2010 IEEE International Conference on Robotics and Automation
Anchorage Convention District
May 3-8, 2010, Anchorage, Alaska, USA

Registration of 3D Facial Surfaces Using Covariance Matrix Pyramids

Moritz Kaiser*, Bogdan Kwolek*, Christoph Staub†, and Gerhard Rigoll*

*Institute for Human-Machine
Communication
Technische Universität München
Arcisstr. 21, 80333 Munich, Germany
{moritz.kaiser,bkwolek,rigoll}@tum.de

†Robotics and Embedded Systems,
Department of Informatics
Technische Universität München
Arcisstr. 21, 80333 Munich, Germany
staub@in.tum.de

***Abstract*— Registration of 3D facial surfaces means establishing point-to-point correspondence between two 3D facial surfaces. Difficulties typical for the registration of 3D facial surfaces are varying illumination, pose or viewpoint changes, varying facial expressions, and different appearance of individuals. In this work we propose to use a covariance matrix as descriptor for the neighborhood of a salient point in a face. It encodes the variance of the channels, such as red, green, blue, depth, etc., their correlations with each other, and spatial layout, while filtering out the influence of the disturbing effects mentioned above. A pyramidal approach is applied where first the location of a corresponding point is computed roughly and then the position is gradually refined. The method does not require any training. Particle Swarm Optimization makes the search for corresponding points more efficient. Results with a challenging dataset confirm that the approach works greatly for a variety of disturbing effects.**

I. INTRODUCTION

Registration of 3D surfaces is the process of establishing point-to-point correspondence between two surfaces. There are many applications in computer graphics and robotics that stand or fall on the exact non-rigid registration of 3D facial surfaces. Examples include detection and tracking of facial expressions [1] or 3D face pose [2], memorizing faces [3], creating face models [4], or facial animation [5].

In [4], Blanz and Vetter proposed to apply the Kanade-Lucas-Tomasi (KLT) feature tracker [6] for the registration of 3D facial surfaces. Later works, such as [7], [8], and [1] modified the way of mapping the 3D surface into the 2D plane but also applied an optical flow method in combination with a Gaussian image pyramid.

However, a pyramidal feature tracker often fails in practice. In contrast to registration of images of the same scene or tracking problems, where up to several frames per second are obtained, registration of 3D facial surfaces poses a unique challenge.

- The two individuals might appear very different (facial hair, skin color, sex, etc.).
- The two individuals might have a different facial expression.
- The illumination under which the two individuals have been recorded may vary considerably.
- Different pose or position of the face might cause large distances between corresponding points requiring many pyramid levels in a Gaussian image pyramid.

At coarse levels in a Gaussian image pyramid, relatively huge image regions are described by only one weighted mean value. If illumination and appearance of the two considered individuals vary, the weighted mean might be meaningless.

In this work we show that if at coarse levels of an image pyramid the structure of an image region instead of only the weighted mean is regarded, the registration accuracy can be greatly improved. An elegant way to describe the structure of an image region is to consider the covariance matrix of this region, which contains the variance of the channels (red, green, blue, depth, etc.) and, even more important, the covariance between those channels. Finding a corresponding point with a similar covariance matrix in another image is a constrained nonlinear optimization problem that we solve with constrained Particle Swarm Optimization (PSO).

The approach has many practical properties regarding registration of 3D faces. A covariance matrix is a natural and simple way to fuse conventional channels, such as red, green, blue and new types of channels, such as depth, resulting in a strong and robust indicator for point-to-point correspondence. Variations in illumination or appearance of the faces that change only the mean do not affect the covariance matrix. Variations in pose or viewpoint also do not change the covariance matrix considerably. Noise corrupting image regions is largely filtered out. Moreover, a Gaussian pyramid is not needed any more. For coarser pyramid levels only the size of the region for which the covariance matrix is computed is increased.

The paper is organized as follows. In Sec. II the covariance matrix based region descriptor is defined. Our pyramidal registration method is described in Sec. III. The results of our registration method are presented in Sec. IV. Section V gives a conclusion and outlines future work.

II. COVARIANCE MATRIX BASED REGION DESCRIPTOR

The covariance matrix based region descriptor was presented in [9] in the context of texture classification. Let $\boldsymbol{I} \in \mathbb{R}^{H \times W \times C}$ be an image with height H, width W, and C channels or components, such as red, green, blue, etc. At each pixel location $\boldsymbol{x} = (x, y)^T$ a component vector $\boldsymbol{h}(\boldsymbol{x}) \in \mathbb{R}^C$ is extracted. Let $\mathcal{N}$ be a squared region of the image and $\boldsymbol{x} \in \mathcal{N}$ all points inside $\mathcal{N}$. The covariance

978-1-4244-5040-4/10/$26.00 ©2010 IEEE

matrix that describes region $\mathcal{N}$ is

$$\boldsymbol{C}_{\mathcal{N}} = \frac{1}{|\mathcal{N}|-1} \sum_{\boldsymbol{x}\in\mathcal{N}} (\boldsymbol{h}(\boldsymbol{x}) - \boldsymbol{\mu})(\boldsymbol{h}(\boldsymbol{x}) - \boldsymbol{\mu})^T, \qquad (1)$$

where $\boldsymbol{\mu} = \frac{1}{|\mathcal{N}|} \sum_{\boldsymbol{x}\in\mathcal{N}} \boldsymbol{h}(\boldsymbol{x})$ denotes the mean vector of the component vectors and $|\mathcal{N}|$ stands for the number of points in region $\mathcal{N}$. A total of $C = 10$ channels was employed:

$$\boldsymbol{h}(\boldsymbol{x}) = \Big(x, y, z(x, y), R(x, y), G(x, y), B(x, y), \\ |\frac{\partial I(x,y)}{\partial x}|, |\frac{\partial I(x,y)}{\partial y}|, |\frac{\partial^2 I(x,y)}{\partial x^2}|, |\frac{\partial^2 I(x,y)}{\partial y^2}|\Big)^T, \qquad (2)$$

where z is the depth, R, G, B are the red, green, blue color values, respectively, and I is the intensity.

The covariance matrix is a well-suited descriptor for the neighborhood of salient points in faces. It is very informative containing information about the variance of the channels, correlations between channels, and spatial layout in the neighborhood. While picking out information that is relevant for correspondence estimation, it largely filters out disturbing effects, such as varying illumination, pose or viewpoint changes, noise caused by varying expressions, and different appearance of individuals.

As distance measure between two covariance matrices we employ the metric for covariance matrices proposed in [10]:

$$\rho(\boldsymbol{C}_1, \boldsymbol{C}_2) = \sqrt{\sum_{i=1}^{C} \ln^2 \lambda_i(\boldsymbol{C}_1, \boldsymbol{C}_2)}\ , \qquad (3)$$

where $\{\lambda_i(\boldsymbol{C}_1, \boldsymbol{C}_2)\}_{i=1\ldots C}$ denote the generalized eigenvalues of $\boldsymbol{C}_1$ and $\boldsymbol{C}_2$. The generalized eigenvalues are defined by $\lambda_i \boldsymbol{C}_1 \boldsymbol{v}_i = \boldsymbol{C}_2 \boldsymbol{v}_i$, with $\boldsymbol{v}_i \neq \boldsymbol{0}$. A generalized eigenvalue problem can be converted into a normal eigenvalue problem: $\boldsymbol{C}_1^{-1} \boldsymbol{C}_2 \boldsymbol{v}_i = \lambda_i \boldsymbol{v}_i$.

III. PYRAMIDAL REGISTRATION METHOD

Our registration method performs the following task with the help of the covariance matrix region descriptor. In a reference face, denoted by $\boldsymbol{I}_{\text{ref}}$, an arbitrary number of salient points is defined either by a corner detector, e.g. [11], or by hand. Those points are found in other face images, i.e., images that should be registered, which are denoted by $\boldsymbol{I}_{\text{reg}}$. First, the 3D point cloud of each face is mapped into the 2D plane in order to obtain the images $\boldsymbol{I}_{\text{ref}}$ and $\boldsymbol{I}_{\text{reg}}$. Then, a simple pyramidal approach is applied. At lower levels, the neighborhood around a salient point for which the covariance matrix is computed and the search region in which a corresponding salient point is searched are huge. At higher levels the neighborhood and the search region become smaller. At each level, a corresponding point is found via a modified version of Particle Swarm Optimization (PSO) and this location is used as starting point at the next level.

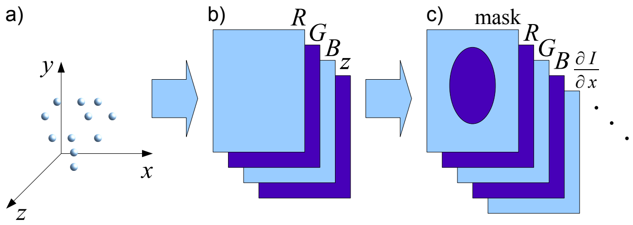

Fig. 1: The 3D facial surface (a) is mapped into the 2D plane (b) and subsequently 10 channels are computed (c).

A. Mapping 3D Facial Surface into 2D Plane

Several databases of 3D facial surfaces exist. They are typically recorded with a 3D scanner whose output is a 3D point cloud. We directly employ the x- and y-coordinates of the 3D points to determine their position in the 2D image $\boldsymbol{I}$. As the x- and y-coordinates are not whole numbers and the points are not equidistant from each other, which results in holes in the image, barycentric interpolation is used to assign a value to each pixel. Let $\boldsymbol{x}_1, \boldsymbol{x}_2, \boldsymbol{x}_3$ be the 2D coordinates of the three vertices of a triangle and $\boldsymbol{x} = (x, y)^T$ a pixel position (whole numbers) inside this triangle. The barycentric coordinates of $\boldsymbol{x}$ are:

$$\begin{aligned} b_1(\boldsymbol{x}) &= A(\boldsymbol{x}, \boldsymbol{x}_2, \boldsymbol{x}_3)/A(\boldsymbol{x}_1, \boldsymbol{x}_2, \boldsymbol{x}_3), \\ b_2(\boldsymbol{x}) &= A(\boldsymbol{x}, \boldsymbol{x}_3, \boldsymbol{x}_1)/A(\boldsymbol{x}_1, \boldsymbol{x}_2, \boldsymbol{x}_3), \qquad (4) \\ b_3(\boldsymbol{x}) &= A(\boldsymbol{x}, \boldsymbol{x}_1, \boldsymbol{x}_2)/A(\boldsymbol{x}_1, \boldsymbol{x}_2, \boldsymbol{x}_3), \end{aligned}$$

where $A(\cdot)$ means area of triangle. The red component of pixel $\boldsymbol{x}$ is

$$R(\boldsymbol{x}) = \sum_{k=1}^{3} b_k(\boldsymbol{x}) R(\boldsymbol{x}_k). \qquad (5)$$

The other components or channels are computed alike. In the resulting image $\boldsymbol{I}$ each pixel has a red, green, blue and z value. We employed a 256×192 image as 2D plane. Subsequently, the 10 channels of Eq. 2 are computed. It is convenient to also store a mask which labels each pixel as foreground (face) or background. The whole process is depicted in Fig. 1.

More sophisticated methods to map 3D surfaces into the 2D plane have been presented in the past. In [8], Least Squares Conformal Mapping (LSCM) was applied. The authors of [1] suggested to use harmonic mapping and in [7] a cost function that minimizes length and area distortion is employed. We tried our method also with LSCM and harmonic mapping but, although computationally much more intensive, the registration accuracy could not be improved. Therefore, the simple but efficient direct projection was used.

B. PSO-Based Correspondence Estimation

Let $\mathcal{N}_{\boldsymbol{x}}$ be the neighborhood of a salient point $\boldsymbol{x}$ in the reference image $\boldsymbol{I}_{\text{ref}}$ and let $\boldsymbol{C}_{\boldsymbol{x}}$ be the covariance matrix computed over this region. A corresponding point $\boldsymbol{y}$ that has a similar covariance matrix $\boldsymbol{C}_{\boldsymbol{y}}$ is searched inside the image to register $\boldsymbol{I}_{\text{reg}}$:

$$\boldsymbol{y} = \arg\min_{\boldsymbol{y}} f(\boldsymbol{y}) \equiv \arg\min_{\boldsymbol{y}} \rho(\boldsymbol{C}_{\boldsymbol{x}}, \boldsymbol{C}_{\boldsymbol{y}}). \qquad (6)$$

It is convenient to search $\boldsymbol{y}$ not in the whole image but only inside a search region $\mathcal{S}$. Note that if $\mathcal{N}$ is huge, e.g. 32×32, the correspondence estimation is robust but not very precise,

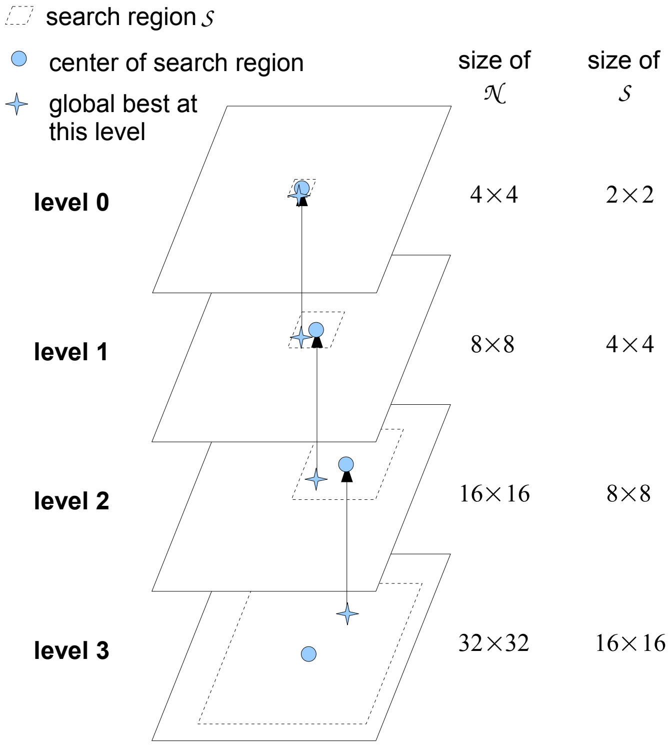

Fig. 2: Covariance matrix pyramid.

while with a small $\mathcal{N}$, e.g. 4×4, the estimate is more accurate but also more susceptible to noise and outliers.

Finding a corresponding position $\boldsymbol{y}$ in $\boldsymbol{I}_{\text{reg}}$ is a constrained nonlinear optimization problem. PSO, first introduced in [12], is a simple technique that has proven to be very fast and efficient for many optimization problems [13]. *Particles* are moving points in multidimensional space (here 2D positions in $\boldsymbol{I}_{\text{reg}}$) that are drawn towards the positions of their own previous best position and the global best. If the nonlinear optimization problem has additional constraints, they can easily be integrated into the PSO [13].

Let N be the number of particles, each with a location $\boldsymbol{y}_n \in \mathbb{R}^2$ and velocity $\boldsymbol{v}_n \in \mathbb{R}^2$. Let $\hat{\boldsymbol{y}}_n$ be the current best position of each particle and $\hat{\boldsymbol{g}}$ be the global best. Position $\boldsymbol{y}_n$ is constrained to lie inside a certain search region $\mathcal{S}$. The algorithm has the following form:

```
    // Initialize each particle's velocity and position, and local and global best
 1: $\boldsymbol{v}_n = \boldsymbol{0}$ and $\boldsymbol{y}_n \in \mathcal{S}$ is chosen randomly
 2: $\hat{\boldsymbol{y}}_n \leftarrow \boldsymbol{y}_n$ and $\hat{\boldsymbol{g}} = \arg\min_{\boldsymbol{y}_n} f(\boldsymbol{y}_n)$
 3: while number of iter. $< M$ do
 4:     for all particles $n$ do
 5:         $\boldsymbol{v}_n \leftarrow \omega\boldsymbol{v}_n + c_1\boldsymbol{r}_1 \circ (\hat{\boldsymbol{y}}_n - \boldsymbol{y}_n) + c_2\boldsymbol{r}_2 \circ (\hat{\boldsymbol{g}} - \boldsymbol{y}_n)$
 6:         $\boldsymbol{y}_n \leftarrow \boldsymbol{y}_n + \boldsymbol{v}_n$        // Update particle location
 7:         if $\boldsymbol{y}_n \in \mathcal{S}$ then
 8:             if $f(\boldsymbol{y}_n) < f(\hat{\boldsymbol{y}}_n)$, $\hat{\boldsymbol{y}}_n \leftarrow \boldsymbol{y}_n$      // local best
 9:             if $f(\boldsymbol{y}_n) < f(\hat{\boldsymbol{g}})$, $\hat{\boldsymbol{g}} \leftarrow \boldsymbol{y}_n$        // global best
10:         end if
11:     end for
12: end while
```

The inertia weight ω and the two constants c_1 and c_2 at line 5 balance the influence of the particle's previous velocity ($\boldsymbol{v}_n$), its local best ($\hat{\boldsymbol{y}}_n$), and the global best ($\hat{\boldsymbol{g}}_n$), respectively. We set $\omega = 0.9$ and $c_1 = c_2 = 2.05$. The vectors $\boldsymbol{r}_1$ and $\boldsymbol{r}_2$ are vectors of random numbers in the range $[0, 1]$ which are generated in each iteration according to a uniform probability distribution. The operator $\circ$ denotes element-wise multiplication. In order to prevent excessive growth of the velocity in line 5 it is multiplied by k: $\boldsymbol{v}_n \leftarrow k\boldsymbol{v}_n$, where

$$k = \frac{2}{|2 - \phi\sqrt{\phi^2 - 4\phi}|}, \quad \phi = c_1 + c_2, \tag{7}$$

as suggested in [14].

C. Covariance Matrix Pyramids

As mentioned at the beginning we have to deal with varying illumination and different appearance and expressions of the individuals, so we cannot expect to find a position with $\boldsymbol{C}_{\boldsymbol{y}} = \boldsymbol{C}_{\boldsymbol{x}}$. If the PSO-based search is directly applied corresponding points far away from the correct location are often detected. There is a tradeoff between choosing a large neighborhood $\mathcal{N}$ over which the covariance matrix is computed, which allows us to find the rough location robustly, and a small $\mathcal{N}$, that allows us to find the position of the salient point more accurately but also unreliably. Therefore, a pyramidal approach is applied. At lower levels, $\mathcal{N}$ and the search region $\mathcal{S}$ are huge. Hence, the location of the salient point in the face is roughly determined (eye, eye brow, mouth, nose, etc.). At the next higher level, (i) the search region is centered around the location found at the lower level, (ii) the size of the search region is decreased, and (iii) the size of the neighborhood is also decreased to make the solution more precise. The constrained PSO combined with the pyramidal approach results in the following algorithm:

```
$\hat{\boldsymbol{g}}_L = \boldsymbol{x}$              // Initialize center for search region
for level $l = L - 1$ to $0$ do
    // Initialize search region, neighborhood, particle's velocity and position, and local and global best
    $|\mathcal{S}_l| = 2^{l+1} \times 2^{l+1}$ centered at $\hat{\boldsymbol{g}}_{l+1}$
    $|\mathcal{N}_l| = 2^{l+2} \times 2^{l+2}$
    $\boldsymbol{v}_n = \boldsymbol{0}$ and $\boldsymbol{y}_n \leftarrow$ Gaussian probability distribution
    $\hat{\boldsymbol{y}}_n \leftarrow \boldsymbol{y}_n$ and $\hat{\boldsymbol{g}} = \arg\min_{\boldsymbol{y}_n} f(\boldsymbol{y}_n)$
    for $M$ iterations do
        for all particles $n$ do
            $\boldsymbol{v}_n \leftarrow k \cdot (\omega\boldsymbol{v}_n + c_1\boldsymbol{r}_1 \circ (\hat{\boldsymbol{y}}_n - \boldsymbol{y}_n)$
                                  $+ c_2\boldsymbol{r}_2 \circ (\hat{\boldsymbol{g}} - \boldsymbol{y}_n))$
            $\boldsymbol{y}_n \leftarrow \boldsymbol{y}_n + \boldsymbol{v}_n$     // Update particle location
            if $\boldsymbol{y}_n \in \mathcal{S}_l$ then
                if $f(\boldsymbol{y}_n) < f(\hat{\boldsymbol{y}}_n)$, $\hat{\boldsymbol{y}}_n \leftarrow \boldsymbol{y}_n$ // local best
                if $f(\boldsymbol{y}_n) < f(\hat{\boldsymbol{g}})$, $\hat{\boldsymbol{g}} \leftarrow \boldsymbol{y}_n$  // global best
            end if
        end for
    end for
end for
```

The constrained PSO is repeated at each level exactly as explained in Sec. III-B. Only a few things change at each level. The search region is centered at the global best $\hat{\boldsymbol{g}}_{l+1}$

from the previous level and scaled down (Line 3), for the lowest level the center is the position $\boldsymbol{x}$ of the salient point in the reference face (Line 1), the size of the neighborhood over which the covariance matrix is computed is scaled down (Line 4), the velocity of each particle is initialized with zero and the particle location is chosen randomly according to a Gaussian probability distribution (Line 5). The mean of the Gaussian probability distribution is set to the current global best and the standard deviation is empirically set to half of the size of the search region. It can be seen that any further constraint can easily be integrated at line 12. We additionally required the position $\boldsymbol{y}_n$ to not lie in the background which can be verified with the foreground-background mask computed previously (Sec. III-A). The pyramidal approach is depicted in Fig. 2. Four levels are employed and we begin at level 3. At the right side the size of the neighborhood and the size of the search region are shown, which are both step by step decreased. The global best at each level is indicated by a star. If larger distances are anticipated, the pyramid should have more levels. In the evaluation in Sec. IV we show results for various numbers of particles (N) and various maximum numbers of iterations (M).

Observe that the positions found via PSO are continuous. Therefore, bilinear interpolation is applied, if the x- and y-coordinates are not whole numbers. For example, the red channel at position (x, y) can be computed by

$$\begin{aligned} R(x,y) =& (1-\alpha_x)(1-\alpha_y)R(x_0,y_0) \\ &+ \alpha_x\alpha_y R(x_0+1,y_0+1) \\ &+ \alpha_x(1-\alpha_y)R(x_0+1,y_0) \\ &+ (1-\alpha_x)\alpha_y R(x_0,y_0+1), \end{aligned} \tag{8}$$

where $x = x_0 + \alpha_x$ and $y = y_0 + \alpha_y$, and x_0 and y_0 are the integer parts of x and y. Note that then the pixels in the neighborhood $\mathcal{N}$, which are needed to compute $\boldsymbol{C_y}$, must also be interpolated.

IV. RESULTS

A. Evaluation Scheme

The Bosphorus Database [15] was employed for the evaluation. The data is labeled with 22 landmarks per face. These landmarks were considered as ground truth. For those 22 spatial points the accuracy of the registration process is measured. The average distance $\bar{d} = \frac{1}{K}\sum_{k=1}^{K} d_i$, with $d_i = \sqrt{\Delta x_i^2 + \Delta y_i^2 + \Delta z_i^2}$, between the position estimated by our method and the true coordinates of this landmark was computed. Subsequently, the average distance was normalized to the height of the reference face, so that e.g. a distance of 0.05 stands for 5 % of the height of the reference face. Images with missing landmarks due to occlusion or lateral point of view were sorted out. The remaining set of facial surfaces contains $K = 2760$ shots from 105 individuals with all kinds of facial expressions and head poses. Figure 5 (a) shows individual number 000 from the database with neutral expression, which was chosen as reference face, with the 22 landmarks.

B. Evaluation of Covariance Matrix Pyramids

In Tab. I the effects of the covariance matrix pyramid are shown. The statements of Sec. III-C can be confirmed. The first row shows the poor results for a small neighborhood ($|\mathcal{N}| = 4 \times 4$) without pyramidal approach with $N = 5$ particles and $M = 20$ iterations. The variance $\sigma^2 = \frac{1}{K}\sum_{k=1}^{K}(d_i - \bar{d})^2$ is relatively large, indicating that some of the corresponding points have been found in a completely wrong place while others could be located quite accurately. The second row displays the results for a larger neighborhood ($|\mathcal{N}| = 32 \times 32$) with $N = 5$ particles and $M = 20$ iterations. The average distance is also poor but σ^2 is smaller, indicating that the region of the landmark in the face has been found reliably but the exact location could not be determined. Here the pyramidal approach applies. Rows three to five show that if subsequently the size of the neighborhood and the size of the search region are decreased step by step, the position can be computed more accurately.

C. Evaluation for Different Numbers of Particles and Iterations

At each pyramid level a certain number N of particles and number M of iterations is employed. Figure 3 shows the influence of N on the average distance $\bar{d}$ and the computation time t per face for $M = 40$. All reported computation times are for a C++ implementation running on a 3GHz Intel® Pentium® Dual-Core processor and 3GB working memory. The results show that for more than $N = 20$ particles the accuracy does not improve significantly, which is obvious, since the particle locations have only two dimensions. Figure 4 shows the effect of M on the registration accuracy and the computation time t per face for $N = 5$. It can be seen that it is not necessary to employ more than $M = 100$ iterations. We conclude that if optimal performance is required, our method should have $N = 20$ and $M = 100$, but also smaller values can be chosen to reduce the computation time without too much loss in registration accuracy.

D. General Evaluation

The KLT feature tracker [6] as suggested in [4] was implemented as baseline system. Table II displays the results for landmarks situated in different face regions, namely eye brows, eyes, nose, mouth, and chin, and the overall results. The average distance $\bar{d}$ for the baseline system, for our registration method with $N = 5$ particles and $M = 20$ iterations, and for our registration method with $N = 10$ particles and $M = 100$ iterations are shown. Compared to the baseline system with an average distance of 0.0667, our method could decrease the average distance by $(0.0667 - 0.0457)/0.0667 = 31\%$. Some faces from the database with the corresponding points found by our method are shown in Fig. 5 (b). It can be seen that the landmarks are detected reliably. Note that our approach does not require any training. Considering that the test set includes individuals with different facial expressions, sex, and ethnic background, with or without facial hair, and with varying illumination conditions and poses, an average distance of 0.0457 is remarkable.

pyramid level	$\vert\mathcal{N}\vert$	$\vert\mathcal{S}\vert$	$\bar{d}$	σ^2
no pyram. appr.	4×4	16×16	0.0577	0.00171
3	32×32	16×16	0.0538	0.00146
2	16×16	8×8	0.0499	0.00154
1	8×8	4×4	0.0469	0.00159
0	4×4	2×2	0.0468	0.00159

TABLE I: Average $\bar{d}$ and variance σ^2 of the distances d_i between the position of landmarks estimated by our method ($N = 5$, $M = 20$) and their true position for 2760 faces each with 22 landmarks. The distances are normalized to the height of the reference face.

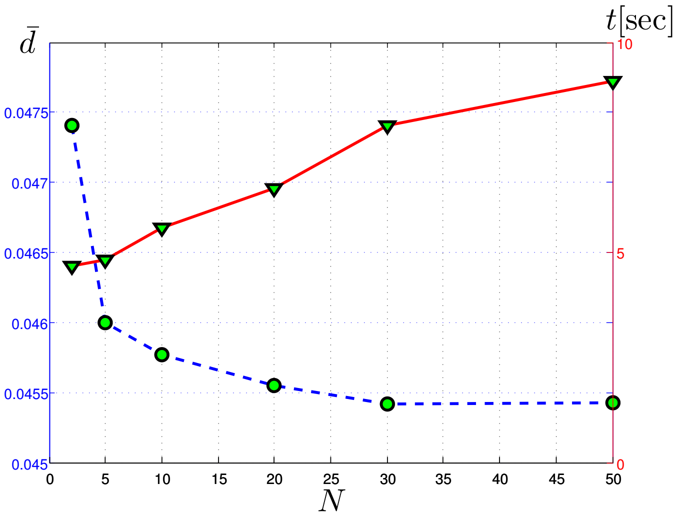

Fig. 3: Average distance $\bar{d}$ and computation time t per face against number of particles N for $M = 40$ iterations.

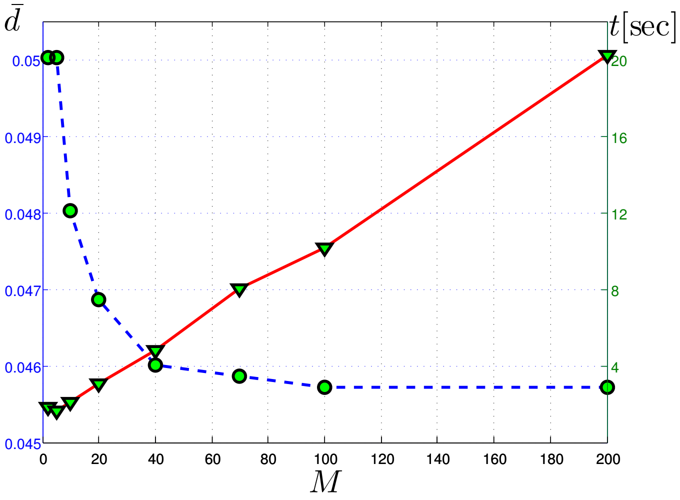

Fig. 4: Average distance $\bar{d}$ and computation time t per face against number of iterations M for $N = 5$ particles.

V. CONCLUSION AND FUTURE WORK

In this work a registration method is presented that is able to cope with the challenges of the registration of 3D facial surfaces, such as varying illumination, pose or viewpoint changes, varying facial expressions, and different appearance of individuals. A salient point in a face is described by the variance of all channels (red, green, blue, depth, etc.) and the covariance between those channels in its neighborhood. A corresponding point in another face with a similar covariance matrix is found via Particle Swarm Optimization (PSO). Covariance matrix pyramids are applied that gradually refine the location of the spatial point. The results demonstrate the successive improvement of the location for a pyramidal approach. With a challenging dataset it is shown that our registration method performs remarkably for a variety of disturbing effects. Our method improves the registration accuracy by 31% compared to the method [4] which was chosen as baseline system.

In our ongoing research, we will investigate possibilities to add a training phase to the application of covariance matrix pyramids and the use of further channels obtained by Gabor filters or Edgelet filters.

REFERENCES

[1] Y. Wang, M. Gupta, S. Zhang, S. Wang, X. Gu, D. Samaras, and P. Huang, “High resolution tracking of non-rigid motion of densely sampled 3D data using harmonic maps,” *International Journal of Computer Vision*, vol. 76, no. 3, pp. 283–300, 2008.

[2] F. Dornaika and B. Raducanu, “Detecting and tracking of 3D face pose for human-robot interaction,” in *ICRA*, 2008, pp. 1716–1721.

[3] M. Hanheide, S. Wrede, C. Lang, and G. Sagerer, “Who am i talking with? A face memory for social robots,” in *ICRA*, 2008, pp. 3660–3665.

[4] V. Blanz and T. Vetter, “A morphable model for the synthesis of 3D faces,” in *SIGGRAPH*, 1999, pp. 187–194.

[5] L. Zhang, N. Snavely, B. Curless, and S. M. Seitz, “Spacetime faces: High resolution capture for modeling and animation,” *ACM Trans. Graph.*, vol. 23, no. 3, pp. 548–558, 2004.

[6] C. Tomasi and T. Kanade, “Detection and tracking of point features,” Carnegie Mellon University, Tech. Rep., April 1991.

[7] N. Litke, M. Droske, M. Rumpf, and P. Schröder, “An image processing approach to surface matching,” in *Symposium on Geometry Processing*, 2005, pp. 207–216.

[8] A. Savran and B. Sankur, “Non-rigid registration of 3D surfaces by deformable 2D triangular meshes,” in *Computer Vision and Pattern Recognition Workshops*, June 2008, pp. 1–6.

[9] O. Tuzel, F. Porikli, and P. Meer, “Region covariance: A fast descriptor for detection and classification,” in *European Conference on Computer Vision*, 2006, pp. 589–600.

[10] W. Förstner and B. Moonen, “A metric for covariance matrices,” University of Stuttgart, Tech. Rep., 1999.

[11] J. Shi and C. Tomasi, “Good features to track,” in *Computer Vision and Pattern Recognition*, 1994, pp. 593–600.

[12] J. Kennedy and R. Eberhart, “Particle swarm optimization,” in *Proc. of the IEEE International Conference on Neural Networks*, 1995, pp. 1942–1948.

[13] X. Hu and R. Eberhart, “Solving constrained nonlinear optimization problems with particle swarm optimization,” in *6th World Multiconference on Systemics, Cybernetics and Informatics*, 2002, pp. 203–206.

[14] M. Clerc and J. Kennedy, “The particle swarm - explosion, stability, and convergence in a multidimensional complex space,” *IEEE Transaction on Evolutionary Computation*, vol. 6, no. 1, pp. 58–73, 2002.

[15] A. Savran, N. Alyüz, H. Dibeklioglu, O. Çeliktutan, B. Gökberk, B. Sankur, and L. Akarun, “Bosphorus database for 3D face analysis,” in *BIOID*, 2008, pp. 47–56.

average distance $\bar{d}$	landmarks at eye brows	landmarks at eyes	landmarks at nose	landmarks at mouth	landmarks at chin	overall
KLT [6] as suggested in [4]	0.0591	0.0511	0.0592	0.0856	0.0995	0.0667
our method ($N = 5$, $M = 20$)	0.0378	0.0312	0.0527	0.0525	0.0635	0.0468
our method ($N = 10$, $M = 100$)	**0.0361**	**0.0291**	**0.0514**	**0.0525**	**0.0634**	**0.0457**

TABLE II: Average distance $\bar{d}$ between the position of landmarks estimated by the registration method and their true position (2516 faces with 22 landmarks). The distances are normalized to the height of the reference face. Our method with ($N = 10$, $M = 100$) could decrease the average distance by 31% compared to a registration with the KLT feature tracker [6] as suggested in [4].

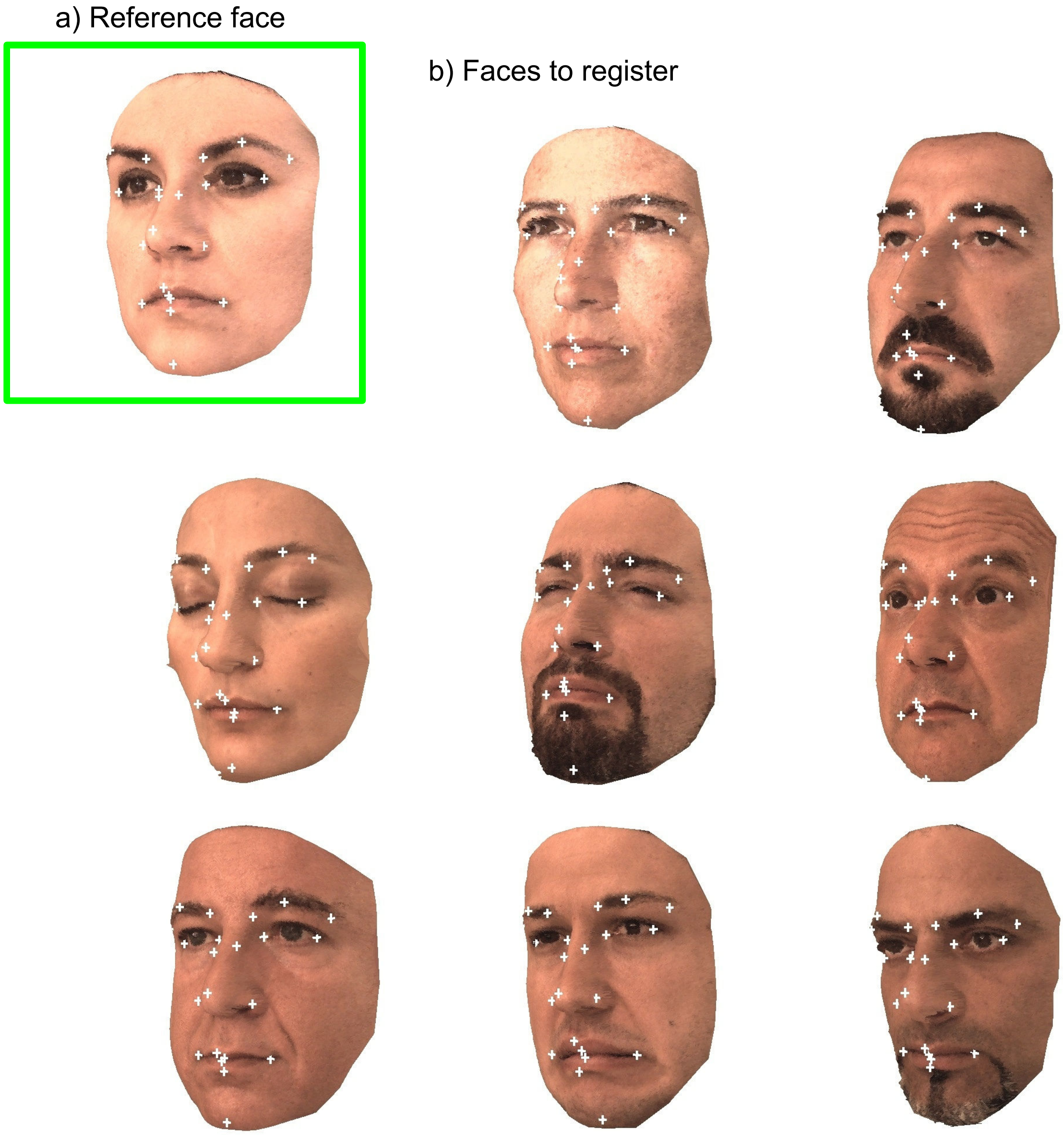

Fig. 5: Some 3D faces from the testing database. (a) shows the face that has been used as reference face with landmarks. (b) shows faces from the database with the landmarks found by our registration method.