot be used during a cardiac arrest emergency. Algorithms for smartwatches have been developed to calculate CCR and CCD. However, there is concerns that these smartwatch algorithm will not calculates CCD correctly if the user lifts their hands during a CPR compression[3]. For this reason this technique may not be practical for those who do not have adequate CPR training as medical professionals do.

Computer vision techniques could be used to track the CPR performer and possibly calculate quality parameters. Today many people have mobile phones with cameras. This means a computer vision algorithm capable of calculating the CPR quality parameters would be accessible to a considerably larger number of people than other solutions. The issue with using a phone's camera is that the distance the CPR performer is from the camera must be estimated. This can lower the accuracy of the results if the camera is at an angle while filming the CPR performer or the object in the scene of a known size, used to estimate size of a pixel in a real world metrics e.g. mm, is measured incorrectly.

A solution to this problem is a depth camera. Recently depth cameras have been incorporated into smart phones e.g. Samsung Galaxy S10[4], Huawei P30 Pro[5], and Apple IPhone XR[6]. A depth camera could calculate the distance the CPR performer is from the camera and the size of a pixel in real world metrics. This could result in a more accurate calculation of the CCD.

1.3 Project Overview

The report for this project is organised in the following way:

- Chapter 2, Background, gives an explanation of cardiopulmonary resuscitation (CPR) and the parameters used to quantify the quality of the CPR. It also details the previous work carried out by others and the results that were achieved.

- Chapter 3, Implementation, outlines the implementation of the proposed algorithm. The hardware, software and computer vision methods used are explained.

- Chapter 4, Evaluation reports the results achieved by the proposed algorithm. The results are analysed and compared with the ground truth. The method of extracting ground truth is described.

- Chapter 5, Conclusions, discusses the results obtained by the algorithm and possible further work.

2 Background

2.1 Cardiopulmonary Resuscitation (CPR)

CPR is an emergency technique used to maintain blood circulation in a person suffering from a cardiac arrest. CPR involves the use of chest compressions and artificial ventilation, thirty chest compressions followed by two rescue breaths. CPR is used both during out-of-hospital cardiac arrest (OHCA) and in hospital emergency situations, before a defibrillator can be administered. CPR should be performed on a hard surface, such as the ground, to ensure compressions are delivered correctly. Compression only CPR is suggested by the American Heart Association for laypeople. It is simpler to learn and potentially increases the likelihood of a layperson performing CPR [7].

Figure 2.1: The steps involved in CPR[8]

2.2 CPR Quality Parameters

The American Heart Association describes compressions as "forceful rhythmic applications of pressure over the lower half of the sternum" [9]. There are three quality parameters, chest compression rate (CCR), chest compression depth (CCD), and chest compression fraction (CCF) that govern the effectiveness of the CPR performed.

2.2.1 Chest Compression Fraction (CCF)

CCF is the percentage of time chest compressions are performed on the patient. Chest compressions may be stopped during CPR to carry out rescue breaths, check the patient's pulse and analysing the heart's rhythm with an Automated External Defibrillator (AED). The American Heart Association advises a CCF above 80% [10].

2.2.2 Chest Compression Rate (CCR)

CCR is the rate of chest compressions per minute. It is recommended by the American Heart Association to complete between 100 to 120 chest compressions per minute [10].

2.2.3 Chest Compression Depth (CCD)

CCD is the depth the chest is pressed down by the CPR performer. A study conducted by Duval *et al.*[1] to find the optimal combination of CCR and CCD found 107 chest compressions per minute and a compression depth of 4.7cm increased the chance of survival. Within 20.0% of the CCR and CCD increased survival by 6.0%. This combination of CCR and CCD was effective regardless of the patient's age, gender, presenting cardiac rhythm or cardiac adjunct used.

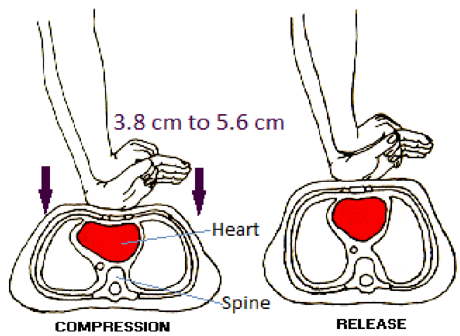

Figure 2.2: Compression placement and CCD range

2.3 Previous Work

The CCR and CCD are vital to the quality of the CPR being preformed. The use of technology to measure and report these quality parameters is advantageous for both training and emergency circumstances. Medical devices have been developed that are capable of measuring both parameters accurately[11][12]. However, these devices are costly and cumbersome. As a result, the likelihood of a bystander, present during a cardiac arrest, having access to such devices is low. These devices are mainly used by medical professionals and for training purposes. However, the cost of the devices is a deterrent for laypeople as it increases the cost of a CPR course. The increasing availability of smart devices, such as smartwatches and smartphones, has led to efforts to accurately measure and report the CCR and CCD during CPR.

Smartwatch applications have been developed to measure the CCR and CCD parameters[13][14]. These applications take advantage of the accelerometer present in smartwatches to calculate these parameters. A study conducted by Lu *et al.*[15] concluded that the use of smartwatches to measure and relay the quality parameters to medical professionals improved their CPR performance. It was noted that without the device CPR was preformed too fast and shallow. This highlights the benefit of having access to real time feedback of the quality of CPR being preformed, even for trained professionals. However, it should be recognised that this method of calculating the CCD does not consider the position in which the chest compressions are being executed[14]. In some cases, when performing CPR people lift the back of their hands during the chest compressions[3]. This increases the vertical movement of their wrists which will cause an inaccurate calculation of the CCD, if calculated using a smartwatch app. Therefore, a system capable of analysing more of the CPR performer than the wrist motion may yield more accurate CCD results.

The CCR can be accurately measured using the camera of a smartphone. Corkery[16] studied the possibility of using a smartphone camera to evaluate CPR delivery. It was found that 99% of chest compressions and 100% of rescue breaths were recognised by the system developed. The CCR was accurately determined using the velocity vectors of the pixels, determined using dense optical flow[17]. The app produced real time feedback, audio and visual, to the CPR performer. The smartphone camera was positioned facing the CPR performer so that the whole body of the performer was visible to the camera i.e. the head, shoulders, wrists and hands of the CPR performer. This vantage point has the advantage of

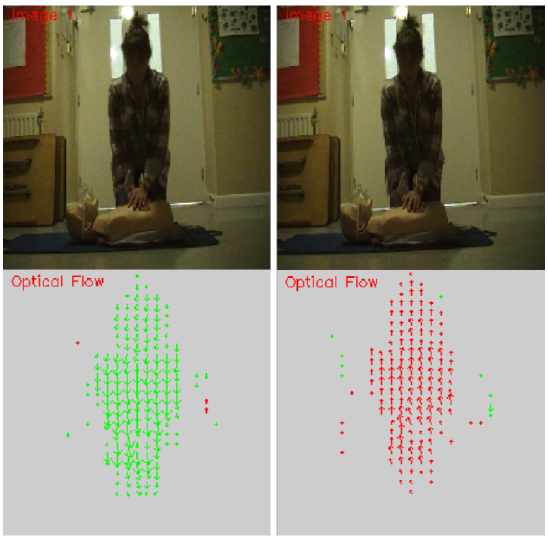

Figure 2.3: Dense optical flow showing downward movement (left) and upward movement (right)[16]

allowing the position of the person executing the chest compressions to be analysed. However, the disadvantage of this view point is the possible difficulty of appropriately positioning the camera in an emergency situation.

Measuring the CCD using a smartphone is challenging for a number of reasons. The system must locate the region of interest (ROI) of the CPR performer precisely and then accurately track this ROI during chest compressions and rescue breaths. The system must also be able to convert the number of pixels the ROI has moved in a chest compression to physical distance i.e mm.

A study administered by Wittam[2] investigated measuring CCD using a smartphone camera. Similar to the study conducted by Corkery[16] the camera was positioned facing the CPR performer and the patient so that the whole front profile of the CPR performer was visible. The system designed tracked the wrists of the CPR performer. This was achieved by the user manually selecting a sample of their clothes. This is used to locate the CPR performer in the frame and hence the end of the sleeves of the piece of clothing. It is at the sleeve ending that the chest compressions are tracked. Therefore, this method tracks the wrist movement during a chest compression, provided the CPR performer is wearing long sleeves. The issue of finding the physical distance is overcome by using the length of the CPR performers arm, manually inputted into the system, to estimate the CCD measurement. A table tennis ball attached to the CPR performers wrist was used to provide ground truth. The ball was tracked and the number of pixels it moves was compared to the system's result.

This approach did have shortcomings. As it was the bottom of the CPR performers sleeve that was located, the wrists of those with shorter sleeves will not be found. There is also an issue with how the CCD in pixels is converted to a real world measurement (mm). The users must measure their arm so that it can be used to estimate the size in mm of a pixel in the frame. However, if users do not measure their arm correctly the CCD measurement will be incorrect. Error in the conversion from pixel to mm also occurs if the camera is not parallel to the CPR performer as the whole arm of the CPR performer will not be in the camera's field of view. The study used one video for

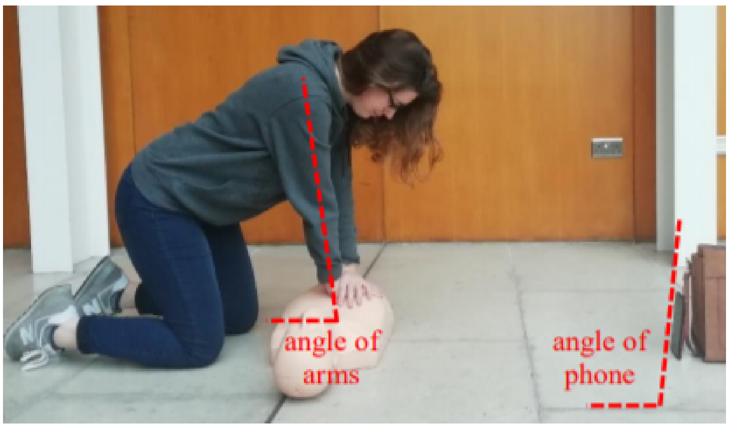

Figure 2.4: Conversation from pixel to mm, camera position issue[2]

testing. Further videos should be tested to confidently evaluate the system proposed. It was found that the system had an error of $\pm$ 3.86 mm.

Øyvind *et al.*[18] proposed a system to measure CCD which used a smartphone camera placed on the ground beside the patient's arm. The camera faced towards the sky capturing the CPR performer's shoulder and head in the frame (see figure 2.5). Accumulative Difference Images (ADI) were used to automatically detect the CPR performer. A template of a CPR performer's shoulders and head was used to locate the CPR performer in the ADI. The CCD was calculated by compensating for the CPR performers position in the frame and the camera angle. A motion band (the number of rows at the region of interest (see figure 2.6)) is used to estimate the CCD. The system calculated the CCD every 15 frames. This is every half second as the videos tested had a frame rate of 30 fps. This system will only calculate the CCD at the correct time if the CPR performer is abiding by the ideal CCR (100 to 120 compressions per minute). All the tests performed in the study used the same CPR performer with the distance from the shoulders to the ground measured in advance. The camera placement was vital in obtaining accurate results. If the camera was not placed close to the patients arm the shoulder region was not detected accurately. The study tested the performance of automatically and manually detecting the CPR performer. It was found that the smallest error the system achieved was $\pm$ 2.5 mm (automatic detection) and $\pm$ 2.8 mm (manual detection). A Resusci Anne mannequin, which has a device in the mannequin's chest to record CCR and CCD, was used to provide ground truth.

Figure 2.5: Position of camera used by Øyvind *et al.*[18]

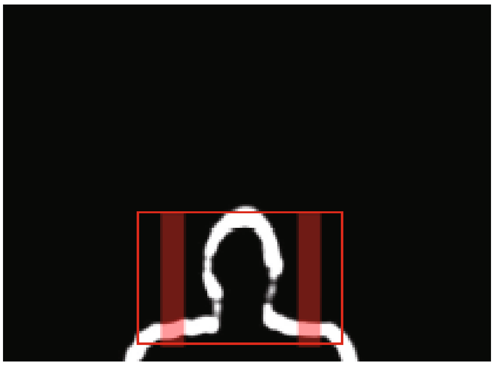

Figure 2.6: ADI of region of interest[18]

Lins *et al.*[19] used motion capture data from a Microsoft Kinect v2 to derive both the CCR and CCD. The Microsoft Kinect v2 can extract a skeleton from the video data captured of the CPR performer on the device. Software developed for the Kinect, RESKIN, is used to establish the position of the skeleton joints i.e. the hands, wrists, elbows and shoulders. The Kinect software supplies spatial coordinates of the joints making it possible to calculate CCD. Sinusoids, using the Differential Optimisation algorithm, are fitted to the data to determine the CCR and CCD. It was found that while the CCR could be measured quite accurately the CCD had a higher error. The error of the CCD for the system was ± 11.8 mm. A Resusci Anne mannequin was used to provide ground truth.

2.3.1 Summary

To measure CCD previous work indicates that using a depth camera and choosing a ROI other than the wrist could achieve an accurate result.

Table 2.1: Summary of previous work using a camera

System	Camera Position	videos tested	No. participants	Depth camera /sensor	ROI	CCD Error (mm)
Øyvind *et al.* [18]	beside patient facing up	9 (different camera positions)	1	No	shoulder	± 2.5 (automatic) ± 2.8 (manual)
Lins *et al.*[19]	facing CPR performer straight on	28	28	Yes	shoulder elbow wrist hand (skeleton form)	± 11.8
Wittam[2]	facing CPR performer straight on	1	1	No	wrist	± 3.86

3 Implementation

3.1 Software

The proposed algorithm was developed using C++. The OpenCV 4.1.2 [20] and Intel RealSeanse SDK 2.0 [21] libraries were used to produce the algorithm. The matching algorithm for the chamfer matching was provided by Dr. Kenneth Dawson-Howe[22].

3.2 Hardware

The Intel RealSense Depth Camera D435i[23] was used to record the CPR performers. The camera uses active IR stereo to estimate the depth for each pixel. Stereo is influenced by human vision. Two images taken from slightly different vantage points are searched for features present in both (see figure 3.1). The disparity between the features is used to calculate the depth.

To locate features texture is required. To increase the likelihood of features being located the camera uses an infrared light projector to create texture. The resulting depth image is shown in figure 3.2. The camera has a minimum depth distance of 0.105m.

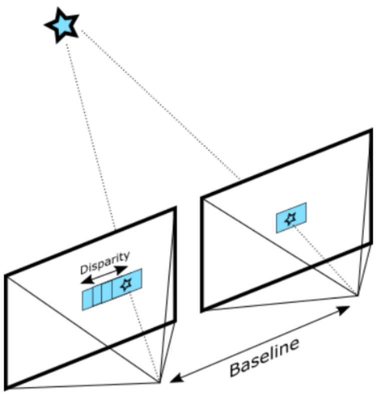

Figure 3.1: How stereo calculates depth

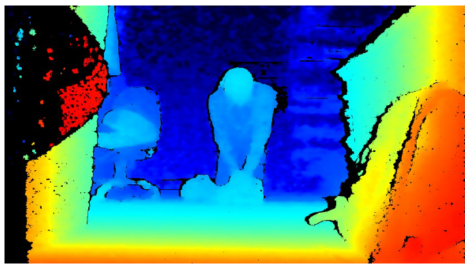

Figure 3.2: depth image calculated by the Intel RealSense Depth Camera D435i

3.3 Computer Vision Methods

3.3.1 Dense Optical Flow

Optical flow computes a motion field for the image. This motion field contains a motion vector, with magnitude and direction, for every pixel in the image[24]. Optical flow uses the brightness in the scene to compute the vectors. It is assumed in a video that the change in brightness of pixels from one frame to the next does not vary. For this reason optical flow is sensitive to sudden changes in brightness. Texture in the image is also required.

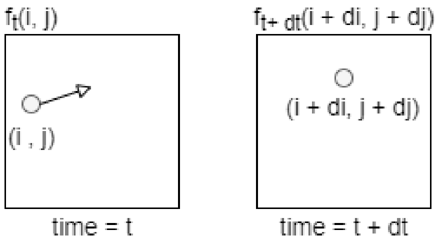

Figure 3.3: Movement of object in two frames

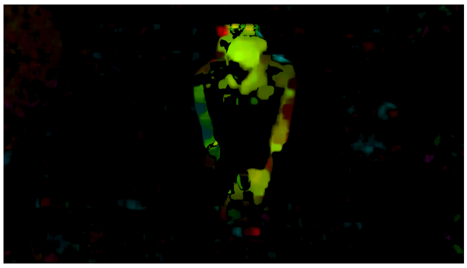

Figure 3.4: Motion captured in figure 3.6 by optical flow (green is up and red is down)

OpenCV uses the Farnebäck method[17]. This method employs the use of quadratic polynomials to estimate each of the frame's pixels neighbourhoods. The displacements can be calculated by the changes in the polynomials under translation[24].

3.3.2 Canny Edge Detection

Canny edge detection is a technique used to identify the edges in an image. The steps to the algorithm are outlined below.

1. The grey-scale image is convolved with a Gaussian filter to smooth and remove noise.

2. The orientation and magnitude of the edge is estimated by finding the partial first derivative of the smoothed grey-scale image. OpenCV uses the Sobel partial first derivatives[25] (see figure 3.5). The edge magnitude/gradient is calculated by finding the root mean square of the vertical and horizontal partial derivatives (see eq.1). The edge orientation is found by calculating the arctangent of the vertical and horizontal partial derivatives (see eq.2).

$$\text{Magnitude} = \sqrt{h_1^2 + h_3^2} \tag{1}$$

$$h_1(i,j) = \begin{bmatrix} 1 & 2 & 1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{bmatrix} \qquad h_3(i,j) = \begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix}$$

Figure 3.5: Sobel partial first derivatives. Horizontal partial first derivative (left) and Vertical partial first derivative (right)[26]

$$\text{Orientation} = \arctan\left(\frac{h_1}{h_3}\right) \tag{2}$$

3. To ascertain central edges non-maxima suppression using zero-crossing is used. This involves placing edges where points of maxima are. A local maximum can be found when the gradient function is at a peak i.e. the derivative of the gradient function is zero. To do this, two points orthogonal to an edge are examined. If the gradient at a point (i, j) is less than the gradient of either of the two orthogonal points, this point is set to zero. Otherwise the gradient at this point remains the same.

4. Thresholding using hysteresis is applied to the image. The thresholding method hysteresis is used to minimise streaking. It involves using two threshold values, a high gradient threshold and a low one. Points above the high threshold are edge points while points below the low threshold are not edge points. Points between the high and low threshold that are connected are considered edge points.

5. This can be done with multiple scales and used in conjunction with feature synthesis to combine the edges.

Figure 3.6: grey-scale image convolved with a Gaussian filter

Figure 3.7: edges detected by canny edge detector

3.3.3 Chamfer Matching

Chamfer matching is a technique used to locate objects in an edge image. A template of the object is used. It is common for object to appear slightly different to the template in images due to noise or the object orientation. Chamfer matching is not as sensitive to these as other matching methods such as template matching. The steps of chamfer matching are outlined below.

1. The edges of the image are found using an edge detector method such as Canny edge detector.

2. A chamfer image is created using the edge image. A chamfer image is an image composed of chamfer values, the distance from a pixel to the nearest edge[27].

Figure 3.8: Chamfer image of edge image shown in figure 3.7

3. The object template, a binary image (an image consisting of only white and black pixels), is used to compute a matching space. The template is compared to every possible location in the chamfer image. The position of the template non-zero pixels, in relation to where the template is being compared in the chamfer image, are summed. This creates a matching space containing the summed chamfer values corresponding to the position of the template. A value of zero in the matching space signifies the object is present in the image.

3.8	2.8	2.4	2	1	0	1	2
3.4	2.4	1.4	1	1	0	1	1.4
3	2	1	0	0	0	0	1
3	2	1	0	0	1	0	1
3	2	1	0	1	1	0	1
3	2	1	0	1	1	0	1
3	2	1	0	0	0	0	1
3.4	2.4	1.4	1	1	1	1	1.4

Chamfer Image

1	1	1	1
1			1
1			1
1			1
1	1	1	1

Template

27.4	19.6	12.8	8	27.4
23.2	16.8	10.4	5	9.4
21	14	8	0	6
23.2	16.8	10.4	5	9.4

Matching Space

Figure 3.9: Example of chamfer matching[27]. White pixels in the template have a value of 1

4. Local minima are located in the matching space. This is the position of the object in the original image.

3.4 Proposed Algorithm

The flow of the algorithm proposed is shown in figure 3.13. The depth and RGB frames must be aligned. The depth and RGB frames are different resolutions as the data is captured from different sensors which are located at separate locations. This means if the frames are not aligned a pixel in the depth frame will not be present in the RGB frame at the same location.

To locate the CPR performer dense optical flow is used. CPR is comprised of up/down motion during compressions. As a result there is a large area in the frame that has either an up or a down motion. Corkery[16] used dense optical flow to locate, track and record the CCR. The proposed algorithm also uses dense optical flow to locate and track the CPR performer. Dense optical flow can be slow, given it is calculating the motion vector for every pixel. To combat this and allow the algorithm to run in real time the resolution of the frame is reduced. The original frame resolution is 720p (1280 x 720). This is reduced to 256 x 144. This is an 80% reduction in resolution.

An edge image of the RGB frame is created as outlined in section 3.3.2. This edge image is used to locate the CPR performer's shoulders. Only the CPR performer is present in the edge image as the dense optical flow is used to segment the performer from objects in the rest of the frame. If the CPR performer is wearing clothing that has creases the edge image contains these. This can cause an issue with the chamfer matching part of the algorithm as the creases in the clothing change with movement. To negate this and improve the number of cases the algorithm works for, the edges within the CPR performers outline are removed. The outline is improved by closing any gaps.

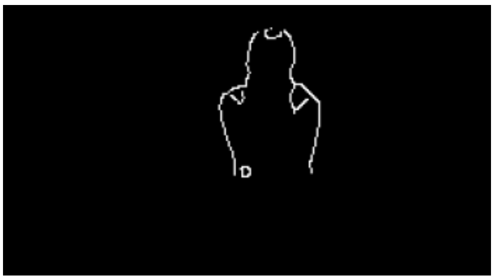

Figure 3.10: Edge image of segmented CPR performer

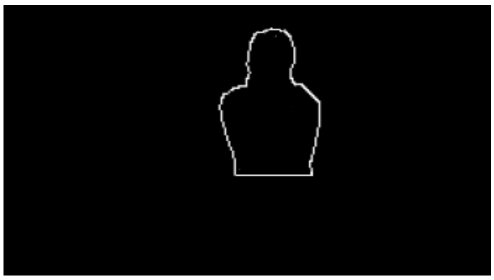

Figure 3.11: Creases of clothing removed from CPR performer edge image

The shoulders are found by splitting the outline of the CPR performer in half, the left side and the right side. The slope between each outer edge point is calculated (see eq. 3) until the widest edge point of the outline of the CPR performer. The maximum slope is the beginning of the shoulder. To process the image as seen in fig. 3.12 the rise of the slope uses the rows (y) and the run uses the columns (x) of the image.

$$\text{slope} = \frac{x_2 - x_1}{y_2 - y_1} \quad (3)$$

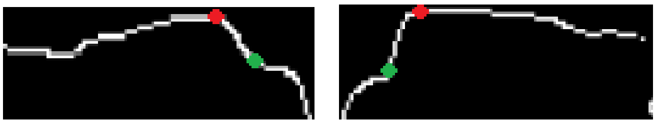

Figure 3.12: finding shoulders. Widest point (red), maximum slope (green), left side (left) and right side (right) of outline rotated 90°

A template of both the left and right shoulder is used during chamfer matching (see section 3.3.3) to locate the appropriate shoulder in frames. The shoulder is located again after each rescue breath.

The location of the shoulders found is stored. The maximum and minimum locations are used to calculate the pixel movement of the CPR performer. This value is converted into millimetres using the depth frame. The height of a depth frame pixel (mm) at the CPR performer's shoulder is multiplied by the number of pixels moved (see eq. 4).

$$\text{movement(mm)} = movement(pixels) \times depthpixelhight(mm) \quad (4)$$

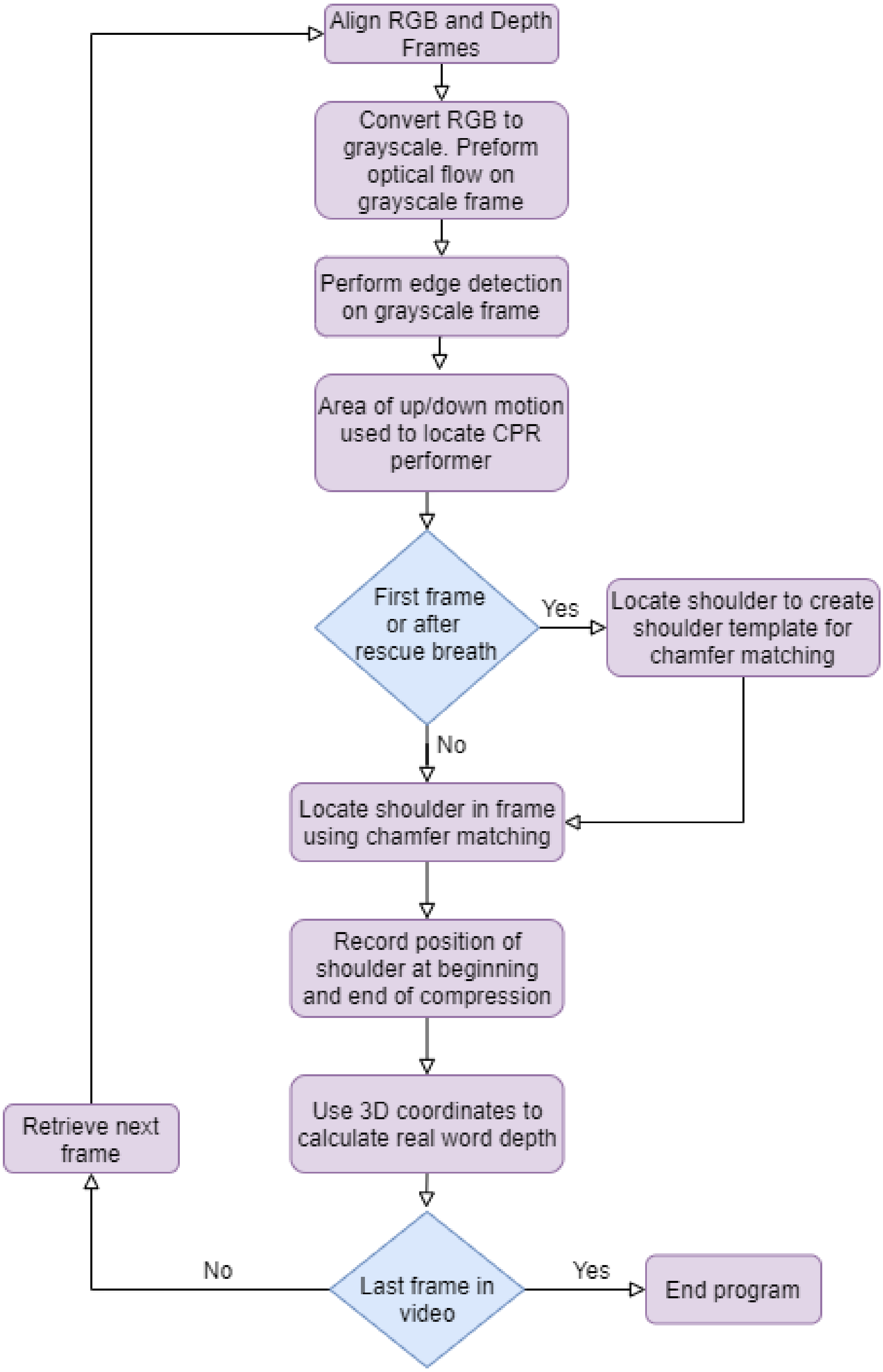

Figure 3.13: Flow of proposed algorithm

4 Evaluation

4.1 Ground Truth

The performance of the algorithm is tested using the ground truth. The method of obtaining ground truth was the same as Whittam[2]. A ping pong ball is attached to the CPR performer's arm. This ball is tracked for each compression to calculate the number of pixels moved. The Hough circle transform (see section 4.1.1) locates circles in images and is used to locate the ping pong ball. Due to motion blur the ping pong ball can often only be located at the top and the bottom of a compression. However, this is not a problem as these are the parts of the compression needed to calculate CCD.

4.1.1 Hough Circle Transform

The Hough circle transform[26] is a computer vision technique used to locate circles in an image. The steps involved in the Hough circle transform are outlined below:

1. The edges of the image are found using edge detection such as Canny edge detector (see section 3.3.2).

2. The equation of a circle can be described by eq. 1. Circles with a radius between a set range are searched for.

$$(x - a) + (y - b)^2 = r^2 \tag{1}$$

where (a, b) is the centre of the circle and r is the radius

3. The image is converted from image space (i, j) to Hough space (a, b). Hough space represents the likelihood a circle is present.

4. A 2D accumulator is initialised to zero. At every edge point the accumulator is

incremented according to the likelihood the point is the centre of a circle.

5. Local maxima are located to find the position of the circles in the image.

4.1.2 Reliability

The ground truth can track the vertical movement of the arm. Therefore, provided the arm remains straight, the ground truth can track the vertical shoulder movement. However, there is a concern that this method of ground truth may not be reliably calculating the CCD. This method assumes the vertical movement of the arm is equal to the depth the chest is pressed down.

The most accurate way to measure the CCD would be to test the algorithm using a training device such as the Resusci Anne mannequin. A depth measurement device inside the mannequin's chest is the most reliable way to record the CCD.

4.2 Reliability of the Intel RealSense Depth Camera D435i

A number of tests were performed to assess how reliable the depth data collected by the depth camera was. The orientation of the data recorded by the camera is as follows; the z-axis the the distance from the camera to an object, the y-axis is the vertical distance, and the x-axis is the horizontal distance. This project is concerned with the z-axis and the y-axis.

4.2.1 Testing z-axis

The accuracy of the distance from the camera to an object was tested. A box was placed a known distance from the camera. The distance reported by the Intel RealSense camera was recorded. This was repeated, moving the box further from the camera, six more times. The set up of the experiment can be seen in fig. 4.1.

Fig. 4.2 shows that the error between the ground truth and the depth camera increases as the object is moved further from the camera. The highest error was reported for position 7 (1.7m) at 80mm. This is a concern as to fully view the CPR performer in the test videos the camera was placed between 1.3m and 1.6m from the CPR performer. This means there could be a substantial difference between what the camera has reported and the actual distance. Considering the range in quality CCD is between 38mm and 56mm (see section 2.2.3) the error found for distances greater than 0.565m is not acceptable for this project.

Section 4.2.2 it was found that a depth camera calculated within 2mm of the measured vertical movement. The camera was placed 0.88mm from the ping pong ball. This suggests that the z-axis error found by the depth camera may not have as much as an effect on the accuracy of the y-axis.

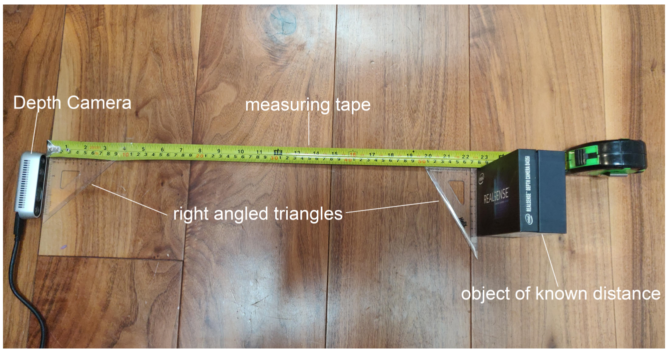

Figure 4.1: set up of testing the z-axis of depth camera

Table 4.1: Results of testing z-axis

Position	Ground Truth (m)	Depth Camera (m)	Difference (mm)
1	0.565	0.56	5
2	0.755	0.74	15
3	0.9	0.88	19
4	1.135	1.11	25
5	1.325	1.28	45
6	1.52	1.45	70
7	1.7	1.62	80
8	2.08	2.01	70

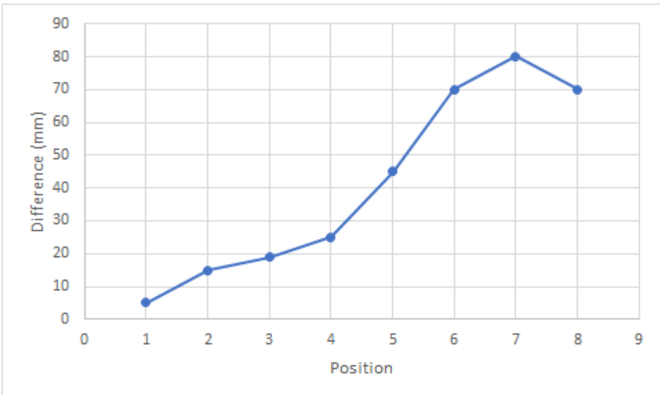

Figure 4.2: graph showing the difference between the ground truth and the depth camera for the 7 positions of the object

4.2.2 Testing y-axis

The y-axis (vertical) distance was tested by moving a ping pong ball 100mm up, and then down 100mm. The set up of the test is shown in the video cameraTest.avi accompanying this report and also fig. 4.3. The ball was moved 100mm 8 times.

It was found the ball moved 108 pixels each of these times. The depth image was used to find the height of a pixel where the ball was placed. This was found to be 0.93mm. Therefore, the ball moved 98mm (calculated using eq. 4, section 3.4). This is an accurate measurement using the depth camera as a ruler has an absolute error of ±1mm. The ping pong ball was approximately 0.84m from the depth camera.

Figure 4.3: set up of testing the y-axis of depth camera

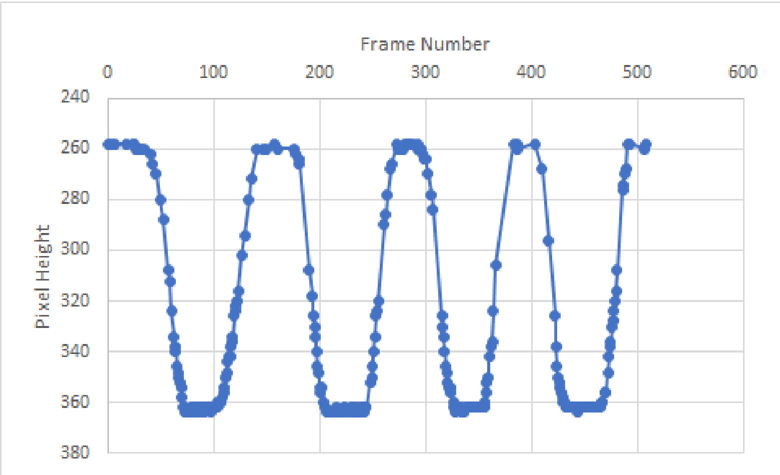

Figure 4.4: graph showing the position of the ping pong ball during y-axis test

4.3 Algorithm Performance

The algorithm was tested with 8 videos of a person performing CPR on a computer with a Intel Core i7-7500U. There were two CPR performers. One performer was filmed for 6 videos (test3.avi to test8.avi) and the other for 2 videos (test1.avi and test2.avi). The algorithm calculates the CCD by tracking the shoulder of the CPR performer. Therefore, a CPR performer wearing an item of clothing that misshapen the shoulders, or if something obscures the shoulders, such as hair, the algorithm may not calculate the CCD correctly. In video 7 (test7.avi) the CPR performer has her hair down and over her shoulders. In video 8 (test8.avi) the CPR performer is wearing a coat which changes the shape of the shoulders. In Videos 1 to 6 (test1.avi to test6.avi) the CPR performers are wearing long sleeved jumpers or tops.

The performance of the algorithm was assessed by finding the mean absolute error (MAE, see eq. 2) and the percentage error (PE, see eq. 3) between the ground truth and the results of the algorithm for each video. The results are shown in table 4.2.

$$\mathrm{MAE} = \frac{1}{n}\sum_{t=1}^{n} \mid \mathit{difference}_t \mid \tag{2}$$

$$\mathrm{PE} = \frac{\mathit{algorithm}}{\mathit{groundtruth}} \times 100 \tag{3}$$

Table 4.2: Algorithm performance for videos 1 to 8 (CCD)

Video	MAE (pixels)	MAE (mm)	PE (pixels)	PE (mm)
1	7.2	14.48	35.2	55.7
2	12.4	16.7	203.3	93.3
3	19.4	16.7	109.2	78.0
4	9.9	17.7	54.6	75.3
5	6.6	19.7	36.3	53.0
6	6.8	10.8	31.6	38.3
7	8.5	25.7	65.3	70.9
8	35.3	11.3	178.1	36.6

Table 4.3: Algorithm performance for videos 1 to 8 (percentage of compressions detected)

Video	Percentage of compressions detected
1	96.6
2	93.5
3	65.2
4	73.5
5	91..5
6	87.1
7	96.2
8	51.1

The proposed algorithm found 87% or more of the compressions for 70% of the test videos (see table 4.3). Test video 8 has the lowest percentage of compressions detected. This is the video where the CPR performer is wearing a coat (see test video 8 accompanying this report and fig. 4.7). This distorts the shape of the shoulder and as a result makes it harder to detect. Test video 3 and 4 are of the same CPR performer and were taken on the same day. The fleece the performer is wearing has creases, detected during edge detection, that may not be removed adequately, or the outline closed correctly, to find a match.

The CCD results obtained from table 4.2 show that the algorithm does not achieve accurate results. There are a number of reasons this may have occurred.

1. The Intel RealSense Camera constantly dropped frames while recording videos. It appears that this is an issue other users of the the camera have experienced. Intel RealSense use the issue feature on GitHub to provide support to users. In response to the issue of dropping frames during recording[28][29] it was recommended to disable the auto exposure feature and set it manually. This was tested but the reduction in frames dropping was not significant. Users found that frames were not dropped if the video was recorded with a Ubuntu system. The test videos were recorded using Windows 10. The dropping frames could lead to error in both the ground truth and the algorithm. It is important the CCD is accurate to within millimetres. Therefore, the loss of frames could result in vital vertical movement being lost.

2. The scaling of the original image by 80% may introduce error. The image was scaled in an attempt increase the time taken to locate the CPR performer, locate the shoulders, and track the vertical movement to find the top and bottom of the compression. Increasing the resolution will increase the time taken to process each frame. It should be noted that increasing the resolution results in an edge image that has a broken outline of the CPR performer. This means the shoulders cannot be detected as often.

Fig. 4.5 of video 6 shows pixel position of the left and right shoulders and the ground truth for each compression. This test video had a high compression detection percentage of 96.2%. It also has the lowest MAE (10.8mm) of the 8 videos. It can be seen from viewing the video that the CPR performer is wearing a long sleeved top and there is nothing obstructing the shoulders.

In video 2 rescue breaths are performed. Fig. 4.6 shows the algorithms capability to find the shoulder after a rescue breath has been performed. During a rescue breath the shoulders are not tracked. This is shown by the empty space between the tracked compressions. In the event a shoulder is not detected the pixel height is recorded on the graph as 0.

Video 7 performed the worst of the 8 test videos with a MAE of 25.7mm. The CPR performer in this video has most her shoulders obstructed by hair. The CPR performers right shoulder (the left shoulder to the viewer) can be tracked as the hair is not completely obstructing it. This allowed for the algorithm to detect 96.2% of the compressions. However, the CCD could not be calculated accurately.

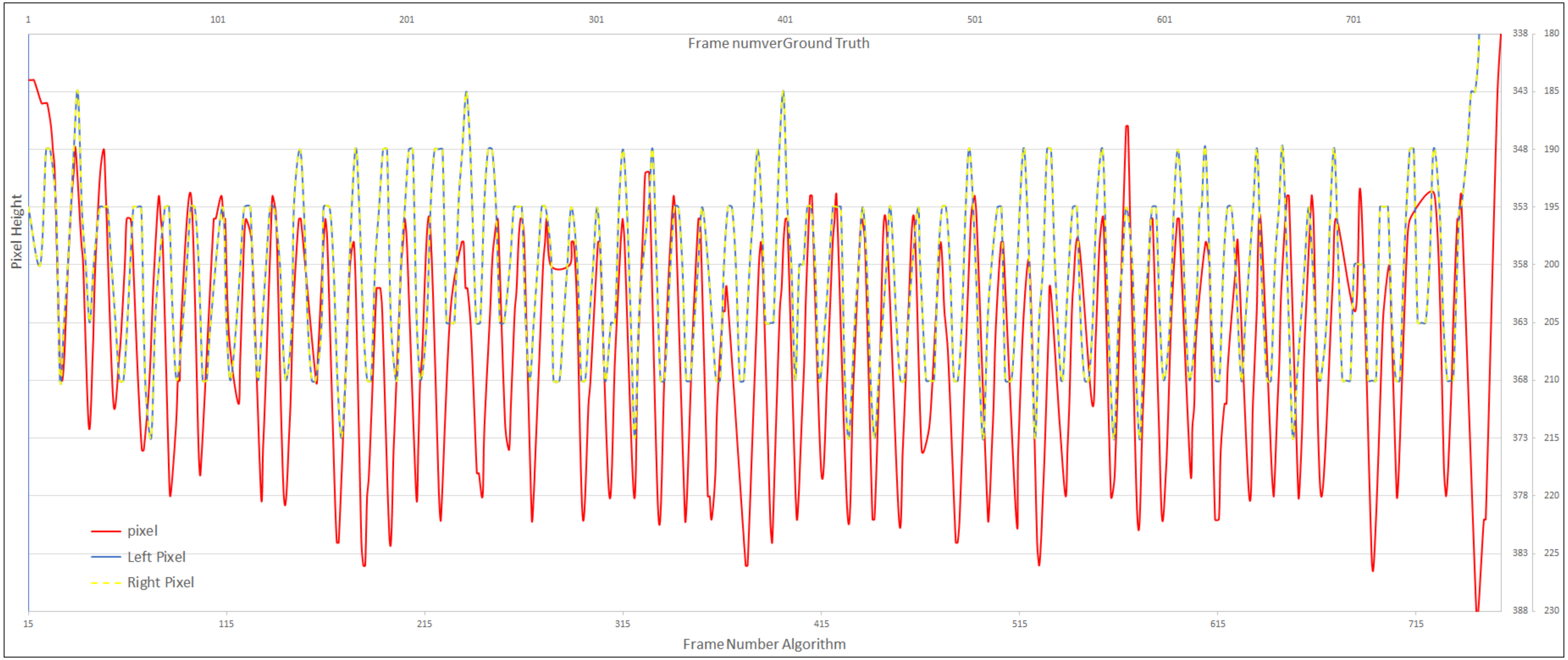

Figure 4.5: graph showing test video 6 tracked compressions of the algorithm(yellow and blue) and the ground truth (red)

Figure 4.6: graph showing test video 2 tracked compressions, of the algorithm(yellow and blue) and the ground truth (red)

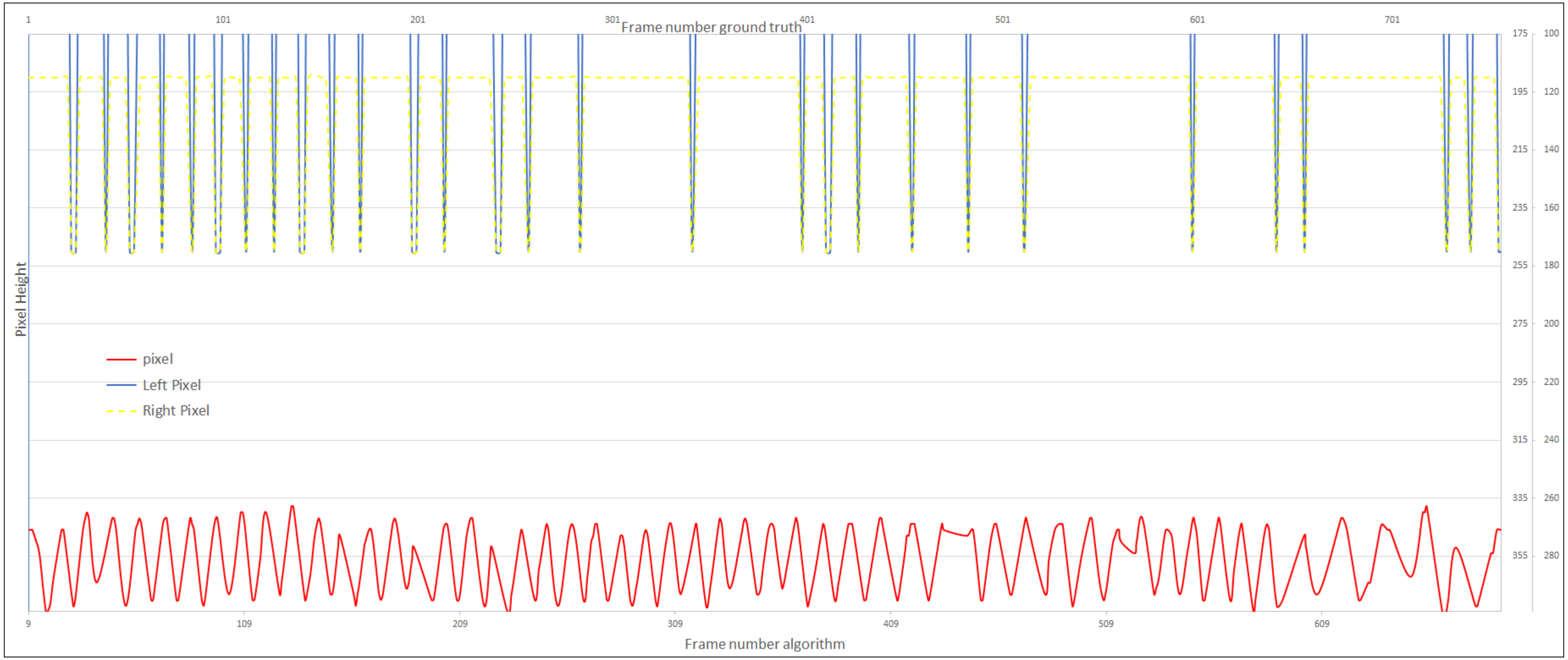

Figure 4.7: graph showing test video 8 tracked compressions, of the algorithm(yellow and blue) and the ground truth (red)

5 Conclusion

5.1 Quality of Proposed Algorithm

The project aimed to calculate the CCD of a CPR performer using a depth camera and by doing so prove a proof of concept. The range of CCD that will increase survival is between 38mm and 56mm, a range of 18mm. This mean that the CCD must be calculated having as small of a millimetre error as possible. Therefore, the proposed algorithm is not suitable for measuring CCD. A MAE of ± 16.6 is not acceptable. Comparing this to the previous work in the field the algorithm performed the worst. This is likely due to the reasons described in section 4.3.

The Intel RealSense camera was tested and it was found the camera had less error the closer to an object it was. This suggests the accuracy of the CCD may be improved by moving the camera closer to the CPR performer, if possible. It would also be beneficial to film the videos on a Ubuntu system as other users of the camera found the dropping frames issue did not occur with Ubuntu.

While the CCD was not calculated successfully the percentage of compressions detected was above 90% for 60% of the test videos. Graphing the tracked shoulders showed a sinusoidal trend, consistent with CPR, implying chamfer matching was an appropriate computer vision technique to locate the shoulders.

The method of detecting the CPR performer's outline could benefit from being improved. Currently the frame resolution is reduced by 80%. In doing so the edge image is more compacted and the outline is less broken. The image must be scaled to allow dense optical flow to execute in real time.

Table 5.1 compares the previous work to the proposed algorithm. While it is clear that the algorithm does not perform sufficiently. The test outlined in section 4.2.2 showed the depth camera could accurately measure vertical movement.

In conclusion, the proposed algorithm was not successful but tests of the Intel RealSense camera suggest that a depth camera could accurately measure the CCD. Provided the method of tracking the CPR performer is reliable.

Table 5.1: Comparing previous work with proposed algorithm

System	Camera Position	videos tested	No. participants	Depth camera /sensor	ROI	CCD Error (mm)
Øyvind *et al.* [18]	beside patient facing up	9 (different camera positions)	1	No	shoulder	± 2.5 (automatic) ± 2.8 (manual)
Lins *et al.*[19]	facing CPR performer straight on	28	28	Yes	shoulder elbow wrist hand (skeleton form)	± 11.8
Wittam[2]	facing CPR performer straight on	1	1	No	wrist	± 3.86
Proposed Algorithm	facing CPR performer straight on	8	2	Yes	shoulder	±16.6

5.2 Further Work

Further work could be carried out regarding the topic of CPR and analysing its quality parameters.

5.2.1 CPR Positioning

The position of CPR performer is not considered in this project. It is best during CPR for the performer to keep his or her arms straight. Designing a system that analysed the CPR performer's position could be very helpful for training and emergency situations.

5.2.2 Integrated Application

An application combining the analysis of the CCR and CCD could be developed to help CPR performers during training or an emergency. Audio outputs could guide the CPR performer to improve the quality of the CPR.

5.2.3 Further testing

To reliably test the algorithm the ground truth could be improved. Currently the ground truth is generated by tracking a ping pong ball on the CPR performer's arm. This method tracks the vertical movement by not the actual CCD. The use of a device like the Resusci Anne mannequin.

Bibliography

[1] S. Duval, P. Pepe, T. Aufderheide, J. Goodloe, G. Debaty, J. Labarère, A. Sugiyama, and D. Yannopoulos, "Optimal combination of compression rate and depth during cardiopulmonary resuscitation for functionally favorable survival", *JAMA Cardiology*, pp. 900–908, Aug. 2019. DOI: 10.1001/jamacardio.2019.2717.

[2] J. Whittam, "Visual measurement of chest compression depth in cardiopulmonary resuscitation", Unpublished Bachelor Dissertation, School of Computer Science and Statistics, Trinity College Dublin, 2019.

[3] Ø. Meinich-Bache, K. Engan, T. Eftestøl, and I. Austvoll, "Kinect modelling of chest compressions - a feasibility study for chest compression depth measurement using digital strategies", Oct. 2018, pp. 913–917. DOI: 10.1109/ICIP.2018.8451387.

[4] Samsung. (2020). Galaxy s10 5g, [Online]. Available: https://www.samsung.com/global/galaxy/galaxy-s10/galaxy-s10-5g/. (accessed: 20/04/2020).

[5] Hauwei. (2020). Huawei p30 pro specifications, [Online]. Available: https://consumer.huawei.com/ie/phones/p30-pro/specs/. (accessed: 20/04/2020).

[6] Apple. (2020). Iphone xr, [Online]. Available: https://www.apple.com/iphone-xr/specs/. (accessed: 20/04/2020).

[7] American Heart Association. (2015). Part 14: Education, [Online]. Available: https://eccguidelines.heart.org/circulation/cpr-ecc-guidelines/part-14-education/?strue=1&id=8-1. (accessed: 11/04/2020).

[8] British Heart Foundation. (2018, April). Steps to cpr poster, [Online]. Available: https://www.bhf.org.uk/informationsupport/publications/emergency-life-support/steps-to-cpr-poster. (accessed: 11/04/2020).

[9] American Heart Association. (2015). Part 5: Adult basic life support (BLS) and cardiopulmonary resuscitation (CPR) quality, [Online]. Available: https://eccguidelines.heart.org/circulation/cpr-ecc-guidelines/part-5-adult-basic-life-support-and-cardiopulmonary-resuscitation-quality/. (accessed: 11/04/2020).

[10] ——, (2020). High-quality cpr, [Online]. Available: https://cpr.heart.org/en/resuscitation-science/high-quality-cpr. (accessed: 11/04/2020).

[11] Philips. (2020, Jan). Q-CPR measurement and feedback tool, [Online]. Available: https://www.usa.philips.com/healthcare/product/HCNOCTN89/qcpr-measurement-and-feedback-tool-cpr-meter. (accessed: 16/01/2020).

[12] Zoll. (2019, Dec). AED plus, [Online]. Available: https://www.zoll.com/products/aeds/aeds-for-public-access/aed-plus-for-public-access. (accessed: 16/01/2020).

[13] A. Grünerbl, G. Pirkl, E. Monger, M. Gobbi, and P. Lukowicz, "Smart-watch life saver: Smart-watch interactive-feedback system for improving bystander cpr", Sep. 2015.

[14] T.-C. Lu, Y. Chen, T.-W. Ho, Y.-T. Chang, Y.-T. Lee, Y.-S. Wang, Y.-P. Chen, C.-M. Fu, W.-C. Chiang, M. Ma, C.-C. Fang, F. Lai, and A. Turner, "A novel depth estimation algorithm of chest compression for feedback of high-quality cardiopulmonary resuscitation based on a smartwatch", *Journal of Biomedical Informatics*, vol. 87, pp. 60–65, Sep. 2018. DOI: 10.1016/j.jbi.2018.09.014.

[15] T.-C. Lu, Y.-T. Chang, T.-W. Ho, Y. Chen, Y.-T. Lee, Y.-S. Wang, Y.-P. Chen, C.-L. Tsai, M. Ma, C.-C. Fang, F. Lai, H. Meischke, and A. Turner, "Using a smartwatch with real-time feedback improves the delivery of high-quality cardiopulmonary resuscitation by healthcare professionals", *Resuscitation*, vol. 140, May 2019. DOI: 10.1016/j.resuscitation.2019.04.050.

[16] G. Corkery and K. Dawson-Howe, "A smartphone tool for evaluating cardiopulmonary resuscitation (cpr) delivery", Jan. 2019, pp. 489–496. DOI: 10.5220/0007258904890496.

[17] G. Farnebäck, "Two-frame motion estimation based on polynomial expansion", vol. 2749, Jun. 2003, pp. 363–370. DOI: 10.1007/3-540-45103-X_50.

[18] Ø. Meinich-Bache, K. Engan, T. Eftestøl, and I. Austvoll, "Detecting chest compression depth using a smartphone camera and motion segmentation", May 2017, pp. 53–64, ISBN: 978-3-319-59128-5. DOI: 10.1007/978-3-319-59129-2_5.

[19] C. Lins, D. Eckhoff, A. Klausen, S. Hellmers, A. Hein, and S. Fudickar, "Cardiopulmonary resuscitation quality parameters from motion capture data using differential evolution fitting of sinusoids", *Applied Soft Computing Journal*, vol. 79, pp. 300–309, Mar. 2019.

[20] OpenCV. (2020). Releases, [Online]. Available: https://opencv.org/releases/. (accessed: 20/04/2020).

[21] Intel RealSense Technology. (2020). Intel realsense sdk 2.0, [Online]. Available: https://www.intelrealsense.com/sdk-2/. (accessed: 20/04/2020).

[22] Kenneth Dawson-Howe. (2020). Code samples - recognition, [Online]. Available: https://www.intelrealsense.com/sdk-2/. (accessed: 20/04/2020).

[23] Intel RealSense Technology. (2020). Intel® realsense™ depth camera d435i, [Online]. Available: https://www.intelrealsense.com/depth-camera-d435i/. (accessed: 20/04/2020).

[24] K. Dawson-Howe, *A Practical Introduction to Computer Vision with OpenCV*, 2nd ed. Hoboken, NJ: Wiley, 2014, pp. 182–185.

[25] OpenCV. (2019, Nov). Canny edge detector, [Online]. Available: https://docs.opencv.org/2.4/doc/tutorials/imgproc/imgtrans/canny_detector/canny_detector.html. (accessed: 20/04/2020).

[26] K. Dawson-Howe, *A Practical Introduction to Computer Vision with OpenCV*, 2nd ed. Hoboken, NJ: Wiley, 2014, ch. 6.

[27] ——, *A Practical Introduction to Computer Vision with OpenCV*, 2nd ed. Hoboken, NJ: Wiley, 2014, pp. 137–140.

[28] Intel RealSense. (Oct, 2018). Viewer dropping frames during recording, [Online]. Available: https://github.com/IntelRealSense/librealsense/issues/2535. (accessed: 20/04/2020).

[29] ——, (Sep, 2018). Rgb at half framerate, [Online]. Available: https://github.com/IntelRealSense/librealsense/issues/2436. (accessed: 20/04/2020).

A1 Appendix

A1.1 Appendix numbering

- Test videos 1 to 8
- C++ code for proposed algorithm
- C++ code for ground truth