Image and Vision Computing 32 (2014) 707–727

Contents lists available at ScienceDirect

Image and Vision Computing

journal homepage: www.elsevier.com/locate/imavis

Online learning and fusion of orientation appearance models for robust rigid object tracking ☆

Ioannis Marras [a,*], Georgios Tzimiropoulos [a,b], Stefanos Zafeiriou [a], Maja Pantic [a,c]

[a] Imperial College London, Department of Computing, SW7 2AZ, UK
[b] University of Lincoln, School of Computer Science, Lincoln LN6 7TS, UK
[c] University of Twente, Department of Computer Science, Enschede 7522 NB, The Netherlands

ARTICLE INFO

Article history:
Received 5 June 2013
Received in revised form 14 February 2014
Accepted 5 April 2014
Available online 20 May 2014

Keywords:
Rigid object tracking
Fusion of orientation appearance models
Subspace learning
Online learning
Face analysis
RGB-D

ABSTRACT

We introduce a robust framework for learning and fusing of orientation appearance models based on both texture and depth information for rigid object tracking. Our framework fuses data obtained from a standard visual camera and dense depth maps obtained by low-cost consumer depth cameras such as the Kinect. To combine these two completely different modalities, we propose to use features that do not depend on the data representation: angles. More specifically, our framework combines image gradient orientations as extracted from intensity images with the directions of surface normals computed from dense depth fields. We propose to capture the correlations between the obtained orientation appearance models using a fusion approach motivated by the original Active Appearance Models (AAMs). To incorporate these features in a learning framework, we use a robust kernel based on the Euler representation of angles which does not require off-line training, and can be efficiently implemented online. The robustness of learning from orientation appearance models is presented both theoretically and experimentally in this work. This kernel enables us to cope with gross measurement errors, missing data as well as other typical problems such as illumination changes and occlusions. By combining the proposed models with a particle filter, the proposed framework was used for performing 2D plus 3D rigid object tracking, achieving robust performance in very difficult tracking scenarios including extreme pose variations.

1. Introduction

Depth or range cameras have been developed for several years and are available to researchers for certain applications for about a decade. With the development of 3D capturing equipment, it has become faster and easier to obtain 3D shape and 2D texture information to represent a real 3D object in a scene. Subspace learning techniques have been widely used for fusing the two modalities. These techniques have provided valuable tools for understanding and capturing the intrinsic non-linear structure of visual data encountered in many important machine vision problems. At the same time, there has been a substantially increasing interest in related applications such as appearance-based object recognition and rigid object tracking. A fundamental problem of the majority of subspace learning techniques (both linear and non-linear) for appearance-based object representation is that they are not robust. In this paper, we are extending existed subspace techniques in a robust framework for learning and fusing of orientation appearance models based on both texture and depth information.

Outliers are common not only because of illumination changes, occlusions or cast shadows but also because the depth measurements provided by an depth camera could be very noisy and the obtained depth maps usually contain "holes" or missing parts. It should be mentioned here that there are several cases that should be considered as "partial occlusions", like extreme facial expressions, or when a hand/object covers partially the tracked object. Fig. 1 depicts a few examples for all these common cases by using Kinect for obtaining both texture and depth information. In contrary to the texture information, there are cases of partially object occlusions where there are no significant differences in the depth information when a hand/object touches an object, as the raw Kinect depth data is of low resolution with high noise levels. Furthermore, as it is shown in Fig. 1, there are many cases when the estimation of the nose tip in the 3D space is very possible to fail, such as cases of extreme facial expressions or when a hand covers partially the object by touching it. This is a problem for all methods for both

☆ This paper has been recommended for acceptance by Lijun Yin.

* Corresponding author at: Imperial College, Department of Computing, London, U.K.
E-mail addresses: i.marras@imperial.ac.uk (I. Marras), gt204@imperial.ac.uk (G. Tzimiropoulos), s.zafeiriou@imperial.ac.uk (S. Zafeiriou), m.pantic@imperial.ac.uk (M. Pantic).

http://dx.doi.org/10.1016/j.imavis.2014.04.017
0262-8856/

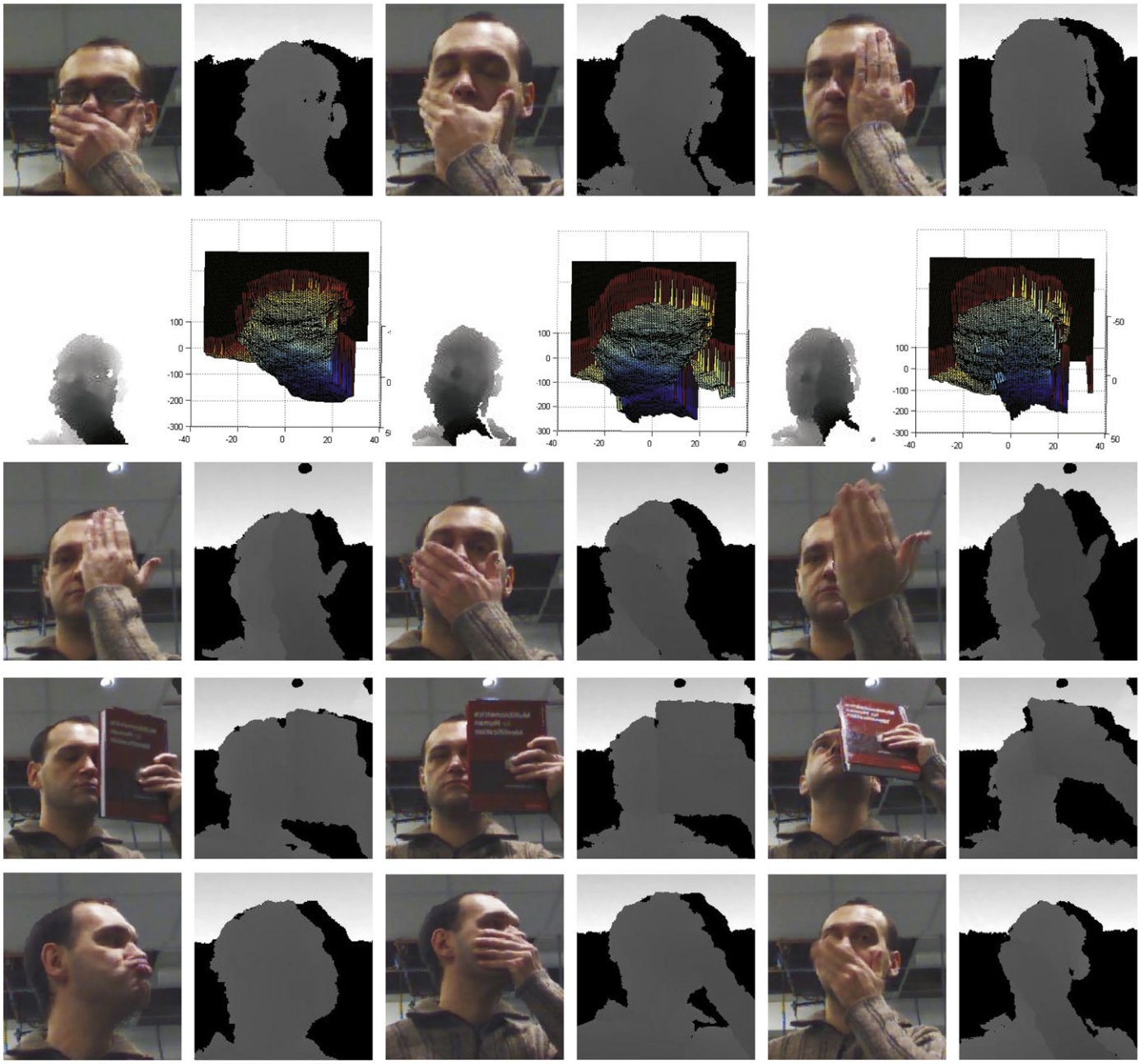

Fig. 1. Common examples of partial occluded faces, in both cropped texture and depth information by using Kinect. First row: a hand is touching the face. In contrary to the texture information, there are cases of partially object occlusions where there are no significant differences in the depth information when a hand/object touches the tracked object, as the raw Kinect depth data is of low resolution with high noise levels, Second row: depth information only for the face regions that are depicted in the first row, as well as their triangulated meshes without any mesh filtering, Third row: a hand, which is far from the face, is covering a face part. In this case, there are missing face parts inside the parallelepiped containing the face, Fourth row: in general an object is covering a face part, and Fifth row: cases where the nose tip estimation in the 3D space is not performing well, thus making a 3D tracking/pose estimation procedure, which is based on this estimation, to fail.

3D tracking/pose estimation and 3D face recognition such as [1,2] that are based on accurate nose tip estimation. In [2], the use of random regression forests for real time head pose estimation from high quality range scans, is introduced. Note that subspace learning for visual tracking requires robustness, efficiency and online adaptation. This combined problem has been rarely studied in literature. For example, in [3], the subspace is efficiently learned online using incremental ℓ_2^2 norm PCA [4]. Nevertheless, the ℓ_2^2 norm enjoys optimality properties only when image noise is independent and identically distributed (i.i.d.) Gaussian; for data corrupted by outliers, the estimated subspace can be arbitrarily skewed. On the other hand, robust reformulations of PCA [5–7] typically cannot be extended for efficient online learning.

3D shape information can be used to produce algorithms which are able to handle many challenges such as inaccurate face alignment, pose variations, measurement noise, missing data, facial expressions and partial occlusion. Many different approaches were proposed for dealing with the aforementioned problems [8,9,1,10–12]. Early approaches, such as [1], use specific face regions that are not affected by the presence of facial deformations caused by facial expressions, such as the nose and the area around it. Subspace learning algorithms for 3D mesh normals, such as Principal Component Analysis (PCA), employ low-dimensional representation of surfaces [13–16]. In its simplest form, PCA on surface normals has been applied on the concatenation of normal coordinates [14]. One attempt to exploit the special structure of normals (i.e., that lie on a sphere) was conducted in [15].

The question of how to fuse intensity with depth has been rarely addressed in tracking literature. In particular, although there are attempts to use both modalities [17,18], no particular fusion strategies have been proposed. The majority of the studies regarding fusion of intensity and depth can be found in face recognition literature, where two main lines of research can be identified, i.e. decision and feature level fusion [14,19]. Decision level fusion mainly concerns combination of scores produced from the two different modalities using off-the-shelf classifiers [20–24]. Similarly, feature level fusion is usually performed by extracting and concatenating features that are extracted from the two modalities [25], such as the magnitude of multiscale Gabor filter responses (such features are relatively computationally expensive to be computed in an on-line tracking scenario) or concatenating features that have been produced by applying a subspace analysis (such as Principal Component Analysis (PCA), Independent Component Analysis (ICA) or Nonnegative Matrix Factorization (NMF) [14,19]) on simple intensity and depth features [14,19] (which features are rarely in the same unit or correspond to similar measurements).

1.1. Visual tracking

Visual tracking aims to accurately estimate the location and possibly the orientation in 3D space of one or more objects of interests in video. Most existing methods are capable of tracking objects in well-controlled environments. However, tracking in unconstrained environments is still an unsolved problem. The definition of "unconstrained" varies with the application. For example, in unconstrained real-word face analysis, the term refers to robustness against appearance changes caused by illumination changes, occlusions, non-rigid deformations, abrupt head movements, and pose variations. For example, in surveillance from a static camera, the aim is to roughly locate and maintain the position of humans usually in crowded environments. For this purpose, tracking-by-detection with data association (see for example [26] and the references therein) has been quite a successful approach for coping with similar appearances and complicated interactions which often result in identity switches. However the usefulness of such methods for problems such as face tracking in human computer interaction where accuracy is as significant as robustness is yet to be fully appraised. Popular examples include subspace-based techniques [27,28], gradient descent [29], mixture models [30,31], discriminative models for regression and classification [32–35], and combinations of the above [3,36–40]. In [28], a method of interpreting images using an Active Appearance Model (AAM) was used. An AAM contains a statistical model of the shape and gray-level appearance of an object of interest which can generalize to almost any valid example. In [3], a tracking method that incrementally learns a low-dimensional subspace representation, efficiently adapting online to changes in the appearance of the target, was presented. The model update, based on incremental algorithms for principal component analysis, includes two important features: a method for correctly updating the sample mean, and a forgetting factor to ensure less modelling power is expended fitting older observations. Slight inaccuracies in a tracker can lead to incorrectly labelled training examples, which degrades a classifier and can cause further drift. In [34], it is showed that using Multiple Instance Learning (MIL) instead of traditional supervised learning avoids these problems, and can lead to a more robust tracker with fewer parameter tweaks. In [35], a method called Online Multi-Class LPBoost (OMCLP) which is directly applicable to multi-class problems, is presented. This algorithm tries to maximize the multi-class soft-margin of the samples. In order to solve the LP problem in online settings, an efficient variant of online convex programming, which is based on primal-dual gradient descent–ascent update strategies, is performed. In [36], a kernel method for performing a non-linear form of half-quadratic PCA (HQ-PCA) is developed to deal with non-linearly distributed data. In [26], the problem of automatically detecting and tracking a variable number of persons in complex scenes using a monocular, potentially moving, uncalibrated camera, is investigated. Previous methods for face tracking based on 3D information require an off-line training process for creating object-specific models [41,18,42,43], do not explicitly deal with outliers [42], do not cope with fast head movements [17], or require the face to be detected at each frame [2]. In [17], the problem of 3D deformable face tracking with such commodity depth cameras, was studied. A regularized maximum likelihood deformable model fitting (DMF) algorithm is developed, with special emphasis on handling the noisy input depth data.

We are interested in investigating how to incorporate 3D information provided by commercial depth cameras such as the Kinect within subspace-based methods for online appearance-based rigid face tracking. This combination appears to be very beneficial because on one hand subspace methods have been remarkably successful for maintaining a compact representation of the target object [27,37,28,40] which in many cases can be efficiently implemented online [4,39,3,36], on the other hand they appear to be susceptible to large pose variations. The main reason for this is that, in most cases, object motion is described by very simple parametric motion models such as similarity or affine warps while pose variation is incorporated into the object appearance. Clearly, it is very difficult to learn and maintain an updated model for *both* pose and appearance.[1] By using 3D information and a more accurate 3D motion model as proposed in this paper, pose and appearance are decoupled, and therefore learning and maintaining an updated model for appearance *only* are feasible by using efficient online subspace learning schemes [4]. Finally, once this subspace is learned, robust tracking can be performed by a "recognition-by-minimizing-the-reconstruction-error" approach, which has been very recently shown to be extremely discriminative [44].

In this work, an approach for learning and fusing appearance models computed from these different modalities for robust rigid object tracking, is proposed. The main problem now is how the appearance subspace can be efficiently and robustly learned and updated when data is corrupted by outliers. To achieve this task, we propose:

(1) to use features that do not depend on the data representation: angles. More specifically, our method learns orientation appearance models from image gradient orientations as extracted from intensity images and the directions of surface normals computed from dense depth fields provided by the Kinect.
(2) to incorporate these features in a robust learning framework, by using the recently proposed robust Kernel PCA method based on the Euler representation of angles [45,46]. The employed kernel enables us to cope with gross measurement errors, missing data as well as other typical problems in visual tracking such as illumination changes and occlusions. As it was shown also in [45], the kernel can be also efficiently implemented online.
(3) to capture the correlations between the learned orientation appearance models using a fusion approach motivated by the original Active Appearance Model of [28].

Thus, the proposed learning and fusing framework is robust, exact, computationally efficient and does not require off-line training. By combining the proposed models with a particle filter, the proposed tracking framework achieved robust and accurate performance in videos with non-uniform illumination, cast shadows, occlusions and most importantly large pose variations. Furthermore, during the tracking procedure the proposed framework, based on the 3D shape information, can estimate the 3D object pose something very important for numerous applications. To the best of our knowledge, this is the first time that subspace methods are employed successfully to cope with such cumbersome conditions.

[1] One of the ways to work around this problem is to generate a dense set of object instances in different poses just before the tracking is about to start; this obviously turns out to be a very tedious process.

2. Object representations of appearance models

2.1. Object representations

The shape of the object **S** is represented by a 3D triangulated mesh of n points $\mathbf{s}_k = [x\ y\ z]^T \in \Re^3$, i.e. $\mathbf{S} = [\mathbf{s}_1| \ldots |\mathbf{s}_n] \in \Re^{3\times n}$. Along with its shape, the object is represented by an intensity image $\mathbf{I}(\mathbf{u})$, where $\mathbf{u} = [u\ v]^T$ denotes pixel locations defined within a 2D texture-map. In this texture map, there is a 2D triangulated mesh each point of which is associated with a vertex of the 3D shape.

2.2. Advantages and disadvantages of the object representation

Most equipment based on active stereo vision is robust enough to illumination variations, thus the obtained 3D shape represents the actual information irrespective of lighting. Moreover, complete transformations between different 3D images can be computed in the 3D space, removing efficiently the transformation out of the image plane, which is very difficult in the 2D pixel domain [47]. In general, both texture and depth information have advantages and disadvantages. For example, in contrary to the texture information, the depth information is more robust to illumination changes (Fig. 2). The depth sensor in Kinect captures video data in 3D under any ambient light conditions. In contrary to the depth information the texture information is more robust when an object is moving far from the camera (Fig. 3). In this case, a 3D mesh denoising filtering could make the problem more difficult by producing 3D shape object representation close to planar shape representation. In contrary to the texture information, the depth information can also help to segment the 3D objects in a scene (Fig. 4), while in contrary to the depth information it is not possible for the texture information to introduce missing object parts (Fig. 5), which is a big problem for subspace learning methods. Thus, it is more powerful if those two different kinds of information are combined in a unified framework. In addition, this combination appears to be very beneficial because subspace methods have been remarkably successful for maintaining a compact representation of an object.

2.3. Appearance models

Assume that we are given a data population of m shapes and textures $\mathbf{S}i$ and $\mathbf{I}i$, $i = 1,\ldots m$. A compact way to *jointly* represent this data is to use the approach proposed in the original AAM of [28]: Principal Component Analysis (PCA) is used twice to obtain one subspace for the shapes and one for the textures. The resulted models $\left\{\overline{\mathbf{S}}_i, \mathbf{U}_S\mathbf{p}\right\}$ and $\{\overline{\mathbf{I}}_i, \mathbf{U}_I\mathbf{c}\}$ can be used to represent the *i-th* shape and texture, respectively, as

$$\hat{\mathbf{S}}_i = \overline{\mathbf{S}}_i + \mathbf{U}_S\mathbf{p}, \mathbf{p} = \mathbf{U}_S^T\left(\mathbf{S}_i - \overline{\mathbf{S}}_i\right) \text{ and } \quad \hat{\mathbf{I}}_i = \overline{\mathbf{I}}_i + \mathbf{U}_I\mathbf{c}, \mathbf{c} = \mathbf{U}_I^T(\mathbf{I}_i - \overline{\mathbf{I}}_i). \tag{1}$$

For each data sample, the embedding of its shape and texture is computed, appropriately weighted and then concatenated in a single vector. Next, a third PCA is applied to the concatenated vectors so that possible correlations between the shape and the texture are captured. There are two problems related to the above approach. First, it seems unnatural to combine the two subspaces because shape and texture are measured in different units although a heuristic to work around the problem is proposed in [28]. Second, it is assumed that data samples are outlier-free which justifies the use of standard ℓ_2^2-norm PCA. While this assumption is absolutely valid when building an AAM offline, it seems to be completely inappropriate for online learning when no control over the training data exists at all. To alleviate both problems, we propose to learn and fuse orientation appearance models. The key features of our method are summarized in the next sections.

3. Orientation features

3.1. Image gradient orientations

Given the texture **I** of an object, we covert it to grayscale image and we apply a Gaussian filtering for image de-noising and afterwards we extract image gradient orientation from

$$\Phi^g(\mathbf{u}) = \arctan\frac{\mathbf{G}_y(\mathbf{u})}{\mathbf{G}_x(\mathbf{u})}, \tag{2}$$

where $\mathbf{G}_x = \mathbf{H}_x \star \mathbf{I}$, $\mathbf{G}_y = \mathbf{H}_y \star \mathbf{I}$ and $\mathbf{H}_x$, $\mathbf{H}_y$ are the differentiation filters along the horizontal and vertical image axis respectively (Fig. 6). Possible choices for $\mathbf{H}_x$, $\mathbf{H}_y$ include central difference estimators and discrete approximations to the first derivative of the Gaussian.

3.2. Azimuth angle of surface normals

Given the depth image information of an object, we apply a Gaussian filtering for image de-noising and afterwards the triangulated 3D mesh information is created. Afterwards, we calculate the azimuth angle of surface normals (Fig. 6). Mathematically, given a continuous surface $z = f(\mathbf{x})$ defined on a lattice or a real space $\mathbf{x} = (x,y)$, normals $\mathbf{n}(\mathbf{x})$ are defined as

$$\mathbf{n}(\mathbf{x}) = \frac{1}{\sqrt{1 + \frac{\partial f^2}{\partial x} + \frac{\partial f^2}{\partial y}}}\left(-\frac{\partial f}{\partial x}, -\frac{\partial f}{\partial y}, 1\right)^T. \tag{3}$$

Normals $\mathbf{n} \in \Re^3$ do not lie on a Euclidean space but on a spherical manifold $\eta \in \mathcal{S}^2$, where $\mathcal{S}^2$ is the unit 2-sphere. On the unit sphere, the surface normal $\mathbf{n}(\mathbf{x})$ at $\mathbf{x}$ has azimuth angle defined as

$$\Phi^a(\mathbf{x}) = \arctan\frac{n_y(\mathbf{x})}{n_x(\mathbf{x})} = \arctan\frac{\frac{\partial f}{\partial y}}{\frac{\partial f}{\partial x}}. \tag{4}$$

Methods for computing the normals of surfaces can be found in [48]. In many cases, surface reconstruction methods do not recover the actual surface but the needle map. Such methods include Shape from X (SfX) and Photometric Stereo (PS) algorithms [49].

3.3. PCA of orientation appearance models

The use of a complex PCA of image gradient orientations for performing 2D face recognition robust to occlusions and illuminations was proposed in [50]. Inspired by this work and based on our previous work presented in [51], we will show how this concept can be applied also in the case of the azimuth angles of normals. We call this PCA of azimuth angles of normals as Azimuth Angle Principal Component Analysis (AAPCA). To our knowledge, this is the first time that this concept is introduced.

Let us denote by ϕ_i the n-dimensional vector obtained by writing either Φ_i^a or Φ_i^g (the orientation maps computed from shape $\mathbf{S}_i$ or texture $\mathbf{I}_i$ information, correspondingly) in lexicographic ordering. Vectors ϕ_i are difficult to use directly in optimization problems for learning. For example, writing such a vector as a linear combination of a dictionary of angles seems to be meaningless. To use angular data, we first map them onto the unit sphere by using the Euler representation of complex numbers [45]

$$\mathbf{e}(\phi_i) = \frac{1}{\sqrt{n}}\left[cos(\phi_i)^T + jsin(\phi_i)^T\right]^T, \tag{5}$$

where $\cos(\phi_i = [\cos(\phi_i(1)),\ldots, \cos(\phi_i(n))]^T)$ and $\sin(\phi_i = [\sin(\phi_i(1)),\ldots, \sin(\phi_i(n))]^T)$, $\sin(\phi_i(n))]^T$. Note that similar features have been proposed in [52], but here we avoid the normalization based on gradient

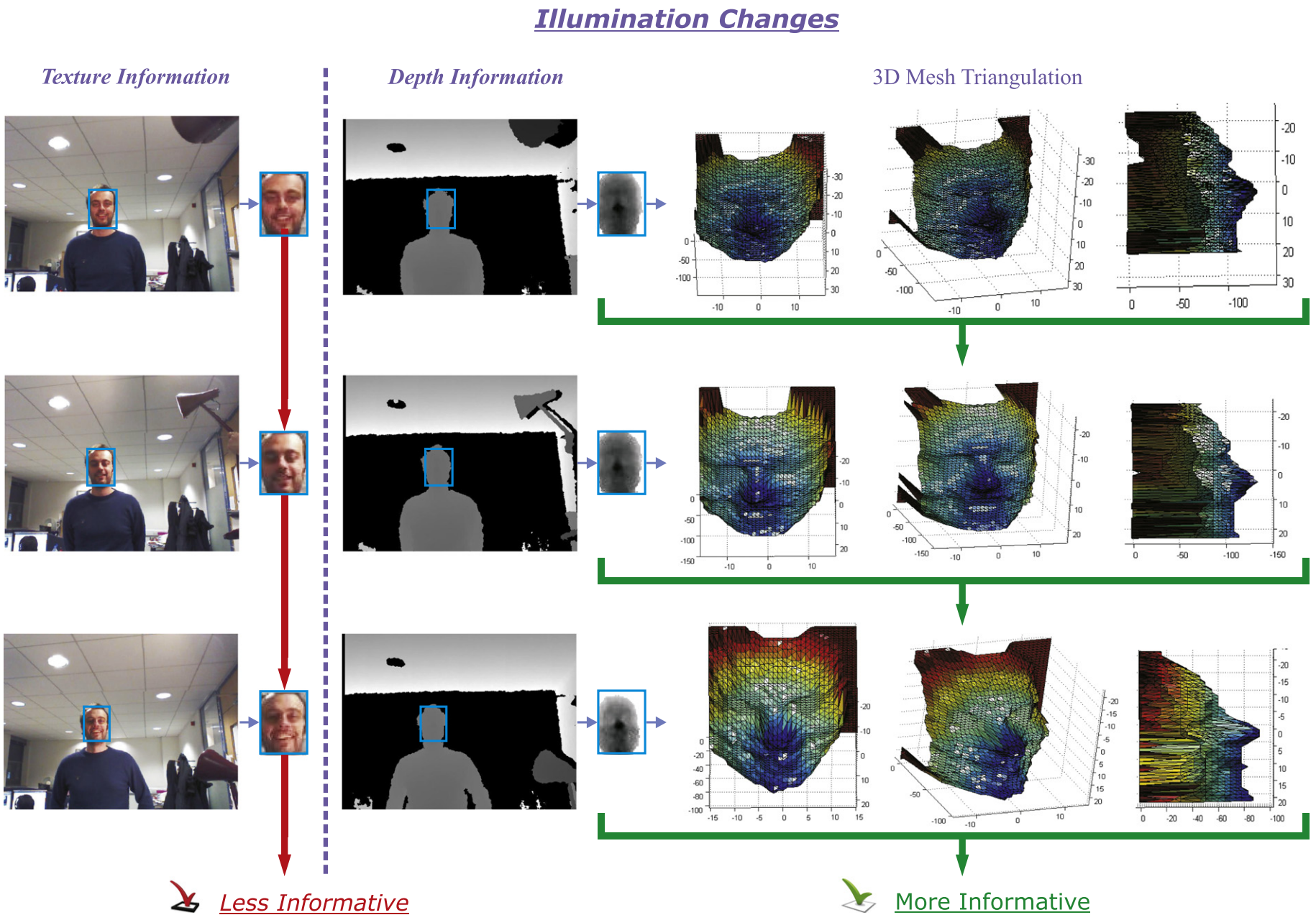

Fig. 2. In contrary to the texture information, the depth information is more robust to illumination changes.

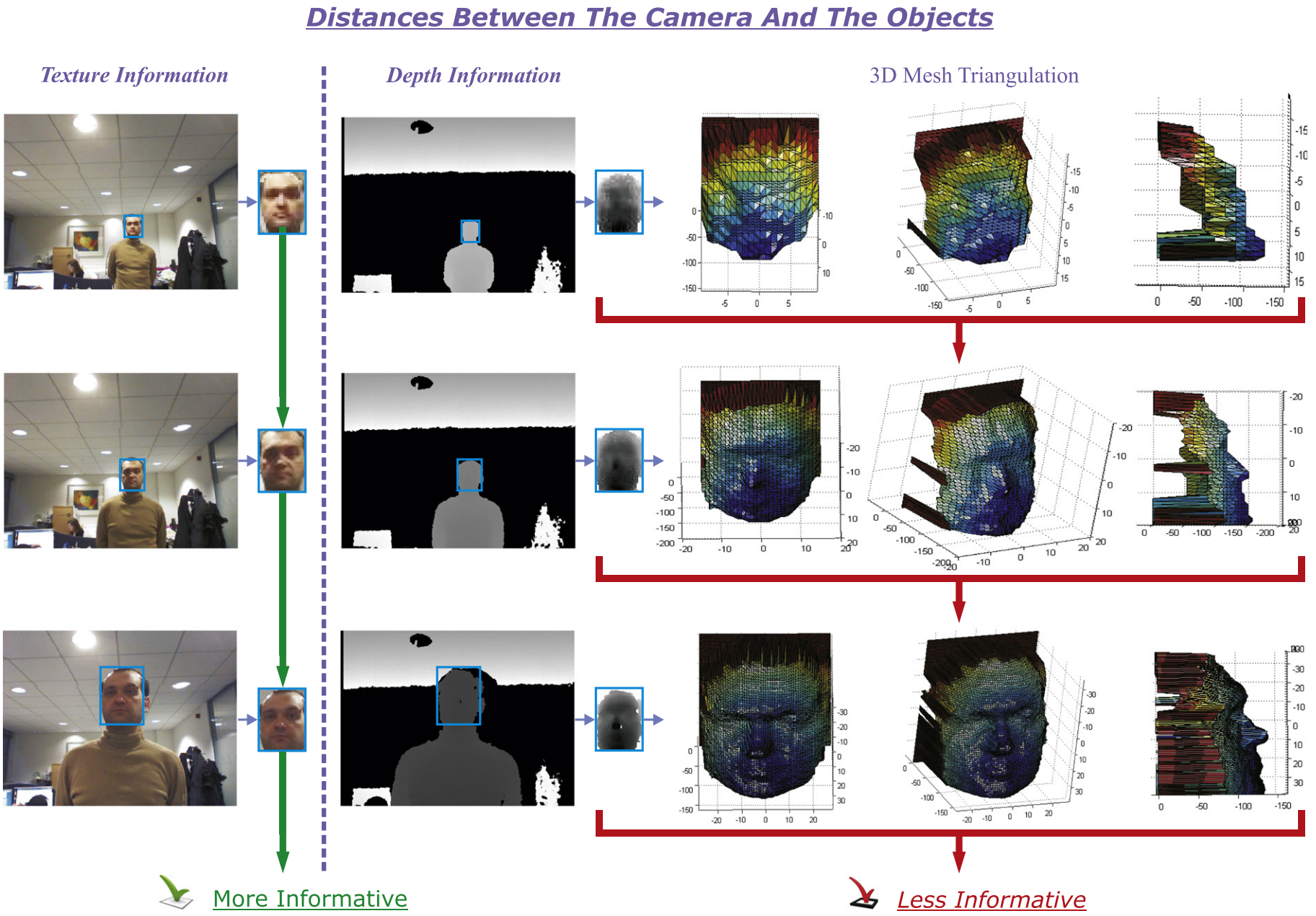

Fig. 3. In contrary to the depth information the texture information is more robust when an object is moving far from the camera. This is more obvious in the right most column where the profiles of the triangulated 3D faces are depicted.

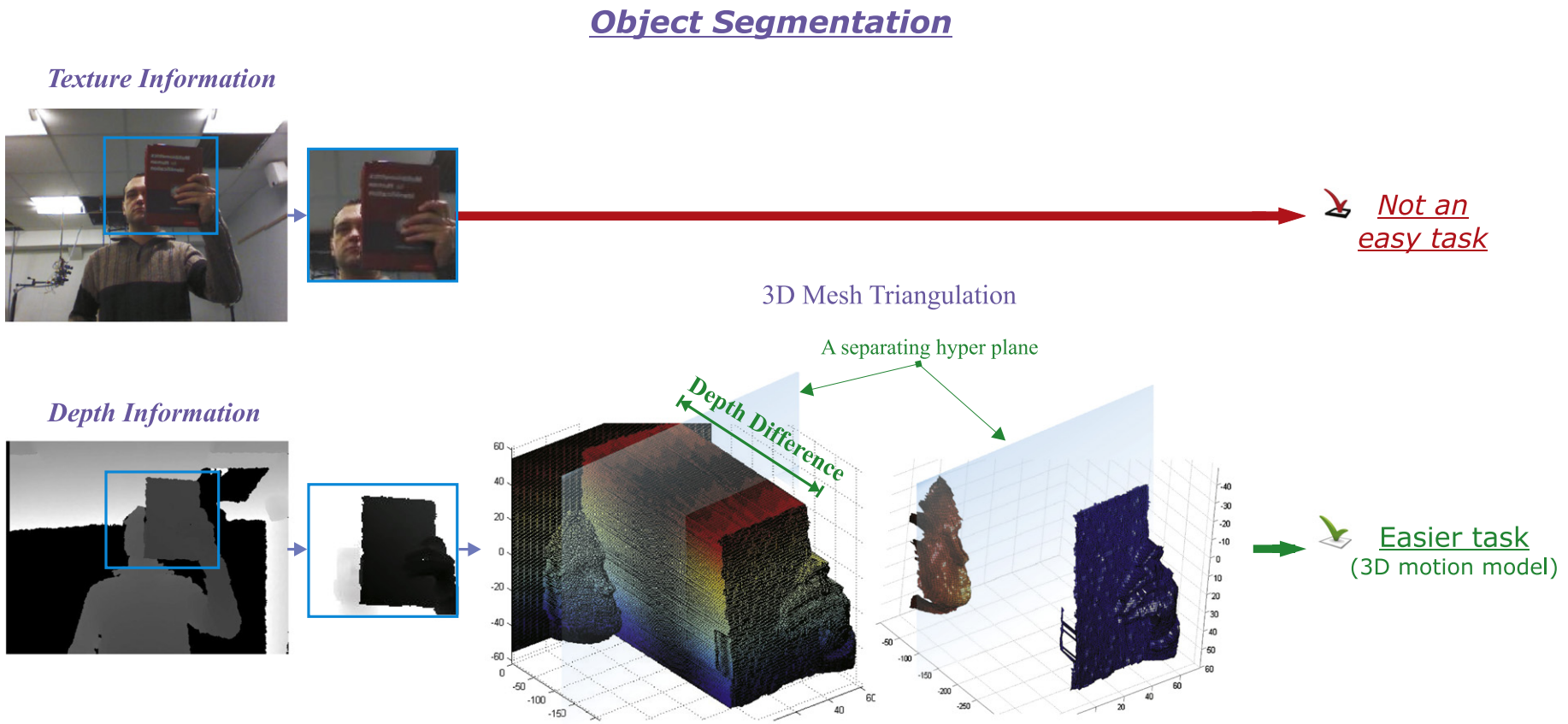

Fig. 4. In contrary to the texture information, the depth information can help to segment the 3D objects in a scene.

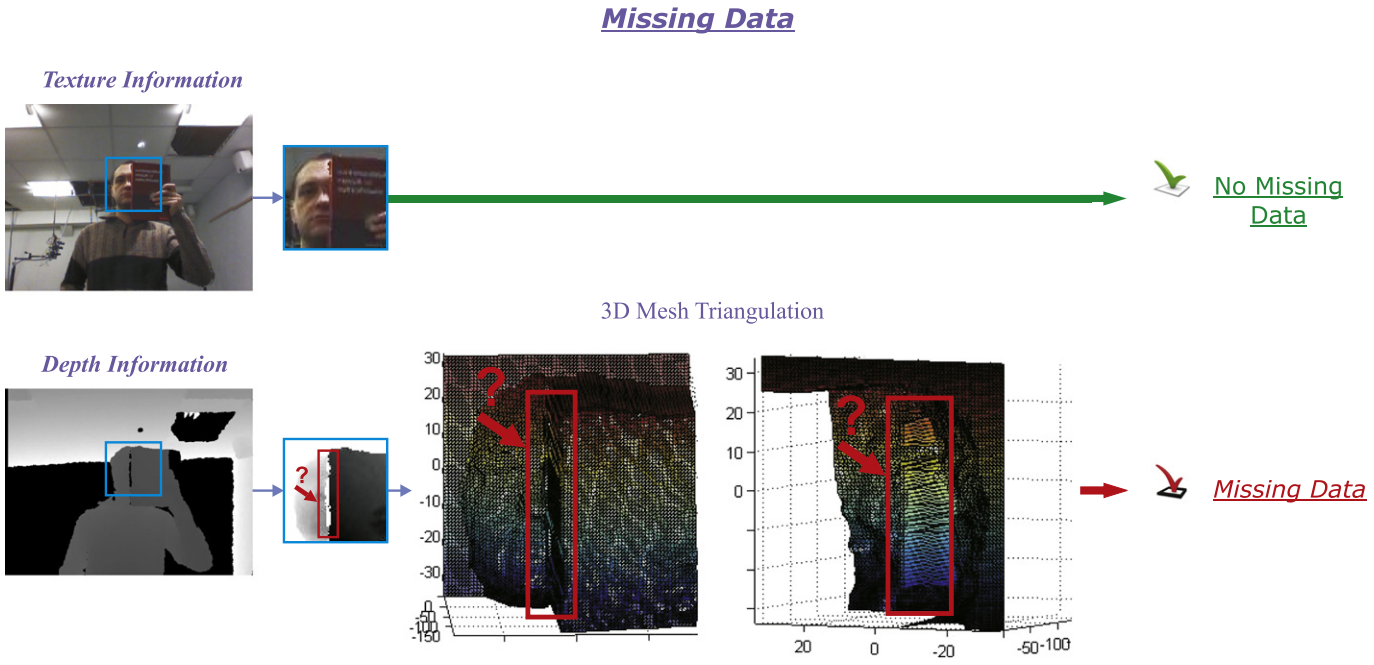

Fig. 5. In contrary to the depth information, it is not possible for the texture information to introduce missing object parts.

magnitude suggested in [52] because it makes them more sensitive to outliers and removes the kernel properties as described in [45]. Using $\mathbf{e}_i \equiv \mathbf{e}(\phi_i)$, correlation can be measured using the real part of the familiar inner product [53,45,54]

$$c\left(\mathbf{e}_i, \mathbf{e}_j\right) \triangleq \Re\left\{\mathbf{e}_i^H \mathbf{e}_j\right\} = \frac{1}{n}\sum_{k=1}^{n} cos[\Delta\phi(k)], \tag{6}$$

where $\Delta\phi \triangleq \phi_i - \phi_j$. As it can be observed, the effect of using the Euler representation is that correlation is measured by applying the *cosine kernel* to angle differences. From Eq. (6), we observe that if $\mathbf{S}_i \simeq \mathbf{S}j$ or $\mathbf{I}_i \simeq \mathbf{I}j$, then $\forall k \Delta\phi(k) \simeq 0$, and therefore $c \to 1$.

Assume now that either $\mathbf{e}_i$ or $\mathbf{e}_j$ is partially corrupted by outliers. Let us denote by P_o the region of corruption. Then, as it was shown in [45] it holds

$$\sum_{k \in P_o} cos[\Delta\phi(k)] \simeq 0, \tag{7}$$

which in turn shows that (unlike other image correlation measures such as correlation of pixel intensities) outliers vanish and do not bias arbitrarily the value of c. We refer the reader to [45] for a detailed justification of the above result for the case of image gradient orientations.

The cosine-based dissimilarity measure between two vectors of angles $\phi_i = [\phi_i(1,1) \ldots \phi_i(M_1, M_2)]^T$ and $\phi_j = [\phi_j(1,1) \ldots \phi_j(M_1, M_2)]^T$ is:

$$d^2\left(\phi_i, \phi_j\right) \triangleq \sum_{\mathbf{x}} \left\{1 - cos\left[\phi_i(\mathbf{x}) - \phi_j(\mathbf{x})\right]\right\} = \frac{1}{2}\left|\left|e^{j\phi_i} - e^{j\phi_j}\right|\right|^2, \tag{8}$$

where $e^{j\phi_i} = \left[e^{j\phi_i(1)}, \ldots, e^{j\phi_i(p)}\right]^T$ where $e^{ja} = cosa + \sqrt{-1}sina$. We define the mapping from $[0, 2\pi)^{M_1 M_2}$ to a subset of complex sphere $\mathbf{z}_i(\phi_i) = e^{j\phi_i}$ with radius $\sqrt{M_1 M_2}$. After transforming the data, PCA is applied on $\mathbf{z}_i$.

A kernel PCA based on the cosine of orientation differences for the robust estimation of orientation subspaces is obtained by using the mapping of (6) and then by applying linear complex PCA to the transformed data [45]. More specifically, we look for a set of $p < m$ orthonormal bases $\mathbf{U} = [\mathbf{u}_1| \ldots |\mathbf{u}_p] \in \mathbb{C}^{n \times p}$ by solving

$$\mathbf{U}_o = \arg\max_{\mathbf{U}} \operatorname{tr}\left[\mathbf{U}^H \mathbf{E}\mathbf{E}^H \mathbf{U}\right] \quad \text{subject to (s.t.)} \quad \mathbf{U}^H\mathbf{U} = \mathbf{I}, \tag{9}$$

where $\mathbf{E} = [\mathbf{e}_1| \ldots |\mathbf{e}_m] \in \mathbb{C}^{n \times m}$. The solution is given by the p eigenvectors of $\mathbf{E}\mathbf{E}^H$ corresponding to the p largest eigenvalues. Finally, the p-dimensional embedding $\mathbf{C} = [\mathbf{c}_1| \ldots |\mathbf{c}_n] \in \mathbb{C}^{p \times n}$ of $\mathbf{E}$ is given by $\mathbf{C} = \mathbf{U}^H\mathbf{E}$.

We denote by $\mathbf{E}^a \in \mathbb{C}^{n \times m}$ and $\mathbf{E}^g \in \mathbb{C}^{n \times m}$ the Euler representation of these two angular representations. Then, we denote the learned subspaces by $\mathbf{U}^a \in \mathbb{C}^{n \times p_a}$ and $\mathbf{U}^g \in \mathbb{C}^{n \times p_g}$ and the corresponding embeddings by $\mathbf{C}^a \in \mathbb{C}^{p_a \times m}$ and $\mathbf{C}^g \in \mathbb{C}^{p_g \times m}$ respectively.

3.4. Demonstrating the robust properties of AAPCA

For simplicity, hereinafter, we assume that we have a set $\mathcal{J} = \{\mathbf{J}_1, \ldots, \mathbf{J}_N\}$ of 3D (or 2.5D) of facial surfaces from a range camera sampled over a grid of resolution $M_1 \times M_2$. For the i-th facial surface, at each point of the grid $\mathbf{x}$, we compute the normal vector $\mathbf{G}_i = [\mathbf{n}_i(\mathbf{x})] \in \mathcal{P}^{M_1 \times M_2}$ where $\mathcal{P}$ is the pure subset of $\Re^3$ of the vector that lies on the unit sphere. In PS and SfX algorithms the set $\mathcal{G} = \{\mathbf{G}_1, \ldots, \mathbf{G}_N\}$ is computed directly.

Prior to describing our algorithm, we will briefly outline two popular PCA methodologies on surface normals that take into account the special structure of surface normals (i.e., that lie in a unit sphere). The first one is based on Azimuthal Equidistant Projection (AEP) [15] which was proposed and applied to surface normals prior to the application of PCA. The second one is based on Principal Geodesic Analysis (PGA) [13] for nonlinear statistical analysis, taking into consideration the non-Euclidean actual manifold of objects' surfaces.

- *PCA using Azimuthal Equidistant Projection (AEP-PCA)*: In order to formulate AEP we first need to define the mean elevation and azimuthal angle of $\mathcal{G}$ at each spatial location $\mathbf{x}$. In [15], the mean elevation and azimuthal angles at $\mathbf{x}$ were defined as $\widetilde{\theta}(\mathbf{x}) = \frac{\pi}{2} - \arcsin(\widetilde{n}_z(\mathbf{x}))$

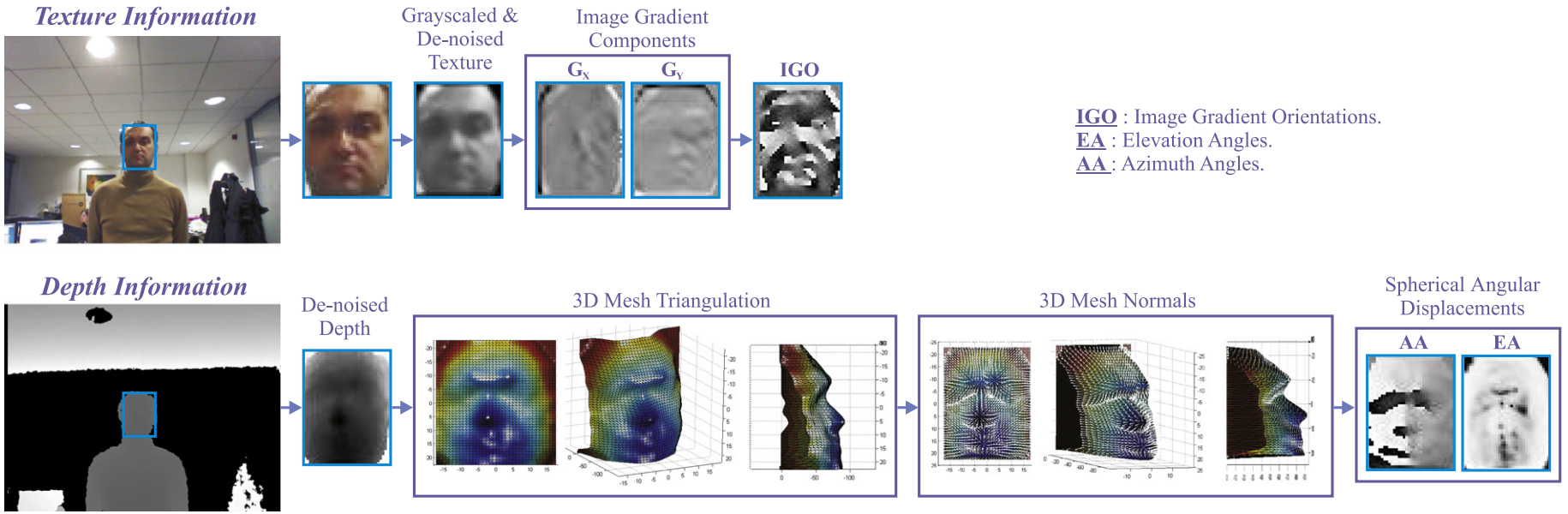

Fig. 6. Image gradient orientations and azimuth angle of surface normals of a face calculated from its texture and depth information, respectively.

and $\widetilde{\phi}(\mathbf{x}) = arctan\frac{\widetilde{n}_y(\mathbf{x})}{\widetilde{n}_x(\mathbf{x})}$, where $\widetilde{\mathbf{n}}(\mathbf{x}) = (\widetilde{n}_x(\mathbf{x}), \widetilde{n}_y(\mathbf{x}), \widetilde{n}_z(\mathbf{x}))$ is the mean representation of the normal at $\mathbf{x}$. For spherical data, such as surface normals, the intrinsic mean is in many cases represented by the spherical median [55]. For computation ease, in [15], instead of the spherical median, the average surface normal at $\mathbf{x}$ $\hat{\mathbf{n}}(\mathbf{x}) = \frac{1}{K}\sum_{i=1}^{K}\mathbf{n}_i(\mathbf{x}), \widetilde{\mathbf{n}} = \frac{\hat{\mathbf{n}}(\mathbf{x})}{||\hat{\mathbf{n}}(\mathbf{x})||}$ was used.

In order to build the AEP, the tangent plane to the unit-sphere at the location corresponding to the mean-surface normal is computed. A local coordinate system is then established on this tangent plane. The origin is the point of contact between the tangent plane and the unit sphere. The x-axis is aligned parallel to the local circle of latitude on the unit-sphere. The AEP maps the normal $\mathbf{n}_i(\mathbf{x})$ at $\mathbf{x}$ to the new vector $\mathbf{v}_i(\mathbf{x}) = (v_x^i(\mathbf{x}), v_y^i(\mathbf{x}))$ (for more details on how to compute the AEP the interested reader may refer to [15]).

After applying the AEP transform to all the samples of $\mathcal{G}$, the matrix $\mathbf{U} = [\mathbf{u}_1 \ldots \mathbf{u}_N]$ where $\mathbf{u}_i = \left[v_x^i([1,1]) \ldots v_x^i([M_1, M_2]) \;\; v_y^i([1,1]) \ldots v_y^i([M_1, M_2])\right]^T \in \Re^{M_1 M_2}$ is formed. Finally, the method [15] proceeds by computing the eigenvectors of the covariance matrix $\Sigma_{AOP} = \frac{1}{N}\mathbf{U}\mathbf{U}^T$. These eigenvectors $\mathbf{P}$ are used in order to represent facial shape and used as a prior for SfX algorithms. In order to project a novel sample to the subspace of $\mathbf{P}$, we first transform the test normal field $\mathcal{G}$ and $\widetilde{\mathbf{n}}(\mathbf{x})$ into $\mathbf{u}$ and then project using $\mathbf{b} = \mathbf{P}^T\mathbf{u}$.

- *Principal Geodesic Analysis (PGA)*: PGA is another statistical analysis method suitable for data that do not naturally lie in a Euclidean space. In standard PCA, the lower-dimensional subspaces form a linear subspace in which the data lies. In PGA, this notion is replaced by a geodesic submanifold. In other words, while each principal axis in PCA is a straight line, in PGA each principal axis is a geodesic curve. In the spherical case this corresponds to a circle. PGA utilizes the so-called log and exponential transforms in order to map the normals, that originally lie in a unit sphere, to a space where computing linear variations from the eigenanalysis of a covariance matrix could be meaningful.

In order to formulate PGA first we need to define the exponential and log maps on the sphere and a mean representation of the normals at $\mathbf{x}$. Let $\nu \in T_n\mathcal{S}^2$ be a vector on the tangent plane to $\mathcal{S}^2$ at $\eta \in \mathcal{S}^2$ and $\nu \neq 0$. The exponential map of ν, denoted by $\mathrm{Exp}_\eta(\nu)$, is the point on $\mathcal{S}^2$ along the geodesic in the direction of ν at distance $||\nu||$ from η. The log map is the inverse transform of the exponential map, that is $\mathrm{Log}_\eta(\mathrm{Exp}_\eta(\nu)) = \nu$. The geodesic distance between two points $\eta_1 \in \mathcal{S}^2$ and $\eta_2 \in \mathcal{S}^2$ can be expressed in terms of the log map, i.e. $d(\eta_1, \eta_2) = ||\mathrm{Log}_{\eta_1}(\eta_2)||$.

Instead of computing the mean of spherical directional data in [13,16], PGA finds the intrinsic mean, or the so-called spherical median $\mu = [\mu(\mathbf{x})]$ using the Exp and Log mappings. The point μ cannot be found analytically and a gradient descent procedure was used. For details on the computation of the intrinsic mean and the spherical median, the interested reader may refer to [13].

Having computed the spherical median μ it was shown that principal geodesics can be approximated by applying linear PCA on the vectors $\mathbf{u}_\mu = \left[\nu_\mu(1,1)^T \ldots \nu_\mu(M_1, M_2)^T\right]^T \in \Re^{2M_1M_2}$ where $\nu_\mu(\mathbf{x}) = \mathrm{Log}_{\mu(x)}(\mathbf{n}_i(\mathbf{x})) \in \Re^2$. After the transformation of the training set $\mathcal{G}$ into the matrix $\mathbf{U} = [\mathbf{u}_\mu^1 \ldots \mathbf{u}_\mu^N]$ and then we compute the principal components of $\Sigma_{PGA} = \frac{1}{N}\mathbf{U}\mathbf{U}^T$. A normal field of a novel sample $\mathbf{G}$ is transformed using Log_μ into $\mathbf{u}_\mu$ and then projected into the subspace $\mathbf{b} = \mathbf{P}^T\mathbf{u}_\mu$.

In order to demonstrate the robustness of the proposed AAPCA we have conducted a series of experiments using artificially generated data. More specifically, we used 21 images of one of the subjects of the FRGC v2 database [56]. The database contains 3D face scans acquired using a Minolta 910 laser scanner that produces range images with a resolution of 640 × 480 in pixels. Along with this set of images, we created a second one containing artificially occluded images. In particular, 20% of the images were artificially occluded by a 3D cloth patch placed at random spatial locations (the cloth patch has been taken from one of the images of the FRGC v2 database).

For both the original and the corrupted set, we applied standard ℓ_2^2 PCA on the depth images, PGA and AEP-PCA on the normals, and the proposed AAPCA. Then, we reconstructed the depth, the normals and the azimuth angles using the first 4 principal components of the employed ℓ_2^2 PCA, AEP-PCA, PGE, and AAPCA, respectively. Fig. 7 illustrates the quality of reconstruction for one example. The first row of Fig. 7 shows the original images of (from left to right) depth, the first two components of the normals and the azimuth angle. The second row of Fig. 7 shows (from left to right) the corresponding occluded images (by the piece of cloth). The third row of Fig. 7 shows the reconstruction of the images in the first row using the 4 principal components of the non-corrupted subspaces. That is, the first image in the third row shows the reconstruction of the non-occluded depth image (the first image in the first row) using the 4 principal components of ℓ_2^2 PCA. Similarly, the second and third images of the third row show the reconstruction of the first two components of the normals (the second and third images in the first row) using PGA (we show only the PGA result due to space limitation, similar results obtained from AEP-PCA). Finally, the fourth image of the third row shows the reconstruction of azimuth angle using AAPCA. In a similar spirit, the last row of Fig. 7 shows the reconstruction of the occluded set (second row) from the subspaces learned from the corrupted set. For this case, as we may see the reconstruction results, *except for the case of the proposed AAPCA*, suffer from artifacts.

This result is well justified by looking at the first 4 principal components of each method obtained for both the original and the occluded scenarios. For the latter case, ideally, a robust method would produce eigenvectors that match as closely as possible to the ones obtained from the former (original) case. As Fig. 8 shows, this is not the case however for ℓ_2^2 PCA using depth and PGA. More specifically, in this figure, the first and third rows show the subspace generated by the original images while the second and fourth rows show the subspace generated by the occluded data set. We may observe that in both methods, occlusions result in corrupted subspace. Fig. 8 shows also the results of the proposed AAPCA. From the sixth row, it is evident that the principal subspace appears to be artifact-free and therefore disocclusion is possible.

We also evaluated the robust performance of AAPCA quantitatively and compared it with that of PCA on depth values, AEP, and PGA. Because these methods operate on different domains, we used a performance measure which does not depend on the specific domain. More specifically, for each of these methods, we computed a measure of total similarity between the principal subspace for the noise-free case $\mathbf{U}_{\mathrm{noise-free}}$ and the principal subspace for the noisy case $\mathbf{U}_{\mathrm{noisy}}$ as follows:

$$Q = \sum_{i=1}^{k}\sum_{j=1}^{k} cos\,\alpha_{ij}, \tag{10}$$

where α_{ij} is the angle between each of the k eigenvectors defining the principal components of $\mathbf{U}_{\mathrm{noise-free}}$ and each one of $\mathbf{U}_{\mathrm{noisy}}$ [57]. The value Q lies between k (coincident spaces) and 0 (orthogonal spaces) [57]. The mean Q values over 20 repetitions of the experiment (random placement of the occlusion) and for all tested methods are depicted in Fig. 9. The mean values of Q shows that the proposed AAPCA is far more robust than all the other tested methods.

4. Fusion of orientation appearance models

As it was defined in Subsection 3.3, $\mathbf{E}^a \in \mathbb{C}^{n \times m}$, $\mathbf{E}^g \in \mathbb{C}^{n \times m}$, $\mathbf{U}^a \in \mathbb{C}^{n \times p_a}$, $\mathbf{U}^g \in \mathbb{C}^{n \times p_g}$, $\mathbf{C}^a \in \mathbb{C}^{p_a \times m}$ and $\mathbf{C}^g \in \mathbb{C}^{p_g \times m}$ are the Euler representations,

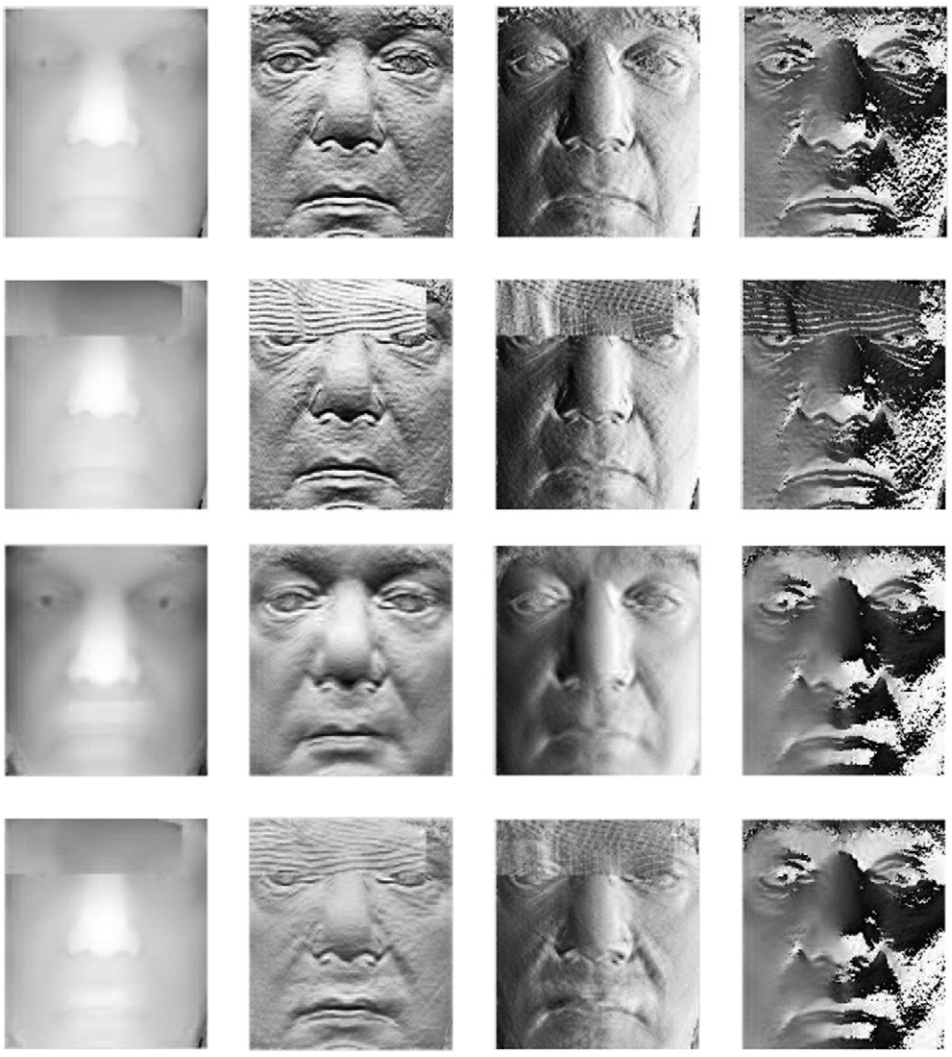

Fig. 7. Quality of reconstruction for ℓ_2^2 PCA, PGA, and AAPCA. First row: the original images of (from left to right) depth, the first two components of the normals and the azimuth angle. Second row: (from left to right) the corresponding occluded images (by the piece of cloth). Third row: the reconstruction of the images in the first row using the 4 principal components of the non-corrupted subspaces. The first image in the third row shows the reconstruction of the non-occluded depth image (the first image in the first row) using the 4 principal components of ℓ_2^2 PCA. Similarly, the second and third images of the third row show the reconstruction of the first two components of the normals (the second and third images in the first row) using PGA. Finally, the fourth image of the third row shows the reconstruction of azimuth angle using AAPCA. Fourth row: the reconstruction of the occluded set (second row) from the subspaces learned from the corrupted set. For this case, as we may see the reconstruction results, except for the case of the proposed AAPCA, suffer from artifacts.

the learned subspaces and the corresponding embeddings of two angular representations. Because $\mathbf{U}^a$ and $\mathbf{U}^g$ are learned from data (angles) measured in the same units (radians), we can capture further correlations between shapes and textures by concatenating

$$\mathbf{C} = \left[\left(\mathbf{C}^a\right)^H \ \left(\mathbf{C}^g\right)^H \right]^H, \in \mathbb{C}^{(p_a+p_g)\times m} \tag{11}$$

and then apply a further linear complex PCA on $\mathbf{C}$ to obtain a set of p_f bases $\mathbf{V} = \left[\mathbf{v}_1 | \ldots | \mathbf{v}_{p_f}\right] \in \mathbb{C}^{(p_a+p_g)\times p_f}$. Then, these bases can be used to compute p_f-dimensional embeddings $\mathbf{B} = \mathbf{V}^H\mathbf{C} \in \mathbb{C}^{p_f \times m}$ controlling the appearance of *both* orientation models. To better illustrate this fusing process, let us consider how the orientations of a test shape $\mathbf{S}_y$ and texture $\mathbf{I}_y$ denoted by $\mathbf{y} = [(\mathbf{e}_y^a)^H \ (\mathbf{e}_y^g)^H]^H$ are reconstructed by the subspace. Let us first write $\mathbf{V} = [(\mathbf{V}^a)^H \ (\mathbf{V}^g)^H]^H$. Then, the reconstruction is given by

$$\widetilde{\mathbf{y}} \approx \begin{bmatrix} \mathbf{U}^a\mathbf{V}^a \\ \mathbf{U}^g\mathbf{V}^g \end{bmatrix} \mathbf{b}_y, \tag{12}$$

where

$$\mathbf{b}_y = \mathbf{V}^H\mathbf{c}_y = \mathbf{V}^H \begin{bmatrix} \mathbf{c}_y^a \\ \mathbf{c}_y^g \end{bmatrix} = \mathbf{V}^H \begin{bmatrix} (\mathbf{U}^a)^H\mathbf{e}_y^a \\ (\mathbf{U}^g)^H\mathbf{e}_y^g \end{bmatrix}. \tag{13}$$

Thus, the coefficients $\mathbf{b}_y$ used for the reconstruction in Eq. (4), are computed from the fused subspace $\mathbf{V}$ and are common for *both* orientation appearance models as can be easily seen from Eq. (13). Finally, note that, in contrast to [28], no feature weighting is used in the proposed scheme.

4.1. Online learning

A key feature of the proposed subspace learning algorithm of orientation appearance models is that it can be implemented online which means that it can continually update the learned orientation appearance models using newly processed data. It is evident that the batch version of PCA is not suitable for this purpose because, each time, it requires to process all data (up to the current one) in order to generate an updated subspace. For this purpose, prior work [3] efficiently updates the

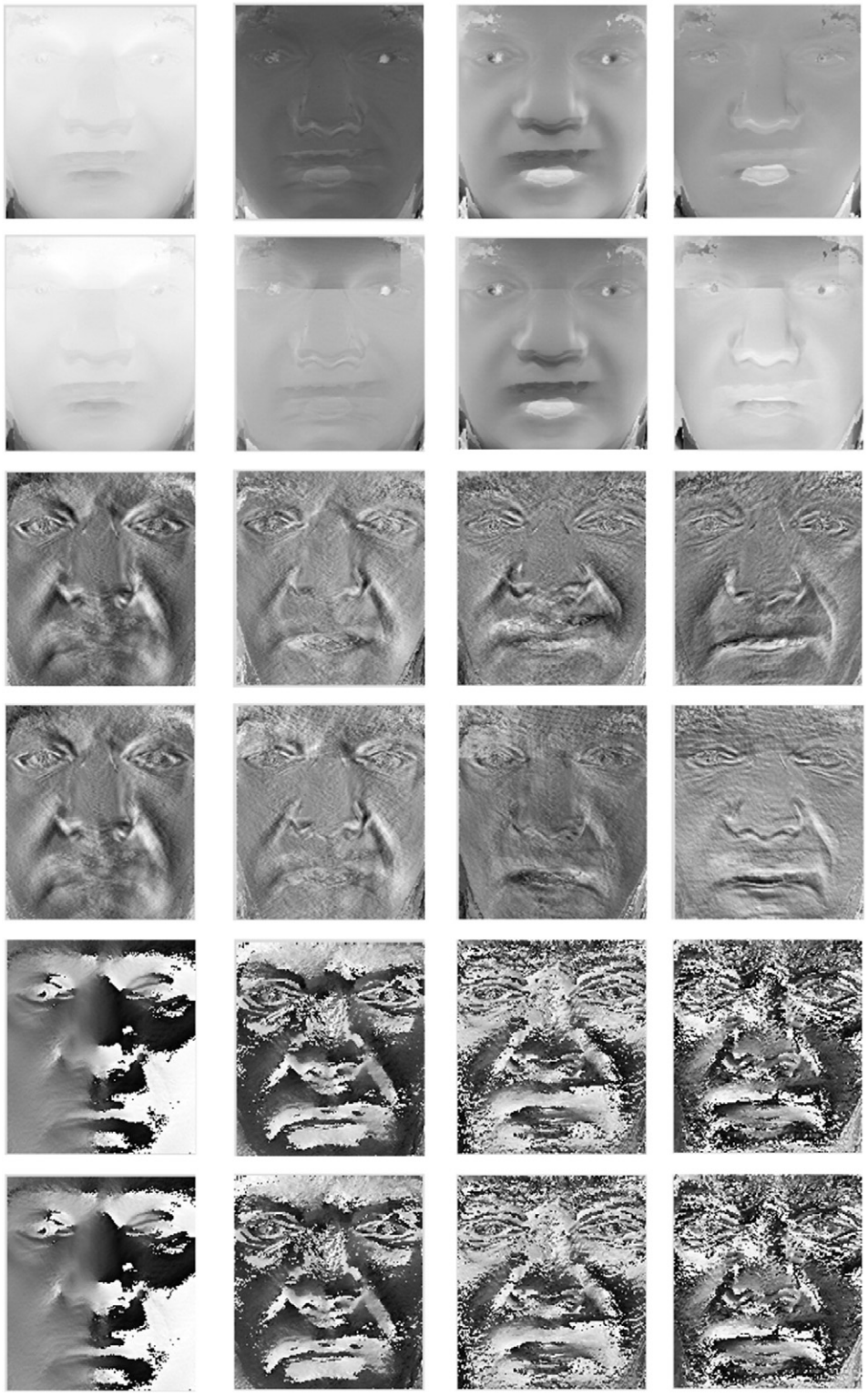

Fig. 8. The 4 principal components of ℓ_2^2 norm PCA on depth, First row: Original data, Second row: Corrupted data. The 4 principal components of PGA, Third row: Original data, Fourth row: Corrupted data. The 4 principal components of the proposed AAPCA, Fifth row: Original data, and Sixth row: Corrupted data.

subspace using the incremental ℓ_2^2 norm PCA proposed in [4]. The kernel-based extension to [4] has been proposed in [36], however the method is inexact because it requires the calculation of pre-images and, for the same reason, it is significantly slower. Fortunately, because the kernel PCA described above is direct, i.e. it employs the explicit mapping of Eq. (5), an exact and efficient solution is feasible. The proposed algorithm is summarized as follows [45].

Let us assume that, given m shapes $\{\mathbf{S}_1,\ldots,\mathbf{S}_m\}$ or textures $\{\mathbf{I}_1,\ldots,\mathbf{I}_m\}$, we have already computed the principal subspace $\mathbf{U}_m$ and $\Sigma_m = \Lambda_m^{1/2}$. Then, given l new data samples our target is to obtain $\mathbf{U}_{m+l}$ and Σ_{m+l}

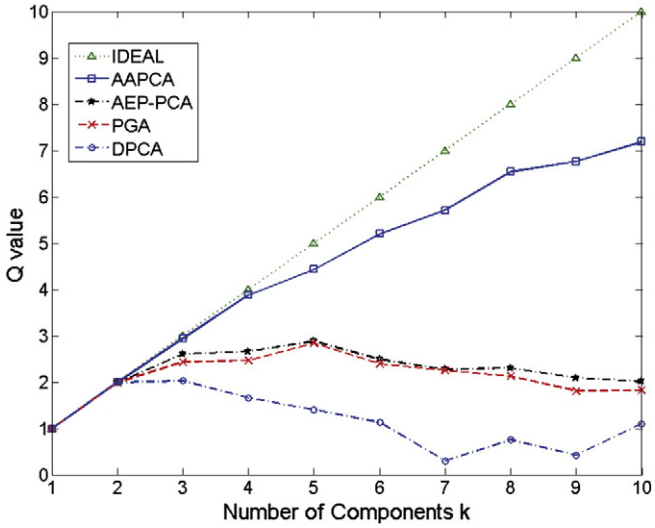

Fig. 9. The Q values obtained for all methods as a function of the number of principal components.

corresponding to $\{\mathbf{I}_1,\ldots,\mathbf{I}_{m+l}\}$ or $\{\mathbf{S}_1,\ldots,\mathbf{S}_{m+l}\}$ efficiently. The steps of the proposed incremental learning algorithm are summarized in Algorithm 1.

Algorithm 1. Online learning of orientation appearance model

Inputs: The principal subspace $\mathbf{U}_m$ and $\Sigma_m = \Lambda_m^{1/2}$, a set of new orientation maps $\{\Phi_{m+1},\ldots,\Phi_{m+l}\}$ and the number p of principal components.

Step 1. Use Eq. (5) to compute the matrix of the transformed data $\mathbf{E}_m = [\mathbf{e}_{m+1}| \ldots |\mathbf{e}_{m+l}]$.

Step 2. Compute $\widetilde{\mathbf{E}} = \text{orth}\left(\mathbf{E}-\mathbf{Q}\mathbf{Q}^H\mathbf{E}\right)$ and $\mathbf{R} = \begin{bmatrix} \Sigma_m & \mathbf{Q}^H\mathbf{E} \\ 0 & \widetilde{\mathbf{E}}^H\left(\mathbf{E}-\mathbf{Q}\mathbf{Q}^H\mathbf{E}\right) \end{bmatrix}$ (where orth performs orthogonalization).

Step 3. Compute $\mathbf{R} \overset{svd}{=} \widetilde{\mathbf{U}}\Sigma_{m+l}\widetilde{\mathbf{Y}}^H$ (where Σ_{m+l} are new singular values).

Step 4. Compute the new principal subspace $\mathbf{U}_{m+l} = \left[\mathbf{U}_m \quad \widetilde{\mathbf{E}}\right]\widetilde{\mathbf{U}}$.

In case of long sequence videos, in the online learning of the subspaces procedure a forgetting factor is used. In numerous vision applications it can be desirable to pay more attention on recently-acquired images, which are more representative regarding the current conditions (i.e. illumination changes, non-rigid deformations etc.), and less on earlier observations. For example, when tracking a target with a changing appearing, it is likely that recent observations will be more indicative of its appearance than would more distant ones. Thus, down-weighting the contribution of earlier observations plays an important role in online learning. As time progresses the observation history can become very large, to the point of overwhelming the relative contribution of each block of new data, rendering the learner "blind" to changes in the observation stream. One way to moderate the balance between old and new observations is to incorporate a forgetting factor in the incremental eigenbasis update [3].

5. Online learning of view-based orientation models for rigid object tracking

The proposed subspace learning technique for learning from orientation appearance models was used to perform 3D rigid object tracking. We combine the proposed fused orientation appearance models with a 3D motion model and standard particle filter methods for rigid object tracking [3]. The texture and depth information needed for feature extraction were provided by Kinect. By using 3D object information and an accurate 3D motion model, pose and appearance are decoupled, and therefore learning and maintaining an updated model for appearance *only* are feasible by using efficient online robust subspace learning schemes [4]. For the fusion of the orientation appearance models, the incremental ℓ_2^2 norm PCA proposed in [4] were used. When pre-learned multiple view-based subspaces are not used, for a reasonably small number of frames, all eigenspaces were generated using the batch mode of the kernel PCA of [45] and standard ℓ_2^2-norm PCA for the fusion step. When pre-learned multiple view-based subspaces are used, a batch version of PCA was used offline for the creation of the view-based subspaces. These subspaces are initially object-independent, while after a certain number of online updates during the tracking procedure, they will become object-specific. When the algorithm switches to the online mode, then for each newly tracked frame, Algorithm 1 is used to update each one of the orientation appearance eigenspaces, $\mathbf{U}^g$ and $\mathbf{U}^a$. The embedding of the new sample is also calculated which is then used to update the eigenspace $\mathbf{V}$ using the method in [4]. The motion model as well as the procedure for creation of pre-learned multiple view-based subspaces that were used in our tracking procedure, will be described later on.

5.1. Motion model

The provided 3D shape information enables us to use 3D motion models. With respect to the origin, given a set of 3D parameters the shape is first warped, $\mathbf{S}_W$, by

$$\mathbf{S}_W = \mathbf{TRS}, \quad \mathbf{R} = \mathbf{R}_\phi\mathbf{R}_\theta\mathbf{R}_\varphi, \tag{14}$$

where $\mathbf{T}$ is the translation matrix and $\mathbf{R}_\phi,\mathbf{R}_\theta,\mathbf{R}_\varphi$ are rotation matrices around the three main axes. In other words, poses of rigid body are represented as a 6 dimensional vector $\varepsilon = [T_xT_yT_z\Omega_\phi\Omega_\theta\Omega_\varphi]$, consisting of the translation parameters and the three rotation angles for each one of the three main axes. The warped shape $\mathbf{S}_W$ is then used for extracting surface normals and the corresponding azimuth angles. Finally, $\mathbf{S}_W$ is projected, $\Gamma\mathbf{S}_\mathbf{W}$, using a scale orthographic projection Γ to obtain the mapped 2D points $\mathbf{u}$. Overall, given a set of motion parameters, each vertex $\mathbf{s}_k = [x\ y\ z]^T$ of the object's shape $\mathbf{S}$ is projected to a 2D vertex. Finally, in the usual way, the texture is generated from the piecewise affine warp defined by the original 2D triangulated mesh and the one obtained after the projection. Then, this texture is used to calculate the image gradient orientations.

When a 3D motion model is used, then during the tracking procedure the 3D pose of an object can be estimated in each frame. The 3D pose of the object can be well estimated if and only if the tracking procedure performs well. Thus, a good object pose estimation is an indication of a good tracking procedure. Among the others, in our experiments we show that our approach can handle real data presenting large 3D object pose changes, partial occlusions, and facial expressions without calculation or a-priori knowledge of the camera calibration parameters. Later on we evaluate our system on a publicly available database on which we achieve state-of-the-art performance.

5.2. Multiple view-based subspace learning

Beside the fact that we have a 3D motion model and 2.5D object information it is very difficult to build the entire 3D object structure. In other words, it is very difficult to create a unique subspace of the entire 3D object based on partial views with missing parts of it. However, a view-based approach has several advantages for both texture and depth information. For example, the relative pose of constituent range observations can easily represent varying levels of detail on an object and can directly capture non-Lambertian appearance on the surface of an object [58]. In addition, the view-based eigenspaces presented in [59]

have also shown that separate eigenspaces perform better than using a combined eigenspace of the pose-varying images. This approach highly depends on the number of views chosen to sample the viewing sphere and of the accuracy of the alignment of the views. In general, the majority of systems presented in the past have shown that separating the shape information from the texture information yields additional performance enhancements. Based on the above conclusions, multi-view orientation appearance 2D models that describe shape and texture variations in certain poses, were constructed in this work. For example, in case of tracking of a human face, we can initially generate prior person-independent view-based subspaces under a fixed lighting condition. More specifically, the subspaces for the adjacent views were produced based on 200 different aligned faces. In Fig. 10 an example of both texture and depth face pose information for one person is depicted. The adjacent views were separated by 15° for both pitch and yaw rotations, ignoring the roll rotation since there is no significant changes in this case. The purpose was these discrete subspaces to cover rotations in the range of [−60° 60°], for both pitch and yaw rotations. For each view i, we define its set of parameters, E_i, as

$$E_i = \{\mathbf{U}_i^g, \mathbf{U}_i^a, \mathbf{V}_i, \varepsilon_i\}, \tag{15}$$

where ε_i is the pose, $\mathbf{U}_i^g$, $\mathbf{U}_i^a$ and $\mathbf{V}_i$ are the subspaces for orientation of gradients, of 3D mesh normals and of their fusion (see Section 4), respectively, for this view. For each one of the 81 discrete face poses, i, the set of parameters E_i was created. The dimensions of the subspaces for all poses were the same, while the image resolution was 60×60 pixels. As an alternative option, the morphable model presented in [60] was used for creation of the multiple view-based subspaces, by producing 24.200 facial aligned images for the same 81 discrete poses that correspond to 200 different persons.

5.3. Tracking with orientation appearance models

During the tracking initialization stage, the user has to define ε at the beginning, i.e. the center position of the object to be tracked in the 3D space, the parallelepiped contains the 2.5D information of the tracked object based on its center, as well as its orientation. In Fig. 11 the main steps of the rigid object tracking procedure, are depicted.

In general, a particle filter calculates the posterior distribution of a system's states based on a transition model and an observation model. In our tracking framework, the transition model is described as a Gaussian Mixture Model around an approximation of the state posterior distribution of the previous time step:

$$p\left(M_t^i, M_{t-1}^{1:P}\right) = \sum_{i=1}^{P} w_{t-1}^i \mathcal{N}\left(M_t; M_{t-1}^i, \Xi\right) \tag{16}$$

where M_t^i is the 3D motion defined by particle i at time t, $M_{t-1}^{1:P}$ is the set of P transformations of the previous time step, the weights of which are denoted by $w_{t-1}^{1:P}$, and Ξ is a diagonal covariance matrix. In the first phase, P particles are drawn. In the second phase, the observation model is applied to estimate the weighting for the next iteration (the weights are normalized to ensure $\sum_{i=1}^{P} w_t^i = 1$). Furthermore, the most probable sample is selected as the state M_t^{best} at time t. Thus, the estimation of the posterior distribution is an incremental process and utilizes a hidden Markov model which only relies on the previous time step.

Finally, our observation model computes the probability of a sample being generated by the learned orientation appearance model. More specifically, we follow a "recognition-by-minimizing-the-reconstruction-error" approach, which has been very recently shown to be extremely discriminative for the application of face recognition in [44], and model this probability as

$$p\left(\mathbf{y}_t^i | \mathbf{M}_t^i\right) \propto e^{\frac{\|\mathbf{y}_t^i - \tilde{\mathbf{y}}_t^i\|_f^2}{\sigma}}, \tag{17}$$

where $\tilde{\mathbf{y}}_t^i$ is given by Eq. (13).

In case multiple view-based subspaces were used, during the tracking procedure only these pre-learned subspaces that correspond to the

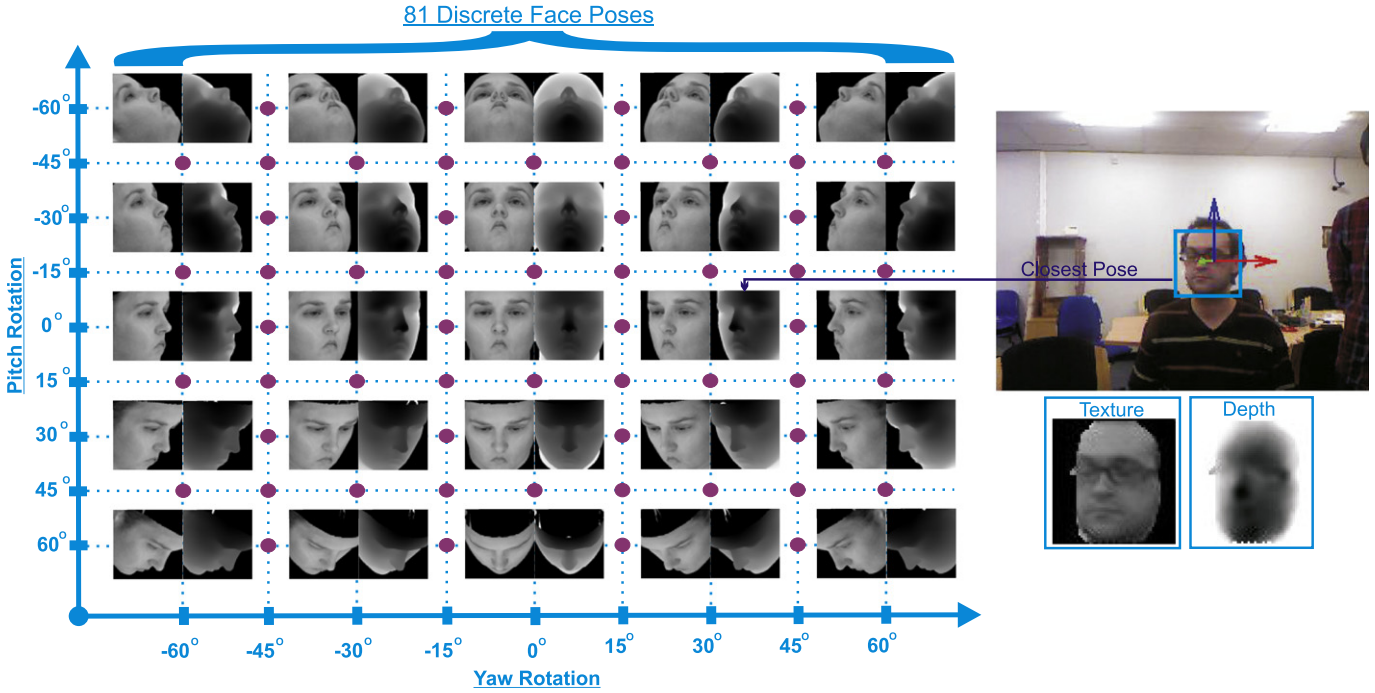

Fig. 10. An example of both texture and depth face pose information for one person that used for the offline creation of the multiple view-based subspaces for a face tracking procedure.

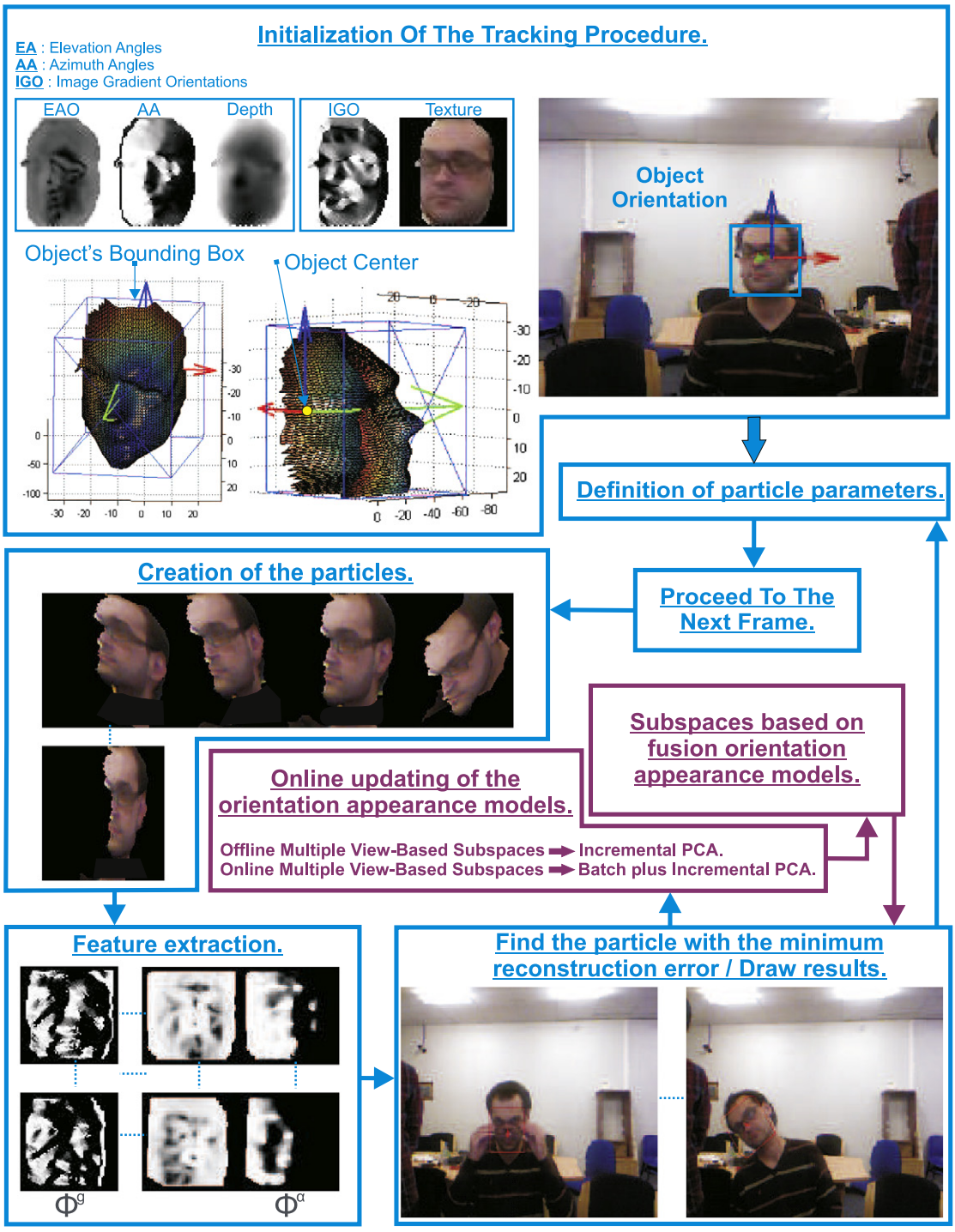

Fig. 11. The main steps of the rigid object tracking procedure.

four closest face poses, based on pitch and yaw angles, were used in each particle in order to estimate the position and the orientation of the tracked object, while this cross matching enhances the stability and accuracy of the tracking procedure.

6. Experimental results

Evaluating and comparing different tracking approaches are rather tedious tasks. A fair comparison requires not only a faithful reproduction of the original implementation but also tweaking of the related parameters and training on similar data. In this work, we chose to evaluate the proposed algorithm and compare it with (a) similar subspace-based techniques and (b) the state-of-the-art method of [2]. For the purposes of (a), we used the following variants of the proposed scheme:

(1) 3D motion model + image gradient orientations only. We call this tracker 3D + IGO.
(2) 3D motion model + azimuth angles only. We coin this tracker 3D + AA.

Table 1
Experimental results for the Biwi Kinect Head Pose Database. Mean and standard deviations of the angular errors are shown together. The last column shows the percentage of images where the angular error was below 10°. In bold are shown the results of the proposed method.

Methods	Yaw error	Pitch error	Roll error	Direction estimation accuracy
Method in [2]	11 ± 12.1°	9.9 ± 10.8°	9.1 ± 10.1°	81.0%
3D + IGO + AA	**9.2 ± 13.0°**	**9.0 ± 11.1°**	**8.0 ± 10.3°**	**89.9%**

Table 2
Mean S values for all trackers and videos. The proposed tracker is coined 3D + IGO + AA. In bold are shown the results of the proposed method.

	3D + IGO	3D + AA	**3D + IGO + AA**	3D + I + D	2D + IGO
Video 1	0.1822	0.2645	**0.1598**	0.8644	0.9221
Video 2	0.1827	0.1572	**0.1127**	0.2760	0.3912
Video 3	0.2884	0.4254	**0.2531**	0.9081	0.9001

(3) 3D motion model + fusion of image gradient orientations with azimuth angles. This is basically the tracker proposed in this work. We call this tracker 3D + IGO + AA.
(4) 2D motion model + image gradient orientations only. We call this tracker 2D + IGO.

We additionally used 3D motion model + fusion of pixel intensities with depth. We coin this tracker 3D + I + D. This tracker is particularly included for performing comparison with standard ℓ_2^2-norm PCA methods. A simplified version of this tracker which uses 2D motion and pixel intensities only has been proposed in [3].

To compare all above variants of subspace-based tracking techniques, we used 3 representative videos. The first video contains face expressions. The second video contains extreme face pose variations of two subjects while illumination variations appear only for the second subject. The third video contains face occlusions with extreme pose variations. All parameters related to the generation of particles remained constant for all methods and videos. In this way, we attempted to isolate only the motion model and the appearance model used, so that concrete conclusions can be drawn. Finally, we evaluated all trackers using a 2D bounding box surrounding the face region. This is the standard approach used in 2D tracking; we followed a similar approach because of its ease to generate ground truth data and in order to be able to compare with trackers using 2D motion models. We measure tracking accuracy from $S = 1 - \frac{\#\{D \cap G\}}{\#\{D \cup G\}}$, where D and G denote the detected and manually annotated bounding boxes and respectively, and # {} is the number of pixels in the set (the smallest S is the more overlap we have). Table 2 shows the mean values of S for all trackers and videos respectively. Figs. 12, 13 and 14 plot S for all methods and videos as a function of the frame number. Figs. 15, 16 and 17 illustrate the

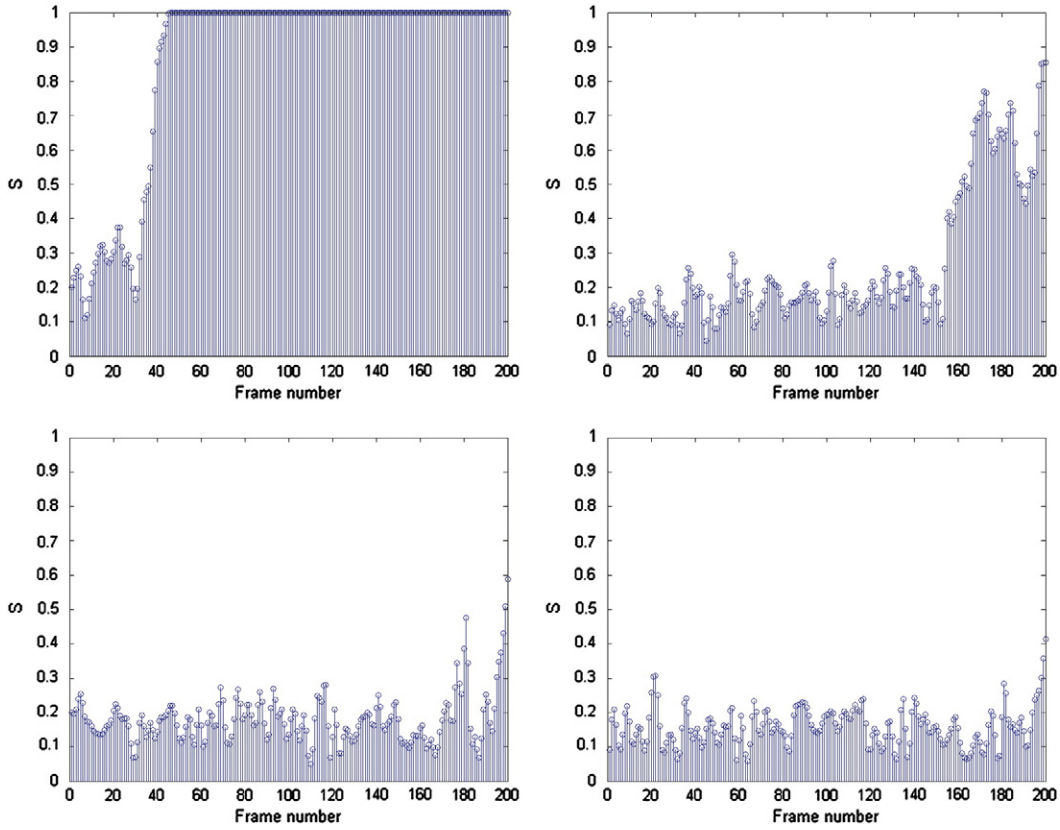

Fig. 12. S value vs the number of frames for the first video. First row: First image: 3D + I + D. Second image: 3D + AA. Second row: First image: 3D + IGO. Second image: 3D + IGO + AA.

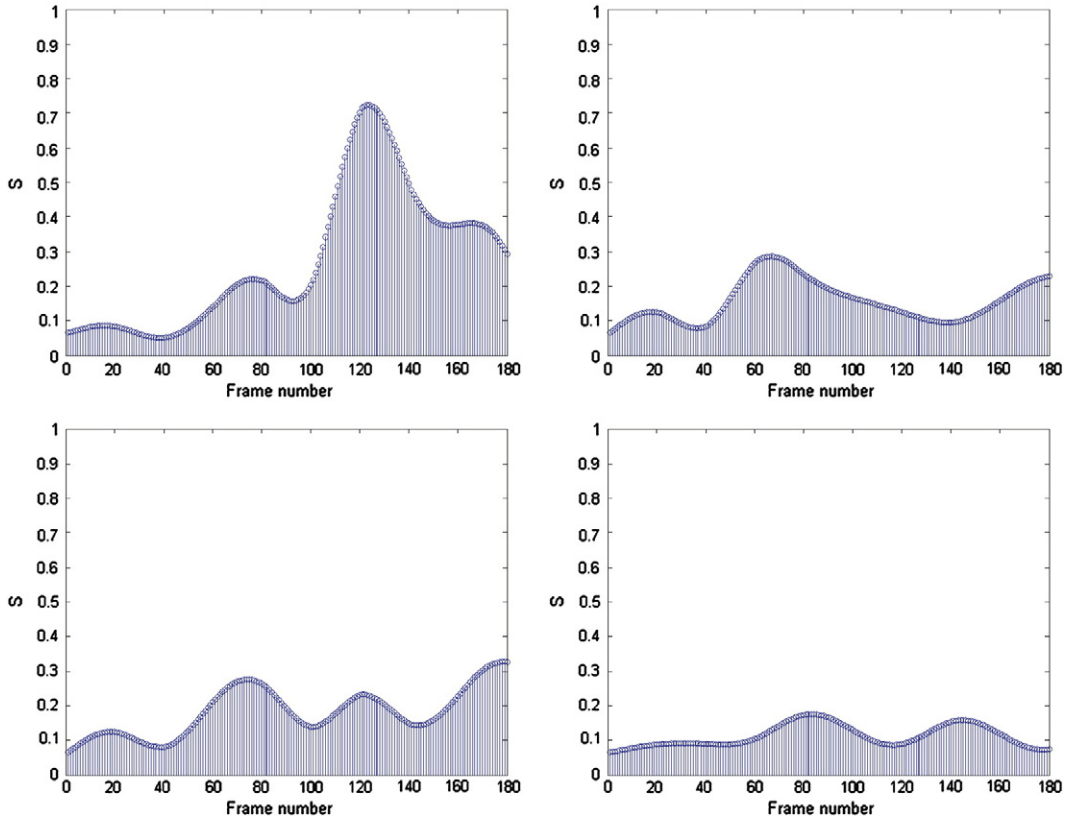

Fig. 13. *S* value vs the number of frames for the second video. First row: First image: 3D + I + D. Second image: 3D + AA. Second row: First image: 3D + IGO. Second image: 3D + IGO + AA.

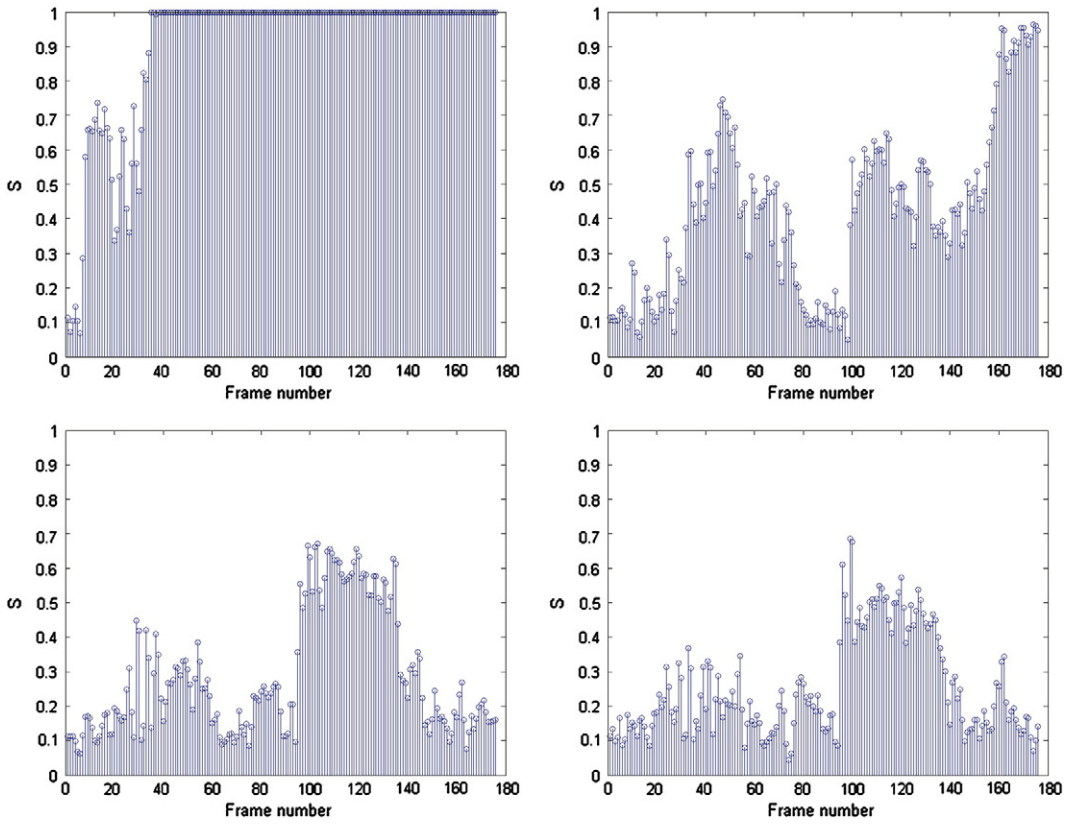

Fig. 14. *S* value vs the number of frames for the third video. First row: First image: 3D + I + D. Second image: 3D + AA. Second row: First image: 3D + IGO. Second image: 3D + IGO + AA.

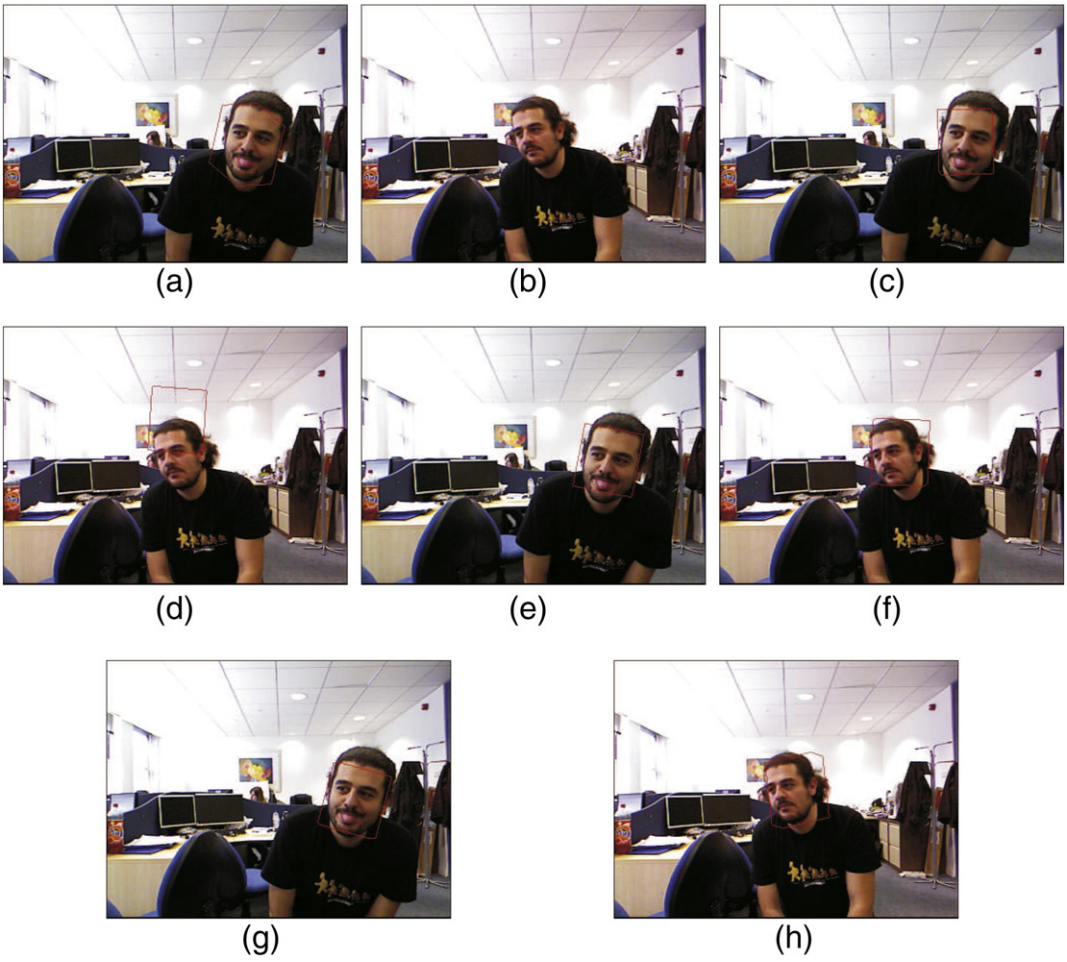

Fig. 15. Tracking examples from the first video. (a),(b): 3D + I + D. (c),(d): 3D + AA. (e),(f): 3D + IGO. (g),(h): 3D + IGO + AA.

Fig. 16. Tracking examples for the second video. (a): 3D + I + D, (b): 3D + AA, (c): 3D + IGO, (d): 3D + IGO + AA.

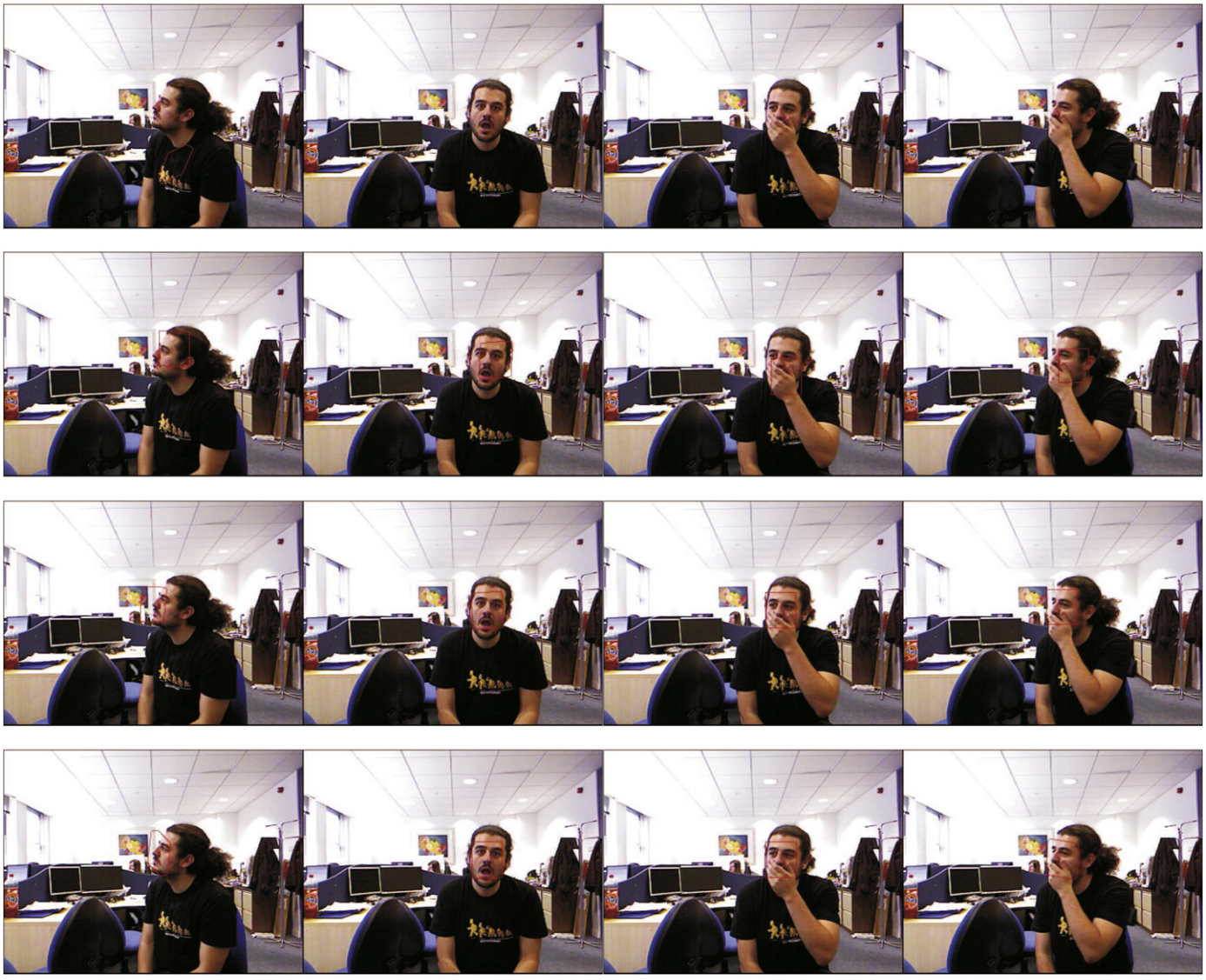

Fig. 17. Tracking examples for the third video. First row: 3D + I + D. Second row: 3D + IGO. Third row: 3D + AA. Fourth row: 3D + IGO + AA.

performance of the proposed tracker for some cumbersome tracking conditions. Finally, Fig. 18 depicts examples of the proposed method tracking and face pose estimation performance on specific frames derived by the second video.

By exploiting the 3D motion model, the proposed framework was used to estimate, during the tracking procedure, the center and the rotation angles of the tracked object in the 3D space. In order to assess the performance of our algorithm, we used the Biwi Kinect Head Pose Database [61,62]. The dataset contains over 15 K images of 20 people (6 females and 14 males — 4 people were recorded twice) recorded while sitting about 1 m away from the sensor. For each frame, a depth image, the corresponding texture image (both 640 × 480 pixels), and the annotation is provided. The head pose range covers about ±75 degrees yaw and ±60 degrees pitch. The subjects were asked to rotate their heads trying to span all possible ranges of angles their head is capable of. Ground truth is provided in the form of the 3D location of the head and its rotation. In this database, the texture data are not aligned with the depth data, while in many videos the problem of the frame dropping exists. Because of that, we were able to test our method only on 10 videos in which the misalignment difference of texture and depth in pixels was almost constant and the number of the dropped frames was quite small. The best configuration of our method (3D +IGO + AA) was compared to the state-of-the art method presented in [2] which is based on discriminative random regression forests: ensembles of random trees trained by splitting each node so as to simultaneously reduce the entropy of the class labels distribution and the variance of the head position and orientation. The results are given in Table 1, where mean and standard deviations of the angular errors are shown together. The last column shows the percentage of images where the angular error was below 10°. Since the head pose estimation is a hard procedure, for these experiments more than one subspace were created based on the tracked object. More specifically, based on the tracked object one subspace was created for discrete face poses, i.e. for every 15° of yaw or/and roll rotations.

From our results, we verify some of our speculations in the introduction section. More specifically, from our results below it is evident that:

(1) 3D motion models + subspace learning outperforms 2D motion models + subspace learning, especially for the case of large pose variations. This proves our argument that decoupling pose from appearance greatly benefits appearance-based tracking.
(2) 3D motion models + subspace learning works particularly well when only learning is performed in a robust manner. This is

Fig. 18. Examples of the proposed method tracking and face pose estimation performance on specific frames derived by the second video.

illustrated by the performance of the proposed combinations: 3D + IGO, 3D + AA, 3D + IGO + AA.

(3) The proposed fusion scheme 3D + IGO + AA performs the best among all subspace-based methods and outperforms even the state-of-the-art method [2]. This justifies the motivation behind the proposed scheme.

Furthermore, the time consuming procedure in the proposed framework is affected by the creation of the particles. However, its implementation in CUDA™ is a real time procedure. Therefore, the proposed method can be used as a real time tracking procedure.

7. Conclusion

We proposed a learning and fusing framework for multimodal visual tracking that is robust, exact, computationally efficient and does not require off-line training. Our method learns orientation appearance models from image gradient orientations and the directions of surface normals. These features are incorporated in a robust learning framework, by using a robust Kernel PCA method based on the Euler representation of angles which enables an efficient online implementation. Finally, our method captures the correlations between the learned orientation appearance models using a fusion approach motivated by the original AAM. By combining the proposed models with a particle filter, the proposed tracking framework achieved robust and accurate performance in videos with non-uniform illumination, cast shadows, significant pose variation and occlusions. To the best of our knowledge, this is the first time that subspace methods are employed successfully to cope with such cumbersome conditions.

Acknowledgments

The research presented in this paper has been funded by the European Community 7th Framework Programme [FP7/2007–2013] under grant agreement no. 288235 (FROG). The work of Maja Pantic is funded by the European Research Council under the ERC Starting Grant agreement no. ERC-2007-StG-203143 (MAHNOB). The work of Stefanos Zafeiriou and partially the work of Ioannis Marras were funded by the EPSRC project EP/J017787/1 (4D-FAB).

References

[1] K. Chang, K. Bowyer, P. Flynn, Multiple nose region matching for 3D face recognition under varying facial expression, IEEE Trans. Pattern Anal. Mach. Intell. (2006) 1695–1700.
[2] G. Fanelli, J. Gall, L.V. Gool, Real time head pose estimation with random regression forests, IEEE Conference On Computer Vision and Pattern Recognition (CVPR), 2011, pp. 617–624.
[3] D. Ross, J. Lim, R.S. Lin, M.H. Yang, Incremental learning for robust visual tracking, Int. J. Comput. Vis. 77 (2008) 125–141.
[4] A. Levy, M. Lindenbaum, Sequential Karhunen–Loeve basis extraction and its application to images, IEEE Trans. Image Process. 9 (2000) 1371–1374.
[5] F. de la Torre, M. Black, A framework for robust subspace learning, Int. J. Comput. Vis. 54 (2003) 117–142.
[6] N. Kwak, Principal component analysis based on L1-norm maximization, IEEE Trans. Pattern Anal. Mach. Intell. 30 (2008) 1672–1680.
[7] E. Candès, X. Li, Y. Ma, J. Wright, Robust principal component analysis&quest, J. ACM 58 (3) (2011) 11.
[8] K. Bowyer, K. Chang, P. Flynn, A survey of approaches and challenges in 3D and multi-modal 3D + 2D face recognition, CVIU 101 (1) (2006) 1–15.
[9] S. Berretti, A. Del Bimbo, P. Pala, 3D face recognition using isogeodesic stripes, IEEE Trans. Pattern Anal. Mach. Intell. 32 (12) (2010) 2162–2177.
[10] T. Faltemier, K. Bowyer, P. Flynn, A region ensemble for 3-d face recognition, IEEE TIFS 3 (1) (2008) 62–73.
[11] A. Bronstein, M. Bronstein, R. Kimmel, Expression-invariant representations of faces, IEEE TIP 16 (1) (2007) 188–197.
[12] I.A. Kakadiaris, G. Passalis, G. Toderici, N. Murtuza, Y. Lu, N. Karampatziakis, T. Theoharis, Recognition in the presence of facial expressions: an annotated deformable model approach, IEEE Trans. Pattern Anal. Mach. Intell. 29 (4) (2007) 640–649.
[13] P.T. Fletcher, C. Lu, S.M. Pizer, S. Joshi, Principal geodesic analysis for the study of nonlinear statistics of shape, IEEE Trans. Med. Imaging 23 (8) (2004) 995–1005.
[14] B. Gokberk, H. Dutagaci, A. Ulas, L. Akarun, B. Sankur, Representation plurality and fusion for 3-d face recognition, IEEE Trans. Syst. Man Cybern. 38 (1) (2008) 155–173.
[15] W.A.P. Smith, E.R. Hancock, Recovering facial shape using a statistical model of surface normal direction, IEEE Trans. Pattern Anal. Mach. Intell. 28 (12) (2006) 1914–1930.
[16] W. Smith, E. Hancock, Facial shape-from-shading and recognition using principal geodesic analysis and robust statistics, IJCV 76 (1) (2008) 71–91.
[17] Q. Cai, D. Gallup, C. Zhang, Z. Zhang, 3D deformable face tracking with a commodity depth camera, European Conference on Computer Vision (ECCV), 2010, pp. 229–242.
[18] L. Morency, P. Sundberg, T. Darrell, Pose estimation using 3D view-based eigenspaces, Faces & Gesture, 2003, pp. 45–52.
[19] S. Zafeiriou, G.A. Atkinson, M.F. Hansen, W.A.P. Smith, V. Argyriou, M. Petrou, M.L. Smith, L.N. Smith, Face recognition and verification using photometric stereo: the photoface database and a comprehensive evaluation, IEEE Trans. Inf. Forensics Secur. 8 (1) (2013) 121–135.

[20] C.C.Y. Wang, Y. Ho, Facial feature detection and face recognition from 2D and 3D images, Pattern Recogn. Lett. 23 (2002) 1191–1202.
[21] S.M.F. Tsalakanidou, M. Strintzis, Face localization and authentication using color and depth images, IEEE Trans. Image Process. 14 (2) (2005) 152–168.
[22] S.G.M. Hüsken, M. Brauckmann, C. Malsburg, Strategies and benefits of fusion of 2D and 3D face recognition, IEEE Workshop on Face Recognition Grand Challenge Experiments, 2005, pp. 174–181.
[23] M.B.A.S. Mian, R. Owens, An efficient multimodal 2D–3D hybrid approach to automatic face recognition, IEEE Trans. Pattern Anal. Mach. Intell. 29 (11) (2007) 1927–1943.
[24] K.B.K.I. Chang, P. Flynn, An evaluation of multi-model 2D + 3D biometrics, IEEE Trans. Pattern Anal. Mach. Intell. 27 (4) (2005) 619–624.
[25] V.C.J. Cook, C. Mccool, S. Sridharan, Combined 2D/3D face recognition using log-gabor templates, IEEE Conference on Video and Signal Based Surveillance, 2006, pp. 83–90.
[26] M. Breitenstein, F. Reichlin, B. Leibe, E. Koller-Meier, L. Van Gool, Online multiperson tracking-by-detection from a single, uncalibrated camera, IEEE Trans. Pattern Anal. Mach. Intell. 33 (9) (2011) 1820–1833.
[27] M. Black, A. Jepson, Eigentracking: robust matching and tracking of articulated objects using a view-based representation, Int. J. Comput. Vis. 26 (1998) 63–84.
[28] T. Cootes, G. Edwards, C. Taylor, Active appearance models, IEEE Trans. Pattern Anal. Mach. Intell. 23 (2001) 681–685.
[29] B. Lucas, T. Kanade, An iterative image registration technique with an application to stereo vision, International Joint Conference on Artificial Intelligence (IJCAI), vol. 3, 1981, pp. 674–679.
[30] A. Jepson, D. Fleet, T. El-Maraghi, Robust online appearance models for visual tracking, IEEE Trans. Pattern Anal. Mach. Intell. (2003) 1296–1311.
[31] S. Zhou, R. Chellappa, B. Moghaddam, Visual tracking and recognition using appearance-adaptive models in particle filters, IEEE Trans. Image Process. 13 (2004) 1491–1506.
[32] S. Avidan, Support vector tracking, IEEE Trans. Pattern Anal. Mach. Intell. 26 (2004) 1064–1072.
[33] H. Grabner, M. Grabner, H. Bischof, Real-time tracking via on-line boosting, British Machine Vision Conference (BMVC), 2006, pp. 47–56.
[34] B. Babenko, M. Yang, S. Belongie, Visual tracking with online multiple instance learning, IEEE Conference On Computer Vision and Pattern Recognition (CVPR), 2009, pp. 983–990.
[35] A. Saffari, M. Godec, T. Pock, C. Leistner, H. Bischof, Online multi-class lpboost, IEEE Conference On Computer Vision and Pattern Recognition (CVPR), 2010, pp. 3570–3577.
[36] T.J. Chin, D. Suter, Incremental kernel principal component analysis, IEEE Trans. Image Process. 16 (2007) 1662–1674.
[37] G. Hager, P. Belhumeur, Efficient region tracking with parametric models of geometry and illumination, IEEE Trans. Pattern Anal. Mach. Intell. 20 (1998) 1025.
[38] S. Baker, I. Matthews, Equivalence and efficiency of image alignment algorithms, IEEE Conference On Computer Vision and Pattern Recognition (CVPR), 2001, pp. 1090–1097.
[39] I. Matthews, T. Ishikawa, S. Baker, The template update problem, IEEE Trans. Pattern Anal. Mach. Intell. 26 (2004) 810–815.
[40] I. Matthews, S. Baker, Active appearance models revisited, Int. J. Comput. Vis. 60 (2004) 135–164.
[41] R. Yang, Z. Zhang, Model-based head pose tracking with stereovision, Face & Gesture Recognition, 2002, pp. 255–260.
[42] T. Weise, H. Li, L. Van Gool, M. Pauly, Face/off: live facial puppetry, SIGGRAPH/Eurographics Symposium on Computer Animation, 2009, pp. 7–16.
[43] T. Weise, S. Bouaziz, H. Li, M. Pauly, Realtime performance-based facial animation, ACM Trans. Graph. 30 (4) (2011).
[44] I. Naseem, R. Togneri, M. Bennamoun, Linear regression for face recognition, IEEE Trans. Pattern Anal. Mach. Intell. 32 (11) (2010) 2106–2112.
[45] G. Tzimiropoulos, S. Zafeiriou, M. Pantic, Subspace learning from image gradient orientations, IEEE Trans, Pattern Anal. Mach. Intell. 34 (12) (2012) 2454–2466.
[46] G. Tzimiropoulos, S. Zafeiriou, M. Pantic, Principal component analysis of image gradient orientations for face recognition, Face & Gesture, 2011, pp. 553–558.
[47] K.C.K. Bowyer, P. Flynn, A survey of approaches and challenges in 3D and multi-modal 3D + 2D face recognition, Comput. Vis. Image Underst. 101 (1) (2006) 1–15.
[48] J. Foley, Computer Graphics: Principles and Practice, Addison-Wesley Professional, 1996.
[49] S. Barsky, M. Petrou, The 4-source photometric stereo technique for three-dimensional surfaces in the presence of highlights and shadows, IEEE Trans. Pattern Anal. Mach. Intell. 25 (10) (2003) 1239–1252.
[50] G. Tzimiropoulos, S. Zafeiriou, M. Pantic, Subspace learning from image gradient orientations, IEEE Trans. Pattern Anal. Mach. Intell. 34 (12) (2012) 2454–2466.
[51] I. Marras, S. Zafeiriou, G. Tzimiropoulos, Robust learning from normals for 3D face recognition, Computer Vision, ECCV 2012. Workshops and Demonstrations, 2012, pp. 230–239.
[52] T. Cootes, C. Taylor, On representing edge structure for model matching, IEEE Conference On Computer Vision and Pattern Recognition (CVPR), 2001.
[53] A. Fitch, A. Kadyrov, W. Christmas, J. Kittler, Orientation correlation, British Machine Vision Conference (BMVC), 2002, pp. 133–142.
[54] G. Tzimiropoulos, V. Argyriou, S. Zafeiriou, T. Stathaki, Robust fft-based scale-invariant image registration with image gradients, IEEE Trans. Pattern Anal. Mach. Intell. 32 (2010) 1899–1906.
[55] R. Fisher, Dispersion on a sphere, Proc. R. Soc. Lond. A Math. Phys. Sci. 217 (1130) (1953) 295–305.
[56] P.J. Phillips, P. Flynn, T. Scruggs, K.W. Bowyer, J. Chang, K. Hoffman, J. Marques, J. Min, W. Worek, Overview of the face recognition grand challenge, IEEE Conf. Comput. Vis. Pattern Recognit. (2005) 947–954.
[57] W. Krzanowski, Between-groups comparison of principal components, J. Am. Stat. Assoc. (1979) 703–707.
[58] L.P. Morency, A. Rahimi, T. Darrell, Adaptive view-based appearance model, IEEE Conference On Computer Vision and Pattern Recognition (CVPR), 2003, pp. 803–810.
[59] B. Moghaddam, A. Pentland, Face recognition using view-based and modular eigenspaces, Autom. Syst. Identif. Inspection Hum., SPIE 2277 (1994) 1–7.
[60] V. Blanz, T. Vetter, A morphable model for the synthesis of 3D faces, SIGGRAPH Conference Proceedings, 1999, pp. 187–194.
[61] G. Fanelli, M. Dantone, A. Fossati, J. Gall, L.V. Gool, Random forests for real time 3D face analysis, Int. J. Comput. Vis. (2012).
[62] G. Fanelli, T. Weise, J. Gall, L.V. Gool, Real time head pose estimation from consumer depth cameras, 33rd Annual Symposium of the German Association for Pattern Recognition (DAGM), 2011.