Online Learning and Fusion of Orientation Appearance Models for Robust Rigid Object Tracking

Ioannis Marras, Joan Alabort Medina, Georgios Tzimiropoulos, Stefanos Zafeiriou and Maja Pantic

Abstract— **We present a robust framework for learning and fusing different modalities for rigid object tracking. Our method fuses data obtained from a standard visual camera and dense depth maps obtained by low-cost consumer depths 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 method combines image gradient orientations as extracted from intensity images with the directions of surface normals computed from dense depth fields provided by the Kinect. To incorporate these features in a learning framework, we use a robust kernel based on the Euler representation of angles. This kernel enables us to cope with gross measurement errors, missing data as well as typical problems in visual tracking such as illumination changes and occlusions. Additionally, the employed kernel can be efficiently implemented online. Finally, we propose to capture the correlations between the obtained orientation appearance models using a fusion approach motivated by the original AAM. 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 performance in very difficult tracking scenarios including extreme pose variations.**

I. Introduction

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. The approach to be followed is also imposed by the application as well as the assumed setting. 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 [5] 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.

In this work, we are interested in accurately and robustly tracking large rigid head motions. We focus on the appearance-based approach to visual tracking which has been the de-facto choice for this purpose. Popular examples include subspace-based techniques [4], [9], gradient descent [22], mixture models [19], [35], discriminative models for regression and classification [1], [2], [17], [28], and combinations of the above [3], [8], [18], [23], [24], [27].

Our main aim in this work is how to incorporate 3D information provided by commercial depth cameras such as the Kinect within subspace-based methods for online appearance-based face tracking.

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, while in contrary to the depth information the texture information is more robust when an object is moving far from the camera. The depth information can also help to remove the background information in a scene. Thus, it is more powerful if those two different kind of information are combined in a unified framework. In addition, 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 [4], [9], [18], [23] which in many cases can be efficiently implemented online [8], [21], [24], [27], 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* is feasible by using efficient online subspace learning schemes [21]. 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

I. Marras, J. A. Medina, G. Tzimiropoulos, S. Zafeiriou and M. Pantic are with Department of Computing, Imperial College London, 180 Queen's Gate. London SW7 2AZ, U.K. {i.marras, ja310, gt204, s.zafeiriou, m.pantic}@imperial.ac.uk

G. Tzimiropoulos is also with School of Computer Science, University of Lincoln, Lincoln LN6 7TS, U.K.

M. Pantic is also with Faculty of Electrical Engineering, Mathematics and Computer Science, University of Twente, The Netherlands.

[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.

shown to be extremely discriminative [26].

The main problem now is how the appearance subspace can be efficiently and robustly learned and updated when data is corrupted by outliers. Outliers are common not only because of illumination changes, occlusions or cast shadows but also because the depth measurements provided by the Kinect could be very noisy and the obtained depth maps usually contain “holes”. Note that subspace learning for visual tracking requires robustness, efficiency and online adaptation. This combined problem has been vary rarely studied in literature. For example, in [27], the subspace is efficiently learned online using incremental ℓ_2 norm PCA [21]. Nevertheless, the ℓ_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 [7], [11], [20] typically cannot be extended for efficient online learning.

Previous methods for face tracking based on 3D information require an off-line training process for creating object-specific models [25], [32]–[34], do not explicitly deal with outliers [33], do not cope with fast head movements [6], or require the face to be already detected [13]. Finally, the question of how to fuse intensity with depth has been rarely addressed in literature. Although there are attempts in literature to use both modalities [6], [25], no particular fusion strategies have been proposed.

Our main contribution in this work is an approach for learning and fusing appearance models computed from these different modalities for robust rigid object tracking. 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 [30], [31]. 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 [31], 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 [9].

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.

II. Online Learning and Fusion of Robust Orientation Appearance Models

A. Object representations

We are interested in the problem of rigid object tracking given measurements of the object’s shape and texture. The shape of the object $\mathbf{S}$ is represented by a 3D triangulated mesh of n points $\mathbf{s}_k = [x\ y\ z]^T \in \Re^3$, i.e. $\mathbf{S} = [\mathbf{s}_1 | \cdots | \mathbf{s}_n] \in \Re^{3\times n}$. 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.

B. 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 [9]: Principal Component Analysis (PCA) is used twice to obtain one subspace for the shapes and one for the textures. For each data sample, the embedding of its shape and texture are 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. In this work, we follow a similar approach but use different features and a different computational mechanism for PCA. Another difference is that we use dense depth measurements.

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 [9]. Second, it is assumed that data samples are outlier-free which justifies the use of standard ℓ_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.

1) Orientation Features: **Azimuth Angle of Surface Normals.** We used the azimuth angle of surface normals. 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

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

Normals $\mathbf{n} \in \Re^3$ do not lie on a Euclidean space but on a spherical manifold $\boldsymbol{\eta} \in \mathcal{S}^2$, 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

$$\mathbf{\Phi}^a(\mathbf{x}) = \arctan \frac{n_y(\mathbf{x})}{n_x(\mathbf{x})} = \arctan \frac{\frac{\partial f}{\partial y}}{\frac{\partial f}{\partial x}}. \tag{2}$$

Methods for computing the normals of surfaces can be found in [16].

Image Gradient Orientations. Given the texture $\mathbf{I}$ of an object, we extract image gradient orientation from

$$\mathbf{\Phi}^g(\mathbf{u}) = \arctan \frac{\mathbf{G}_y(\mathbf{u})}{\mathbf{G}_x(\mathbf{u})}, \tag{3}$$

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. Possible choices for $\mathbf{H}_x, \mathbf{H}_y$ include central difference estimators and discrete approximations to the first derivative of the Gaussian.

2) Orientation Appearance Models: Let us denote by $\boldsymbol{\phi}_i$ the $n-$dimensional vector obtained by writing either $\mathbf{\Phi}_i^a$ or $\mathbf{\Phi}_i^g$ (the orientation maps computed from $\mathbf{S}_i$, $\mathbf{I}_i$) in lexicographic ordering. Vectors $\boldsymbol{\phi}_i$ 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 [31]

$$\mathbf{e}(\boldsymbol{\phi}_i) = \frac{1}{\sqrt{n}}[\cos(\boldsymbol{\phi}_i)^T + j\sin(\boldsymbol{\phi}_i)^T]^T, \tag{4}$$

where $\cos(\boldsymbol{\phi}_i) = [\cos(\boldsymbol{\phi}_i(1)), \ldots, \cos(\boldsymbol{\phi}_i(n))]^T$ and $\sin(\boldsymbol{\phi}_i) = [\sin(\boldsymbol{\phi}_i(1)), \ldots, \sin(\boldsymbol{\phi}_i(n))]^T$. Note that similar features have been proposed in [10], but here we avoid the normalization based on gradient magnitude suggested in [10] because it makes them more sensitive to outliers and removes the kernel properties as described in [31]. Using $\mathbf{e}_i \equiv \mathbf{e}(\boldsymbol{\phi}_i)$, correlation can be measured using the real part of the familiar inner product [15], [29], [31]

$$\begin{aligned} c(\mathbf{e}_i, \mathbf{e}_j) &\triangleq \Re\{\mathbf{e}_i^H \mathbf{e}_j\} \\ &= \frac{1}{n}\sum_{k=1}^{n} \cos[\Delta\boldsymbol{\phi}(k)], \end{aligned} \tag{5}$$

where $\Delta\boldsymbol{\phi} \triangleq \boldsymbol{\phi}_i - \boldsymbol{\phi}_j$. 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 (5), we observe that if $\mathbf{S}_i \simeq \mathbf{S}_j$ or $\mathbf{I}_i \simeq \mathbf{I}_j$, then $\forall k \; \Delta\boldsymbol{\phi}(k) \simeq 0$, 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 [31] it holds

$$\sum_{k \in P_o} \cos[\Delta\boldsymbol{\phi}(k)] \simeq 0, \tag{6}$$

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 [31] for a detailed justification of the above result for the case of image gradient orientations. We assume here that similar arguments can be made for the case of the azimuth angles of the surface normals.

A kernel PCA based on the cosine of orientation differences for the robust estimation of orientation subspaces is obtained by using the mapping of (5) and then by applying linear complex PCA to the transformed data [31]. More specifically, we look for a set of $p < m$ orthonormal bases $\mathbf{U} = [\mathbf{u}_1|\cdots|\mathbf{u}_p] \in \mathbb{C}^{n \times p}$ by solving

$$\begin{aligned} \mathbf{U}_o &= \arg\max\nolimits_{\mathbf{U}} \operatorname{tr}\left[\mathbf{U}^H \mathbf{E}\mathbf{E}^H \mathbf{U}\right] \\ \text{subject to (s.t.)} \quad & \mathbf{U}^H\mathbf{U} = \mathbf{I}. \end{aligned} \tag{7}$$

where $\mathbf{E} = [\mathbf{e}_1|\cdots|\mathbf{e}_m] \in \mathbb{C}^{n \times m}$. 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 $\mathbf{C} = [\mathbf{c}_1|\cdots|\mathbf{c}_n] \in \mathbb{C}^{p \times n}$ of $\mathbf{E}$ are given by $\mathbf{C} = \mathbf{U}^H\mathbf{E}$.

Finally, we propose to apply the above kernel PCA to learn orientation appearance models for both azimuth angles of surface normals and image gradient orientations. More specifically, 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) Fusion of Orientation Appearance Models: 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

$$\mathbf{C} = [(\mathbf{C}^a)^H \; (\mathbf{C}^g)^H]^H, \; \in \mathbb{C}^{(p_a+p_g) \times m} \tag{8}$$

and then apply a further linear complex PCA on $\mathbf{C}$ to obtain a set of p_f bases $\mathbf{V} = [\mathbf{v}_1|\cdots|\mathbf{v}_{p_f}] \in \mathbb{C}^{(p_a+p_g) \times p_f}$. Then, these bases can 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

$$\tilde{\mathbf{y}} \approx \begin{bmatrix} \mathbf{U}^a\mathbf{V}^a \\ \mathbf{U}^g\mathbf{V}^g \end{bmatrix} \mathbf{b}_y, \tag{9}$$

where

$$\mathbf{b}_y = \mathbf{V}^H\mathbf{c}_y = \mathbf{V}^H \begin{bmatrix} \mathbf{c}_y^a \\ \mathbf{c}_y^g \end{bmatrix} = \mathbf{V}^H \begin{bmatrix} (\mathbf{U}^a)^H\mathbf{e}_y^a \\ (\mathbf{U}^g)^H\mathbf{e}_y^g \end{bmatrix}. \tag{10}$$

Thus, the coefficients $\mathbf{b}_y$ used for the reconstruction in (II-B.3), are computed from the fused subspace $\mathbf{V}$ and are common for *both* orientation appearance models as can be easily seen from (10). Finally, note that, in contrast to [9], no feature weighting is used in the proposed scheme.

4) Online learning: A key feature of the proposed algorithm is that it continually updates the learned orientation appearance models using newly processed (tracked) frames. It is evident that the batch version of PCA is not suitable for this purpose because, each time, it requires to process all frames (up to the current one) in order to generate the updated subspace. For this purpose, prior work [27] efficiently updates the subspace using the incremental ℓ_2 norm PCA proposed in [21]. The kernel-based extension to [21] has been proposed in [8], 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 (4), an exact and efficient solution is feasible. The proposed algorithm is summarized as follows [31].

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 $\mathbf{\Sigma}_m = \mathbf{\Lambda}_m^{1/2}$. Then, given l new data samples our target is to obtain $\mathbf{U}_{m+l}$ and $\mathbf{\Sigma}_{m+l}$ 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 $\mathbf{\Sigma}_m = \mathbf{\Lambda}_m^{1/2}$, a set of new orientation maps $\{\mathbf{\Phi}_{m+1}, \ldots, \mathbf{\Phi}_{m+l}\}$ and the number p of principal components.

Step 1. Using (4) compute the matrix of the transformed data $\mathbf{E}_m = [\mathbf{e}_{m+1}|\ldots|\mathbf{e}_{m+l}]$.

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

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

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

Finally, for the fusion of the orientation appearance models, we used the incremental ℓ_2 norm PCA proposed in [21]. Overall, the algorithm proceeds as follows. Initially and for a reasonably small number of frames, all eigenspaces are generated using the batch mode of the kernel PCA of [31] and standard ℓ_2-norm PCA for the fusion step. When the algorithm switches to the online mode, then for each newly tracked frame, algorithm 1 is used to update the orientation appearance models. The embedding of the new sample is also calculated which is then used to update the eigenspace $\mathbf{V}$ using the method in [21].

III. Motion Model

The provided 3D shape information enables us to use 3D motion models. In this way, pose and appearance are decoupled, which we believe that it is crucial for the robustness of subspace-based tracking methods. Given a set of 3D parameters the shape is first warped by

$$\mathbf{S}_W = \mathbf{R}_\phi \mathbf{R}_\theta \mathbf{R}_\varphi \mathbf{S} + \mathbf{t}_w, \qquad (11)$$

where $\mathbf{t}_w$ is a $3D$ translation and $\mathbf{R}_\phi, \mathbf{R}_\theta, \mathbf{R}_\varphi$ are rotation matrices. The warped shape $\mathbf{S}_W$ is then used for extracting surface normals and the corresponding azimuth angles. Finally, $\mathbf{S}_W$ is projected using a scale orthographic projection P 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. We have thoroughly evaluated our system on a publicly available database on which we achieve state-of-the-art performance.

IV. Tracking with Orientation Appearance Models

We combine the proposed fused orientation appearance models with the 3D motion model earlier described and standard particle filter methods for rigid object tracking [27]. 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:

$$p(M_t^i, M_{t-1}^{1:P}) = \sum_{i=1}^{P} w_{t-1}^i \mathcal{N}(M_t; M_{t-1}^i, \Xi) \qquad (12)$$

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 [26], and model this probability as

$$p(\mathbf{y}_t^i|\mathbf{M}_t^i) \propto e^{\frac{||\mathbf{y}_t^i - \tilde{\mathbf{y}}_t^i||_f^2}{\sigma}}, \qquad (13)$$

where $\tilde{\mathbf{y}}_t^i$ is given by (10).

V. Results

Evaluating and comparing different tracking approaches is a rather tedious task. 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 [13]. 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.
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-norm PCA methods. A simplified version of this tracker which uses 2D motion and pixel intensities only has been proposed in [27].

To compare all above variants of subspace-based tracking techniques, we used 3 representative videos. The first video contains face expressions, the second one contains extreme face pose variations and illumination variations, while 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 II shows the mean (median) values of S for al trackers and videos respectively. Fig. 4,5 and 6 plots S for all methods and videos as a function of the frame number. Finally, Figs. 1,2 and 3 illustrates the performance of the proposed tracker for some cumbersome tracking conditions.

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 [12], [14]. The dataset contains over 15K images of 20 people (6 females and 14 males - 4 people were recorded twice) recorded while sitting about 1 meter away from the sensor. For each frame, a depth image, the corresponding texture image (both 640x480 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 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 [13] 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 I, 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 degrees.

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 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 [13]. This justifies the motivation behind the proposed scheme.

	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

TABLE II

Mean (Median) S values for all trackers and videos. The proposed tracker is coined **3D+IGO+AA**.

VI. 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

TABLE I

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 DEGREES.

Methods	Yaw error	Pitch error	Roll error	Direction estimation accuracy
Method proposed in [13]	11 ± 12.1^o	9.9 ± 10.8^o	9.1 ± 10.1^o	81.0%
Our approach 3D+IGO+AA	9.2 ± 13.0^o	9.0 ± 11.1^o	8.0 ± 10.3^o	89.9%

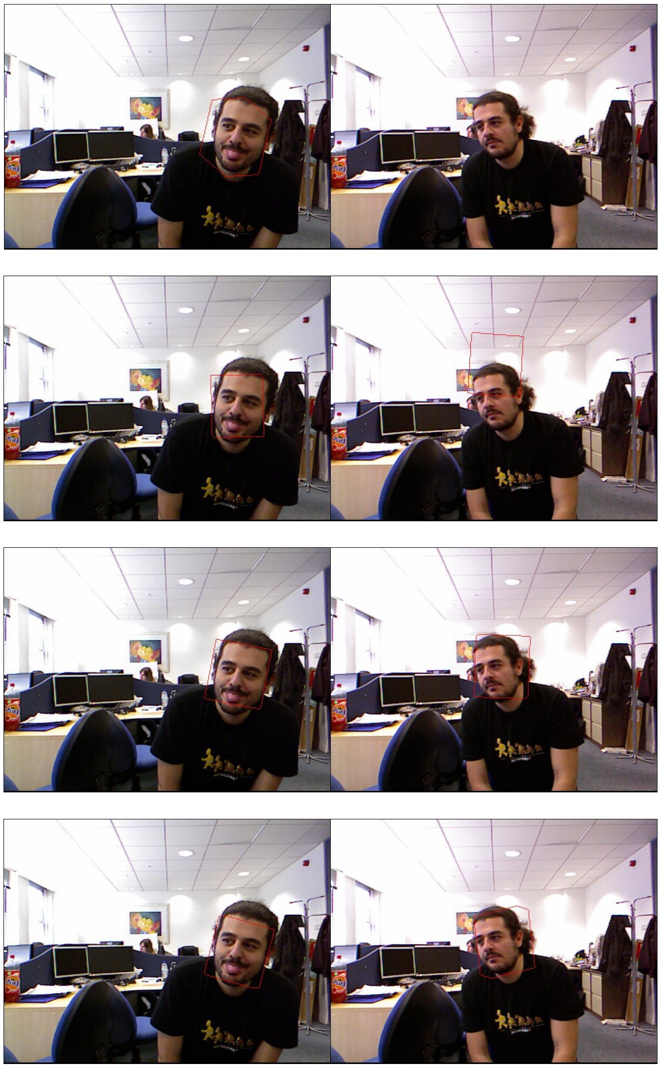

Fig. 1. Tracking examples from the first video. First row: 3D+I+D. Second row: 3D+AA. Third row: 3D+IGO. Fourth row: 3D+IGO+AA

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.

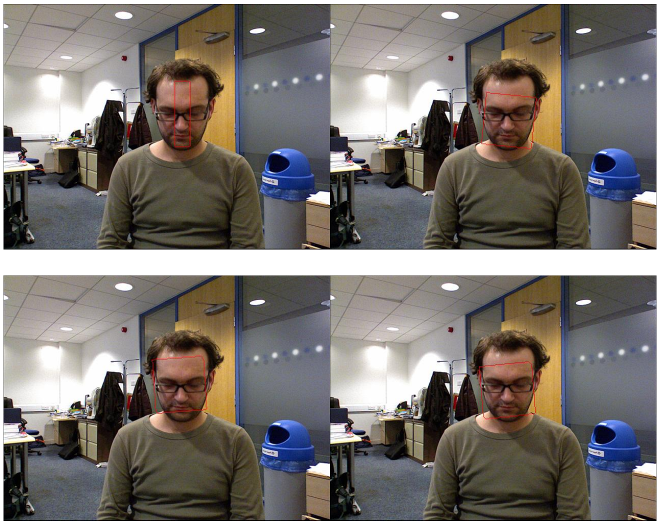

Fig. 2. Tracking examples for the second video. First Row: First image: 3D+I+D. Second image: 3D+AA. Second row: First image: 3D+IGO. Second image: 3D+IGO+AA.

VII. ACKNOWLEDGEMENTS

The research presented in this paper has been funded by the European Communitys 7th Framework Programme [FP7/2007-2013] under grant agreement no. 288235 (FROG). The work by Maja Pantic is funded by the European Research Council under the ERC Starting Grant agreement no. ERC-2007- StG-203143 (MAHNOB). The work by Stefanos Zafeiriou is partially funded by the Junior Research Fellowship of Imperial College London.

REFERENCES

[1] S. Avidan. Support vector tracking. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 26:1064 – 1072, 2004.

[2] B. Babenko, M. Yang, and S. Belongie. Visual Tracking with Online Multiple Instance Learning. In *Computer Vision and Pattern Recognition (CVPR)*, pages 983 – 990, 2009.

[3] S. Baker and I. Matthews. Equivalence and Efficiency of Image Alignment Algorithms. In *Computer Vision and Pattern Recognition (CVPR)*, pages 1090 – 1097, 2001.

[4] M. Black and A. Jepson. Eigentracking: Robust matching and tracking of articulated objects using a view-based representation. *International Journal of computer Vision (IJCV)*, 26:63 – 84, 1998.

[5] M. Breitenstein, F. Reichlin, B. Leibe, E. Koller-Meier, and L. Van Gool. Online multiperson tracking-by-detection from a single, uncalibrated camera. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 33(9):1820–1833, 2011.

[6] Q. Cai, D. Gallup, C. Zhang, and Z. Zhang. 3d deformable face tracking with a commodity depth camera. In *European Conference on Computer Vision (ECCV)*, pages 229–242, 2010.

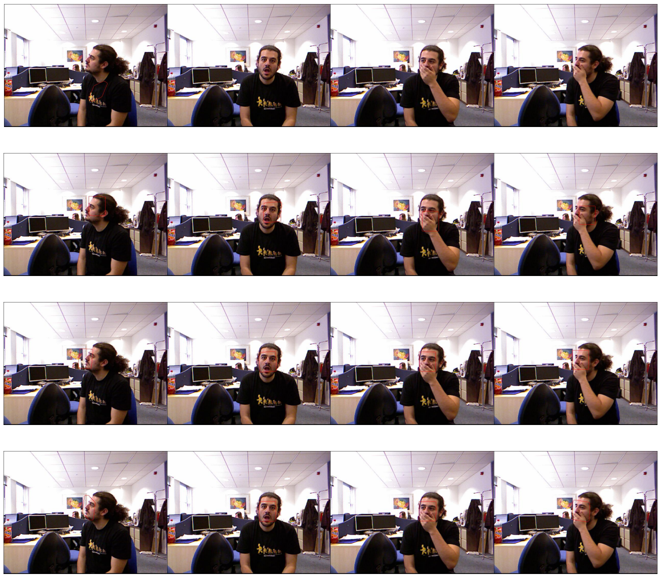

Fig. 3. Tracking examples for the third video. First row: 3D+I+D. Second row: 3D+IGO. Third row: 3D+AA. Fourth row: 3D+IGO+AA.

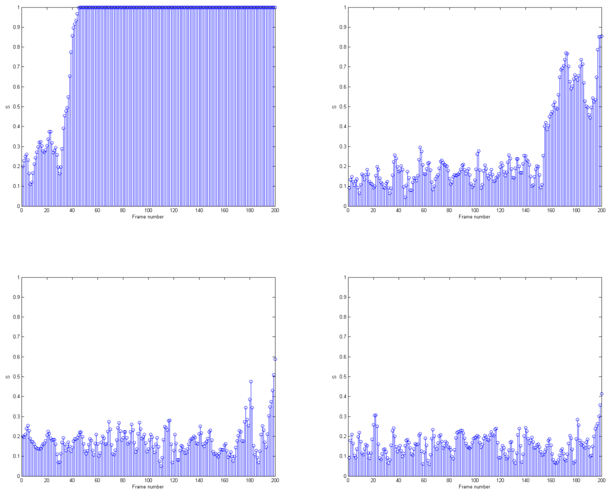

Fig. 4. S value vs the number of frames for the first video. First Row: First image: 3D+I+D. Second image: 3D+AA. Second row: First image: 3D+IGO. Second image: 3D+IGO+AA.

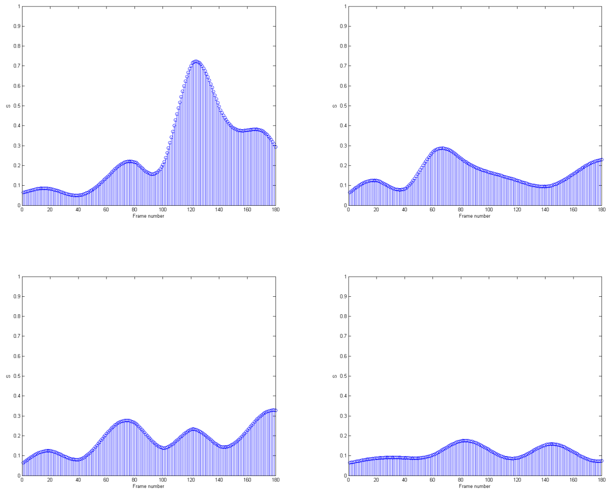

Fig. 5. S value vs the number of frames for the second video. First Row: First image: 3D+I+D. Second image: 3D+AA. Second row: First image: 3D+IGO. Second image: 3D+IGO+AA.

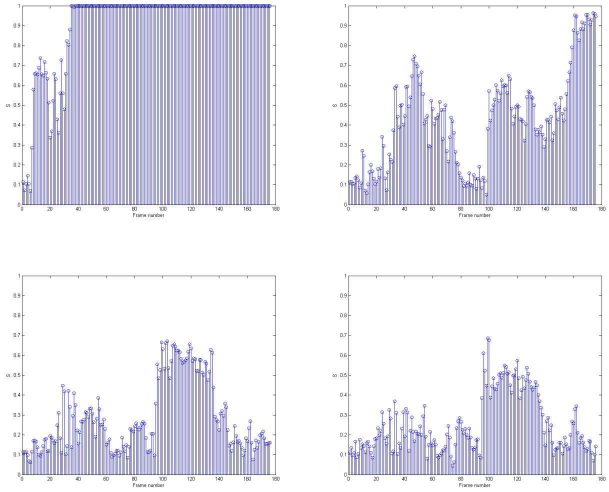

Fig. 6. S value vs the number of frames for the third video. First Row: First image: 3D+I+D. Second image: 3D+AA. Second row: First image: 3D+IGO. Second image: 3D+IGO+AA.

[7] E. Candès, X. Li, Y. Ma, and J. Wright. Robust principal component analysis&quest. *Journal of The ACM (JACM)*, 58(3):11, 2011.

[8] T.-J. Chin and D. Suter. Incremental Kernel Principal Component Analysis. *IEEE Transactions on Image Processing (TIP)*, 16:1662 – 1674, 2007.

[9] T. Cootes, G. Edwards, and C. Taylor. Active Appearance Models. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 23:681 – 685, 2001.

[10] T. Cootes and C. Taylor. On representing edge structure for model matching. In *Computer Vision and Pattern Recognition (CVPR)*. IEEE, 2001.

[11] F. de la Torre and M. Black. A Framework for Robust Subspace Learning. *International Journal of computer Vision (IJCV)*, 54:117 – 142, 2003.

[12] G. Fanelli, M. Dantone, A. Fossati, J. Gall, and L. V. Gool. Random forests for real time 3d face analysis. *International Journal of computer Vision (IJCV)*, 2012.

[13] G. Fanelli, J. Gall, and L. V. Gool. Real time head pose estimation with random regression forests. In *Computer Vision and Pattern Recognition (CVPR)*, pages 617–624, June 2011.

[14] G. Fanelli, T. Weise, J. Gall, and L. V. Gool. Real time head pose estimation from consumer depth cameras. In *33rd Annual Symposium of the German Association for Pattern Recognition (DAGM)*, September 2011.

[15] A. Fitch, A. Kadyrov, W. Christmas, and J. Kittler. Orientation correlation. In *British Machine Vision Conference (BMVC)*, pages 133–142, 2002.

[16] J. Foley. *Computer graphics: principles and practice*. Addison-Wesley Professional, 1996.

[17] H. Grabner, M. Grabner, and H. Bischof. Real-time tracking via online boosting. In *British Machine Vision Conference (BMVC)*, pages 47–56, 2006.

[18] G. Hager and P. Belhumeur. Efficient Region Tracking with Parametric Models of Geometry and Illumination. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 20:1025, 1998.

[19] A. Jepson, D. Fleet, and T. El-Maraghi. Robust Online Appearance Models for Visual Tracking. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, pages 1296 – 1311, 2003.

[20] N. Kwak. Principal Component Analysis Based on L1-Norm Max-

imization. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 30:1672 – 1680, 2008.

[21] A. Levy and M. Lindenbaum. Squential Karhunen-Loeve Basis Extraction and its Application to Images. *IEEE Transactions on Image Processing (TIP)*, 9:1371 – 1374, 2000.

[22] B. Lucas and T. Kanade. An iterative image registration technique with an application to stereo vision. In *International Joint Conference on Artificial Intelligence (IJCAI)*, volume 3, pages 674 – 679, 1981.

[23] I. Matthews and S. Baker. Active Appearance Models Revisited. *International Journal of computer Vision (IJCV)*, 60:135 – 164, 2004.

[24] I. Matthews, T. Ishikawa, and S. Baker. The Template Update Problem. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 26:810 – 815, 2004.

[25] L. Morency, P. Sundberg, and T. Darrell. Pose estimation using 3d view-based eigenspaces. In *Faces & Gesture*, pages 45–52, 2003.

[26] I. Naseem, R. Togneri, and M. Bennamoun. Linear regression for face recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 32(11):2106–2112, 2010.

[27] D. Ross, J. Lim, R.-S. Lin, and M.-H. Yang. Incremental Learning for Robust Visual Tracking. *International Journal of computer Vision (IJCV)*, 77:125 – 141, 2008.

[28] A. Saffari, M. Godec, T. Pock, C. Leistner, and H. Bischof. Online multi-class lpboost. In *Computer Vision and Pattern Recognition (CVPR)*, pages 3570–3577, 2010.

[29] G. Tzimiropoulos, V. Argyriou, S. Zafeiriou, and T. Stathaki. Robust FFT-Based Scale-Invariant Image Registration with Image Gradients. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 32:1899 – 1906, 2010.

[30] G. Tzimiropoulos, S. Zafeiriou, and M. Pantic. Principal component analysis of image gradient orientations for face recognition. In *Face & Gesture*, pages 553–558, 2011.

[31] G. Tzimiropoulos, S. Zafeiriou, and M. Pantic. Subspace learning from image gradient orientations. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 2012.

[32] T. Weise, S. Bouaziz, H. Li, and M. Pauly. Realtime performance-based facial animation. *ACM Transactions on Graphics*, 30(4), 2011.

[33] T. Weise, H. Li, L. Van Gool, and M. Pauly. Face/off: Live facial puppetry. In *SIGGRAPH/Eurographics Symposium on Computer Animation*, pages 7–16, 2009.

[34] R. Yang and Z. Zhang. Model-based head pose tracking with stereovision. In *Face & Gesture Recognition*, pages 255–260, 2002.

[35] S. Zhou, R. Chellappa, and B. Moghaddam. Visual Tracking and Recognition Using Appearance-Adaptive Models in Particle Filters. *IEEE Transactions on Image Processing (TIP)*, 13:1491 – 1506, 2004.