Canonical Face Depth Map: A Robust 3D Representation for Face Verification

Dirk Colbry and George Stockman
Department of Computer Science and Engineering, Michigan State University
East Lansing, Michigan 48824-1226
{colbrydi, stockman}@cse.msu.edu

Abstract

The Canonical Face Depth Map (CFDM) is a standardized representation for storing and manipulating 3D data from human faces. Our algorithm automates the process of transforming a 3D face scan into its canonical representation, eliminating the need for hand-labeled anchor points. The presented algorithm is designed to be a robust, fully automatic preprocessor for any 3D face recognition algorithm. The experimental results presented here demonstrate that our CFDM is robust to noise and occlusion, and we show that using such a canonical representation can improve the efficiency of face recognition algorithms and reduce memory requirements. Producing the CFDM takes, on average, 0.85 seconds for 320 x 240 pixel scans, and 3.8 seconds for 640 x 480 pixel scans (using a dual AMD Opteron 275, 2.2GHz, with 2MB Cache, and 1GIG RAM). The CFDM enables both 2D and 3D image processing methods - such as convolution and PCA - to be readily used for feature localization and face recognition.

1. Introduction

We have developed a robust representation for storing 3D data gathered from human faces, which we call the Canonical Face Depth Map (CFDM). The CFDM stores registered 3D face data as a depth map and uses face symmetry and surface model fitting to establish a face-based coordinate system. Converting 3D face data into the CFDM allows scans from different people and different times to be compared quickly and easily, and the normalized representation allows for large-scale data comparisons, such as creating an "average" face that combines features from every face scan in the database.

The work discussed in this paper considers points in 3D space ($p \in \mathbb{R}^3$), where p is a triple with x, y, z coordinates ($x, y, z \in \mathbb{R}$). Each point also has an r, g, b value, which is included here to ease visualization but is not currently used for calculating the canonical representation. $\mathcal{R}$ is the set of all $\mathbb{R}^3 \rightarrow \mathbb{R}^3$ rigid transformations. An object in $\mathbb{R}^3$ can be defined by a relationship of the form $f : \mathbb{R}^3 \rightarrow \mathbb{R}$, where a point p is a part of object s if the following surface relationship holds:

$$s \equiv \{p | f(p) = 0\} \tag{1}$$

This definition could represent any set of points in $\mathbb{R}^3$; however, for this paper it will be assumed that the set of points falls on a surface $s \in S$. Two surfaces (s_1, s_2 with relationships f_1, f_2) are equivalent if there exists a rigid transformation $R \in \mathcal{R}$ such that for every point in s_1 the transformed point is also in s_2:

$$s_1 \sim s_2 \text{ if } \exists R_1 \in \mathcal{R} | \forall p \in s_1, f_2(R_1(p)) = 0 \tag{2}$$

When applied to faces, this equivalence relationship will break down in real world situations. For example, faces are not rigid objects and sensors do not provide a continuous ideal surface. 3D scanners provide a 2.5D discrete depth map estimation of the ideal face surface and may have sampling errors, noise or holes in the data.

We developed a surface alignment algorithm to determine the similarity between two 3D face scans [5]. This approach uses a two step algorithm that attempts to minimize the differences between two face surfaces by aligning the region of the face around the nose and eyes, which changes very little with expression. The first step coarsely aligns the face using estimates of key anchor points such as the inside corners of the eyes and the nose tip. The second step refines the coarse estimation by using the Iterative Closest Point (ICP) algorithm [2, 3]. The ICP algorithm samples control points on one scan and finds the closest points on the surface of the second scan. The control points are sampled around the nose and eyes to minimize the effects of expression, and the highest errors are trimmed to reduce the effects of noise and holes [4].

Once the faces are aligned, a surface alignment measure (SAM) is used to determine the closeness of the fit. The

1-4244-1180-7/07/$25.00 ©2007 IEEE

SAM is a measurement of the root mean squared distance between a set of points on one face and the surface of the second face. The SAM score has been shown to be an effective and robust measure of face similarity [5] for face verification. However, the SAM is not practical for recognition tasks, where each face in the database needs to be aligned to every other face in the database in order to have an effective comparison. Even though the SAM calculation is fast, it would take considerable time in a large database.

To address this concern, we have developed a canonical face-based coordinate system, where the goal is to find a function $f^c|S \rightarrow \mathcal{R}$ such that $\forall s_1, s_2 \in S$ where $s_1 \sim s_2$ with rigid transform $R \in \mathcal{R}$ gives:

$$R = f^c(s_1) f^c(s_2)^{-1} \tag{3}$$

A surface $s_c \in S$ is in its canonical representation if $f^c(s_c) = I$ (see Figure 1).

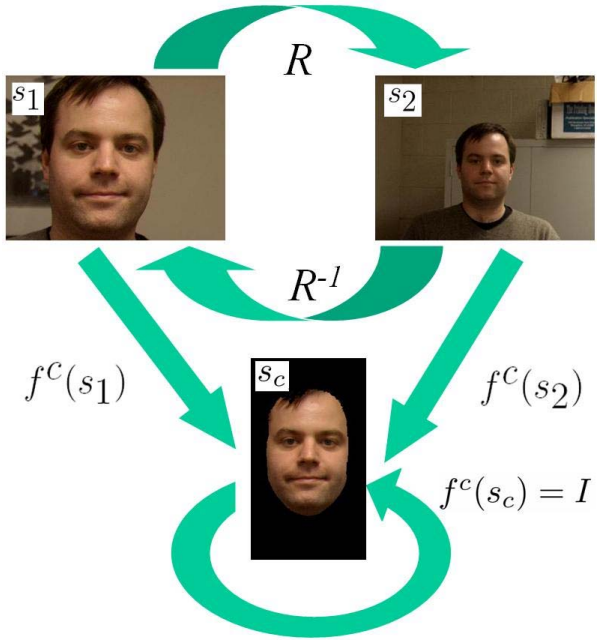


Figure 1. Defining a canonical face function.

2. Development of the CFDM

The canonical representation (CFDM) described in this paper has two main components: a face-based coordinate system and a normalized feature vector based on an orthogonal depth map. The face-based coordinate system developed here is similar to the one presented by BenAbdelkader and Griffin [1]. However, their approach relies on seven manually selected anchor points to properly determine the location of the coordinate system. In contrast, we use ICP to generate a robust bisection of the face and use high sampling rates to develop a good representation of the mid-line plane. This is similar to the approach by Malassiotis and Strintzis [8]; however, they assumed that the nose is always the closest point to the camera, which does not allow for variations in pose.

The second component of the CFDM is a consistent feature vector that can be used to apply space transformation algorithms such as PCA. The CFDM has a similar motivation to the mesh model presented by Xu *et al.* [11], which uses the nose tip as the origin and then conducts a rigid, triangular re-sampling of the data that can be done at different resolutions. PCA is also used in the base algorithm of the Face Recognition Grand Challenge (FRGC) [9], which uses manually selected points to normalize the face into a defined coordinate system.

2.1. Face-Based Coordinate System

This section explores methods for identifying a face-based coordinate system that can account for six degrees of freedom: roll, pitch, yaw and the origin (x_0, y_0, z_0). Having a well defined face-based coordinate system makes it simple to transform data into whatever format or pose an algorithm requires. The goal of creating a face-based coordinate system is to establish a representation that is easily calculated and repeatable for the same face. In addition, face-based coordinate systems should be robust to noise in the data, and it should be possible to estimate the face-based coordinates even when some of the face is occluded or missing.

The set of all possible canonical functions is infinite. So, our goal is to find a canonical function that is easily and robustly found on many different faces. This canonical representation should work in the same way for many different faces and should be robust to expression changes, pose variations and data errors.

For 3D faces, the mid-line plane can be calculated robustly by matching a face scan with a mirror image of itself using the face surface alignment algorithm described in [5]. This algorithm is shown to handle pose variations of up to 10 degrees in roll and pitch, and up to 30 degrees in yaw. For this paper, a mirror image of an object is formally defined as follows:

$$P_M = M \cdot P \tag{4}$$

where P represents the original 3D points in the scan, P_M indicates the mirror points, and M is the mirror transform:

$$M = \begin{bmatrix} -1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

The mirror image is a left-hand projection of the original scan. The process for aligning the original and the mirror scans is the same as for aligning any two faces and uses the

alignment algorithm described in [5]. The resulting transformation between the two scans is defined as the mirror transform and is denoted by T_m.

With the mirror transform, the mid-line plane can be easily calculated. First, an arbitrary point p is chosen. This point does not have to lie on either scan but must not lie on the mid-line plane. Using this point, the surface normal $\vec{N}$ of the mid-line plane is calculated as:

$$p_m = T_m M \cdot p \tag{5}$$

$$\vec{N} = \frac{p_m - p}{|p_m - p|} \tag{6}$$

If a, b, and c are the x, y, and z components of the surface normal and d is the distance along the surface normal from the origin to the plane, then the following equation defines the plane:

$$ax + by + cz + d = 0 \tag{7}$$

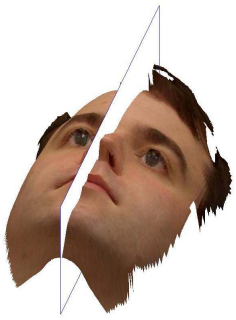
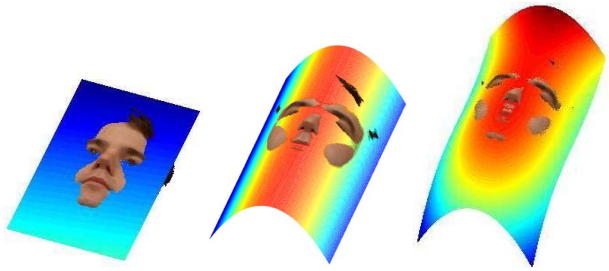


Figure 2. Example of the mid-line plane relative to a face.

The mid-line plane (shown in Figure 2) can be accurately estimated using the plane of symmetry of the face and the surface alignment algorithm. The mid-line plane provides two of the six degrees of freedom and the origin can be estimated by selecting a robust point, such as the tip of the nose. This accounts for five of the six degrees of freedom, yet the pitch (tilt) of the head is still problematic.

To calculate pitch, a series of experiments was conducted on a database of 330 3D scans obtained from Michigan State University [5]. Each scan was fit to a set of parametric surfaces along the plane of symmetry, as shown in Figure 3.

Table 1 shows the pose variations after aligning the canonical faces together and calculating the pose difference. Because these measurements represent the pose variation between scans, they should be considered conservative and the actual results may be half the values in Table 1. Also, a few of the scans did not correctly fit to the surfaces, causing large outliers in the data. Figure 4 shows that these outliers are not significant, and having outliers in the face verification task is not a problem because a subject can always be rescanned. However, outliers are not as easy to determine



(a) Plane (b) Parabolic Cylinder (c) Quadratic

Figure 3. Parametric surface fitting using forced symmetrical surface fitting. **(a)** $ax + by + c$ **(b)** $ax^2 + bx + cy + d$ **(c)** $z = ax^3 + bx^2y + cxy^2 + dy^3 + ex^2 + fxy + gy^2 + hx + iy + j$

Table 1. Improvement of mid-line normalization over database roll, pitch, and yaw differences. In the first row, the original roll, pitch, yaw and distance statistics represent the variations between different scans of the same subjects within the original database. Each subsequent row represents the variation after the different normalization techniques. The Plane, Parabolic Cylinder, Quadratic, and Average face fitting are all applied after the baseline symmetry is established.

	Roll	Pitch	Yaw	Distance
Original	$0.36°$	$1.62°$	$1.52°$	$63.69mm$
std dev	$\pm 0.49°$	$\pm 2.25°$	$\pm 1.99°$	$\pm 66.08mm$
Symmetry	$0.01°$	$0.00°$	$0.00°$	$3.10mm$
std dev	$\pm 0.53°$	$\pm 3.44°$	$\pm 0.56°$	$\pm 2.14mm$
Plane	$0.01°$	$0.10°$	$0.02°$	$5.12mm$
std dev	$\pm 0.52°$	$\pm 3.15°$	$\pm 0.54°$	$\pm 2.48mm$
Parabola	$0.02°$	$0.10°$	$0.00°$	$2.06mm$
std dev	$\pm 0.53°$	$\pm 2.59°$	$\pm 0.50°$	$\pm 3.33mm$
Quad	$0.00°$	$0.01°$	$0.01°$	$12.67mm$
std dev	$\pm 0.53°$	$\pm 3.42°$	$\pm 0.51°$	$\pm 5.31mm$
Average	$0.01°$	$0.01°$	$0.00°$	$2.90mm$
std dev	$\pm 0.58°$	$\pm 2.01°$	$\pm 0.79°$	$\pm 7.81mm$

in a face recognition task and should be detected before attempting recognition on a large database.

The parabolic cylinder seems to be the best surface fitting approach, given our goal to minimize the variance or standard deviations in the results in Table 1 and Figure 4. As a comparison, the results for surface fitting and pitch normalization are also given for an average face model, generated by averaging all faces in the database. Figure 4 suggests that fitting a surface to an average face is the most reliable pitch correction method, but this approach breaks down when it encounters a face that is significantly larger or smaller than the average. (Note that our average face is generated from the same data set, so it is probably biased toward this data.) Overall, the approaches shown in Figure 4 are accurate for changes of up to $\pm 2°$ in the pitch axis, which is much larger than the $\pm 0.5°$ for the other two axes.

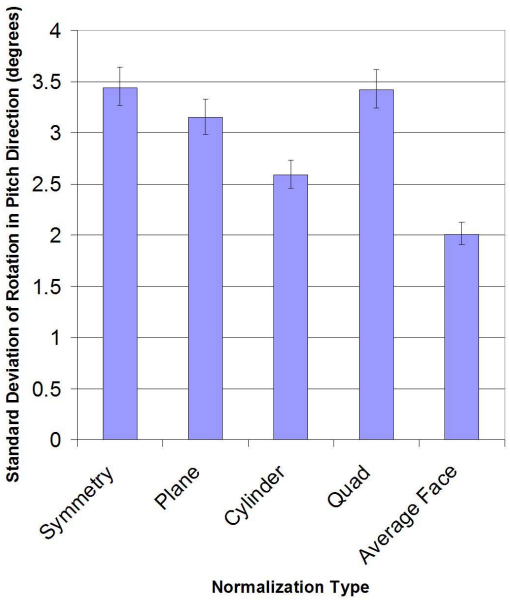
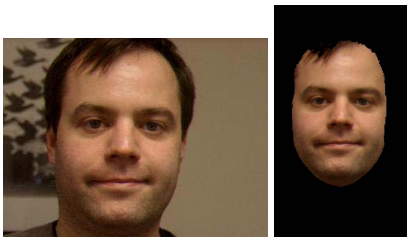


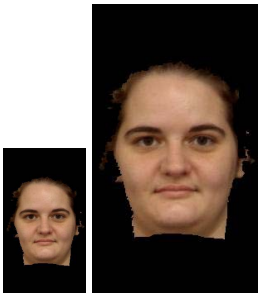
Figure 4. Comparison of pitch over different fitting techniques.



(a) Original Scan (b) CFDM

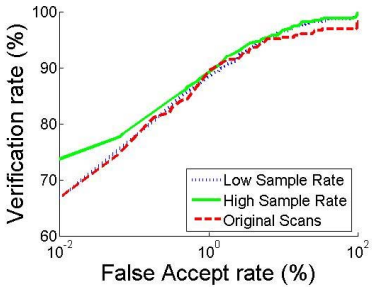
(c) Original Scan (d) CFDM

Figure 5. (a) and (c) are scans with different camera offset and resolution, (b) and (d) show how using a depth map normalizes the resolution.



(a) 200 × 350

(b) 400 × 700



(c) ROC Performance

Figure 6. Depth maps sampled at two different resolutions. The higher resolution data improves ICP performace by allowing for incremental changes during each iteration of the algorithm.

2.2. Normalized Feature Vector

Once a face-based coordinate system is calculated, the data are re-sampled using the orthogonal depth map, and this re-sampling is at the same scale and dimension regardless of the input scan resolution. For example, the images in Figure 5 are at different camera offsets. After normalizing the scans into the CFDM representation (Figures 5b and 5d), they can be directly compared. In contrast, 2D images have an unknown scale and must be normalized based on some common point distance, such as the distance between the eyes. However, different people have different between-eye distances, so the effect is a comparison of faces that must be scaled up or down to match each other.

Another advantage of using a Canonical Face Depth Map is that the depth map can be sampled at any resolution (see Figure 6a and b). Lower resolutions can be used to speed up some algorithms, while higher resolutions can be used to increase accuracy. For example, improved accuracy in the ICP-based surface alignment algorithm can be achieved by super sampling the CFDM and creating a highly dense target scan in which ICP can operate. Figure 6c shows two ROCs for face depth maps sampled at different scales. A database of 330 scans was used with the SAM as a simple matching score. The higher sampling rate increases the accuracy of the algorithm.

Because the CFDM uses an orthogonal projection, it is possible to reduce the memory requirements for storage of the file. One goal for the CFDM is to be able to store it on a smart card. A standard smart card has 512 kilobytes of memory [7]. The Minolta VIVID910 scanner has two scanning modes: fine mode (640 x 480) and fast mode (320 x 240). The database collected by MSU uses the fast mode while the database collected for the FRGC uses the fine mode. Both data types can be processed by the CFDM algorithm. A comparison of the computation time and memory requirements are shown in Table 2. A 320×240 raw scan takes up 1.58MB of memory, while a compressed 400×700 CFDM file only requires 0.24MB of space. At this resolution it is possible to store many scans on a single memory card, allowing for non-rigid variations, such as smiles.

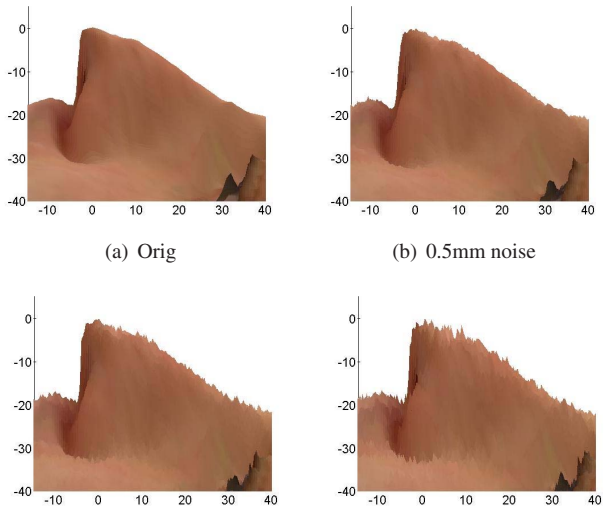
3. Robustness Experiments

The experiments described in this section test the robustness of the CFDM to noise and holes in the data. In the first experiment, Gaussian noise was added to the z component of the data set. The noise was generated in MATLAB

Table 2. Comparison of CFDM generation from both the fine mode and fast mode of the Minolta Vivid 910 scanner.

	320 x 240	640 x 480
Scan Time	$\approx$ 2 sec	$\approx$ 4 sec
Curvature Calculation	0.42 sec	2.48 sec
Symmetry alignment (ICP)	0.23 sec	0.83 sec
Orthogonal Projection	0.22 sec	0.47 sec
Total CFDM Generation	0.85 sec	3.8 sec
Raw File (Minolta cdm)	1.58 MB	3.61 MB
Compressed Raw File	0.88 MB	2.1 MB
400x700 CFDM file (ppm)	1.68 MB	
400x700 Compressed CFDM	0.24 MB	

using a normal distribution and then multiplied by one of three noise amplitudes (0.5mm, 1.0mm and 1.5mm). This range was chosen because the Minolta scanner accuracy is about 0.5mm and the SAM verification threshold described in [10] is about 1.0mm. Smaller errors would not be significant, and any higher level of noise would result in inaccurate matching. Figure 7 shows some examples of these noise levels on a typical scan. The results for the noise experiments in Table 3 show that the mean roll, pitch and yaw stay about zero degrees, but as the amount of noise increases the standard deviation of these values also increases. Fortunately this increase is very small, indicating that the CFDM is robust to Gaussian noise.



(a) Orig (b) 0.5mm noise

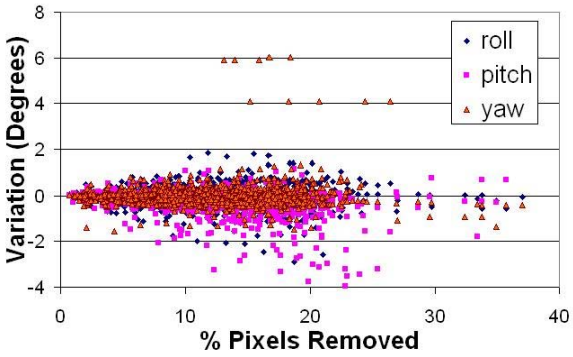
(c) 1.0mm noise (d) 1.5mm noise

Figure 7. Example noise on the surface of the face.

The second experiment was designed to test the CFDM against holes in the data. These trials removed a Gaussian hole from the scan; this hole covers a circular area of approximately 5mm radius taken from the region of the scan used by the surface alignment algorithm. Each hole removes a different number of pixels depending on the scale of the scan and whether the holes overlap with previous holes in the data. This process is repeated for a total of ten iterations, with each iteration adding another hole to the scan. Figure 8b shows an example of holes added to the scan. Figure 8a displays the percentage of valid pixels removed by the holes from the fitting region (around the eyes and nose) for 1030 scans (103 scans with 10 iterations each). The y axis shows the difference in the roll, pitch and yaw between the CFDM face processed without the holes and the CFDM face processed with the holes.

Table 3. Error with respect to noise

	Roll	Pitch	Yaw	Distance
Parabola	0.02°	0.10°	0.00°	$2.06mm$
std dev	$\pm 0.53^\circ$	$\pm 2.59^\circ$	$\pm 0.50^\circ$	$\pm 3.33mm$
$0.5mm$	0.01°	0.06°	0.01°	$2.57mm$
std dev	$\pm 0.56^\circ$	$\pm 2.37^\circ$	$\pm 0.89^\circ$	$\pm 4.84mm$
$1.0mm$	0.00°	0.07°	0.01°	$3.80mm$
std dev	$\pm 0.62^\circ$	$\pm 2.49^\circ$	$\pm 1.00^\circ$	$\pm 5.99mm$
$1.5mm$	0.00°	0.06°	0.01°	$5.32mm$
std dev	$\pm 0.67^\circ$	$\pm 2.32^\circ$	$\pm 1.03^\circ$	$\pm 6.24mm$



(a) Rotation Errors

(b) Example Holes

Figure 8. Effect of holes on the CFDM angles.

When less than 10% of the data is removed, the angle variation is no more than two degrees. As the number of holes increases, there are larger errors. In most cases, the angle difference is still well under 2 degrees of variation. However, there are cases where the errors spike, normally

due to holes placed in key locations that interfere with anchor point detection or that cause large asymmetries in the data.

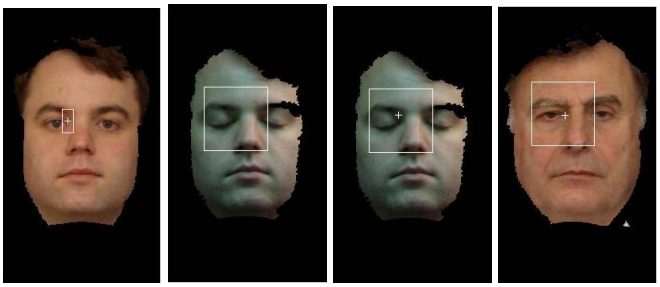
4. Applications

The output of the Canonical Face Depth Map algorithm is trivially transformed into a data structure with the same format as the original face scan. Maintaining the data format allows the Canonical Face Depth Map to be used as a preprocessor to other algorithms. This section presents two common image processing algorithms, correlation and PCA, and applies the CFDM developed in this paper as a preprocessor for these algorithms. Correlation and PCA were chosen because they provide a good baseline, have potential for effective identification, and perform best if the input data are normalized and of fixed size resolution.

4.1. Correlation

For each window region, the points are normalized to their average depth. The correlation algorithm is similar to traditional correlation and minimizes the L2-norm, but takes into account missing data (due to flags) and trims errors in the z data to make the algorithm more robust to spike noise. Several experiments were conducted to compare feature regions on a probe scan to those on a target scan; these experiments used the following procedure:

1. Identify anchor points on the target scan.
2. Choose a region around each point as an independent mask. (See Figure 9a)
3. Find the anchor points on the probe scan.
4. Choose a larger search region about each point on the probe scan. (See Figure 9b)
5. Convolve the masked region in the search region for each corresponding point. (See Figure 9c and d)
6. Record the minimum correlation score and the row and column of the points.



(a) Mask Region (b) Search Region (c) Point found on subject (d) Point found on different subject

Figure 9. Example of the correlation point algorithm.

The correlation algorithm was tested on the depth channel, shape index, and color channel of a data set of 330 scans from 111 subjects. All scans were normalized using the CFDM algorithm. The L2-norm (average root mean squared error over the entire correlation window) for each anchor point (eye corners, nose tip, mouth corners, bridge of the nose, and chin) was used as a similarity score. The best performance was found using a mask around the bridge of the nose and applying it to the shape index. The shape index is a ratio of the maximum and minimum curvature values at each point, as defined in [6].

4.2. PCA Analysis

In addition to providing a data set, the Face Recognition Grand Challenge (FRGC) also provides a programming interface called the Biometrics Experimental Environment (BEE). The BEE system gives FRGC competitors a common interface for running independently-evaluated algorithms. It also provides an implementation of the CMU PCA matching algorithm to use for baseline comparison. PCA is a well-established 2D color face matching method. It is not state-of-the-art, but has been shown to perform acceptably when there are minor variations in pose and lighting.

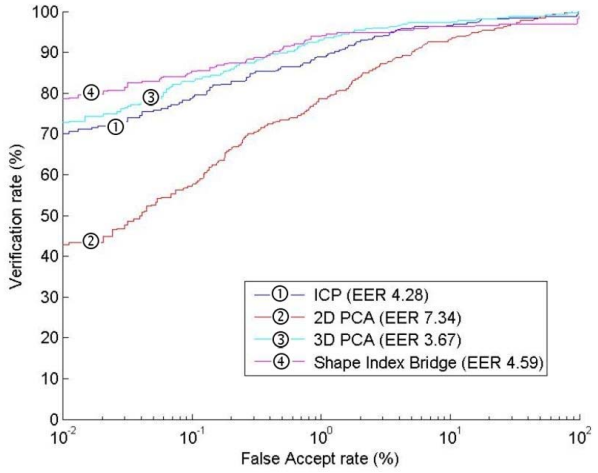


Figure 10. Comparing ROC curves on 330 mostly frontal scans using ICP (SAM), PCA and Convolution.

Figure 10 shows four different recognition algorithms for comparison. The CFDM-corrected PCA algorithm did not perform as well as the others, mostly because there were some large lighting variations in the FRGC data set. The ICP-based SAM score did not use the CFDM for alignment yet it performed well without the need for any training data. The normalized 3D PCA algorithm performs even better than the SAM, which is expected because the PCA algorithm is trained on the same data set.

5. Concluding Discussion

The 3D Canonical Face Depth Map automatically and robustly removes the effects of pose and scaling variations in 3D depth maps (see examples in Figure 11). Producing the CFDM takes, on average, 0.85 seconds for 320 x 240 pixel scans, and 3.8 seconds for 640 x 480 pixel scans (using a dual AMD Opteron 275, 2.2GHz, with 2MB Cache, and 1GIG RAM). The CFDM enables both 2D and 3D image processing methods - such as convolution and PCA - to be readily used for feature localization and face recognition. Experiments demonstrated that the CFDM is robust to both Guassian noise and holes in the data. The CFDM can improve the accuracy of surface alignment algorithms that use the iterative closest point algorithm, and significantly reduce the memory required to store a 3D face model. The canonical representation is also an effective preprocessor for both 2D and 3D face processing algorithms, and we have shown how the fully automatic CFDM can be used with both convolution and PCA.

References

[1] C. BenAbdelkader and P. A. Griffin. Comparing and combining depth and texture cues for face recognition. *Image and Vision Computing*, 23:339–352, 2005.

[2] P. J. Besl and N. D. McKay. A method for registration of 3-D shapes. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 14(2):239–256, 1992.

[3] Y. Chen and G. Medioni. Object modeling by registration of multiple range images. *International Journal of Image and Vision Computing*, 10(3):145–155, 1992.

[4] D. Chetverikov, D. Svirko, D. Stepanov, and P. Kresek. The trimmed iterative closest point algorithm. In *Proceedings of the 16th International Conference on Pattern Recognitionn*, volume 3, pages 545–548, Québec City, Canada, 2002.

[5] D. Colbry and G. Stockman. Identity verification via the 3DID face alignment system. *Proceedings of IEEE Workshop on Applications of Computer Vision (WACV)*, 2007.

[6] C. Dorai and A. K. Jain. Cosmos - a representation scheme for 3D free-form objects. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 19(10):1115–1130, 1997.

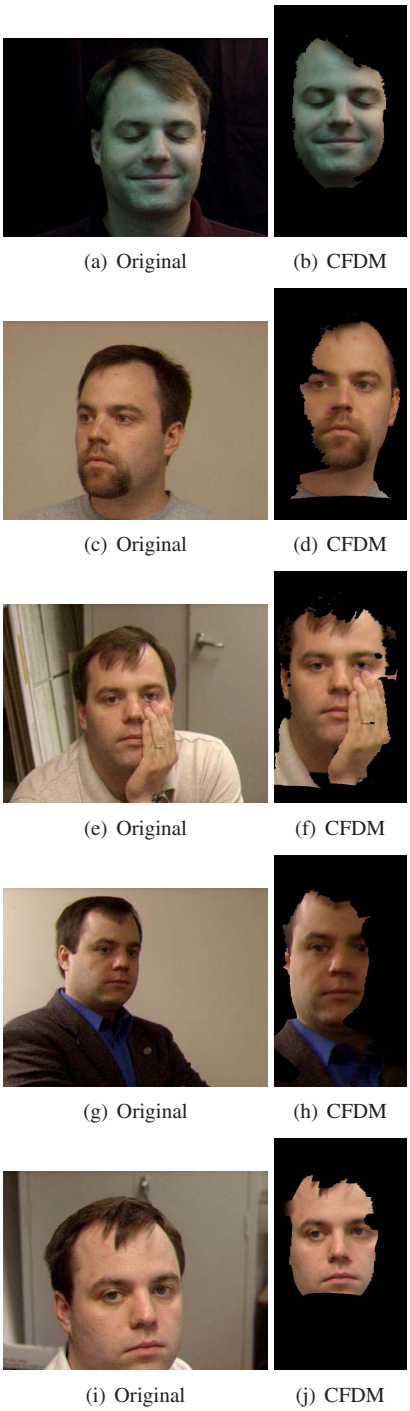
[7] Giesecke and Devrient. Big SIM cards from 512 kb to 512 mb. Technical report, http://www.gi-de.com/.

[8] S. Malassiotis and M. G. Strintzis. Robust face recognition using 2D and 3D data: Pose and illumination compensation. *Pattern Recognition*, 38:2537–2548, 2005.

[9] P. J. Phillips, P. J. Flynn, T. Scruggs, K. W. Bowyer, J. Chang, K. Hoffman, J. Marques, J. Min, and W. Worek. Overview of the Face Recognition Grand Challenge. In *Proceedings IEEE Conference on Computer Vision and Pattern Recognition*, 2005.

[10] G. Stockman, J. Payne, J. Sadler, and D. Colbry. Error analysis of sensor input variations in a 3D face surface matching system. *Sensor Review Journal*, 26(2):116–121, 2006.

[11] C. Xu, Y. Wang, T. Tan, and L. Quan. Face recognition based on 3D mesh models. In *Proceedings of SPIE - The International Society for Optical Engineering.*, Denver, Colorado, 2004.



(a) Original (b) CFDM

(c) Original (d) CFDM

(e) Original (f) CFDM

(g) Original (h) CFDM

(i) Original (j) CFDM

Figure 11. Examples of applying the CFDM algorithm to scans with different levels of difficulty.