Towards using covariance matrix pyramids as salient point descriptors in 3D point clouds

Moritz Kaiser [a,*], Xiao Xu [a], Bogdan Kwolek [a,b], Shamik Sural [a,c], Gerhard Rigoll [a]

[a] Institute for Human-Machine Communication, Technische Universität München, Munich, Germany
[b] Faculty of Electrical and Computer Engineering, Rzeszow University of Technology, Rzeszow, Poland
[c] School of Information Technology, Indian Institute of Technology, Kharagpur, India

A B S T R A C T

In this work, a novel salient point descriptor for 3D point clouds, called Covariance Matrix Pyramids (CMPs), is presented. With CMPs it is possible to compare unstructured and unequal numbers of points which is an important characteristic when working with point clouds. Corresponding points from different scans are matched in a pyramidal approach combined with Particle Swarm Optimization. The flexibility of CMPs is demonstrated on the basis of several databases with objects, such as 3D faces, 3D apples, 3D kitchen scenes, 3D human–machine interaction gesture sequences, and 3D buildings all recorded with different 3D sensors. Quantitative results are given and compared with other state-of-the-art descriptors, whereby CMPs show promising performance.

Keywords:
Salient point descriptor
3D point clouds
Global optimization
Covariance matrix

1. Introduction

1.1. Motivation

In the computer vision domain conventional cameras which output one channel (gray) images or three channel (color) images are increasingly supplemented by information from novel sensors [1,2]. Especially 3D sensors are important to gather necessary information about the environment for all kinds of human–machine interaction applications. Examples for such sensors include the PointGrey Bumblebee XB3, the Velodyne LIDAR used in the DARPA Urban Challenge, the Siemens Structured-Light 3D Scanner, or Microsoft's Kinect sensor. The output of these devices is not conveniently structured as image but as 3D point cloud. The huge success of the NVidia/Google supported Point Cloud Library [3] and Microsoft's Kinect can be seen as indicator that in the near future point clouds will play an important role in the computer vision field and probably even replace conventional images for many applications. However, almost all salient point descriptors rely on dense gray or color images, and only little work has been done on matching points in point clouds. Therefore, we felt the need to present a new point descriptor that is able to cope with 3D point clouds. A possible application for such a descriptor would be automatic labeling of a database. The user could select salient points in, for example, one reference face and the other faces in a database are then automatically labeled.

1.2. Related work

There exists a considerable number of salient point descriptors. Among the most prominent ones are KLT [4], SIFT [5], PCA-SIFT [6], and SURF [7]. In [8], a comparison among state-of-the-art point descriptors is given, in which the SIFT descriptor performs best. Also for tracking, accurate optical flow methods exist, such as [9–12]. The SURF descriptor has been further refined in [13], where the FAIR-SURF descriptor has been proposed. In [14], the authors present a scale invariant method for image matching which applies weighted voting on a 3D affinity matrix.

Covariance matrices have been used in [15,16], where both approaches are applied to conventional images. In [17], the authors propose a similar approach, called Sigma Set, which is computationally less demanding. In [18], Pang et al. applied Gabor-based covariance matrices for face recognition. This approach has been further refined in [19], where the Kernel Gabor Region Covariance Matrix has been presented and also applied for face recognition tasks. In [20], the authors explore smart possibilities to extract features from co-occurrence histograms of oriented gradients (CoHOGs) for person detection. However, all these methods rely on conventional images. Thus, they are not suited for 3D point clouds. In [21], the authors propose an interesting approach where

* Corresponding author. Tel.: +49 89 289 28547; fax: +49 89 289 28535.
E-mail addresses: moritz.kaiser@mytum.de, moritz.kaiser@tum.de (M. Kaiser), xiao.xu@tum.de (X. Xu), bkwolek@prz.edu.pl (B. Kwolek), shamik@sit.iitkgp.ernet.in (S. Sural), gerhard.rigoll@tum.de (G. Rigoll).

SIFT features are adapted for 2.5D range data with image structure and without texture.

There have also been contributions with methods that work directly with point clouds. Frome et al. presented 3D shape contexts and harmonic shape contexts to classify whole shapes without using texture [22]. In [23], the authors introduced a technique for the registration of 3D point clouds and Brostow et al. presented a work on semantic segmentation based on 3D point clouds in [24]. Another promising approach is spin images [25]. Note that the point matching strategy is brute force. Furthermore, spin-images are quite restrictive, i.e., they are designed to match points from exactly the same object, while matching, for example, facial feature points of two different individuals might fail. Rusu et al. [26] presented the Persistent Point Feature Histograms (PFH) for 3D point clouds that are also already available in Willow Garage's Point Cloud Library [3].

1.3. Overview

In this work, covariance matrix pyramids (CMPs), that have been presented in [16], are used for point clouds. Since images and point clouds are structurally different, the method substantially changed in order to work for point clouds. The result is a new, highly flexible salient point descriptor that works directly on 3D point clouds. The method is summarized as follows:

- A list of potential features for the description of the salient point's neighborhood is presented. With a training set, adequate features are selected via Sequential Forward Selection (SFS) with discrete weights (Section 2).
- Features are summarized by a covariance matrix. Employing a covariance matrix as salient point descriptor is practical for matching salient points. In contrast to many previously proposed descriptors (SIFT, SURF, local optical flow, etc.), it provides a convenient way to fuse conventional features (red, green, blue) with non-conventional features (depth, infrared, etc.). Spatial distribution is captured by the covariance between x, y, or z-coordinates of the points and their other features. Furthermore, covariance matrices are, to a certain extent, robust against noise and illumination offset, because both are filtered out by an average filter during covariance computation (Section 3).
- Corresponding points from different scans are matched. To allow for larger displacements covariance matrices are used in pyramids, motivating the name *covariance matrix pyramid*. Particle Swarm Optimization (PSO) is employed to find the best match at each pyramid level (Section 4).

Five application scenarios are given in Section 5. In the first two experiments, salient points in 3D faces are matched. Two publicly available databases with handlabeled landmarks have been employed. With these landmarks as ground truth quantitative results can be given and it can also be shown that PSO reduces computation time while not affecting matching accuracy.

Further, salient points in 3D apples, gesture sequences, kitchen scenes, and buildings are matched. The matching accuracy is compared to another point descriptor for 3D point clouds and two other point descriptors that rely on 2D images. All experiments demonstrate promising performance of CMPs. In Section 6, the work is concluded and future scope is outlined.

2. Adequate features

2.1. Output from sensors

We assume that sensors output an unstructured 3D point cloud. Examples for these sensors include the PointGrey Bumblebee XB3, the Velodyne LIDAR used in the DARPA Urban Challenge, the Siemens Structured-Light 3D Scanner, Inspeck Mega Capturor II 3D, Di3D Dynamic Imaging System, or Microsoft's Kinect sensor. Each point has spatial attributes (x,y,z) and color attributes (r,g,b). If one of the points is selected as salient point, information about this point and its neighborhood must be extracted for its representation. For this purpose, features are extracted, as explained in the next section.

2.2. Feature extraction

For a salient point a set of features is computed. We propose a list of potential features (depicted in Fig. 2 for a face of the Bosphorus database [27]) of which the best features can be selected automatically if a training set is available. Spatial information (x,y,z) can be directly taken. Hue H, saturation S, and value V are computed from each point's rgb-values.

The surface normal $\boldsymbol{n}_i$ for point i, which is depicted in Fig. 1, is computed as follows. The point cloud is triangulated with Delaunay triangulation. The surface normal $\boldsymbol{n}_t$ at the triangle centroid is computed. For the triangle t $(\boldsymbol{p}_1,\boldsymbol{p}_2,\boldsymbol{p}_3)$ the surface normal is

$$\boldsymbol{n}_t = \begin{pmatrix} n_x \\ n_y \\ n_z \end{pmatrix} = (\boldsymbol{p}_2 - \boldsymbol{p}_1) \times (\boldsymbol{p}_3 - \boldsymbol{p}_1). \tag{1}$$

The surface normal $\boldsymbol{n}_i$ of point i is then the average of all surface normals of the triangles of which point i is a vertex:

$$\boldsymbol{n}_i = \frac{1}{\sum_t \omega_t} \sum_t \omega_t \cdot \boldsymbol{n}_t, \tag{2}$$

where ω_t is a weight that depends on the distance between the centroid of triangle t and point i and $\sum_t \omega_t = 1$.

There is no straightforward way to compute the intensity gradient for point clouds, as for conventional images, so an alternative measure is considered. The *intensity normals* $\boldsymbol{g}_i$ are computed similar to the surface normal, except that the third component of the triangle point is the intensity instead of z: $\boldsymbol{p}_j = (x_j, y_j, I_j)^T$.

A further feature is the intensity entropy. To compute the entropy, all points in the neighborhood of point i are taken. We set the neighborhood size to 2% of the object height. A histogram of the intensity values of all points in the neighborhood is created. With this histogram a numerical probability p_g can be assigned to each gray value $g \in (0, 255)$. The intensity entropy is then

$$H(I) = \sum_{g=0}^{255} p_g \cdot \log p_g. \tag{3}$$

We also perform several operations on these features that are inspired by a mean filter, a mean of absolute values filter, and a Laplace filter for conventional images. These three operations are applied to all three components of the surface normal (n_x, n_y, n_z)

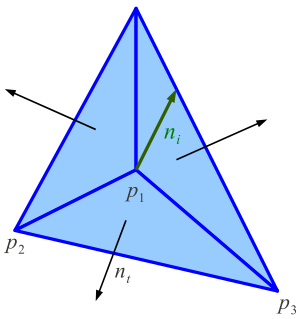

Fig. 1. The surface normal $\boldsymbol{n}_i$ of point i is the average of the surface normals of adjacent triangles.

and also for the three components of the intensity normals (I_x, I_y, I_z). The mean operation, which is denoted by $m(n_x)$, is the mean over all points in the salient point's neighborhood:

$$m(n_{x,i}) = \frac{1}{|\mathcal{N}_i|} \sum_{j \in \mathcal{N}_i} n_{x,j}. \tag{4}$$

We take again 2% of the object height as neighborhood size. Additionally, the mean of absolute values over a certain neighborhood is taken:

$$ma(n_{x,i}) = \frac{1}{|\mathcal{N}_i|} \sum_{j \in \mathcal{N}_i} |n_{x,j}|. \tag{5}$$

The sum of differences operation is the sum of differences between a certain component of point i and the neighboring points:

$$d(n_{x,i}) = \frac{1}{|\mathcal{N}_i|} \sum_{j \in \mathcal{N}_i} (n_{x,i} - n_{x,j})^2. \tag{6}$$

Hence, a total of 35 features can be computed for one point. All potential features are depicted in Fig. 2. If no training material is available, features must be chosen manually, which also works fine, as demonstrated in Section 5 for apples. A better performance is achieved if a training set is available and features can be selected automatically, as illustrated in the next section.

2.3. Feature selection

There are several techniques for the selection of adequate features, such as Sequential Forward Floating Selection [28] or

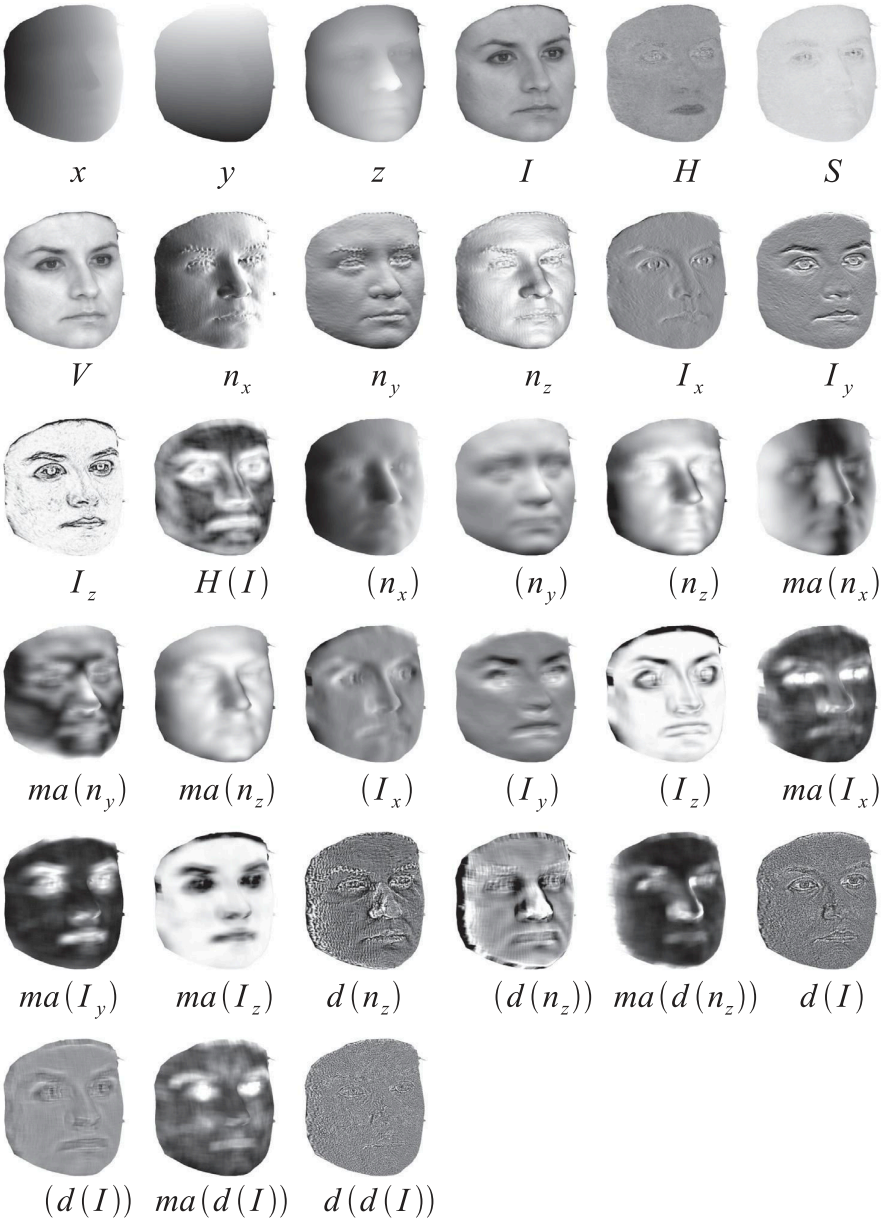

Fig. 2. List of potential features for 3D points in a point cloud.

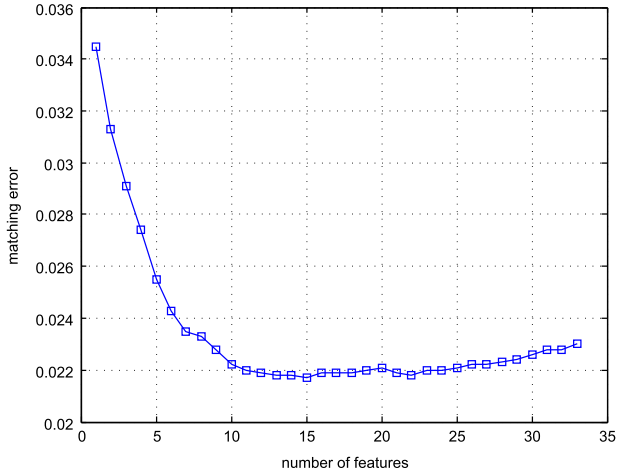

Fig. 3. Matching error $\bar{e}$ over the number of features computed with a training set of 400 facial surfaces.

AdaBoost [29]. However, these methods would require a prohibitively large amount of computation time. Hence, we apply an alternative technique, which we will call Sequential Forward Selection [28] with discrete weights for the selection of adequate features. The method is relatively fast but also computes weights for the features. Let $\mathcal{F}=\{f_1,f_2,\ldots,f_J\}$ be the set of potential features depicted in Fig. 2, where f_j is a particular potential feature and J is the total number of features. The average matching error $\bar{e}(w_1,w_2,\ldots,w_J)$ (explained in Section 5 in more detail) is the measure of matching accuracy. $w_j\in\mathbb{N}_0$ is the weight that corresponds to feature f_j. If w_j is zero, feature f_j is not employed at all. Discrete weights w_j for each potential feature f_j are computed. The method is presented by the following pseudo code.

```
1:  ∀j :  w_j = 0    //Initialization of discrete weights
2:  loop
3:    k = arg min_{1≤j≤J} ē(w_1, ..., w_j + 1, ..., w_J)
4:    w_k ← w_k + 1
5:  end loop
```

All weights w_j are initialized with 0. The best single feature is determined and its weight is set to 1. In subsequent iterations the feature which performs best in combination with the features chosen at previous iterations is added to the set. New features are chosen from the whole feature set $\mathcal{F}$ at each iteration. If feature f_k is chosen twice, three times, etc., simply its weight ω_k is incremented. Therefore, the output of the feature selection method is not only the order of the features in terms of significance, but also a discrete weight for each feature.

Fig. 3 depicts the matching error $\bar{e}$ over the number of features for 400 3D faces of the Bosphorus database. It can be seen that after 15 features the matching error does not improve significantly any more and thus we decided to employ 15 features for the rest of this work. These 15 features provide manifold information about a salient point, or more precisely its neighborhood. This information is summarized with a covariance matrix, which is described in the next section.

3. Covariance matrix as salient point descriptor

In this section the covariance matrix as salient point descriptor for point clouds will be introduced and an appropriate distance measure will be presented.

3.1. Covariance matrix

A covariance matrix is used to describe the neighborhood of a salient point. Let $\mathcal{P}$ be a point cloud with points $\boldsymbol{p}_i$. For each point i a feature vector $\boldsymbol{f}_i\in\mathbb{R}^F$ with F features can be extracted. The neighborhood around point i is denoted by $\mathcal{N}_i$. The covariance matrix that describes the neighborhood of point i is

$$\boldsymbol{C}_i=\frac{1}{|\mathcal{N}_i|-1}\sum_{j\in\mathcal{N}_i}(\boldsymbol{f}_j-\boldsymbol{\mu}_i)(\boldsymbol{f}_j-\boldsymbol{\mu}_i)^T, \tag{7}$$

where $\boldsymbol{\mu}_i=1/|\mathcal{N}_i|\sum_{j\in\mathcal{N}_i}\boldsymbol{f}_j$ denotes the mean vector of the component vectors and $|\mathcal{N}_i|$ stands for the number of points in region $\mathcal{N}_i$. Covariance matrices capture important information about the salient point's neighborhood.

Variance: The covariance matrix encodes each feature by storing its variance computed over the neighborhood of a salient point. For a sufficient number of features this is already a conclusive descriptor. Fig. 4(a) shows two surfaces with the same shape but different texture. The variance of, for example, the hue feature of the two surfaces is different.

Covariance: The variance cannot capture all aspects of the shape and texture. Thus, the covariance between features is another important value. It is a convenient way to fuse features. In Fig. 4(b), two surfaces are shown. For both surfaces the hue feature increases from back to front. The saturation increases from back to front for the upper figure and decreases from back to front for the lower figure. Although the texture is different, the variances of the hue and the saturation features are equal for the two surfaces. However, the covariance between hue and saturation is different and makes it possible to distinguish between the two surfaces.

Spatial layout: The spatial layout of each component is encoded by the covariance with x-, y-, and z-features. Fig. 4(c) shows two different surfaces with equal texture. The variances of the hue-feature are equal for both surfaces. Also the variances of the z-feature are equal, but the covariances between hue and z-feature are different.

Mean shift invariance: The covariance is not affected if the mean of a feature is shifted, so it can cope with illumination changes to a certain extent. Fig. 4(d) shows two surfaces, one with bright illumination and the other one with dim illumination. Nevertheless, the hue feature has the same variance for both surfaces. The same is true for the variance of the saturation and the value.

Noise robustness: Noise is largely filtered out by averaging over the neighborhood while computing the covariance matrix. This is important since during the recording process, mesh generation, and feature computation noise is added to the features. Fig. 4 (e) shows two equal surfaces, but one with and the other one without noise. For both surfaces hue, saturation and value have approximately the same variance.

3.2. Distance measure

Covariance matrices do not lie in Euclidean space and thus most of the distance measures used in machine-learning, such as Euclidean or Mahalanobis distance, cannot be applied. We compare several distance measures that have been proposed for covariance matrices. Assume two covariance matrices $\boldsymbol{C}_1$ and $\boldsymbol{C}_2$, with size $F\times F$, where F is the number of features per point. In [30], two distance measures named log-Euclidean norm

$$\rho(\boldsymbol{C}_1,\boldsymbol{C}_2)=\|\log(\boldsymbol{C}_1)-\log(\boldsymbol{C}_2)\| \tag{8}$$

and log-Euclidean trace

$$\rho(\boldsymbol{C}_1,\boldsymbol{C}_2)=\mathrm{tr}[(\log(\boldsymbol{C}_1)-\log(\boldsymbol{C}_2))^2] \tag{9}$$

have been presented. The matrix logarithm $\log(\boldsymbol{C})$ is computed as follows. First, $\boldsymbol{C}$ is diagonalized: $\boldsymbol{C}=\boldsymbol{R}^T\boldsymbol{D}\boldsymbol{R}$, where $\boldsymbol{R}$ is the rotation

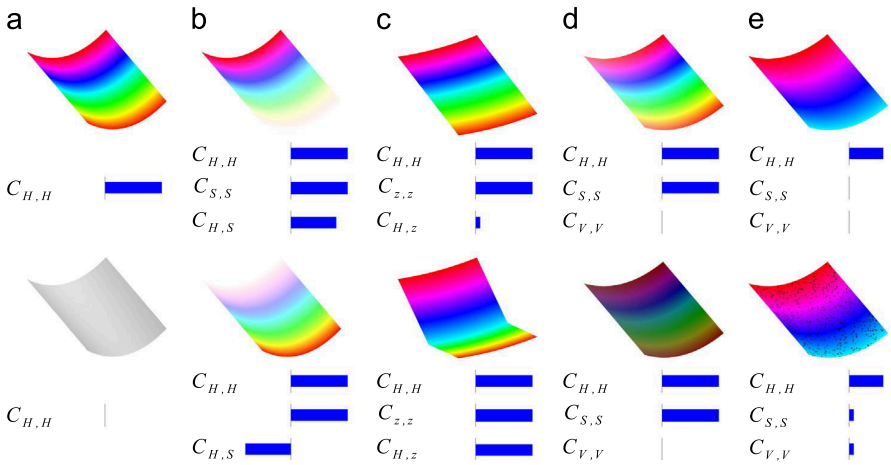

Fig. 4. The covariance matrix is a discriminative salient point descriptor. (a) The variance of features is already informative, (b) the covariance between features provides further information, (c) spatial distribution is encoded by the covariance with *x*-, *y*-, or *z*-features and (d) covariance matrices are robust against mean shift and (e) noise.

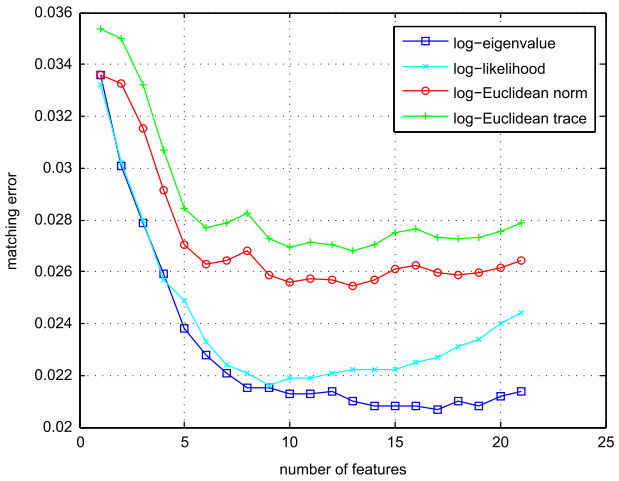

Fig. 5. Matching error $\bar{e}$ over the number of features computed with a training set of 400 facial surfaces. The log-eigenvalue measure performs best.

matrix and $\boldsymbol{D}$ a diagonal matrix. Subsequently, the natural logarithm is applied to each diagonal element of $\boldsymbol{D}$ in order to obtain $\tilde{\boldsymbol{D}}$. Finally, $\log(\boldsymbol{C}) = \boldsymbol{R}^T\tilde{\boldsymbol{D}}\boldsymbol{R}$.

In [31], two metrics based on the generalized eigenvalues are proposed, namely the log-likelihood measure

$$\rho(\boldsymbol{C}_1,\boldsymbol{C}_2) = \frac{1}{F}\sum_{i=1}^{F}(\lambda_i - \log^2(\lambda_i) - 1) \tag{10}$$

and the log-eigenvalue measure

$$\rho(\boldsymbol{C}_1,\boldsymbol{C}_2) = \frac{1}{F}\sum_{i=1}^{F}\log^2(\lambda_i). \tag{11}$$

The generalized eigenvalues of $\boldsymbol{C}_1$ and $\boldsymbol{C}_2$ are denoted by $\{\lambda_i\}_{i=1\ldots F}$ and they 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: $\boldsymbol{C}_1^{-1}\boldsymbol{C}_2\boldsymbol{v}_i = \lambda_i\boldsymbol{v}_i$.

For all distance measures $\rho(\boldsymbol{C}_1,\boldsymbol{C}_2)$, high values mean great difference between the two covariance matrices. If $\rho = 0$, both covariance matrices are equal. We perform the channel selection via SFS with discrete weights, as explained in Section 2, for all four distance measures. Fig. 5 depicts the matching error $\bar{e}$ over the number of features. It can be seen that the log-eigenvalue measure performs best.

4. Matching strategy

The covariance matrix descriptor is used to match points between two surfaces that are represented by 3D point clouds. The point cloud in which one or several reference points is selected will be called *reference point cloud* $\mathcal{P}_{\text{ref}}$ and the point cloud in which corresponding points is searched will be called *new point cloud* $\mathcal{P}_{\text{new}}$. Corresponding points should have similar covariance matrices. Hence, if we want to find a point $j \in \mathcal{P}_{\text{new}}$ that corresponds to point $i \in \mathcal{P}_{\text{ref}}$, we have to minimize

$$j = \arg\min_{\boldsymbol{j}\in\mathcal{P}_{\text{new}}} \rho(\boldsymbol{C}_i,\boldsymbol{C}_j), \tag{12}$$

with $\rho(\cdot,\cdot)$ being the log-eigenvalue distance measure described in Section 3.2. This is a nonlinear optimization problem that we will solve with a pyramidal approach to avoid local minima, where at each level the best match is found via PSO.

4.1. Pyramidal approach

Inspired by multi-resolution pyramids for images we employ a similar coarse-to-fine strategy for 3D point clouds in order to avoid local minima when solving Eq. (12). The covariance matrix descriptor of a point is computed using a subset of points that lie within the neighborhood $\mathcal{N}$ of the point under consideration. The region in which a corresponding point is searched in $\mathcal{P}_{\text{new}}$ is defined as the search region $\mathcal{S}$. There is a tradeoff between a large and a small neighborhood $\mathcal{N}$. A large $\mathcal{N}$ allows us to find the region of the salient point robustly but not the exact location, whereas a small $\mathcal{N}$ allows us to find the location more exactly but not so reliably. Therefore, at first $\mathcal{N}$ and search region $\mathcal{S}$ are quite large (as illustrated in Fig. 6) to determine robustly the area where the corresponding point lies. Once a rough location is found, the position is step-by-step refined. At the next level the search is repeated with the following modifications: (i) the center of the search region $\mathcal{S}$ is the best match found at the previous level, (ii) the size of $\mathcal{S}$ is reduced, and (iii) the size of $\mathcal{N}$ is also reduced to make the solution more precise.

Note that the neighborhoods of two points from $\mathcal{P}_{\text{ref}}$ and $\mathcal{P}_{\text{new}}$ usually comprise different amounts of points, although the neighborhoods have the same size, because of different sensor resolutions, different angles to the projector, different object heights, etc. The advantage of the covariance matrix descriptor is that we can compare subsets with different amounts of points.

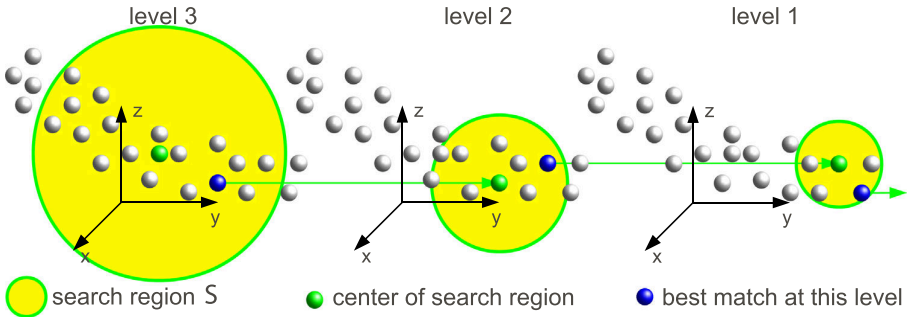

Fig. 6. Illustration of the covariance matrix pyramid for 3D point clouds.

At each level, the point in the search region with the lowest distance measure ρ has to be found. An exhaustive search would be computationally quite expensive. Thus, we propose a more efficient solution to this problem based on PSO.

4.2. Particle swarm optimization

PSO, first introduced in [32], is a simple technique that has proven to be efficient for many nonlinear optimization problems [33]. We perform the search in a *uv*-space, i.e., along the surface, because it is much faster. Hence, also the search of the points that lie within $\mathcal{N}$, which would be computationally intensive, can be sped up considerably. For each 3D point, *uv*-coordinates are computed. There are many mapping methods, such as parallel projection, cylindrical projection [34], Least Squares Conformal Mapping [35], Harmonic Mapping [36], ABF++ [37], etc. Most of these methods were developed in the context of texture mapping. An overview of recent texture mapping methods is given in [38]. The choice of the method depends on the shape of the object the matching is used for. For the results in Section 5 we have employed parallel projection (gesture and kitchen set), cylindrical projection (faces), and ABF++ (apples). Note that for more complex shapes more complex texture mapping methods must be applied. If the shape is too complex, computing *uv*-coordinates might fail. A mesh is laid over the points in *uv*-space and for each hole all points of $\mathcal{P}_{\text{new}}$ with *uv*-coordinates within this hole are registered. Hence, the points inside a certain neighborhood $\mathcal{N}_{\boldsymbol{x}}$ with center $\boldsymbol{x}$ in *uv*-coordinates can be computed quickly.

Let $\mathcal{N}_i$ be the neighborhood of a salient point $i \in \mathcal{P}_{\text{ref}}$ and let $\boldsymbol{C}_i$ be the covariance matrix computed over all points that lie within this region. A location $\boldsymbol{x}$ in *uv*-space that has a similar covariance matrix $\boldsymbol{C}_{\boldsymbol{x}}$ is searched in $\mathcal{P}_{\text{new}}$. This leads to the nonlinear optimization problem

$$\boldsymbol{x} = \arg\min_{\boldsymbol{x}\in\mathcal{S}} E(\boldsymbol{x}), \tag{13}$$

where the energy function is $E(\boldsymbol{x}) = \rho(\boldsymbol{C}_i, \boldsymbol{C}_{\boldsymbol{x}})$. The search is restricted to the search region $\mathcal{S}$. Note that the center $\boldsymbol{x}$ can also lie between points, i.e., $\boldsymbol{x}$ does not have to coincide with an exact point location. A total of N particles are created, each with location $\boldsymbol{x}_n \in \mathbb{R}^2$ and velocity $\boldsymbol{v}_n \in \mathbb{R}^2$ in *uv*-space. At each iteration, particles are drawn toward the positions of their own previous best $\hat{\boldsymbol{x}}_n$ and the global best $\hat{\boldsymbol{g}}$, while they must stay inside the search region $\mathcal{S}$. Our complete method is summarized by the following pseudo code.

```
1:  ĝ ← x_ref                          // Initialize global best
2:  for level l=4 to 1 do
3:    S_l, radius: 2^l% of reference object height, center: ĝ
4:    N_l, radius: 2^l% of reference object height
5:    v_n ← 0 and x_n ← Gaussian distribution, mean: ĝ
6:    x̂_n ← x_n and ĝ = arg min_{x_n} E(x_n)
7:    for M iterations do
8:      for N particles do
9:        v_n ← k · (ω v_n + c_1 r_1 ∘ (x̂_n − x_n) + c_2 r_2 ∘ (ĝ − x_n))
1-        x_n ← x_n + v_n                Update particle location
0:
11:       if x_n ∈ S_l then
1-          if E(x_n) < E(x̂_n), x̂_n ← x_n        Local best
2:
1-          if E(x_n) < E(ĝ), ĝ ← x_n            Global best
3:
1-        end if
4:
1-      end for
5:
1-    end for
6:
17:  end for
```

The center of the search region at the coarsest level (level 4) is initialized with the location of the salient point in the reference point cloud (Line 1). For each level the best match is computed. The velocities of all particles are initialized with zero and their positions are drawn from a Gaussian distribution (Line 5), where the mean is the current global best and the standard deviation is empirically set to the size of the search region divided by 4. The global best $\hat{\boldsymbol{g}}$ is initialized as the best of all particles (Line 6). M iterations are performed (Line 7), where for each particle (Line 8) the velocity is computed according to Line 9. The vectors $\boldsymbol{r}_1$ and $\boldsymbol{r}_2$ are vectors of random numbers in the range $[0, 1]$, which are generated at each iteration according to a uniform probability distribution. The operator $\circ$ denotes element-wise multiplication. We set $k = 0.75$, $\omega = 0.9$, and $c_1 = c_2 = 2.05$, as suggested after an in-depth analysis in [39]. Tests confirmed the suitability of this parameter selection. Particle locations are updated (Line 10) and if necessary the local best $\hat{\boldsymbol{x}}_n$ or the global best $\hat{\boldsymbol{g}}$ are updated (Lines 12 and 13). The corresponding point is then the 3D point that lies closest to $\hat{\boldsymbol{g}}$ after the optimization stops.

Compared to a brute force search, PSO with, for example, $N = 10$ particles and $M = 20$ iterations, reduces the computation time by roughly a factor of 10 while the matching accuracy is not affected considerably, as shown in the next section.

5. Experiments

5.1. Quantitative evaluation and comparison with other state-of-the-art methods

Several experiments have been conducted demonstrating the versatility of CMPs. We provide a quantitative evaluation with a

Table 1
List of the databases that have been used for the evaluation of CMPs.

Database name	Number of scans used for testing	Number of landmarks	Sensor type	Number of scans used for training	2D image available
Bosphorus database [27]	2516	22	Inspeck Mega Capturor II 3D	400	Yes
BU-3DFE database [40]	2100	83	Di3D Dynamic Imaging System	500	Yes
3D apples	15	5	Siemens 3D Scanner	No training	Yes
HMI gesture scenes	300	3–6	Microsoft Kinect	No training	Yes
Kitchen scenes [26]	15	10	Hokuyo UTM-30LX	No training	No
Google 3D warehouse scenes [41]	100	20	Various	No training	Yes

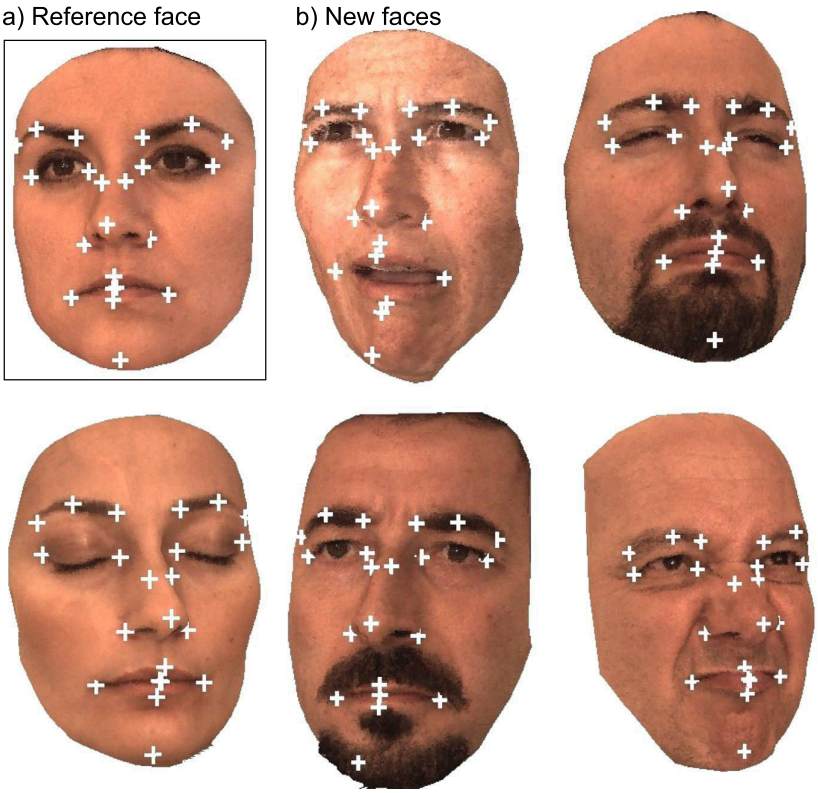

Fig. 7. Some 3D faces from the Bosphorus database. (a) Shows the face that has been used as reference face with landmarks and (b) shows faces from the database with the landmarks found with CMPs.

comparison to other state-of-the-art methods and a qualitative evaluation. The necessary number of particles and iterations is investigated on the basis of the Bosphorus database. The quantitative evaluation is performed as follows. All databases are handlabeled with landmarks which we used as ground truth. For each handlabeled landmark the matching error $e_i = \sqrt{\Delta x_i^2 + \Delta y_i^2 + \Delta z_i^2}$ between the position estimated by the algorithm and the ground truth is computed. The error is averaged over all K landmarks

$$\overline{e} = \frac{1}{K} \sum_{i=1}^{K} e_i. \tag{14}$$

Subsequently, the average error is normalized to the height of the reference object height. If for a certain database *rgb*-images are also available, we can compare our method with conventional point descriptors that work with images (namely local optical flow [34], SIFT-64, and SIFT-128 [5]) and verify if working with 3D point clouds is worthwhile. Because of the normalization to the object height, a fair comparison of the matching error is possible. Furthermore, we compare the results to PFH [3], that rely, similar to CMPs, directly on point clouds.

For the experiments six different databases have been used. Table 1 lists the six databases that have been employed, namely Bosphorus database [27], BU-3DFE database [40], 3D apples, HMI gesture scenes, kitchen scenes [26], and Google 3D warehouse scenes [41].

5.1.1. *Bosphorus database*

In the first experiment, salient points in 3D faces are matched. For this purpose the Bosphorus 3D face database [27] is employed. Each face is handlabeled with 22 landmarks, which we used as ground truth. The *rgb*-images for each face are employed for comparison with other methods that rely on images. The computation time and the numbers of particles and iterations are also investigated. Fig. 7 (a) shows individual number 000 from the database having neutral expression, which was chosen as reference face, with the 22 landmarks.

The test set that has been used includes 2516 faces and it does not contain the 400 faces that have been employed for the feature selection in Section 2. In Table 2, the matching errors normalized to the head height of SIFT-64, SIFT-128 [5], local optical flow [34], PFH [3], CMPs with a brute force matching, and CMPs with PSO are shown. It can be seen that CMPs improve the matching accuracy by 37%, 36%, 16%, and 21% compared to SIFT-64, SIFT-128, local optical flow, and PFH, respectively. The database contains many different individuals. It seems that other descriptors are not able to detect correspondences among individuals as good as CMPs under these circumstances. The table also shows that, with a brute-force search, which is much more time consuming than PSO, the average matching error cannot be considerably improved (0.8%). This justifies the usage of PSO to speed up the matching process.

Fig. 7(a) shows the reference face we used and Fig. 7(b) shows some faces in which the 22 salient points have been marked with CMPs. Some limitations of CMPs can be observed. If a face has an open mouth while the reference face does not, the texture and the depth values are too different. Further, it is difficult to locate points in the chin region, where there is little texture and also depth values are not very descriptive. Note that even for humans it is difficult to choose landmarks at the chin region unambiguously, which can also be observed when looking at the ground truth landmarks. Apart from that, remarkable results can be achieved for the individuals of the database with different facial expressions, sex, ethnic background, with or without facial hair, and with varying illumination conditions and poses, as can be seen in Fig. 7.

5.1.2. BU-3DFE database

The BU-3DFE database [40] contains 2600 faces. The faces have been recorded with a Di3D Dynamic Imaging System. We used 500 scans as training set and 2100 as test set. Each face has 83 handlabeled landmarks. The quantitative results are shown in Table 2. CMPs improve the matching accuracy by 26%, 25%, 14%, and 16% compared to SIFT-64, SIFT-128, local optical flow, and PFH, respectively. In general, detecting the landmarks of this database is more challenging than for the Bosphorus database, since there are less landmarks at edges and corners. SIFT and also PFH have difficulties especially if the landmarks are not situated exactly at corners. The BU-3DFE database also contains scans of many different individuals and it can be seen that CMPs are more appropriate to deal with these differences.

5.1.3. 3D apples

The 3D apples have been recorded with a Siemens Structured-Light 3D Scanner. A total of 15 apples have been used. The apples

Table 2
Average error $\overline{e}$ between the position of landmarks estimated by the registration method and the ground truth. The error is normalized to the height of the object. CMPs are compared to SIFT-64, SIFT-128 [5], local optical flow method of [34], and PFH [3].

Database name	SIFT-64 [5]	SIFT-128 [5]	Local optical flow method of [34]	PFH [3]	CMPs	CMPs, brute force
Bosphorus database [27]	0.0557	0.0549	0.0421	0.0451	0.0353	0.0350
BU-3DFE database [40]	0.0625	0.0617	0.0539	0.0544	0.0456	–
3D apples	0.0624	0.0596	0.0340	0.0551	0.0273	–
HMI gesture scenes	0.0268	0.025	0.0275	0.0247	0.0231	–
Kitchen scenes [26]	–	–	–	0.0325	0.0339	–
Google 3D warehouse scenes [41]	0.0522	0.0513	0.0847	0.0778	0.0532	–

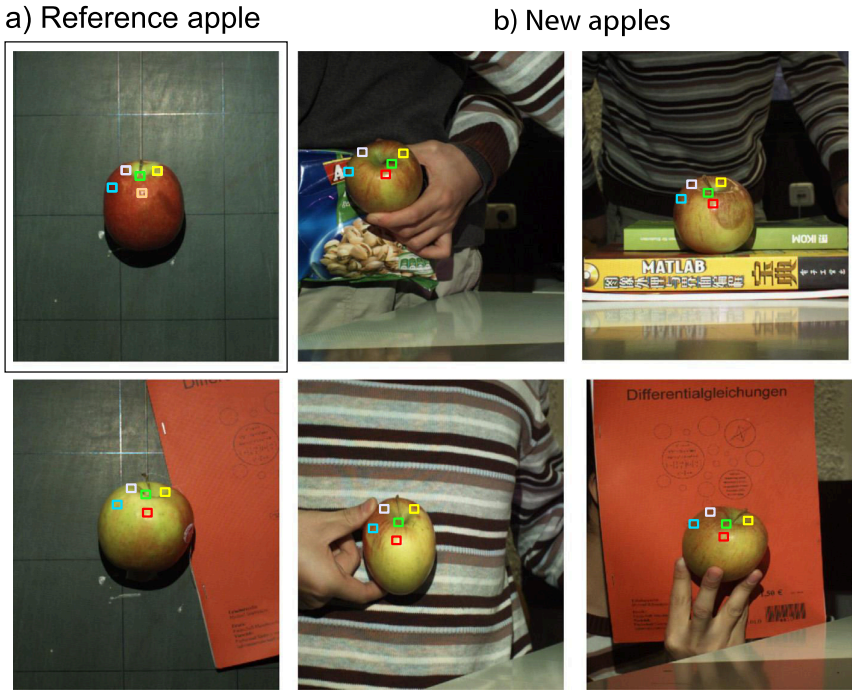

Fig. 8. Scans of apples with a Siemens Structured-Light 3D Scanner. (a) Five points have been selected as reference points and (b) could be found in other apples with CMPs without using color information.

have different colors and sizes. Also the shooting angle is different and there are various backgrounds, such as black board, computer screen, books, and a person holding the apple. Given that no training set is available, features are manually chosen. Color is not very meaningful, because the apples have different colors, such as red, green, and even yellow. Thus, all features based on color information were removed. Fig. 8(a) shows the reference apple for which five points have been chosen manually.

In addition to the point cloud the Siemens Structured-Light 3D Scanner outputs images, so we could also compare CMPs to methods that rely on 2D images. It can be seen that CMPs improve the matching accuracy by 55%, 54%, 21%, and 51% compared to SIFT-64, SIFT-128, local optical flow, and PFH, respectively. The improvement compared to other methods is larger than for the 3D faces. While other methods are developed especially for the description of corners and edges, CMPs are also quite robust for less descriptive regions since they integrate the whole neighborhood.

Fig. 8(b) shows some corresponding points that have been estimated using CMPs. The points could be found correctly. Note that it is already quite difficult for humans to detect these points. Here the flexibility of CMPs can be observed. Color information is not very helpful, so it is ignored. Nevertheless, CMPs find accurate corresponding points by using only features that are based on shape information. Especially the surface normals and the correlation of surface normals with the *xyz*-coordinates provide sufficient information to detect the points robustly.

5.1.4. Human–machine interaction gesture scenes

In this section, salient points are tracked in human–machine interaction gesture scenes. The gestures include arm movements, hand and finger movements and whole body movements. The scenes are recorded with a Point Grey Bumblebee XB3 stereo camera. The output is 25 frames per second video sequence. The database consists of 100 gesture scenes, each of which lasts roughly 3–4 s. We handlabeled 1 frame per second resulting in a total of 300 labeled scans. We used 3–6 landmarks depending on the scene.

For each frame not all pixels have depth values, as can be seen in Fig. 9, where the black pixels in frame 0 are pixels without depth information. No training set is available, so we took the features trained for the 3D faces. Fig. 9 shows some qualitative results. (Some of the videos are also provided as supplementary material). It can be seen that points at the hand, elbow, shoulder, and face are tracked accurately. The quantitative results are shown in Table 2. The landmarks are put only at corners, so also the other descriptors show good performance. CMPs improve the matching accuracy by 11%, 8%, 14%, and 4% compared to SIFT-64, SIFT-128, local optical flow, and PFH, respectively.

5.1.5. Kitchen scenes

For the presentation of the performance of PFH [26] the authors employed several kitchen scans recorded by a Hokuyo UTM-30LX Laser Scanner. The scans show the same room from different viewpoints. For each scan we handlabeled 10 salient points, chose another scan where these points are also visible, and handlabeled the corresponding points as ground truth. Subsequently, we tried to find the corresponding points in another scan automatically and measured the matching error $\bar{e}$ as explained above. The 3D point clouds are recorded with intensity information. No 2D images are provided, so a comparison to other 2D methods is not possible. Fig. 10(a) shows 10 landmarks that were set in this scan and (b) illustrates the corresponding points that have been found automatically with CMPs. It is shown that even in not very descriptive regions a corresponding point can be found quite reliably. In Table 2, the average matching error for PFH [3] and CMPs is shown. Both descriptors show good performance, although PFH can improve the matching accuracy by 2%. PFH are well-suited for finding salient points of the same object recorded from different points of view. However, CMPs are able to perform the correspondence estimation almost as exact as PFH.

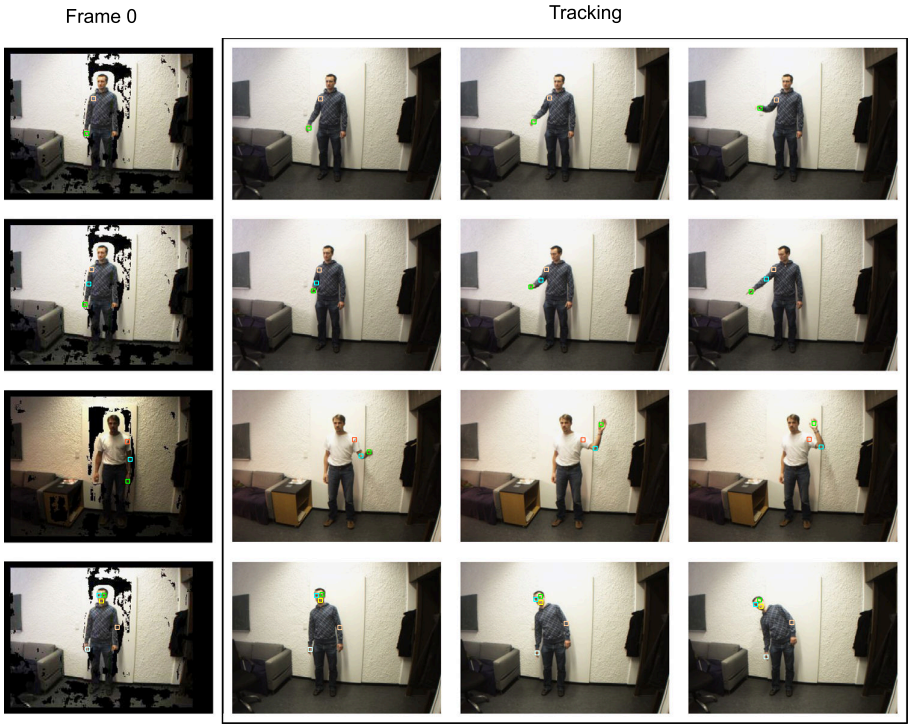

Fig. 9. Human–machine interaction gesture scenes recorded with a Bumblebee XB3 stereo camera. The black pixels in frame 0 have no depth value. Points that are selected in frame 0 are tracked with CMPs.

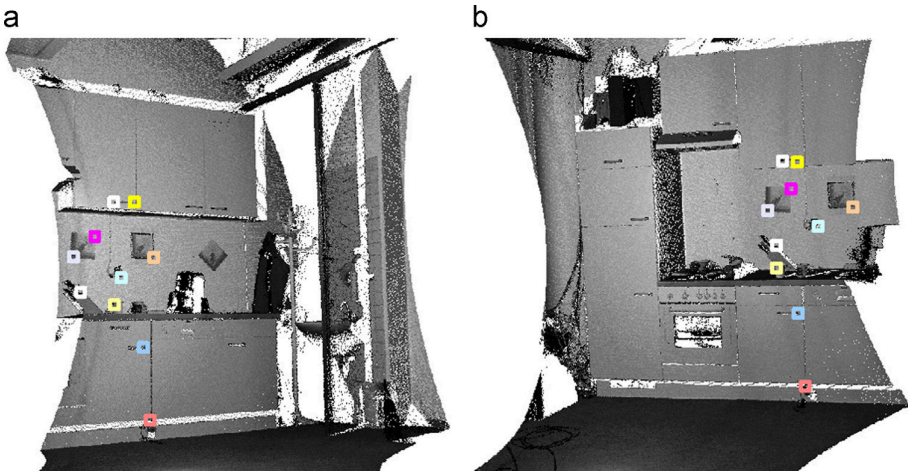

Fig. 10. Two sample scans of the kitchen scenes [26]. In (a) salient points have been selected and in (b) where the same scene is scanned from a different viewpoint the corresponding points have been found automatically with CMPs.

Fig. 11. (a) Shows the Chrysler Building, the Church of Lyon, the Mainz Cathedral, Jefferson Main Street, and the St. Mary Church of Tadcaster and (b) shows the St. Mary Church of Tadcaster from 5 different points of view.

5.1.6. Google 3D warehouse scenes

Google 3D warehouse is a rich source of user created freely available 3D scenes. The scenes can be directly downloaded as collada 3D point cloud. We selected 20 different 3D models of famous buildings, namely Jefferson Main Street (USA), Chrysler Building (USA), United States Capitol (USA), The McLeod Building (USA), 24 and 26 Lichfield Street (UK), Parish Hall of Llanelli (UK), Shire Hall (UK), St. Mary Church of Tadcaster (UK), Holst's House in Barnes (UK), Windsor Guildhall (UK), Royal Castle of Neuschwanstein (Germany), Mainz Cathedral (Germany), Dalberghaus (Germany), Taj Mahal (India), National Stadium of China (China), Mann's Chinese Theatre (China), Church of Lyon (France), Church in Venice (Italy), Catholic Church Via del Plebiscito of Rome (Italy), and Oosterbeek Church (Netherlands). Fig. 11(a) shows several samples from the dataset. For each 3D object 20 points were selected randomly. For every scene 5 different points of view (see Fig. 11 (b)) were employed and these 20 points were searched. As in the previous sections the average matching error was computed with Eq. (14). Since in Google 3D warehouse the color values are also given for the 3D points, the 2D methods can be applied for comparison, too.

The quantitative results are shown in Table 2. CMPs improve the matching accuracy by 37% and 32% compared to local optical flow and PFH, respectively. SIFT-64 and SIFT-128 are 2% and 4% better for this dataset. In general, the results are a little worse for the 3D buildings than for the other sets, which implies that this set is more challenging. Fig. 11 (a) shows several samples from the dataset. It can be seen for the building similar texture blocks are often repeated which is difficult to deal with for the matching methods. Furthermore, the points have not been selected at certain descriptive edges but just randomly which adds further complexity.

5.1.7. Summary of the quantitative evaluation

The applied datasets can be divided into two groups: Same 3D object from different view angles (kitchen scenes [26], Google 3D warehouse scenes [41]) and different 3D objects of the same class

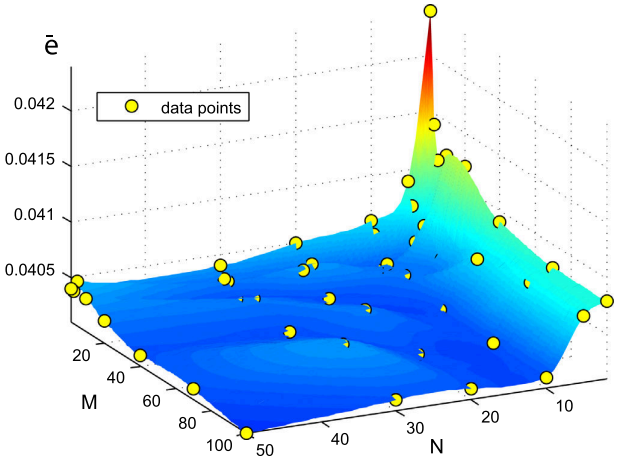

Fig. 12. Matching error $\bar{e}$ over number of particles N and number of iterations M. The error is normalized to the height of the reference face.

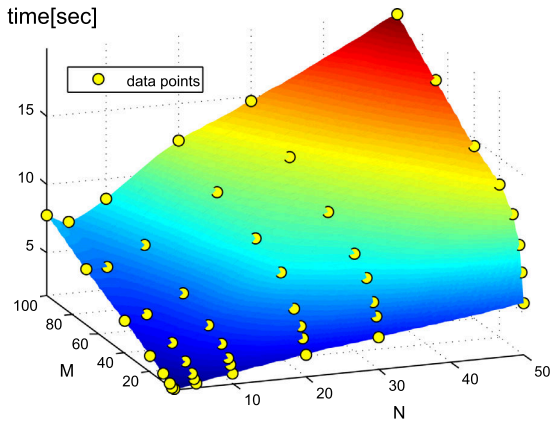

Fig. 13. Computation time per face over number of particles N and number of iterations M.

(Bosphorus database [27], BU-3DFE database [40], 3D apples, HMI gesture scenes). For the first group CMPs show similar or better performance compared to other state-of-the-art descriptors. For the second group CMPs can improve results up to 21% compared to the second best descriptor. This fact implies that CMPs are especially suitable for datasets where objects slightly differ.

5.2. *Evaluation of number of particles and iterations*

At each pyramid level a certain number N of particles and M iterations are employed. Reasonable values for N and M have to be determined. Fig. 12 shows the influence of N and M on the matching error $\bar{e}$ computed for 100 faces from the database. The graph illustrates that beyond certain values for N and M the improvement of the matching accuracy is only minor. Hence, if good performance is the only criterion, a combination of $N=20$ and $M=50$ is a good choice. However, the computation time t is linearly dependent on N and M, which is depicted in Fig. 13. The reported computation times per face are for a Matlab implementation running on an Intel Core i5 processor with 4 GB working memory. Therefore, for all reported matching errors, we have employed $N=10$ particles and $M=20$ iterations, which is a reasonable compromise between efficient computation and matching accuracy.

6. Conclusion and future work

In this article, we have presented a point descriptor for 3D point clouds, called Covariance Matrix Pyramids (CMPs). CMPs are flexible, a list of potential features that are incorporated into the description is presented but it is easy to change or expand this feature set. Results are given for a variety of 3D databases with handlabeled landmarks that are used as ground truth. The proposed method was compared to other point descriptors relying both on 2D images and 3D point clouds. CMPs show promising performance in comparison to the other state-of-the-art methods. Especially, when objects slightly differ from each other CMPs show strong performance and can improve the matching accuracy considerably.

In our ongoing research, we are working on the expansion of the feature set. Furthermore, we have planned to integrate CMPs into more sophisticated tracking methods.

Acknowledgments

This work has been partially funded by the European Projects FP-033812 (AMIDA) and FP-214901 (PROMETHEUS) as well as by an Alexander von Humboldt Fellowship for experienced researchers. We would also like to thank Dr. Rusu from Willow Garage for helping us with the PCL and Zoltan-Csaba Marton from the Intelligent Autonomous Systems Group at TUM for providing us with the kitchen scenes dataset.

Appendix A. Supplementary material

Supplementary data associated with this article can be found in the online version of http://dx.doi.org/10.1016/j.neucom.2012.06.058.

References

[1] M.F. Hansen, G.A. Atkinson, L.N. Smith, M.L. Smith, 3D face reconstructions from photometric stereo using near infrared and visible light, Comput. Vision Image Understanding 114 (8) (2010) 942–951.
[2] S. Zhuo, X. Zhang, X. Miao, T. Sim, Enhancing low light images using near infrared flash images, in: IEEE International Conference on Image Processing, 2010, pp. 2537–2540.
[3] R.B. Rusu, S. Cousins, 3D is here: Point Cloud Library (PCL), in: IEEE International Conference on Robotics and Automation, Shanghai, China, 2011.
[4] C. Tomasi, T. Kanade, Detection and Tracking of Point Features, Technical Report, Carnegie Mellon University, April 1991.
[5] D.G. Lowe, Distinctive image features from scale-invariant keypoints, Int. J. Comput. Vision 60 (2) (2004) 91–110.
[6] Y. Ke, R. Sukthankar, PCA-SIFT: a more distinctive representation for local image descriptors, in: IEEE Conference on Computer Vision and Pattern Recognition, 2004, pp. 506–513.
[7] H. Bay, A. Ess, T. Tuytelaars, L.J.V. Gool, Speeded-up robust features (SURF), Comput. Vision Image Understanding 110 (3) (2008) 346–359.
[8] K. Mikolajczyk, C. Schmid, A performance evaluation of local descriptors, IEEE Trans. Pattern Anal. Mach. Intell. 27 (10) (2005) 1615–1630.
[9] T. Brox, A. Bruhn, N. Papenberg, J. Weickert, High accuracy optical flow estimation based on a theory for warping, in: European Conference on Computer Vision, 2004, pp. 25–36.
[10] X. Shen, Y. Wu, Sparsity model for robust optical flow estimation at motion discontinuities, in: IEEE Conference on Computer Vision and Pattern Recognition, 2010, pp. 2456–2463.
[11] D. Sun, S. Roth, M.J. Black, Secrets of optical flow estimation and their principles, in: IEEE Conference on Computer Vision and Pattern Recognition, 2010, pp. 2432–2439.
[12] L. Xu, J. Jia, Y. Matsushita, Motion detail preserving optical flow estimation, in: IEEE Conference on Computer Vision and Pattern Recognition, 2010, pp. 1293–1300.
[13] Y. Pang, W. Li, Y. Yuan, J. Pan, Fully affine invariant surf for image matching, Neurocomputing 85 (2012) 6–10.
[14] Y. Pang, M. Shang, Y. Yuan, J. Pan, Scale invariant image matching using triplewise constraint and weighted voting, Neurocomputing 83 (2012) 64–71.

[15] O. Tuzel, F. Porikli, P. Meer, Region covariance: a fast descriptor for detection and classification, in: European Conference on Computer Vision, 2006, pp. 589–600.
[16] M. Kaiser, B. Kwolek, C. Staub, G. Rigoll, Registration of 3D facial surfaces using covariance matrix pyramids, in: IEEE International Conference on Robotics and Automation, 2010, pp. 1002–1007.
[17] X. Hong, H. Chang, S. Shan, X. Chen, W. Gao, Sigma set: a small second order statistical region descriptor, in: IEEE Conference on Computer Vision and Pattern Recognition, 2009, pp. 1802–1809.
[18] Y. Pang, Y. Yuan, X. Li, Gabor-based region covariance matrices for face recognition, IEEE Trans. Circuits Syst. Video Techn. 18 (7) (2008) 989–993.
[19] Y. Pang, Y. Yuan, X. Li, Effective feature extraction in high-dimensional space, IEEE Trans. Syst. Man Cybern. Part B 38 (6) (2008) 1652–1656.
[20] Y. Pang, H. Yan, Y. Yuan, K. Wang, Robust coHOG feature extraction in human-centered image/video management system, IEEE Trans. Syst. Man Cybernet. Part B 42 (2) (2012) 458–468.
[21] T.-W.R. Lo, J.P. Siebert, Local feature extraction and matching on range images: 2.5D SIFT, Comput. Vision Image Understanding 113 (12) (2009) 1235–1250.
[22] A. Frome, D. Huber, R. Kolluri, T. Bülow, J. Malik, Recognizing objects in range data using regional point descriptors, in: European Conference on Computer Vision, 2004, pp. 224–237.
[23] A. Makadia, A. Patterson, K. Daniilidis, Fully automatic registration of 3D point clouds, in: IEEE Conference on Computer Vision and Pattern Recognition, 2006, pp. 1297–1304.
[24] G.J. Brostow, J. Shotton, J. Fauqueur, R. Cipolla, Segmentation and recognition using structure from motion point clouds, in: European Conference on Computer Vision, 2008, pp. 44–57.
[25] A.E. Johnson, M. Hebert, Using spin images for efficient object recognition in cluttered 3D scenes, IEEE Trans. Pattern Anal. Mach. Intell. 21 (5) (1999) 433–449.
[26] R.B. Rusu, Z.C. Marton, N. Blodow, M. Beetz, Persistent point feature histograms for 3D point clouds, in: Proceedings of the 10th International Conference on Intelligent Autonomous Systems, Baden-Baden, Germany, 2008.
[27] A. Savran, N. Alyüz, H. Dibeklioglu, O. Çeliktutan, B. Gökberk, B. Sankur, L. Akarun, Bosphorus database for 3D face analysis, in: European Workshop on Biometrics and Identity Management, 2008, pp. 47–56.
[28] J. Schenk, M. Kaiser, G. Rigoll, Selecting features in on-line handwritten whiteboard note recognition: SFS or SFFS? in: International Conference on Document Analysis and Recognition, 2009, pp. 1251–1254.
[29] P.A. Viola, M.J. Jones, Robust real-time face detection, Int. J. Computer Vision 57 (2) (2004) 137–154.
[30] V. Arsigny, P. Fillard, X. Pennec, N. Ayache, Log-euclidean metrics for fast and simple calculus on diffusion tensors, Magn. Reson. Med. 56 (2) (2006) 411–421.
[31] J. Brummer, L. Strydom, An euclidean distance measure between covariance matrices of speech cepstra for text-independent speaker recognition, in: Symposium on Communications and Signal Processing, 1997, pp. 167–172.
[32] J. Kennedy, R. Eberhart, Particle swarm optimization, in: IEEE International Conference on Neural Networks, 1995, pp. 1942–1948.
[33] X. Hu, R. Eberhart, Solving constrained nonlinear optimization problems with particle swarm optimization, in: 6th World Multiconference on Systemics, Cybernetics and Informatics, 2002, pp. 203–206.
[34] V. Blanz, T. Vetter, Face recognition based on fitting a 3D morphable model, IEEE Trans. Pattern Anal. Mach. Intell. 25 (9) (2003) 1063–1074.
[35] B. Lévy, S. Petitjean, N. Ray, J. Maillot, Least squares conformal maps for automatic texture atlas generation, in: SIGGRAPH, 2002, pp. 362–371.
[36] Y. Wang, M. Gupta, S. Zhang, S. Wang, X. Gu, D. Samaras, P. Huang, High resolution tracking of non-rigid motion of densely sampled 3D data using harmonic maps, Int. J. Comput. Vision 76 (3) (2008) 283–300.
[37] A. Sheffer, B. Lévy, M. Mogilnitsky, A. Bogomyakov, ABF++: Fast and robust angle based flattening, ACM Trans. Graphics 24 (2) (2005) 311–330.
[38] A. Sheffer, E. Praun, K. Rose, Mesh parameterization methods and their applications, Found. Trends Comput. Graphics Vision 2 (2) (2006) 105–171.
[39] M. Clerc, J. Kennedy, The particle swarm—explosion, and convergence in a multidimensional complex space, IEEE Trans. Evol. Comput. 6 (1) (2002) 58–73.
[40] L. Yin, X. Wei, Y. Sun, J. Wang, M.J. Rosato, A 3D facial expression database for facial behavior research, in: IEEE International Conference on Automatic Face and Gesture Recognition, 2006, pp. 211–216.
[41] P. Gralla, Build Your own Model Parthenon with Google Sketchup, PC World.

Moritz Kaiser was born in 1983 in Stuttgart (Germany). He studied at the Universities of Ulm (Germany), Madrid (Spain), Notre Dame (USA), and Munich (Germany) and received his Dipl.-Ing. degree from the Technische Universität München in 2008. Currently he is working as a research assistant at the Institute for Human-Machine Communication at the Technische Universität München in order to earn his Ph.D. degree in the field of 3D face modeling and facial animation. He has 8 previous publications.

Xiao Xu was born in 1985 in Sichuan (China). He received his B.Sc. from Shanghai Jiao Tong University in June 2008 and his M.Sc. degree from the Technische Universität München in March 2011. After this he joined the Media Technology Group at the Technische Universität München in April 2011 in order to earn his Ph.D. degree.

Bogdan Kwolek was born in 1963 in Lancut (Poland). He received the M.Sc. from Rzeszow University of Technology in 1988 and his Ph.D. from AGH University of Science and Technology in Cracow in 1998. He was awarded DAAD Scholarships to Bielefeld University and Technische Universität München (Germany) and a scholarship from the French government to INRIA (France). He is an associate professor of computer science at Rzeszow University of Technology. His areas of interest include human–machine communication and multimedia information processing. He is the author of more than 50 papers in the above-mentioned fields.

Shamik Sural was born in Calcutta, India, in 1968. He is an associate professor at the School of Information Technology, IIT Kharagpur India. He received the Ph.D. degree from Jadavpur University in 2000. Before joining IIT, he held technical and managerial positions in a number of organizations both in India as well as in the USA. Dr. Sural has served on the Program Committee of many international conferences. He is a senior member of the IEEE and a recipient of the Alexander von Humboldt Fellowship for Experienced Researchers. He has published more than a hundred research papers in reputed international journals and conferences. His research interests include image processing, data mining and multimedia database systems.

Gerhard Rigoll was born in 1958 in Essen (Germany). He received the Dipl.-Ing. degree in 1982 and the Dr.-Ing. degree in 1986, both from Stuttgart University. After working as a researcher in the USA and Japan for several years, he was appointed full professor of computer science at Gerhard-Mercator-University in Duisburg in 1993. In 2002, he joined Technische Universität München, heading the Institute for Human-Machine Communication. His research interests are human–machine communication, multimedia information processing, speech, handwriting and gesture recognition, face detection and identification, emotion recognition, person tracking, information retrieval, video-indexing, and interactive computer graphics. He is the author and co-author of more than 300 papers in the above-mentioned fields.