Universiteit Leiden

Master Computer Science

Alignment of Facial Images from Infrared and RGB video data

Name:	Marcel Kalmes
Student ID:	s3018520
Date:	25/08/2022
Specialisation:	Advanced Computing and Systems
1st supervisor:	Prof.dr.ir. F.J. Verbeek
2nd supervisor:	Dr.ir. M.P. Janssen

Master's Thesis in Computer Science

Leiden Institute of Advanced Computer Science (LIACS)
Leiden University
Niels Bohrweg 1
2333 CA Leiden
The Netherlands

Contents

1 Introduction

When people donate blood some of these donors may faint or feel unwell. Why this happens exactly is still mostly unknown. The FAINT project, (FAcial INfrared Thermal imaging in the prevention of needle induced fainting" project) aims to predict fainting in blood donors during their donation process. For this project blood donors were recorded during different stages of their donation process using a regular RGB and a thermal infrared camera. For better feature extraction the possibility of aligning both videos will be explored. Lining up both video files allows us to extract a combination of RGB and thermal data for specific areas of the face. The goal is to use this information to train advanced machine learning models to better understand and predict fainting related to the blood donation process. The videos of the donors are recorded before, during and after the blood donation process to get a better understanding of the donors well-being during those stages.
The two video files recorded during a session depict the same donor during the same blood donation but the video files differ in camera angle, zoom, start time, length and centering, as well as frame rate and resolution, therefore, aligning both videos is not straightforward. To align the video files we will look at two methods of detecting similarities between both video files. Once these similarities are found we can use them to align both videos in a way that those similarities align and therefore, both video files are aligned as well.

1.1 Research questions

Three main research questions are answered in this thesis: (1) Can two video files, from the same scene where one is a thermal video and the other one is a regular video, be aligned automatically and accurately? (2) If so, how can this goal be achieved? (3) Can the process of extracting thermal and RGB video recording from a blood donation be improved? These research questions are directly derived from the requirements of the FAINT project to determine the feasibility of accurately aligning the two video files.

1.2 Structure of the thesis

This thesis is structured into 5 chapters starting with the Introduction. The Introduction contains information about the goals and the background of the thesis. The second chapter Materials and Methods outlines the dataset used as well as different tools used in this thesis. After this chapter 3, Design and Implementation describes the design and the implementation of two programs to align a RGB and an infrared video file. The two programs are tested and the results of these experiments are shown and discussed in chapter 4, Results and Experiments. The last chapter, Conclusion and Discussion discusses the results achieved as well as show possible future research regarding video alignment.

1.3 Background

1.3.1 Face detection in RGB images

Face detection is a widely researched topic in computer science and a challenging pattern recognition problem [5]. It performs the basic task of identifying whether something in an image is a face. The way these algorithms and models perform this task varies largely between

the different methods. While earlier face detection methods relied on less computationally expensive methods, modern face detection methods often use large training sets and deep neural networks, which requires large computing power, but improves the accuracy when detecting faces [4]. Face detection has evolved so far that in 2017 Bulat et.al have said that the face alignment problem in 2D for faces in controlled poses has largely been considered solved [2]. Face alignment networks/models try to detect facial features and map them onto the persons face so that the orientation of the face can be captured and, if needed, corrected. To achieve this an active shape model can be used which is a set of dots resembling the shape of the face and containing outlines of different facial landmarks, which can be transformed to fit onto a detected face. This approach performs well even on incomplete or partially covered faces and gives accurate locations for facial landmarks. This face alignment process can be performed using a 2D depiction of the facial features or can be done using a 3D map of the facial features. To aid in the process of solving the 3D face alignment problem the FAN was created so that it could perform 2D predictions as well as 3D predictions very accurately. Using these predictions as well as different data sets with 2D and/or 3D annotations, the FAN is able to create new 3D annotations based on the results of the 2D predictions. The FAN also allows to use different face detectors by specifying them as a parameter. The options for these face detectors are *sfd* and *blazeface*.

1.3.2 Face detection in thermal images

Since deep neural networks are often used, these networks cannot easily be adapted to work with different data such as thermal video data. Most face detection algorithms for RGB images cannot be used to detect faces in thermal images. Since face detection for thermal images has been researched little and face detection for RGB images is a much more researched topic, the few methods that do exist for face detection in thermal images are much less developed.
For example, Cheong et al. (2014) [3] outlined a person's upper body to determine the head region in thermal images whereas Chao et al. (2017) [6] adapted local features for face detection in thermal images. Ribeiro, Fernandes and Neves (2017) [8] used another approach of chamfer matching and contour detection and so far only Kopaczka et al. (2019) [7] used an annotated database of thermal images to train a network similarly to face detection in RGB images. This is also one of the limited datasets of thermal images that are freely available.

1.3.3 Template matching

Template matching describes the process of scoring the similarities of template with areas of a base image based on different algorithms. To perform this, a template is compared to a generally larger base image to see whether there are similarities between the template and the base. These similarities are measured by calculating the overlapping areas and scoring the results. The highest scoring result then contains the best found position for the template in relation to the base image. Template matching algorithms can be as simple as testing every possible template location, but can also be more complicated, such as Borgefors's hierarchical chamfer matching [1].

2 Material and Methods

This chapter contains information about the Dataset and its acquisition as well as details about the programming languages and libraries used.

2.1 Data aquisition

The data was acquired by Sanquin, the non-for-profit organization responsible for blood and plasma donations in the Netherlands, by recording blood donors during different stages of their blood donation process. These videos were recorded for the "FAINT: FAcial INfrared Thermal imaging in the prevention of needle induced fainting" project. The donors were recorded by two different cameras mounted close to each other on two separate tripods. For different recordings the camera perspectives therefore vary. The donors are recorded at different points during their donation process, leading to different recording locations and angles for the same donor in different recordings. The donors were asked to take off their glasses as to not interfere with the infrared video recordings. The cameras were mounted about one meter from the donor and the donors were told to behave normally. At the beginning of each recording both cameras were activated manually with a short delay between the start of the first and the second camera. The cameras were not synchronized.
The cameras that were used to film the recordings are one standard RGB camera, the Nikon Coolpix AW130 which recorded the donor at a resolution of 1920 by 1080 pixels and a frame rate of 25 frames per second. The second camera that was used is an infrared camera for thermal imaging, the FLIR E95 which recorded at different sizes ranging from 748 by 562 pixels up to 1036 by 788 pixels and a frame rate of 30 frames per second. At the beginning of the recording both cameras were adjusted to be centered around the face. This was done manually by adjusting location, angle and zoom. Therefore, the recordings differ in terms of total length, initial starting time, zoom, centering and camera angle.

2.2 Dataset used

The dataset used for this testing consists of 55 sets of videos and two thermal videos that do not have a matching RGB video. All RGB videos are of size 1920 by 1080 pixels while 53 out of 55 thermal videos are recorded at a size of 748 by 562 while the rest are recorded at a size of 1036 by 788 pixels. To effectively work with the acquired video files some issues have to be addressed. As mentioned in section 2.1 there are six main issues identified: (1) The timing of both videos is different. Both cameras were started manually at separate time points causing a difference in the starting time and the total length of the video. (2) The frame rates at which both videos were recorded are different, 25 fps for the RGB video and 30 fps for the thermal video. (3) Both videos were recorded at a separate aspect ratio and resolution. (4) Although both cameras were placed the same distance from the donor, camera zoom is used sometimes in either of the videos after recording has started. (5) The donor's face was manually centered in each video causing differences in relative centering of the face. (6) Both cameras were placed on separate tripods besides each other with varying distance leading to a difference in the recording angle.
Some of these issues can be solved by pre-processing the videos. Those are the timing of the video and the frame rate, while others have to be solved differently for each approach. These are the resolution and aspect ratio, zoom level and relative centering. Zoom level and

resolution lead to one face being made up of more pixels than the other and therefore being larger. This has to be solved by finding reliable size information in the image. This can still be a problem if the video does not depict a full face and is cut off at some point. The problem of the different camera angles can not be solved without losing information.
We can see an example of this video data in Figure 1. We can clearly see the difference in image shape and alignment as well as angle. To improve the readability these images are scaled to fit on the page but the original sizes of these images are 1920 by 1080 pixels for the RGB image and 1036 by 788 pixels for the thermal video.

Figure 1: Unedited images from the dataset

2.3 Programming

The programming language can have a huge impact on the implementation. Not only do different programming languages differ in speed but they also differ largely regarding the availability of different software libraries. To work with video and image data more easily OpenCV was chosen as a library. "OpenCV is an open-source library that includes several hundreds of computer vision algorithms."[9]. OpenCV includes multiple tools that make working with video and image files a lot easier. OpenCV is implemented for Python and C++. It was decided to use Python 3.9.2 and OpenCV version 4.5.5.62.

2.4 Face detection API

Modern face detection algorithms use machine learning models to effectively detect faces. This is often done using deep learning which uses multi layered neural networks. These models have to be trained on a sufficient amount of data to accurately learn and in turn detect faces. There is so far not one "perfect" face detection model that will always accurately detect faces. Therefore multiple approaches have been developed each with advantages and disadvantages. The problem that arises for this project is that these models have been exclusively trained on large datasets of RGB/Grayscale images, which differ fundamentally from thermal images.
A suitable model for aligning the two videos needs to be able to detect accurate facial landmarks in the RGB and the thermal video. One implementation that achieves this is the FAN (Face Alignment Network) described in [2]. The FAN was found when testing different available face detection models which were trained on RGB images. Since the FAN was not trained on thermal images it's performance has to be evaluated. To do so the FAN was tested on the entire dataset and the FAN's confidence score was taken for 25 frames that are within the first minute of the recording. If the FAN was able to detect a face in less than 15 out of 25

Resolution	Average confidence	below 0.5 confidence	no face found
Thermal 1/1	0.658	0	8
Thermal 1/2	0.677	0	6
Thermal 1/4	0.713	4	1
Thermal 1/6	0.631	6	5
Thermal 1/8	0.614	3	7
RGB 1/4	0.826	0	0

Table 1: FAN tested at different resolutions on all 57 files in the dataset

frames the model is deemed unsuitable. The results of this testing can be seen in table 1. The column "below 0.5 confidence" shows the amount of video files where the average confidence after testing 25 frames was below 0.5 and the column "no face found" shows the amount of videos where no face was found in more than 10 out of 25 frames.
We can see that the FAN produces poor results when used with the full resolution thermal image with a confidence score of 0.658. More problematic than the low confidence score is the amount of videos for which no face could be reliably detected. We can see the same issue when using half the original resolution. When using a quarter of the resolution we can see that the model performs well with only 1 out of 57 files not working. Reducing the resolution further leads to worse performance. We can also see that the confidence score of the RGB video file at a quarter of its resolution is 0.826 which is about 0.11 better than when using thermal video data. After these experiments the FAN was deemed suitable for use with thermal video data. Since the image size of the thermal video files is 748 by 562 pixel, the best results were achieved at an image size of 187 by 140 pixel. Some of the thermal videos are recorded at a size of 1036 by 788 pixel they are scaled down to a size of 259 by 197 pixel.

2.5 Template matching API

Template matching is a different approach to face detection, relying on contours of objects in an image to match them to a template. Very capable approaches of this have been shown as early as 1988 by Borgefors in [1]. This method is promising for this project as this approach has been used previously for face detection in thermal images as shown in [6] and [3]. Modern template matching can be performed using different algorithms like chamfer matching. OpenCV supported chamfer matching in older versions, but this has since been replaced by a more basic template matching function called *matchTemplate()*. The OpenCV documentation describes the algorithm as sliding through the image and comparing the size of the overlapping patches to the template using a specified algorithm. There are multiple different algorithms available. The different algorithms available are: (1) *cv2.TM_SQDIFF*, (2)*cv2.TM_SQDIFF_NORMED*, (3) *cv2.TM_CCORR*, (4) *cv2.TM_CCORR_NORMED*, (5) *cv2.TM_CCOEFF* and (6) *cv2.TM_CCOEFF_NORMED*. The details for various algorithms can be found in the OpenCV documentation [9]. After the comparison is done, the best location for the template can be found as global minima using the *minMaxLoc()*.

3 Design and Implementation

To align the two video files as closely as possible we need to find similarities within both videos that can be identified and compared. Since both videos are inherently different but show a similar scene, finding these similarities accurately and reliably is the most difficult part. This chapter shows the design and the implementation of two programs to align the videos using different similarities.

3.1 General Approach

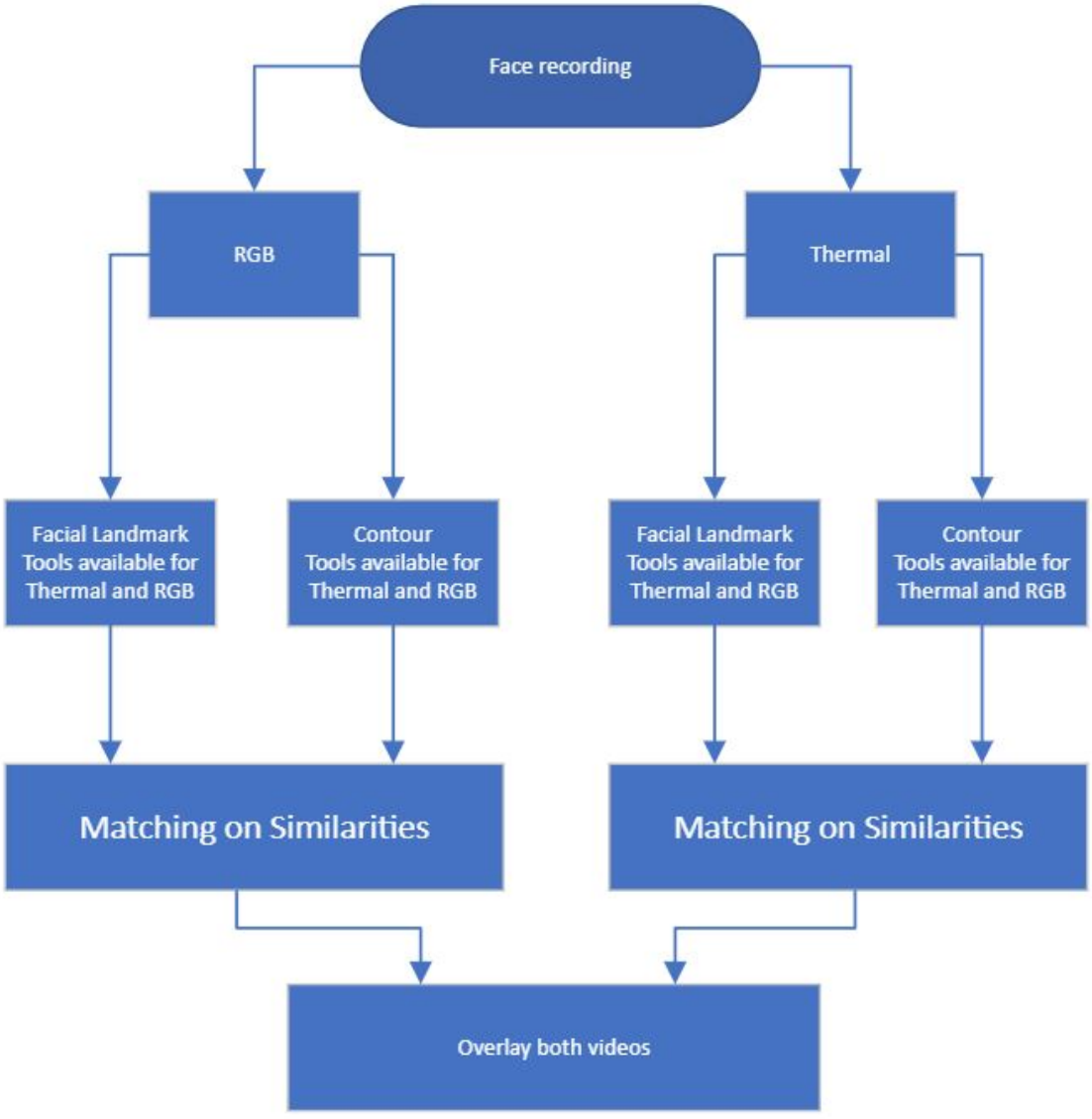

Figure 2: Program outline

In Figure 2 the broad outline of this project is shown. It shows two different approaches to finding similarities within the two videos by (1) using face detection models to detect facial landmarks in both videos and then aligning the videos using these landmarks and (2) using

a form of contour matching to use on the contours of the donor's face in both videos. To implement these approaches a program is to be designed that takes in two different video files, one thermal video and one RGB video and aligns these as accurately as possible. Judging by the availability of tools for finding these similarities, it was expected that this should be quite straightforward.
The first approach focuses on using face detection models to accurately identify the face in each video and then align them using the information gained. For RGB videos there exist many tools and models that can identify faces. Since the goal is to accurately match the faces, algorithms and models that only draw a simple bounding box or detect the presence of a face can not be used. Using such an approach would lead to inaccurate matching due to the simplicity of the bounding boxes. Therefore only models that output exact facial features such as facial landmarks may be used. These models are more sparse and therefore the selection is limited. But since these tools exist, finding landmarks in the RGB video is not an issue.
Since we want to find similarities in both videos, any model used should ideally also work accurately on thermal video data. Since face detection for thermal data is a much less researched topic only Kopaczka's model currently exists [7]. Kopaczkas model however is trained on a relatively small size of thermal images compared to the amount of images used to train a RGB model.
The second approach focuses on contour detection to align both videos. Contour detection was already used by Cheong et.al [8] and Ribeiro et.al [3] to detect faces in thermal video. This approach focuses on contour extraction using edge detection like i.e. Canny edge detection and a contour matching method like Chamfer matching [1]. Since both videos depict a person sitting in front of the camera the contour of this person should be similar in both videos, matching those two contours should be possible. Extracting the contours accurately is difficult since features like a persons hair are depicted completely differently in the two video types.

3.2 Pre-processing

In chapter 2.2 problems with the video data are described. To work with these video files these problems have to be addressed. To solve the issue of different frame rates we can use OpenCV to extract exact time codes for each frame. Using these timecodes we can determine how much time has passed since the start of the recording. Although the videos are of different lengths and have a different starting time we know that each frame in a 25 fps video will last 1/25th of a second while each frame in a 30 fps video will last 1/30th of a second. Since we want to compare each frame of one video to a matching frame in the other video we will have to find the closest matching frame. We determine the closest matching frame to be the frame with the least difference in timecodes. For this we have to assume that both videos are also matched in real time and therefore anything happening at a certain timecode in one video is also happening at the same timecode in the other video. If we assume this, we now have two choices of matching the frame rates. We can either match both videos at 5fps by only keeping every 5th frame of the RGB video and every 6th frame of the thermal video, or we can remove some frames from the thermal video to match them at 25fps. Making both videos 5fps would lose a lot of information since only 1/5th and 1/6th of all frames are kept respectively. Therefore it was decided to match both videos at 25fps by calculating the difference in timecodes that would occur whether a frame on the thermal video is skipped or not. This method will inevitably be slightly inaccurate since an exact match in timecodes can only occur every 5th/6th frame respectively while all other frames will be slightly different

	RGB	Thermal
dlib cnn	yes	no
dlib hog+svm	yes	no
FAN 2D	yes	yes (worse than FAN 3D)
FAN 3D	yes	yes
Kopaczka	sub optimal	yes

Table 2: Different face detection models tested

in timecode but never more than +/- 1/60th of a second. Since both cameras were started manually at different times, an exact match is not possible anyway.
Determining the exact starting point of each video and matching them was done manually by identifying unique frames such as a person walking through the recording or an easily identifiable motion done by the donor. These frames were annotated separately and the time difference between the same scene in the two videos was noted.
Solving the issue of different video lengths is trivial, since each recording consists of a pair of videos, the shorter video is used and the information in the longer video is cut to match the length of the shorter video.

3.3 Alignment using Face detection

So far Kopaczka provides the only model trained on infrared images. With this model they also provide a full dataset of annotated thermal images. For this model Kopaczka used the python library menpo. In this work two functions from the menpo library which are no longer supported in newer menpo versions are used, mainly the hog detector from menpo.feature.hog. This function is no longer supported in versions after menpo version 0.9.1. For his research Kopaczka used menpo 0.8.1 which was released in 2017. Back then menpo was still supported on windows which it no longer is. To run menpo, conda is advised by the developers. Following the instructions of the menpo documentation and the notes in Kopaczkas example, we were able to train a model. However, the trained model was very large in size and only accurate on centered front facing faces. Combining these flaws and the large size of the trained model as well as the problems of being limited to a linux based machine running conda we chose to stop the research on Kopaczka's approach. Since the goal is to get a robust and accurate face detection for both thermal and RGB images using Kopaczkas thermal database to train an effective model on a combination of thermal and RGB images might be useful for further research.
The attention was shifted towards face detection methods that work on RGB to test if they would also succeed on thermal video data. Since many face detection models are trained using different ways of generalising the information of the images like edge detection, they could also work on thermal images. Most of the models that were tested however did not work and would not detect a face in thermal images. The models tested are shown in table 2. We came across the Face Alignment Network (FAN) by Bulat et al. [2]. The model described in this paper is available on Github. The Face Alignment Network accurately depicts facial landmarks and does this even for thermal images, although with less reliable accuracy.

3.4 Alignment using Contour detection

Aligning the frames using contour matching requires the extraction of accurate contours from both frames. Ideally, the contours are as similar as possible so that the matching algorithm finds the optimal position. Extracting an accurate contour from the thermal frame is trivial, since the only part visible in the video is the person being filmed. Since the brightness of each pixel is represented as an integer between 0 and 255 a simple threshold function can be used to detect the contours of the face. The background in each video is almost perfectly black, hence there will be no other contours in the video. Problems can be caused by donors having hairstyles that extend further from the head. The further the hair is away from the donors face the colder it gets. This eventually leads to a low pixel value for the hair and causes it to be removed by the threshold function. A similar problem arises when getting the contours from the RGB frame since different hair colors and their visibility can lead to wrong contour detection. Using a threshold function on the thermal video is the ideal solution to retrieving the contours when the threshold value is set correctly.
Using the same approach on the RGB frame does not work. To retrieve the contours we have to use a form of edge detection. Running the OpenCV function *findContour* will not be able to find significant and relevant edges in the original image otherwise. Therefore, we have to highlight the edges that we want as part of our contours. To achieve this we can use Canny edge detection in combination with other image transformations so that the resulting contours are similar to the contours retrieved from the thermal frame. Canny edge detection uses a small subset of pixels to identify changes in colour or brightness which occur at edges. The accuracy of this can be improved by using Gaussian blur to remove small edges that are not useful. We are interested in acquiring the outside edges of the face, head and shoulders. Features like hair should be taken into consideration but only to the extend that they can be detected in the thermal video.

3.5 Output

The final results should be in form of two video files that have the same resolution and ideally every part of each face overlaps perfectly. Since the final results will be used in training different models, each pixel of the RGB video shall ideally also contain thermal information. When done so, a model could be trained using both the information from the RGB video and the information from the thermal video. Therefore we decided to output the final videos at a size of 500 by 500 pixels with the person centered in the middle. When working with thermal videos, most of the useful information is contained within specific areas of the face. Most commonly used are the cheeks and the nose as well as the forehead. When we define the center of the video to be the center of the nose, the nose can be used as a reference point for finding different areas such as the cheeks and forehead. Since the videos will be used to train a neural network the resolution will likely have to be reduced further than 500 by 500 pixels, therefore there is no reason to have the results in a higher resolution.

3.6 Implementation: Face detection

As mentioned in the Methodology section, matching the two videos using face detection requires the correct detection of facial landmarks. This is done using the FAN network since it is the only model that performs well on RGB and similarly well on thermal videos.

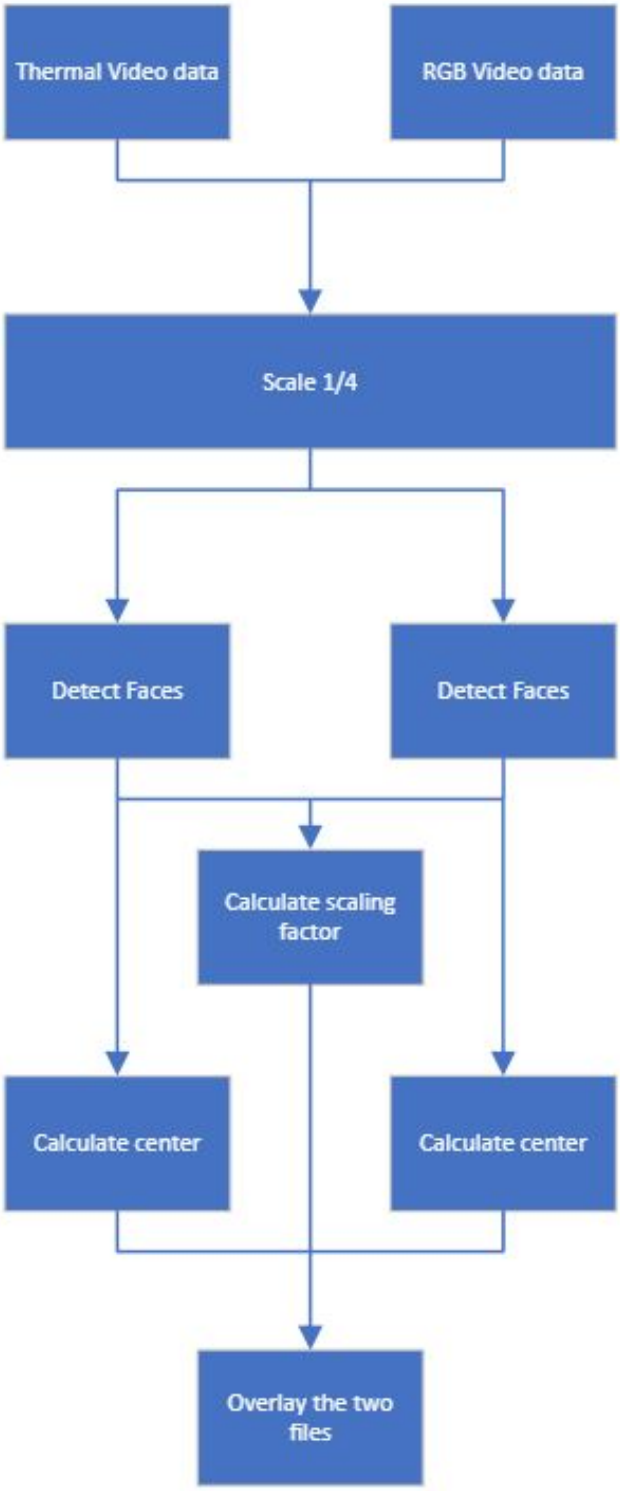

Figure 3: face detection on thermal and RGB images

The face alignment network (`https://github.com/1adrianb/face-alignment`) is able to detect facial features in thermal images quite reliably and since it works on RGB images normally we can use the detected features and compare them. The FAN provides two different models, a 2D and a 3D model that return 2D and 3D landmark locations respectively as well as the option to choose *sfd* or *blazeface* as face detection method. Using the 3D model and the *sfd* face detector provides better results on the thermal video data and therefore this will be used for this approach.

To achieve an accurate detection of the face we first import both videos into our python script using OpenCV. Before looping through each frame we have to synchronize the two videos. This is done using manually annotated time offsets for each video and skipping the correct amount of frames on one video to synchronize them. The first thing that is done in the loop is scaling down both videos to a quarter of their original resolution. This is done so the face detection model runs faster and more accurately. The files are only scaled down to a quarter of the resolution since otherwise the accuracy detecting faces in the thermal video decreases rapidly. Using the down scaled versions of the video, the face alignment network is used to detect faces. Using the gained landmark locations from both videos we calculate a scaling factor to match the size of both faces. Finding this scaling factor is difficult. To calculate it we use the distance between facial landmarks. The accuracy when detecting some landmarks in the thermal video can be poor depending on the donor and therefore varies largely between examples. This can occur if, for example, the hair covers a large area of the frame or the hair covers large parts of the face or neck of the donor. Finding the distance between the landmarks in the RGB image is more accurate since the model is trained on regular images instead of thermal video data. To calculate the scaling factor, multiple different facial landmarks can be used with varying accuracy.

Since the performance of the FAN differs when detecting facial landmarks in the thermal video data, some of the facial features will not be detected perfectly. To understand which facial landmarks could be useful when calculating the scaling factor, we have to identify which facial landmarks are detected incorrectly and in which way they are wrong. When looking at false facial landmark positions in the thermal video of a specific donor we can see that this happens mostly to the outline of the face, the eyes and eyebrows. The nose and mouth are identified correctly using the FAN. Calculating the scaling factor using the mouth and the nose is difficult since the distance between them is relatively small and due to slightly inaccurate classification of both the nose and the mouth, such a small distance leads to huge changes in the scaling factor. To calculate the scaling factor the distance between two easily identifiable facial landmarks is calculated on both the thermal video and the RGB video and then these distances are compared. After this the scaling factor is then used to scale one of the videos by this factor so it matches the other video. In the current implementation the thermal video is scaled up slightly since it has a lower base resolution.

When calculating the distance between two features the average location of this feature is used. The distance between the eyes, eyebrows and both ears were tested. The highest accuracy was achieved using the distance between both eyebrows, although the distance between the eyes performed similarly well. Using the width of the face also produced good results in some examples but failed heavily in others with the jawline being detected far outside a donor's face due to their hairstyle. Using the eyebrows leads to reliable results that can always be detected and while often being slightly inaccurate are always close to the ideal value. The problems with these methods can be seen in Figure 4. We can see that the landmarks are correctly detected in the RGB frame except for the eyebrows which are detected slightly too high. If we compare

this to the results on the thermal frame we see that the eyes and eyebrows are detected too high wile the rest of the face is detected mostly correct. These inaccuracies differ between recordings depending on the donor.
After matching both faces in size the faces are aligned. This is done by first calculating the center of the nose. This works very reliably but is not perfect and commonly shifts in the y-Axis by a few pixels. Using the center of the nose we align this point to be the new center of the frame. To have a resulting 500 by 500 video we first pad all sides of the video with a 250 pixel wide black bar. This is used when the original video is not perfectly centered and would be cut off when centering on the nose. Both videos are then cut to the exact same size with the noses of each frame being the center.

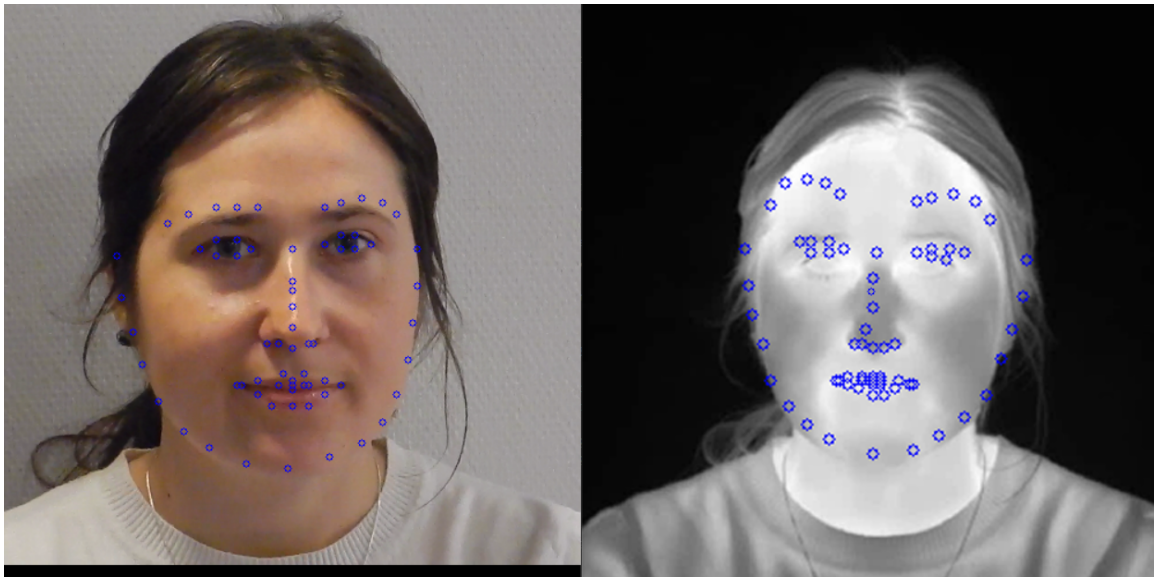

Figure 4: Results from using the FAN on the RGB frame (left) and on the thermal frame (right)

3.7 Implementation: Contour matching

Much like the approach used in older face detection algorithms for thermal data, this approach uses the contours of the recorded faces to match them. This can be done in two ways, either we create an intermediate template that consists of the rough contours of a face and match this template to both the frames of the thermal and the RGB video or we create a template from one frame, either thermal or RGB, and then match this template to the other video. The first method was used successfully shown in [8]. We chose to implement the second approach since it should theoretically be cheaper computationally. This approach is also theoretically more accurate since the template that is created from one frame matches exactly to this frame where an intermediate template will not be an exact match to either video frame.
To implement this we first have to create a template from one of the videos. To get an accurate template from the thermal video is much easier to do than getting an accurate template from the RGB video. To create the template from the thermal video we use a simple threshold function that sets every pixel with a value above 60 to be 1 and every pixel below to be 0. This creates an accurate mask of the donors face. The results of this can be seen in figure 5. Changing the threshold value leads to more of the donors hair being detected.

Figure 5: (left)Mask created using a threshold function on the thermal video. (right) Contours of the mask(left)

Figure 6: Resulting contours after using Gaussian blur and Canny edge detection on the RGB video data

Using this mask we can then get an accurate contour by using the OpenCV function *findContours*. This gives us accurate contours of the donors face that can be used for template matching. Having this template we now need to get similar contours from the RGB video frame. This is more difficult since a regular threshold function will not work. Therefore, we have to rely on edge detection to get the outlines of the donors face. We used Canny edge detection for this combined with 3 by 3 Gaussian blur. The resulting contours can be seen in figure 6.
We can see that these contours also contain the contours of some facial features and that the outside contour is not fully complete. Ideally we would want the contour to only show the outlines of the face and be empty inside these contours but since the contour matching algorithm we are using also works with these contours we will leave it as it is.
Having acquired both contours we now have to match them accurately. OpenCV provides the function *matchTemplate* that allows us to match a template to frame. This *matchTemplate* function has replaced the chamfer matching function that was included in older versions of

OpenCV. The *matchTemplate* function works best when matching exact templates but also returns a best position if no exact match is found. Looking at the template in figure 5(right) we can see that the template contains a lot of empty space around the actual face. This will be a problem when matching this template to a base image since the entire template has to fit within the bounds of the base image that the template is being matched against. To fix this we simply remove empty areas on either sides of the image. Removing these empty areas reduces the size of the template making matching easier.
Since the *matchTemplate* function does not take into account different sizes of the template in regard to the image that it is being compared against, we have to scale both templates to the accurate size before we can match them. This requires a scaling factor similar to the scaling factor used in the face detection approach. For now this scaling is set manually since acquiring it accurately using the templates is very unreliable. The scaling is done by up-scaling the thermal frame by a factor of about 1.8 for the recording used in the images of this thesis. Having up-scaled the template that is derived from the thermal frame we can now use the *matchTemplate* function to align both videos. Using the *minMaxLoc* function on the result of the *matchTemplate* function returns the coordinates of the top left corner of the template in regards to the base image. We chose to align both frames by first matching the templates size to the size of the RGB frame by adding black borders to each side. The width of these borders is calculated using the top left and bottom right corner location of the template in regards to the base image. A simple example for this is: We have a base image of size 400 by 200 pixels and a template of size 200 by 150. The results of the *matchTemplate* function tells us that the top left corner has an offset of 50 and 25. We now know that the template should be placed with its top left corner at the location 50;25 in the base image. To achieve this we can add a black bar of width 50 to the left side of the template and a black bar of width 25 to the top of the template. We can do the same by calculating the bottom right corner location of the template and add black bars to the bottom and right side of the template so that its size matches the size of the base image.
The last step is to cut both videos to size so that they are similar to the results aquired using the FAN approach. To do so we first have to define a new center for the frame. This is ideally the center of the face. Finding the center of the face is difficult if we do not want to use face detection. So we chose to use the center of the mask created from the thermal frame. The center of this will closely resemble the center of the face regarding the x-Axis while the location of the center regarding the y-Axis is dependent on the angle used while filming. If a lot of the donors upper body is visible the center on the y-Axis will be too low. This is not a problem in most of the video data used in this thesis and therefore the center of the mask is used to calculate the new center of the frame. Once we have the center of the frame we can cut the frame to size using the same method as used in the FAN approach.

4 Results and Experiments

Running the two programs gives two different results that can be seen in Figure 7 and Figure 8. Figure 9 shows the results of each approach overlayed. The RGB image is shown on the green color channel and the thermal image is shown on the blue color channel. Since the videos used in this thesis are almost exclusively from donors and cannot be shown in this thesis, one of the researchers let me use her recordings to show the performance of these methods. Aligning two videos as such has not been done before so it is difficult to evaluate the results objectively. Most of this Results section will look into the differences between the results of the two methods.
The results of the two approaches can be seen in Figure 9 with the results of the FAN approach being shown on the left and the results of the template matching approach being shown on the right. For both of these images the RGB frame is being displayed on the green color channel while the thermal frame is being displayed on the blue channel. The final output of the program would still be two separate video files with the same format and alignment but displayed in their usual colors.

Figure 7: Final result created by the FAN approach. Both resulting images are 500 by 500 pixel in size

Looking at the results in Figure 9 we can see that both results are quite similar although there are differences in the alignment of the two frames. We see that the thermal frame in the left image is shifted to the right while the thermal frame in the right image seems to be too far left since the right cheek is not covered by the thermal frame. These differences are likely the result of the camera angle problem. While the FAN approach aligns the face on the center of the nose the template matching approach aligns both frames using their contours. Therefore the contours are matched perfectly in the template matching approach while the nose is perfectly matched using the FAN approach.

Figure 8: Final result created by the template matching approach. Both resulting images are 500 by 500 pixel in size

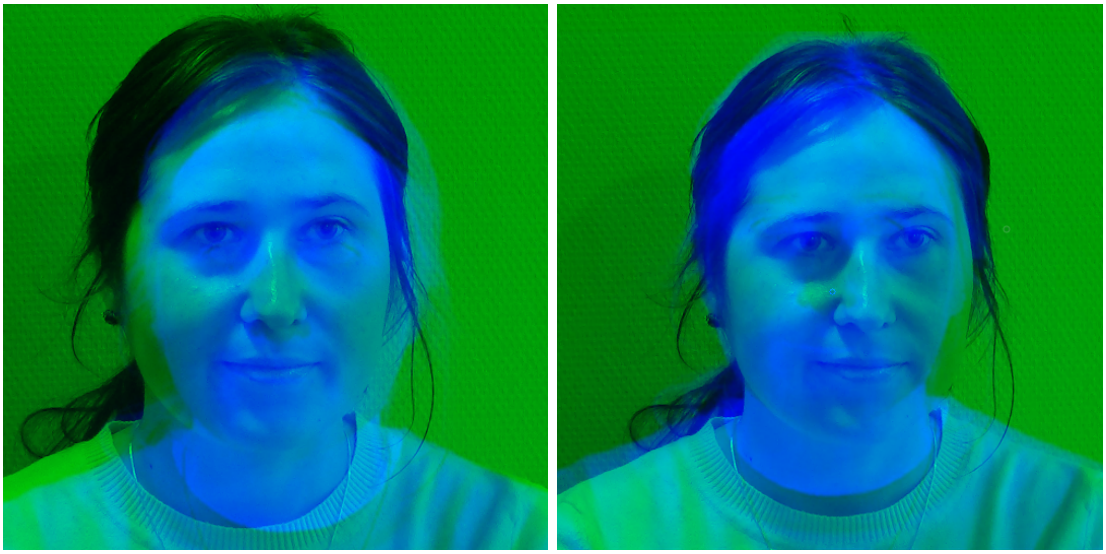

Figure 9: Overlayed thermal and RGB frame, the left image was achieved using the FAN network. The right image was achieved using template matching. The RGB frame is displayed on the green channel while the thermal frame is displayed in the blue channel

Since the fine details of the alignment are difficult to see in Figure 9 the contours of the aligned frames are shown in Figure 10. Here we can see the misalignment's mentioned before more clearly. To create the left image in Figure 10 the functions created for the template matching approach were used and applied to the FAN approaches final images.

4.1 Face detection performance

In Figure 9(left) we can see the resulting frames overlayed. We can see that the match is not perfect with visible discrepancies at the sides of the face. We also see that the scaling has not been performed perfectly since the thermal face is a slightly smaller. This comes from the

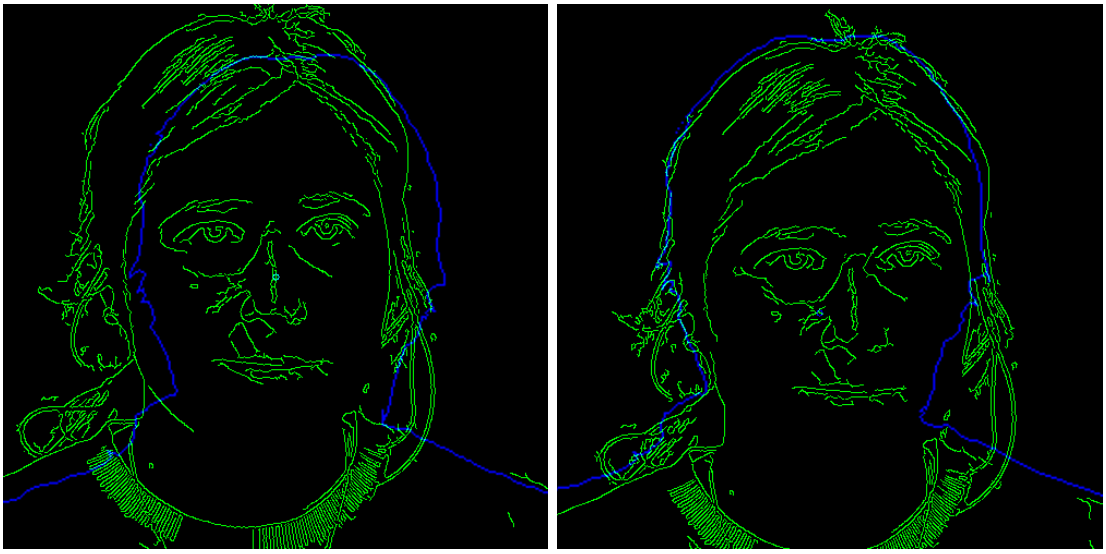

Figure 10: Overlayed thermal and RGB contours, the left image was achieved using the FAN network. The right image was achieved using template matching. The RGB frame is displayed on the green channel while the thermal frame is displayed in the blue channel

inaccuracies in the facial landmark detection for the eyebrows to calculate the size of the face. Since this approach uses the center of the nose to align the videos a difference in camera angle causes the contours of the face to align perfectly if the nose is aligned in the center.
The implementation for this approach works well on the dataset for this thesis. During testing we were able to verify its functionality for 56 out of 57 donor recordings. Due to this approach using face detection it is very robust as long as a face is detected. Dealing with single frames with no detected face is also not a problem since these frames could simply be skipped or estimated by the locations of facial landmarks in the previous frame. However, this approach is limited by the speed of the FAN. When compared to the speed of the template matching it is about 4 times slower although both programs are not optimised for speed. On my local system running an AMD Ryzen 7 4700U the program was able to calculate a frame of a video about every 1.73 seconds while the template matching approach ran at about 0.17 seconds per frame.
This slowdown is mostly due to running the FAN on a quite high resolution image. Reducing the resolution further leads to more inaccuracy when detecting landmarks in thermal frame.

4.2 Contour matching performance

Looking at the results of this approach in Figures 9 and 10 on the right we see that the alignment process works well. The contours are aligned as close as possible which is especially visible in Figure 10. Therefore, the approach of matching the templates was successful. However, this approach is not as sophisticated as the FAN approach since this approach requires the user to manually set the scaling factor since there is no reliable way of calculating it from the video data. A scaling factor could be calculated using the FAN to calculate the distance between landmarks but this would just increase computation time and introduce more points of failure. Since the scaling factor has to be set manually this approach was not tested on all video files.

4.3 Quantifying the results

Quantification of the results is difficult. The main issue lies in measuring objective accuracy. Since no objectively optimal alignment exists we can only measure the performance by comparing the unaligned images with the aligned images. We can see the base images in Figure 1. We can clearly see the issues regarding image shape and size as well as camera angle. Therefore, comparing the misalignment in these original images is not ideal. To compare them anyway, we first have to bring both frames to a comparable size and shape, then calculate the misalignment. We chose to scale the videos in a way that it stays as close to the original video as possible by trying to keep the size of videos comparable. This process is subjective since we chose to reduce the size of the RGB frame so that the size of the donors face is roughly the same size as in the thermal frame. This is probably what the person, setting up the cameras, saw when setting them up.
Since the size of the RGB frame is much larger we chose to scale it down so that it matches the height of the thermal frame. We can also choose to scale it down so that it matches the width of the thermal frame as well. Since scaling down to match the height seems to lead to a more similar size, regarding the face in the recording, we chose to scale it down so that it matches the height of the thermal frame. Now that both videos have the same height we pad both sides of the thermal frame using empty black bars so that both frames now have the same size. Now that both frames are in an 18 by 9 format we can scale them to any resolution. We chose 240 by 426 pixels as final size. Using these frames we can now calculate the misalignment of the videos.

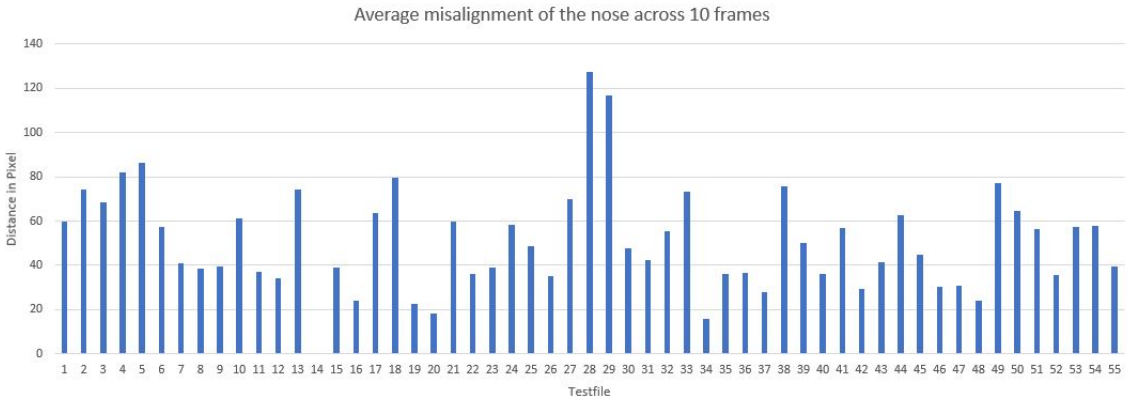

Figure 11: Misalignment across 55 file sets. Each distance was calculated using the average across 10 frames of both video files

The only way to measure the misalignment is by using a parameter that we can identify with a high accuracy. Therefore we will be using the nose as a base. When using the FAN approach the nose will always be in the center of the final frame and therefore the offset is always zero. To further evaluate the performance we will look at one exemplary file. This file will be aligned using both approaches and the runtimes will be tracked as well as the position of the nose. We already know that it will be in the center of the FAN result but we want to look at the offset when using the template matching approach.
Figure 12 shows the alignment of the nose when using the template matching approach. We can see that the nose is misaligned on the X and Y axis. We also see that the rest of the contour is aligned accurately.

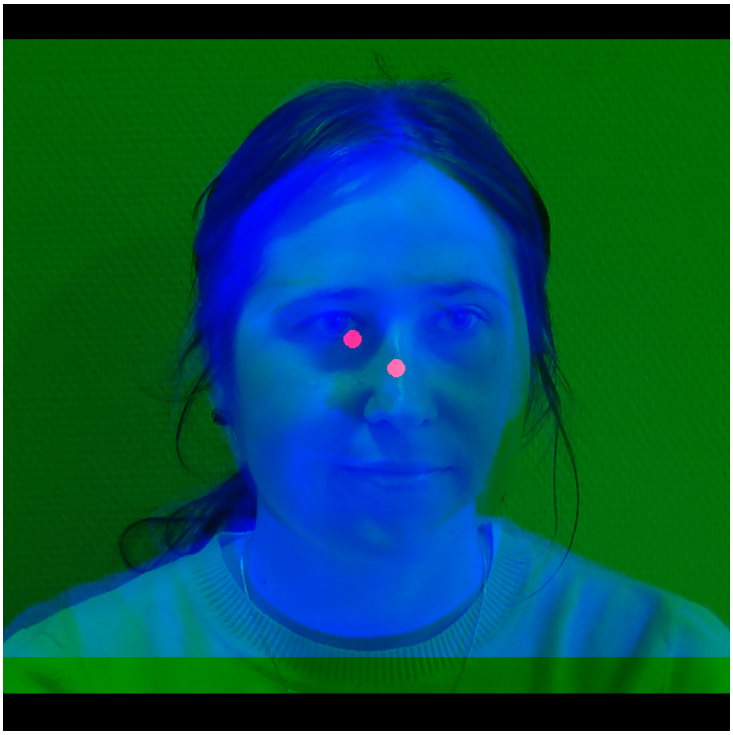

Figure 12: Nose misalignment when using template matching

We can track the misalignment of the nose across the entire video to evaluate the consistency. To do so the final resulting frames of the template matching approach are used. The output is set to produce 600 by 600 pixel videos. Since this image size is too large for accurate predictions by the FAN we have to scale them down. The FAN approach usually scales down the video to a quarter of the original resolution which, for this example, goes from 1036 by 788 pixels to 259 by 197 pixels. Since the final result of the template matching approach went through different scaling transformations during the alignment process, reconstructing the exact same image size is difficult. For the testing we chose a size of 150 by 150 pixels which is a quarter of the 600 by 600 pixels resolution of the original output size. Effectively the size of the video is now smaller than it would usually be when the FAN approach is used. Since most of the videos tested have an original resolution of 748 by 562 pixels and are also scaled down to a quarter of their resolution, scaling down this video even more leads to a closer comparison with the videos which original resolution is lower. The difference in nose position was then plotted across the entire video. The results of this experiment can be seen in Figure 13

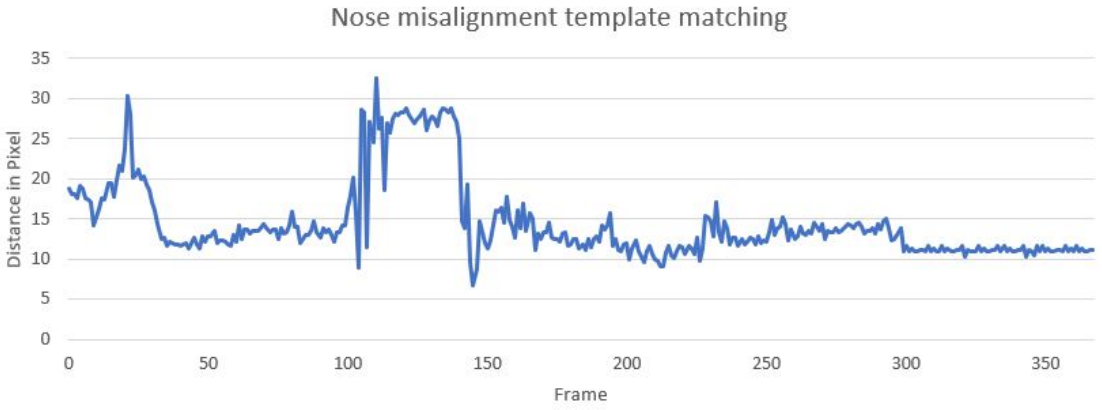

Figure 13: Nose misalignment in example video

In Figure 13 we can see that roughly from frame 100 to 150 the misalignment increases. This is due to a zoom event in the recording where the RGB camera zooms in for about 2 seconds before reverting to the original zoom level. Since the template matching approach does not automatically change its scaling factor, the template and the base image do not match in these frames due to the difference in size. The spike in misalignment in the beginning is due to the focus of the RGB camera being adjusted causing a blurry image.

4.4 Improvements to the videos for better results

The process of integrating RGB and thermal video recordings of the same donor could have been both easier and produced better results if the video files would have been standardised. To improve the quality of the videos, they should firstly be recorded using the same resolution and frame rate. To facilitate synchronization, both videos should be started at exactly the same time, which can be achieved in multiple ways (e.g., by connecting and starting them using a micro controller).
When setting up the cameras they should ideally be mounted together such that they always point in the same direction relative to each other. This can be done by mounting both cameras on a static frame. This frame can then be manually aligned to the donors. Ideally every use of zooming is done before the recording starts and the person being recorded should be fully captured within the frame. Having the person only fill up half the frame is better than having the person being cut off whenever they move or change their seating position. Both cameras should ideally depict the person in the same size. This can be difficult to adjust since camera lenses and the sensor have a significant impact on this. The way both cameras are mounted also has an impact. They should be mounted as close to each other as possible to get the most similar angle. If both cameras are mounted on top of each other the angle will probably have less impact on the recording than when mounted side by side. Mounting them as close as possible together will however improve the matching results. To optimize the video recordings for contour detection a neutral background is ideal. This can be done using a green screen as a background for optimal performance.

5 Conclusion and Discussion

5.1 Discussion

Comparing the results of both approaches gives us no clear winner. Both approaches align the face using different parameters and therefore lead to different results. Since the FAN approach is more sophisticated and can be run on every video that it was tested on without having to manually set scaling factors I personally prefer it over the template matching approach. The template matching approach currently has no way of accurately calculating the scaling factor. Measures like the average width of the contours could theoretically be used to calculate a scaling factor but this solution is currently unreliable due to the inaccurate contours retrieved from the RGB frame. Once the scaling factor can be accurately calculated this approach could work as well as the FAN approach. Looking at the resulting images, the template matching results seem more aligned since the contours overlap almost perfectly. This also means that the facial features such as nose, mouth and eyes are misaligned due to the different recording angles. The FAN approach initially looks like it performed a worse alignment but the facial features are more closely aligned than they are in the template matching approach.
This research was started as part of the FAINT project and the use for these aligned videos will be to predict fainting in blood donors. Using the facial areas for temperature reading is easier if these areas are aligned at the same location across different video recordings. This is achieved when using the FAN approach since the center of the nose can be used as a reference point for different facial areas such as the cheeks or forehead. Getting this information from the results of the template matching approach is more difficult since facial areas are at different locations when taking the center of the image as reference. This is just a preference however since a different use case might lead to a different evaluation. Were the videos recorded perfectly with little to no difference in angle between the recordings both approaches would perform more similar. Overall, the FAN approach is more sophisticated while the results of the template matching arguably align the faces better.
Looking closely at the individual results achieved by the FAN approach we can see that the alignment seems to be inconsistent. Some of the video sets are aligned as accurately as they are using template matching while others seem hardly aligned at all. After testing this approach on all the files we can conclude that the alignment is performed correctly but the difference in camera angle causes severe misalignment in some video sets. Other factors like the donors hand touching their face causes no issues when detecting the facial features and therefore does not affect the alignment. Even incomplete faces can be detected and aligned correctly using the FAN. To improve the performance of this approach further the video recordings need to be improved.

5.2 Conclusion

Aligning RGB and thermal video files depicting the same person seems rather trivial at first sight. To achieve this we have to find elements of the frame that can be accurately recognised in both frames and then we align the frames using the information gained. Looking at Chapter 4 we can see that two approaches to aligning the videos were found. One approach uses facial landmarks to align the videos while the second approach uses template matching to match the contours of both frames. Both approaches perform differently due to the nature of the video data and differences in recording angle.

The first approach uses the FAN (Face Alignment Network) to detect facial landmarks. This face detection model is able to detect facial landmarks reliably for both the thermal and the RGB frame. Since the model is trained on regular RGB images its accuracy decreases for thermal video data but is still accurate enough for alignment. When looking at the final results, we can see that the alignment is not perfect. However, facial features such as the nose, eyes and mouth are aligned quite accurately since the center of the nose is used to calculate the correct alignment. The reason why the contours of the face are not aligned correctly using the FAN is due to the difference in recording angle and can be improved using better recording methods as described in Chapter 4.4.
The other approach that was found uses the contours of both frames and matches them as accurately as possible. The biggest problem with this approach is the creation of accurate contours from the RGB frame and scaling both videos correctly. However, if this is done the alignment works very well with the contours being matched almost perfectly. Since the same issues with the video data exist for this approach, the facial features are not aligned as well as in the FAN approach.
In this thesis two ways of merging a thermal and an RGB video are shown. Although many tools exist for editing both thermal and RGB video, merging both videos by finding similarities in both still proves difficult. The two methods of merging the videos rely on using tools that work but are not optimal which leaves room for improvement.
To answer the three research question posed in the Introduction: (1) Two videos as described in the 2.1 can be overlayed for facial areas to overlap. (2) To do so we have determined two approaches, using facial landmark detection, and using contour detection combined with template matching. (3) Although merging the two video files is possible (with varying accuracy), it could have been made easier and more accurate by changing the way the videos are recorded. This is more closely explained in chapter 4.4.

5.3 Future direction

Aligning two videos in theory is easy but is difficult when thermal video data is used since there are few tools that can work with such data. The two approaches found in this thesis are not perfect with the FAN having problems due to inaccurate facial landmark detection and the template matching approach having no reliable way of scaling both videos accurately. Those problems could be improved by creating a well functioning face detection model for thermal data or ideally, a model that works well on both RGB and thermal video data. It is important that the face detection model detects faces in exactly the same way in the thermal video data as in the RGB video data. The more similar the landmarks are detected in both video files, the more accurate the alignment gets. Since there are currently no well trained face detection models for thermal video data, the FAN was chosen since it performed best. For the template matching approach a template matching algorithm that dynamically scales the image to different sizes when trying to match the template would make this approach more accurate. Getting accurate contours from RGB images can also be difficult depending on the background and the donor and varies between recordings. Having a better contour detection model will also help to make this approach usable.
Since the FAN approach uses face alignment information in 3D it seems obvious to use transformations to match the face alignment of both frames. This is not implemented in this thesis since the alignment information gained about the thermal frame is not accurate enough for such transformations. Overall, both approaches work but each have their own difficulties. Since

the template matching approach is not limited to aligning faces but aligning contours an improved version could be used for different use cases such as aligning multiple people or objects. For this to be possible, the contour detection for the RGB image would have to improve as well as the camera setup.

References

[1] G. Borgefors. "Hierarchical chamfer matching: a parametric edge matching algorithm". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 10.6 (1988), pp. 849–865. DOI: 10.1109/34.9107.

[2] Adrian Bulat and Georgios Tzimiropoulos. "How far are we from solving the 2D & 3D Face Alignment problem? (and a dataset of 230,000 3D facial landmarks)". In: *International Conference on Computer Vision*. 2017.

[3] Yuen Kiat Cheong, Vooi Voon Yap, and Humaira Nisar. "A novel face detection algorithm using thermal imaging". In: *2014 IEEE Symposium on Computer Applications and Industrial Electronics (ISCAIE)*. 2014, pp. 208–213. DOI: 10.1109/ISCAIE.2014.7010239.

[4] Md Khaled Hasan et al. "Human Face Detection Techniques: A Comprehensive Review and Future Research Directions". In: *Electronics* 10.19 (2021). ISSN: 2079-9292. DOI: 10.3390/electronics10192354. URL: https://www.mdpi.com/2079-9292/10/19/2354.

[5] Kaur A. Kumar A. and Kumar M. "Face detection techniques: a review." In: (2019).

[6] Trung NT Ma C and Shimada A Uchiyama H Nagahara H. "Adapting Local Features for Face Detection in Thermal Image". In: *Sensors (Basel)*. 2017, pp. 208–213. DOI: 10.3390/s17122741.

[7] Raphael Kolk Marcin Kopaczka, Felix Burkhard Justus Schock, and Dorit Merhof. "A Thermal Infrared Face Database With Facial Landmarks and Emotion Labels". In: *IEEE Transactions on Instrumentation and Measurement*. 2019.

[8] Ant´onio J. R. Neves Ricardo F. Ribeiro Jos´e Maria Fernandes. "Face Detection on Infrared Thermal Image". In: *Signal*. 2017, pp. 208–213.

[9] Open Source Computer Vision. *Main page*. 2022. URL: https://docs.opencv.org/4.x/index.html (visited on 08/08/2022).