Range Face Segmentation: Face Detection and Segmentation for Authentication in Mobile Device Range Images

Rainhard D. Findling, Fabian Wenny, Clemens Holzmann, René Mayrhofer
firstname.lastname@fh-hagenberg.at

Department of Mobile Computing
University of Applied Sciences Upper Austria
Softwarepark 11, 4232 Hagenberg, Austria

ABSTRACT

Face detection (finding faces of different perspectives in images) is an important task as prerequisite to face recognition. This is especially difficult in the mobile domain, as bad image quality and illumination conditions lead to overall reduced face detection rates. Background information still present in segmented faces and unequally normalized faces further decrease face recognition rates. We present a novel approach to robust single upright face detection and segmentation from different perspectives based on range information (pixel values corresponding to the camera-object distance). We use range template matching for finding the face's coarse position and gradient vector flow (GVF) snakes for precisely segmenting faces. We further evaluate our approach on range faces from the u'smile face database, then perform face recognition using the segmented faces to evaluate and compare our approach with previous research. Results indicate that range template matching might be a good approach to finding a single face; in our tests we achieved an error free detection rate and average recognition rates above 98%/96% for color/range images.

Categories and Subject Descriptors

I.4.8 [**Image Processing and Computer Vision**]: Segmentation; I.5.4 [**Pattern Recognition**]: Applications—*Computer Vision*; G.1.2 [**Numerical Analysis**]: Approximation—*Approximation of surfaces and contours*

Keywords

Range images, mobile device, face detection, face segmentation, template matching, snakes

MoMM2013, 2-4 December, 2013, Vienna, Austria.

1. INTRODUCTION

Face recognition (automatically identifying individuals by their face) is an important task used in a wide variety of security related systems, such as building access, border controls with biometric passports, in video surveillance, or for authentication to computer systems. One of the most important prerequisites to face recognition is face detection – which is finding and segmenting faces from arbitrary perspectives in a given image. Recently, face detection and recognition have received more attention in the domain of mobile devices as one form of authentication to unlock a personal mobile device that does not depend on remembering and entering a PIN/password. However, due to bad image quality, changing illumination conditions and background information included in images, both tasks are generally more complicated in the mobile domain and lead to worse results when compared with e. g. video surveillance settings. Consequently, approaches to still obtain feasible authentication results under these conditions have been developed, such as by combining face recognition with other authentication approaches [16, 25].

When looking at the task of successfully performing face recognition, face detection must provide good results as input to the recognition in terms of a) a high rate of correct detections and b) a good face normalization. When looking at normalization, it is important that all detected faces roughly include the same area of facial information (e. g. from the left to the right ear and from the hair line to the chin) – and that inside the area marking a face, the faces should be positioned equally (such as centered at the nose). When measuring the face detection rate itself, the performance can be stated as amount of correct detections (true positives) and the amount of wrong and missing detections (false positives and false negatives). A low true positive rate means that many faces are missed during detection – which causes the face recognition to have less data available during training and classification. A high false positive rate means that many detections don't actually contain faces – which causes the face classifiers to learn from non-face images. Both cases will decrease the recognition rate and should therefore be avoided.

Unfortunately, the detection rate is not the only factor influencing face recognition. Estimating the detection rate for a specific face detection approach depends on making a binary choice for each of the images if the face was detected correctly or not. This includes a tolerance in terms

of normalization so that faces e. g. slightly shifted to one side, scaled slightly differently or with a certain amount of background information still present will also be counted as correctly detected faces (see figure 1). If the grade of face normalization provided by face detection and segmentation is not sufficient, subsequently applied face classifiers will not only learn the faces' discriminating features, but also discriminating features in normalization[1]. E. g. if the face of subject A is shifted to one side of an image, a classifier will also learn the shift besides learning the face properties. If a face of subject B has the same shift, the classifier will more likely classify this face as originated by subject A.The same applies for background information present in face images after face detection, e. g. with using a rectangular crop area for face segmentation. Again, face classifiers will learn discriminating features in background information additionally to the faces' discriminating features. Consequently, the detection rate itself is a poor indicator for the impact of face detection quality on the subsequent face recognition step, and we therefore evaluate not only the detection rate but also the overall recognition rate and compare it with previous research results.

Figure 1: Face images after face detection showing background information and unequal normalization in size and position [6].

As an additional issue, many face detection and recognition approaches are intentionally designed to work from the frontal face perspective, with only a small set of algorithms being able to also handle other perspectives (such as the profile perspective). To address the mentioned problems of requiring a high correct detection rate and good face normalization, we propose a face detection and segmentation approach based on range images. With range images, each pixel represents the camera-object distance (range images can be obtained e. g. by using stereo cameras and stereo vision algorithms, such as with [12]). Using the known face area from range face segmentation, a grayscale version of the same image can be segmented analogously afterwards. Our approach is intended to be applicable to all perspectives, including frontal and profile faces.

Therefore, this paper makes the following contributions:

- First, we introduce a novel approach to range based face detection for different perspectives, as prerequisite to face recognition on mobile devices. Our approach is based on range face template matching, for which we use templates of the average face's range information per perspective. We explain the template generation and matching in detail with focusing on a high grade of normalization (see section 2).
- Second, we apply gradient vector flow (GVF) snakes after template matching for precise face segmentation along the faces' contours (see section 2.3).
- Finally, we evaluate our face detection and segmentation approach in different stages on range images of the u'smile face database. We a) compare our face segmentation results with previous research and state of the art face detection approaches in terms of detection rate and normalization, and b) apply face recognition on segmented faces in order to measure the actual gain to mobile face recognition (see section 3).

[1]When not learning from geometric but appearance-based features.

2. METHOD

Our range face detection and segmentation approach (see figure 2) performs an initial background removal on input range images. Then, it matches average head range templates of different perspectives to given range images in order to find the head's most probable position. As soon as we know the head's position, we can already perform an approximate segmentation of the face, using the average head's range template contour. As this contour has no possibility to fit the individual face's actual contour, we additionally utilize GVF snakes to precisely segment the face in grayscale and range input images. Using segmented faces as input to face recognition, we measure the quality of our detection and segmentation using classifiers for each perspective and test subject.

2.1 Range Face Template Assembly

This section describes the semi-automatic creation of average head and torso range templates from range images[2]. The template creation toolchain is structured as follows: we first perform a coarse background segmentation to discard information not related to the human face and body. We then normalize the heads' positions so that they are roughly equal in all images. Finally, we create a) average range images, which represent the average range to the subject, and b) "hit count" templates, which – for each pixel – represent the amount of images in which subjects had range information present.

The template creation is not intended to be performed on the mobile device and has to be done only once for each perspective from which face detection should be performed afterwards.

2.1.1 Coarse Background Segmentation

In order to only use range information related to the human head and torso for template creation, we discard all range values bigger than a threshold α. This requires all range images used during template creation to be recorded from roughly the same camera-subject distance, as it is the case with the u'smile face database. Further, α is perspective-dependent: therefore we use histograms of the range value distribution of all images for each perspective to determine the correlating α. The smallest range values (first peak) represent the head and torso, the farthest range values represent background information. Therefore we define α after the first peak (see figure 3).

[2]Assuming heads have been normalized to the same size and rotation, as if the images would have been recorded upright and from the same camera-to-subject distance for all participants.

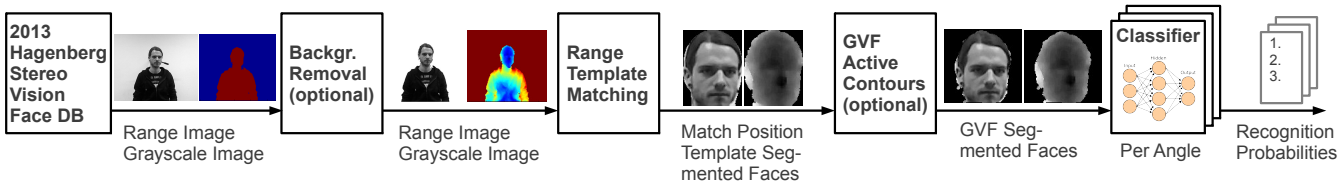


Figure 2: Range face segmentation test setup with optional background removal and GVF snake face contour segmentation.

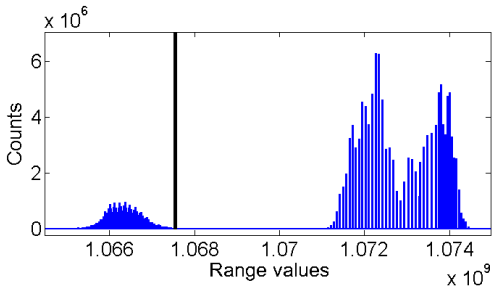
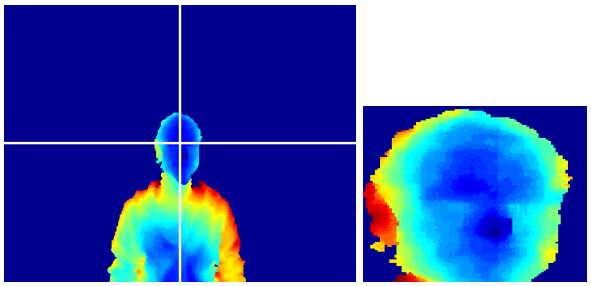


Figure 3: Range value distribution of images in the 0° perspective with α marked bolt.



(a) Central position (b) Boundary area

Figure 4: Head position normalization with a) marked central point and b) a head cut out by it's found boundaries.

2.1.2 *Head Position Normalization*

After coarse background segmentation, we roughly center the heads' positions in the range images. Therefore we search a head's four boundaries and align them along the image's center.

To find the top boundary, we search from the image's top for the first line containing at least N_t range values. N_t can be adjusted to avoid outliers – e. g. for our implementation we used $N_t = 40$. As this line lies beyond the head's top, we go back up 2% of the image height to be sure that all range information is included. The resulting y-coordinate is the head's top boundary. We assume the heads' bottom boundary lies h pixel beyond the found top boundary, with h being hand-picked from the range $[100, 135]$ depending on the perspective. For frontal perspectives the chin is nearly in the same height as the neck. Therefore a smaller h can be used, as the smallest horizontal area filled with range information is higher than in the portrait perspectives, where the chin is beside the neck and cannot be ignored. To find the right and left boundaries, we now crop the image to the top and bottom boundaries in order to discard range information correlated to the torso. Then we use a similar approach as for finding the top boundary: from the image borders on each side we search for the first column containing at least N_s range values, with $N_s = 70$ in our implementation. Again, we then go back 2% of the image's width towards the image's borders for including all relevant range information[3]. As we now know a head's four boundaries (with the boundaries' central point being the head's center point), we can a) shift the center points of all heads of a perspective to the same position and b) cutout the heads (see figure 4).

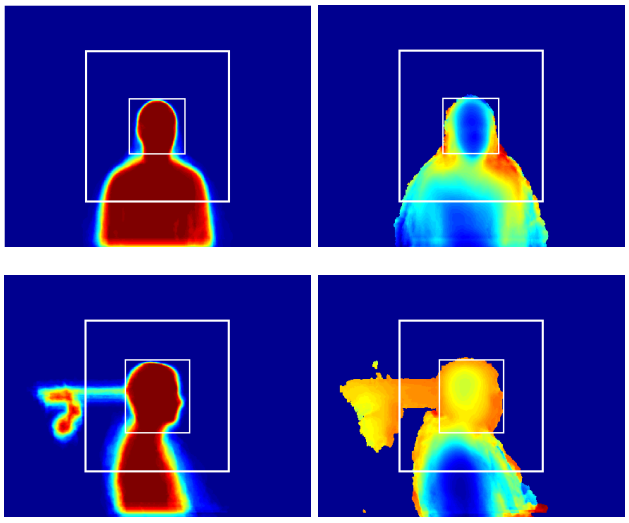
[3]As our test data contains an artifact in portrait perspectives, we have to perform an additional artifact removal in our evaluation implementation at this point (see section 3.1)

2.1.3 *Template Assembly*

Using the head position normalized range images, we can now create four range templates per perspective: a "hit count" torso and face template and an average range torso and face template. The hit count templates represent the amount of images per perspective with range information present at a certain pixel. E. g. as we have at most 620 images per perspective in our test data set, the value range of a hit count template is $[0, 620]$ in our implementation. The average range templates represent the average range information of all images per perspective. We do not consider zero values (=background) for the template creation, therefore the range value at a certain pixel is the average of all images only having a foreground value present. The face templates are composed out of the range images cropped to the face areas determined using the head position normalization. The torso images are composed out of the normalized, not cropped range images. We note that template matching conceptually can be performed based on average range templates as well as hit count templates. In our experiments we achieved worse results throughout using the average range templates and therefore describe our approach using the hit count templates only.

We crop the torso templates to a 300×300 px area around the users head. These cropped torso templates are used during the first stage of template matching which determines the coarse position of the user's head in a range image. The smaller face templates are used during the second stage of template matching, which aims to improve the accuracy of the face position found during the first stage. Therefore, the second stage template matching is performed in a small area around the initially found face position.

Figure 5 shows two examples of head position normalized hit count and average range images. The larger marked region is the 300×300 torso template used for coarse matching, the smaller the face template for fine grained matching.



(a) Hit count (b) Average range

Figure 5: Average torso and face images from frontal and profile perspective. Torso and face templates are marked white.

2.2 Face Template Matching

We perform face template matching based on sliding window principle and template scaling (see figure 6), so that different face sizes and positions can be matched. Our matching approach further consists of two (basically identical) steps: coarse and fine matching, using the created torso and face range templates. During coarse matching, a rough estimate of the heads' position is detected using different scalings and positions of the torso templates. Based on the coarse position we then perform fine matching in the surrounding area using the face templates (using finer sliding window steps with different template scaling and positions again) in order to improve the accuracy of the found face's position.

Comparing how well a template matches a certain area in a range image is done using a template matching metric. Normalizing the template and range data depends on the grade of preprocessing applied to the range input data in order for the metric to work correctly. The metric and normalization for range data without background (see section 2.2.1) is applied in case the background has been removed from input range images (as it has been applied to range images used during template creation). In case no background removal has been applied the metric and normalization for range data without background (see section 2.2.2) are applied instead. We implement and evaluate our approach with range data with both background still present and removed (see section 3).

During our experiments we discovered that searching for correlations between hit count templates and range images is more robust than comparing range values of range images

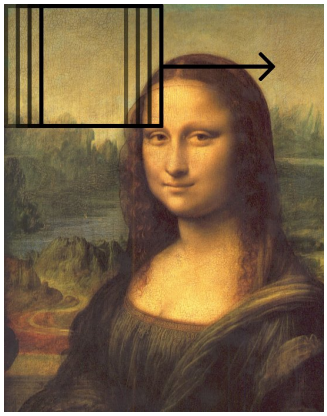


Figure 6: Concept of template matching using a sliding window: similarity between the template and the image is calculated as the template's window slides through the image.

and average range templates. Therefore we only make use of hit count templates subsequent to this point – although average range images might be used as well with different metrics.

2.2.1 *Template Matching Metric: without Background*

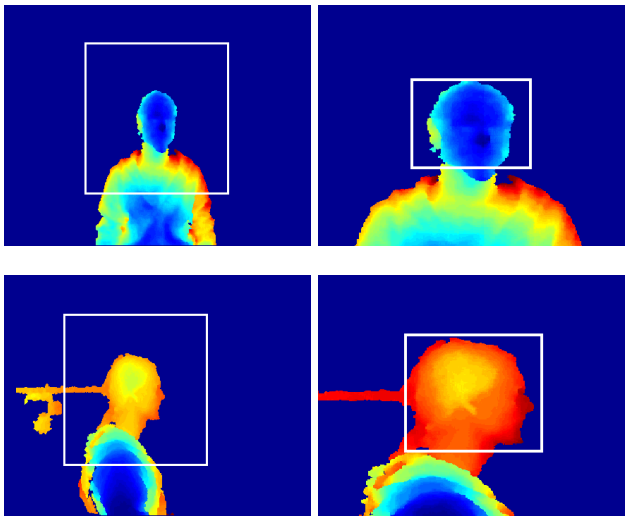
This range data and template normalization requires a background-free range image — similar to the images used during template creation. Initially, we convert the hit count template T to the range $[-1, 1]$ using $T = \frac{T - min(T)}{max(T) - min(T)}$. For matching T with an actual range image I we have to normalize I to the range $[-1, 1]$ too, by setting all pixels containing no range information (background) to -1, and all pixels containing range information to 1.

As a) our metric essentially is a multiplication of the normalized hit count template T with the normalized range image I and b) T by now most likely contains more background (-1) than foreground (1) information, the following effect could be observed: as T contains a bigger background than foreground region it will give a bigger weight to matching the background than to matching the foreground. Therefore not matching the background would lead to a stronger decreased metric than not matching the foreground. This could lead to false detections with large background regions being almost perfectly fitted, but the smaller foreground region being missed completely. Consequently we have to correct T before matching, which we did by scaling all values > 0 so that T sums to 0. As side effect the range of T is no longer $[-1, 1]$, but $[-1, N]$ with $N > 1.0$. e. g. for the perspective of $0°$ we obtained a template range of $[-1, 3.0036]$.

After bringing the range image I and hit count template T to the required range, the metric M can be computed (see equation 1). The current template area $A(T)$ is used to normalize M independently of the size of the currently matched area. Regions outside the current size of template T are not taken into account for M. Higher values indicate better matches.

$$M = \frac{1}{A(T)} \sum_{x,y} T_{x,y} \cdot I_{x,y} \qquad (1)$$

Figure 7 shows the best match found by the template matching metric in a range image without background information. The regions marked white are the best matched position with the torso template (big) and face template (small).



(a) Coarse matching (b) Fine matching

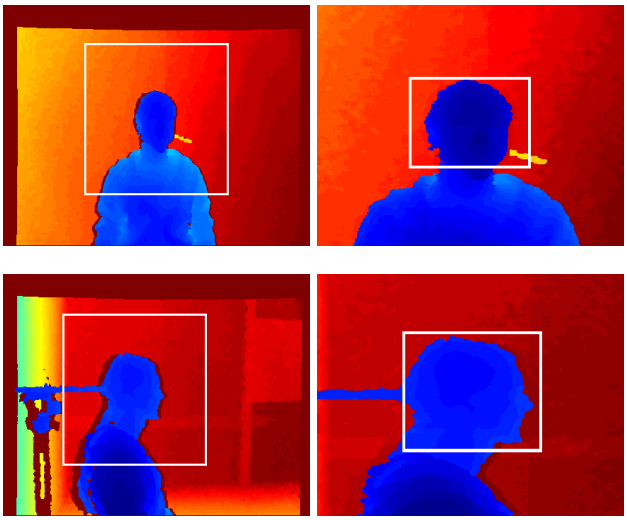
Figure 7: Best matched face area for coarse and fine matching on range images for a frontal and profile perspective with background information removed

2.2.2 *Template Matching Metric: with Background*

In case background removal is not applicable for range images, we use this range data and template normalization which is slightly adapted to work with background information still present in images. Before actually matching template and range image, errors in the range image (regions without information) should be corrected. This is particularly important when matching range images still including background information, as the subsequent application of GVF snakes will likely deliver erroneous results caused by these errors. Especially errors directly at the borders of the face need to be corrected. For the test data used in our evaluation, the real range information of these unknown regions next to the face would be "background". Therefore we apply a hole filling algorithm which fills up these unknown regions with background information.

As for matching range images without background, we first normalize the hit count template T to the range $[-1, 1]$, then scale values > 0 so that T sums to 0. As the range image I contains background information, we cannot apply a simple binarization as for matching range not containing background information. Instead we also normalize I to the range $[-1, 1]$. This requires a strong rise in range from the head (foreground) towards the background, so that the foreground will more likely contain positive values, and the background more likely negative values. We then compute the metric using equation 1.

Figure 8 shows the best match found by the template matching metric on a range image including background information. In comparison to figure 7 the results are approximately the same. In case background range values are overall increasing/decreasing towards a certain direction, small shifts along this direction are to be expected.



(a) Coarse matching (b) Fine matching

Figure 8: Best matched face area for coarse and fine matching on range images for a frontal and profile perspective still containing background information

2.2.3 *Sliding Window and Template Scaling*

We perform face template matching in two stages: coarse matching using the torso template and fine matching using the smaller face template — both from the corresponding perspective. In each stage we perform template matching in a sliding window principle with scaling templates to different sizes.

In the first stage the sliding window starts with step sizes S_x and S_y in x- and y-direction, with $S_x = 40px$ and $S_y = 40px$ in our implementation for performance reasons. For each step, we compute the metric using the current position of the sliding window and the template as described in section 2.2.1 and 2.2.2. After the complete sliding window iteration the result is a list of all metrics with corresponding size and position of the template in the range image. The highest metrics' areas give a first indication where the final best match is located. To find this best match the next step is to call the sliding window again with smaller step sizes and a smaller search area for a more precise search. We use the best four metrics' positions and combine them to the new search area. Additionally, we extend this area with the padding P to ensure that a potential best match at the borders will not be ignored, with a $P = 4px$ in our implementation. For the new and smaller step sizes we use one-tenth of the new search area's length in x- and y-direction.

We call the sliding window algorithm recursively with more precise search areas and smaller step sizes after each iteration until the step size is $1px$ or the search area does not change anymore. Reaching this point we are at the smallest possible search area and have found the best match with the coarse template. Because of different distances between person and camera and therefore different torso and face

sizes, a scaled template could find better matches. Our sliding window technique always uses the same template size for one complete best match search, and we therefore apply a template scaling on top of the sliding window. For each of the different scaled versions of the template, we compute the best match with the sliding window and compare these results in order to find the optimal scaling factor.

For the start of the template scaling algorithm we define three parameters for the scaling range: S_{start}, S_{end} and S_{step}. In our implementation we used $S_{start} = 0.8\%$, $S_{end} = 1.2\%$ and $S_{step} = 0.1\%$. For each scaling step the sliding window algorithm finds the position with the best match. Out of these metrics we use the scaling factor S obtained from the best metric and use it for a more precise scaling factor search. We perform three template scaling iterations to optimize the scaling factor. For each iteration the new search parameters get calculated using equation 2.

$$\begin{aligned} S_{new\_step} &= \frac{S_{step}}{10} \\ S_{new\_start} &= S - 2 \cdot S_{new\_step} \\ S_{new\_end} &= S + 2 \cdot S_{new\_step} \end{aligned} \quad (2)$$

The result of the template matching is the best match of differently scaled torso templates in the range image. We obtain the position of the best match and the scaling factor for the template. After having this information the next step is to search the head in the area of the torso. In order to archive the head position the last step is to repeat the whole process of the sliding window with the head template and the torso area. In our implementation we start again with step sizes of $40px$ and use the resulting scaling factor S from the torso template scaling to scale the head template.

The result of this sliding window is the best match position of the head in the torso area. We cut this head area for the face segmentation.

2.3 Face Segmentation Approaches

After performing range template based face detection we know the position of the face in the image. As next step we segment this face (discard non-face related information). This can either be done by using the range template's contours, or by applying GVF snakes to precisely segment along the individual face contour.

2.3.1 Template Segmentation

A computationally fast and easy to implement approach which delivers feasible results is to segment the face along the hit count template's contour. As the hit count template is very likely larger than the detected face (it contains information in all pixels at which at least one range image contained information during template creation), we only consider pixels for which at least $N\%$ of range images contained information. This leads to the hit count template's contour getting smaller – and fitting the average face better. Again, N depends on the perspective: for the perspective of $0°$ we use only hits with at least $N = 50\%$ appearance in all images of this perspective. For the perspective of $\pm 22.5°$ perspectives we use $N = 60\%$ and for all other perspectives we use $N = 70\%$. When cutting out along the contour of pixels fulfilling the $N\%$ criteria of the hit count template (without using snakes to exactly match the contour), we still can segment faces quite exact (see figure 9).

Figure 9: Faces cut out by the hit count template contour

These faces can be used directly as input to face classifiers. Although this approach is faster – as no further segmentation computation is necessary – it has the major drawback of not fitting the face's actual contour, as it only takes into account the "average face contour" from the corresponding perspective. Therefore, we additionally use GVF snakes to fit the cut out area precisely to the individual's face contour in the next section.

2.3.2 GVF Snakes Segmentation

We apply the GVF snake calculation by Xu et al. [28] to precisely cut out the face on the individual's face contour. We position the initial GVF snake on the contour of the preprocessed hit count template from section 2.3.1 – exactly on the contour, on which a cut out using the template only would take place. From there, the GVF snake should evolve towards the face's actual contour. For this purpose we first create an edge map of the area around the face in the range image, which was found using template matching [4]. Based on the edge map we calculate the GVF field, acting as external forces which pull the snake towards the edge. As GVF parameters we use a regularization coefficient $\mu = 0.2$ and 80 iteration steps. We further specify the following parameters for the GVF snake's internal calculation: $\alpha = 0.05$, $\beta = 0$, $\gamma = 1$ and $\kappa = 0.6$. This results in a trade-off between a quite precise edge matching and fast calculability (see figure 10).

After performing additional GVF snake segmentation, we naturally obtain more precisely segmented faces, which contain less background information than with cutting out faces at the hit count template's contour (see figure 11). Again, these faces are used as input to face classifiers in the next section.

To wrap up: in order to perform range template matching we at first assemble templates of the face and torso area from different perspectives. We therefore first remove background in range images, second perform head position normalization and finally create hit count templates from the range images. Before performing template matching, we normalize input images not containing background information slightly differently than range input images still containing background images in order to handle both variants. We then perform range face detection based on two staged sliding window template matching and template scaling, using the torso template in the first, the face template in the second stage. In each template matching stage, we recursively reduce the granularity of our search by decreasing the sliding window step width in areas of interest. We do this until we have found the most likely position for the torso in the first stage and the face in the second stage, inside the region marked by the torso.

(a) Edge map (b) Initial active contour

(c) Snake deformation (d) Final GVF snake

Figure 10: Step-by-step results from edge map up to the GVF snake.

Figure 11: Faces segmented by GVF snakes after performing range template matching.

3. IMPLEMENTATION / TEST RESULTS

For evaluation, we implemented our face segmentation approach in Matlab. As test data we use the Kinect color and range images from 2013 of the u'smile face database [6]. Using our implementation, we perform face detection and segmentation in 4 different setups:

1. With initial image background removal and without performing GVF snake segmentation.
2. With initial image background removal and with performing GVF snake segmentation.
3. Without initial image background removal and without performing GVF snake segmentation.
4. Without initial image background removal and with performing GVF snake segmentation.

Not performing initial image background removal represents cases in which background removal is not possible for various reasons. Based on the segmented faces we then perform face recognition on range and grayscale images separately and compare our results to previous research [7].

3.1 Test Data Artifact Removal

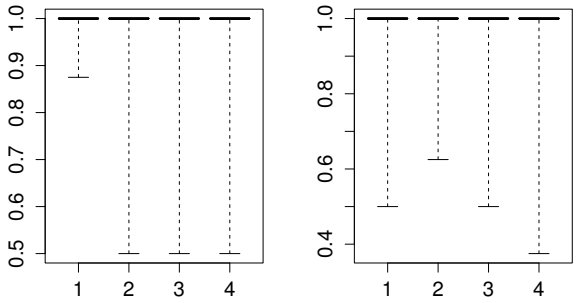
For the image acquisition of the u'smile face database recorded in 2013, a stick behind the head was used to adjust the distance between Kinect sensor and each person. For the head position normalization we need to consider this stick at the back of the head. In the portrait perspectives the side's boundary without the stick can be found in the same way as for the frontal perspectives. Using this boundary plus a width of $140px$ the main stick can be removed from the image and only the head plus a small additional stick remains. After that we remove the remaining stick by searching the first appearance of 70 range values from the side with the stick, because the remaining stick's height is smaller than this threshold and the back of the head's boundary is found.

3.2 Face Classifiers

Based on segmented faces we perform face recognition on grayscale and range faces separately. Therefore, we treat our face recognition as a binary classification problem. When creating the positive and negative data classes, all images of the particular subject form the positive class, and all images of the other subjects form the negative class. As we have 30 people in our test data, we a) compute 30 such binary classification problems and b) have a negative class being 29 times the size of the positive class. For this reason, the recognition rates for the negative class are expected to be better than those of the positive class – we therefore only state the true positive recognition rates in graphs. We use 60% of the data of each class for training and cross validation. The remaining 40% test data are used exclusively for measuring the final recognition rates. For performing face recognition based on our range template based face segmentation results, we use a Support Vector Machine (SVM) from LibSVM [5] and perform a parameter grid search as suggested by Hsu et al. [9]. It turns out that the best classifier for our data is a linear SVM with cost of 10.

3.3 Results

In comparison to the face detection rate of 77.63% from previous research on the u'smile face database [6], which is based on the OpenCV implementation of Viola and Jones [13, 27], we achieve a 100% correct detection rate with all setups of our range template based approach, as we correctly detect all faces.



(a) Colored faces (b) Range faces

Figure 12: Boxplot showing average true positive face recognition rates using segmented faces in a) color and b) range with setup 1-4.

The face recognition results (see figure 12 and 13) clearly show that precise face segmentation using GVF snakes does not improve results (respectively does not improve them significantly). When using initial range image background re-

moval, the average true positive/true negative face recognition rates without GVF snakes are 99.78%/100% for color and 98.21%/99.99% for range, compared to 99.33%/100% for color and 98.7%/99.99% for range with GVF snakes. When looking at face segmentation without performing initial background removal, the average true positive/true negative face recognition rates without using GVF snakes are 99.06%/100% for color and 97.4 %/99.98% for range, compared to 98.61%/100% for color and 96.82%/99.97% for range when using GVF snakes. Also, using range images is on average less reliable than using grayscale images.

We believe the reason for GVF snakes not improving results further is that a) segmentation based on cutting out the head along the templates' borders already showed good results and b) the range images contain undefined areas at or next to the face's real contour from the start. These range errors lead to an edge map showing a contour not completely matching the face's real contour (such as the lower left area in figure 10). Therefore, the GVF snake does not completely match the actual face's contour, but also includes areas of range error – which vary across the subjects and are learned by face classifiers along with the real face features. We believe that GVF snakes face contour segmentation will result in improved face recognition rates (over a cutout along the template contour) when using more accurate range images in the first place than the Kinect can currently deliver.

Altogether, our current face recognition results are still significantly better than from previous research based on grayscale images only and Viola and Jones face detection with a rectangular crop area [6]. When passing all detected faces (including false positive detections) to the classifiers, the average true positive/negative recognition rate was 86.22%/99.57%. Even when only passing correct detections to the classifiers, the average true positive/true negative recognition rate using grayscale images was 97.81%/99.98% – which still is worse than with our current approach. This likely correlates with background information still present in face images and worse normalization in terms of face size and position for the previous approach. We further achieve improved results compared to previous research also based on range template matching [7], where the average true positive/true negative recognition rate was 96.85%/99.97% for color and 93.89%/99.95% for range images – with the most important improvement being the precise data normalization during template creation and matching.

4. RELATED WORK

4.1 Building Blocks

We make use of GVF snakes for precise face segmentation along a face's range contour. Snakes are introduced by Kass et al. [11] as active contour models, that create a line towards features – such as edges – based on internal and external constraint forces. Xu et al. [28, 29] call these forces gradient vector flow (GVF) and compute them as a diffusion of the gradient vectors of an edge map. We create such edge maps from the resulting cutout face range images of the face template matching.

In order to perform face recognition for evaluating our approach, we rely on Support Vector Machines as mentioned e. g. by Phillips [19]. In our test setup we use the implementation of LibSVM [5] with a grid search for parameter tuning as suggested by Hsu et al. [9].

4.2 Related Work

There exist many approaches to face detection, and for a more comprehensive review we refer to [8, 10, 22, 30]. Important research in face detection has been conducted by Turk and Pentland with using Eigenfaces for detection as well as recognition [26]. Rowley et al. use neural networks for detecting the presence of faces [20]. Sung and Poggio use a view-based model cluster approach [24]. Bayesian features are used by Liu to detect faces [14]. Schneidermann and Kanade use wavelet transformation for object detection, including faces [23]. One of the de facto standard approaches to face detection was publicized by Viola and Jones in their object detection framework with Haar-like features [27], with the extension by Lienhart [13]. For their approach there is a measurably decreased detection rate for the profile perspectives, as discussed in [21, 22].

There are a couple of approaches in the 3D face segmentation field. Blanz et al. [1, 2] segment faces with fitting a statistical, morphable model of 3D faces to images in frontal and profile perspectives. In comparison to our approach, they created the model from a set of textured 3D laser scans of the head. Pamplona Segundo et al. [18] propose automatic face segmentation in range images based on edge detection, region clustering using K-means and shape analysis to segment faces. As with many others, their approach is only designed to work from the frontal perspective. Other approaches search for certain features – such as the nose position – to perform segmentation. E. g. Mian et al. [17] perform a nose tip detection, then segment the face using a sphere centered at the nose tip position. Further approaches use a histogram of the range coordinates [3] or segment the head by using a statistical modeling of head and torso points [15].

5. CONCLUSIONS

We are working on a range based face detection and segmentation approach which is intended to be used as prerequisite to face recognition – and can be applied from different perspectives around the head. Our approach focuses on reliability, a high grade of face normalization and precise face contour segmentation. First, we create average range face templates per perspective, which we match with range images to detect the person's head. Then we apply GVF snakes for precise face segmentation along the face's contours. Finally, we perform face recognition to estimate the real-world applicability of our approach.

Our results indicate that face detection and segmentation based on range information might be a very effective approach in general for finding single faces in a mobile device unlock scenario. Using range template matching, we achieve an *error free* face detection on Kinect color and range images of the u'smile face database. Faces segmented by our approach are normalized in position and size and contain very little background information. As both of these are very important to face recognition, we naturally achieve better face recognition results compared to previous research using the same data. Using a linear Support Vector Machine as face classifier we achieve average true positive recognition rates above 98% on grayscale and 96% on range images from our test set.

In order to apply range template based face detection and segmentation in the mobile domain, mobile devices must

be capable of taking such range images, e. g. with stereo cameras. Some current smartphones already contain such cameras mounted on the back side – for more convenient usage they would be required to contain stereo cameras on the front side too. When using stereo cameras, a further prerequisite to successfully performing a range based face unlock in the mobile domain are computationally fast and robust stereo-to-range algorithms that generate range images with only a small amount of erroneous areas. As many existing stereo vision algorithms are either computationally too intensive or deliver inadequate results when applied on data recorded with typical mobile device quality, there is a strong need for improved stereo vision algorithms applicable in the mobile domain as necessary groundwork to successful mobile device stereo vision face unlock.

6. ACKNOWLEDGMENTS

This work has been carried out within the scope of *u'smile*, the Josef Ressel Center for User-Friendly Secure Mobile Environments. We gratefully acknowledge funding and support by the Christian Doppler Gesellschaft, A1 Telekom Austria AG, Drei-Banken-EDV GmbH, LG Nexera Business Solutions AG, and NXP Semiconductors Austria GmbH.

7. REFERENCES

[1] V. Blanz, S. Romdhani, and T. Vetter. Face identification across different poses and illuminations with a 3d morphable model. In *Automatic Face and Gesture Recognition, 2002. Proceedings. Fifth IEEE International Conference on*, pages 192–197, 2002.

[2] V. Blanz and T. Vetter. Face recognition based on fitting a 3D morphable model. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 25(9):1063–1074, 2003.

[3] A. M. Bronstein, M. M. Bronstein, and R. Kimmel. Three-dimensional face recognition. *International Journal of Computer Vision*, 64(1):5–30, Aug. 2005.

[4] J. Canny. A computational approach to edge detection. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, PAMI-8(6):679–698, 1986.

[5] C.-C. Chang and C.-J. Lin. LIBSVM: A library for support vector machines. *ACM Transactions on Intelligent Systems and Technology*, 2(3):1–27, April 2011.

[6] R. D. Findling and R. Mayrhofer. Towards pan shot face unlock: Using biometric face information from different perspectives to unlock mobile devices. *International Journal of Pervasive Computing and Communications*, 9(3), 2013. *(accepted for publication)*.

[7] R. D. Findling and R. Mayrhofer. Towards secure personal device unlock using stereo camera pan shots. In R. Mayrhofer and C. Holzmann, editors, *Second International Workshop on Mobile Computing Platforms and Technologies (MCPT 2013)*, 2013. *(sccepted for publication)*.

[8] E. Hjelmås and B. K. Low. Face detection: A survey. *Computer Vision and Image Understanding*, 83(3):236–274, 2001.

[9] C. W. Hsu, C. C. Chang, and C. J. Lin. *A practical guide to support vector classification*. Department of Computer Science and Information Engineering, National Taiwan University, Taipei, Taiwan, 2003.

[10] D. Huang, C. Shan, M. Ardabilian, Y. Wang, and L. Chen. Local binary patterns and its application to facial image analysis: A survey. *IEEE Transactions on Systems, Man, and Cybernetics, Part C: Applications and Reviews*, 41(6):765–781, Nov. 2011.

[11] M. Kass, A. Witkin, and D. Terzopoulos. Snakes - active contour models. *International Journal Of Computer Vision*, 1(4):321–331, 1987.

[12] K. Konolige. Small vision systems: Hardware and implementation. In Y. Shirai and S. Hirose, editors, *Robotics Research*, pages 203–212. Springer London, 1998.

[13] R. Lienhart and J. Maydt. An extended set of haar-like features for rapid object detection. In *IEEE International Conference on Image Processing 2002*, pages 900–903, 2002.

[14] C. Liu. A bayesian discriminating features method for face detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 25(6):725–740, june 2003.

[15] S. Malassiotis and M. Strintzis. Real-time head tracking and 3d pose estimation from range data. In *Image Processing, 2003. ICIP 2003. Proceedings. 2003 International Conference on*, volume 2, pages II–859–62 vol.3, 2003.

[16] R. Mayrhofer and T. Kaiser. Towards usable authentication on mobile phones: An evaluation of speaker and face recognition on off-the-shelf handsets. In *Proc. IWSSI/SPMU 2012: 4th International Workshop on Security and Privacy in Spontaneous Interaction and Mobile Phone Use, colocated with Pervasive 2012*, June 2012.

[17] A. Mian, M. Bennamoun, and R. Owens. An efficient multimodal 2D-3D hybrid approach to automatic face recognition. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 29(11):1927–1943, 2007.

[18] M. Pamplona Segundo, L. Silva, O. Bellon, and C. Queirolo. Automatic face segmentation and facial landmark detection in range images. *Systems, Man, and Cybernetics, Part B: Cybernetics, IEEE Transactions on*, 40(5):1319–1330, 2010.

[19] P. J. Phillips. Support vector machines applied to face recognition. In M. I. Jordan, M. J. Kearns, and S. A. Solla, editors, *Neural Information Processing Systems*, volume 10, pages 803–809, 1998.

[20] H. A. Rowley, S. Baluja, and T. Kanade. Neural network-based face detection. *IEEE Transactions On Pattern Analysis and Machine intelligence*, 20(1):23–38, 1998.

[21] M. C. Santana, O. Déniz-Suárez, L. Antón-Canalís, and J. Lorenzo-Navarro. Face and facial feature detection evaluation - performance evaluation of public domain haar detectors for face and facial feature detection. In A. Ranchordas and H. Araújo, editors, *International Joint Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications.*, volume 2, pages 167–172, 2008.

[22] M. C. Santana, O. Déniz-Suárez, D. Hernández-Sosa, and J. Lorenzo. A comparison of face and facial feature detectors based on the viola-jones general

object detection framework. *Machine Vision and Applications*, 22(3):481–494, 2011.

[23] H. Schneiderman and T. Kanade. Object detection using the statistics of parts. *International Journal of Computer Vision*, 56(3):151–177, Feb. 2004.

[24] K. Sung and T. Poggio. Example-based learning for view-based human face detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 20:39–51, 1998.

[25] P. Tresadern, T. Cootes, N. Poh, P. Matejka, A. Hadid, C. LeÌĄvy, C. McCool, and S. Marcel. Mobile biometrics: Combined face and voice verification for a mobile platform. *IEEE Pervasive Computing*, 12(1):79–87, 2013.

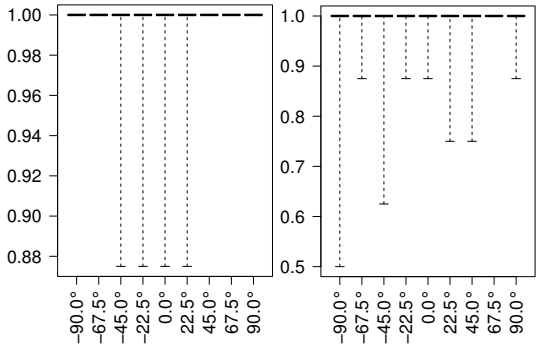
[26] M. Turk and A. Pentland. Eigenfaces for recognition. *Cognitive Neuroscience*, 3(1):71–86, Jan. 1991.

[27] P. Viola and M. Jones. Rapid object detection using a boosted cascade of simple features. *Proceedings of the 2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 1:511–518, 2001.

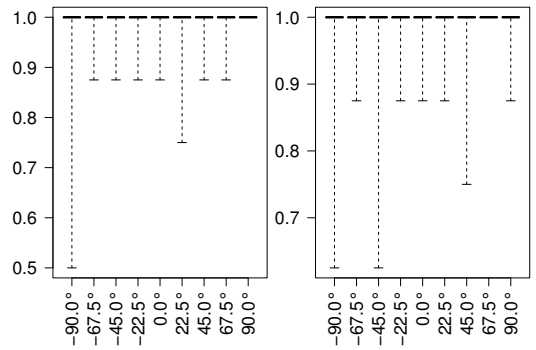
[28] C. Xu and J. Prince. Snakes, shapes, and gradient vector flow. *Image Processing, IEEE Transactions on*, 7(3):359–369, 1998.

[29] C. Xu, J. Yezzi, A., and J. Prince. On the relationship between parametric and geometric active contours. In *Conference on Signals, Systems and Computers, 2000. Conference Record of the Thirty-Fourth Asilomar*, volume 1, pages 483–489 vol.1, 2000.

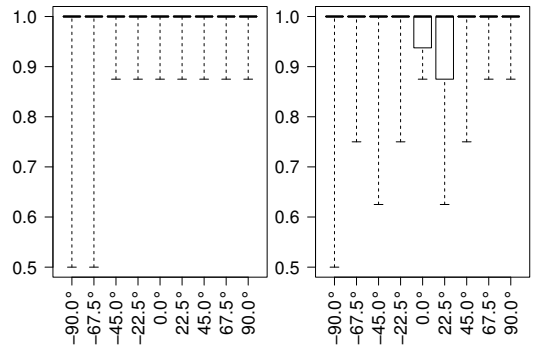
[30] M.-H. Yang, D. Kriegman, and N. Ahuja. Detecting faces in images: a survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24(1):34–58, Jan. 2002.



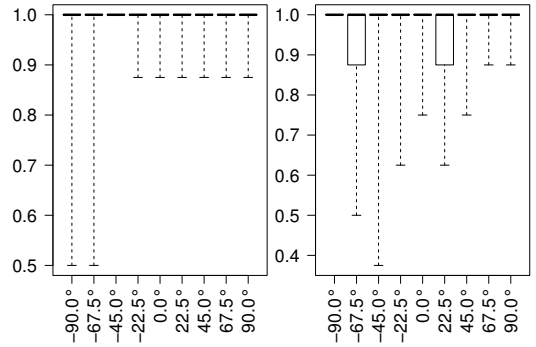
(a) Background removal, without GVF snake segmentation



(b) Background removal, with GVF snake segmentation



(c) Background removal, without GVF snake segmentation



(d) Background removal, with GVF snake segmentation

Figure 13: Boxplot showing true positive face recognition results separated for perspectives, using segmented faces in color (left) and range (right).