3D Head Pose Estimation Using Tensor Decomposition and Non-linear Manifold Modeling

Dmytro Derkach
Pompeu Fabra University
Barcelona, Spain
dmytro.derkach@upf.edu

Adria Ruiz
Inria, LJK
Grenoble, France
adria.ruiz-ovejero@inria.fr

Federico M. Sukno
Pompeu Fabra University
Barcelona, Spain
federico.sukno@upf.edu

Abstract

Head pose estimation is a challenging computer vision problem with important applications in different scenarios such as human-computer interaction or face recognition. In this paper, we present an algorithm for 3D head pose estimation using only depth information from Kinect sensors. A key feature of the proposed approach is that it allows modeling the underlying 3D manifold that results from the combination of pitch, yaw and roll variations. To do so, we use tensor decomposition to generate separate subspaces for each variation factor and show that each of them has a clear structure that can be modeled with cosine functions from a unique shared parameter per angle. Such representation provides a deep understanding of data behavior and angle estimations can be performed by optimizing combination of these cosine functions. We evaluate our approach on two publicly available databases, and achieve top state-of-the-art performance.

1. Introduction

Head pose estimation is a relevant problem for several computer vision applications, including human-computer interaction, video conferencing, face recognition and facial motion analysis [7]. Head pose estimation has traditionally been performed on 2D images, but recent advances in 3D acquisition systems have led to a growing interest in methods that operate on 3D data [28]. These methods are less sensitive to changes in illumination and viewpoint than 2D image-based approaches, which makes them more accurate and robust. Therefore, in this work we focus on head pose estimation from 3D data.

The goal of head pose estimation is to predict the relative orientation between the camera and a 3D mesh of the target head. This orientation is usually represented by three angles: rotation around vertical axis (yaw angle), around lateral axis (pitch angle), and around longitudinal axis (roll angle). Despite the fact that standard features used to represent 3D meshes lie in high-dimensional spaces, a key observation to solve this problem is that the aforementioned angles define a lower-dimensional manifold with only 3 degrees of freedom. This fact makes tensor decomposition and manifold learning appealing frameworks for the estimation of the orientation parameters. In particular, factorization methods such as multi-linear decomposition [32], are able to separate the variations produced by the different factors (i.e. angles) into separate subspaces, thus obtaining specific parametrizations for each of them. On the other hand, manifold learning [34] can be used to find the low-dimensional manifold structure defined by the orientation angles.

In this context, previous works have attempted to use the described frameworks for head pose estimation. Concretely, methods such as Isomap [27] or Local Linear Embedding [14] have been explored in order to learn the underlying manifold structure defined by the orientation parameters. Even though the cited methods are able to learn generic low-dimensional data representations, the resulting manifold is only defined implicitly and, therefore, it is difficult to introduce specific constraints to model the inherent structure defined by pose changes.

In order to address this limitation, we propose a novel approach to learn the manifold defined by 3D rotations. In particular, our method is able to explicitly model its underlying structure with an analytic form which takes into account the specific constraints imposed by orientation variations. For this purpose, we use multi-linear decomposition to split the pose variation factors (i.e. yaw, pitch and roll) and obtain a set of subspaces whose coefficients are governed by an unique parameter. These coefficients define a continuous curve in each of the sub-spaces that corresponds to the head pose variation along one of the rotation angles. We further show that these curves can be modeled in terms of trigonometric functions, which are indeed the bases to explain rotation effects. Thus, we introduce a minimization framework for pose estimation based on tensor decomposi-

tion constrained by trigonometric functions so that the solutions obtained are always compatible with the underlying manifold.

We start by studying a motivating example using 2D images that capture out-of-plane rotations of simple objects along the vertical axis. Then, we formalize our minimization framework generalizing it to 3D rotations in any of the three axes and demonstrate its usefulness by applying it to head pose estimation. Our experiments on the SASE and BIWI databases [20, 12], two large and publicly available 3D face corpora, show that the proposed framework can achieve state of the art performance for head-pose estimation.

The rest of the paper is organized as follows. Section 2 introduces a brief review of the existing approaches for head pose estimation. in Section 3 we provide the required background on tensor theory and Section 4 details the proposed manifold modeling framework for head pose estimation. The experimental setups and results are covered in Section 5. Section 6 concludes the paper.

2. Related work

2.1. Manifold-based methods

Many methods have considered the underlying manifold structure of head pose variations [34]. The main idea behind these methods is that, regardless of the dimensionality of the input features, there should be at most 3 degrees of freedom for head pose variation, thus defining a 3D manifold [27]. However, in general, this manifold is embedded non-linearly in the ambient space defined by the features, which has led researchers to explore non-linear manifold learning methods such as Locally Linear Embedding [14], Isomap [27], Synchronized Submanifold Embedding [38], Homeomorphic Manifold Analysis [26], Neighborhood Preserving Embedding or Locality Preserving Projection [4] for head pose estimation from 2D images.

An interesting possibility to enhance the embedding results is the use of head pose labels. For example, Balasubramanian *et al*. [2] present a Biased Manifold Embedding (BME) framework in which the distance metric between data points is modified so that heads under similar poses are brought closer to each other than they would be under the unbiased (unsupervised) case. Similarly, Wang and Song [35] consider head-pose information to constrain the distances between data points and present a regression variant of Fisher Discriminant Analysis (FDA), which they call supervised neighborhood-based FDA. An alternative approach is followed by Benabdelkader [4], who firstly apply unsupervised manifold learning methods and then employ the head pose information to train regressors in the resulting low-dimensional manifolds.

Liu *et al*. [18] argue that a single manifold is not enough for head pose estimation and that appearance variations such as changes in identity, scale and illumination make it necessary the use of multiple different manifolds to model pose parameters. Thus, authors presented a clustering method to construct multiple manifolds, each of which characterizes the underlying subspace of some subjects. Peng *et al*. [26] also learn multiple manifolds; they use Homeomorphic Manifold Analysis to build a separate manifold for each subject and learn non-linear mappings to relate each subject-manifold with a common pose-manifold whose topology is predefined as a unit circle or sphere (for addressing rotations about one or two axes, respectively).

The most similar work to ours is probably the one from Takallou and Kasaei. [31], who learn a non-linear tensor model based on multi-linear decomposition for head pose estimation from 2D images. They build a three-way tensor to account for identity, pose and pixels information, targeting only yaw rotations. During training, they find individual-dependent mappings between each training pose and a unified pose manifold based on tensor decomposition. At test time, each query image is projected into pose and identity subspaces, which results in as many pose coefficients as identities in the training set. The final pose estimate is obtained by validating the available pose coefficients in terms of compliance with the unified pose manifold (e.g. inversely to the distance to training samples).

We see that, in contrast to our work, all of the above methods use 2D images and most of them do not target rotations about the three spacial axes. Moreover, none of them provides an analytic formulation for the pose manifolds.

2.2. 3D methods review

Head pose estimation has traditionally been performed on 2D images, but recent advances in 3D acquisition systems have led to a growing interest in methods that operate on 3D data. These methods are less sensitive to changes in illumination and viewpoint than 2D image-based approaches, which makes them more accurate and robust.

An important distinction between different approaches is the type of input data that is used. Firstly, very few methods use only depth information, typically relying on curvatures, symmetry planes or most salient facial landmarks, such as the nose tip [6, 30, 16].

In contrast, a majority of head pose estimation algorithms working in 3D, use also RGB data as additional source of information, facilitating aspects such as face detection and estimation of fiducial points. In this category we find approaches based on the fusion of 2D and 3D features (e.g. SIFT, HOG) to train regressors [33], template fitting, such as 3D Morphable Models [36], or depth features initialized by 2D face detection [25].

Finally, it is also common to take advantage of temporal information for tracking the head pose across sequences of

frames, which considerably improves performance. However, tracking-based algorithms often benefit from the fact that test sequences usually start with nearly frontal head poses and their accuracy to detect initial head poses other than frontal is not clear. Thus, when comparing our results, we will focus on methods that provide estimation results on a per-frame bases, without tracking.

Interestingly, we see that previous methods targeting head pose estimation from 3D data have not taken advantage of the underlying manifold structure of 3D head rotations.

3. Tensor decomposition

In this section, we give a review of tensor decomposition methods, especially focusing on the higher order SVD (HOSVD) [5, 9, 15].

In many scenarios, data can be naturally represented as multidimensional arrays and, therefore, it is beneficial to take into account its inherent structure in order to analyze it. For this purpose, the use of tensors is a natural solution. In particular, a tensor is also known as a *n*-way array or a *n*-mode matrix. Vectors and matrices can be considered as first and second order tensors, respectively.

The starting point of our derivation of multilinear SVD will be to remind the simple SVD decomposition. For matrix $\mathcal{A} \in \mathbb{R}^{m\times n}$ we recall the SVD as being:

$$\mathcal{A} = U\Sigma V^T = \sum_{k=1}^{r} \sigma_k u_k v_k^T = \sum_{k=1}^{r} \sigma_k u_k \otimes v_k \quad (1)$$

and for the elements $\mathcal{A}_{ij}$ of $\mathcal{A}$ we have

$$\mathcal{A}_{ij} = \sum_{k=1}^{r} U_{ik}\Sigma_{kk}V_{ik} \quad (2)$$

Here $\otimes$ denotes the tensor (or outer product) $x \otimes y \triangleq xy^T$; Σ is a diagonal $(r \times r)$ matrix with nonzero singular values of $\mathcal{A}$ (the square roots of the eigenvalues of $\mathcal{A}^T\mathcal{A}$) on its diagonal; u_k and v_k are the orthonormal columns of the matrix U $(m \times r)$ and V $(n \times r)$, respectively, with v_k being the eigenvectors of $\mathcal{A}^T\mathcal{A}$ and $u_k = \mathcal{A}v_k/\sigma_k$ [5].

The SVD is useful whenever we have a two-dimensional data set $\mathcal{A}_{ij}$, which is naturally expressed in term of a matrix $\mathcal{A}$ (second order tensor). In the application of this paper we will deal with cases where the dimension is bigger than two, particularly is equal five (fifth order tensor). The SVD may be generalized to higher order tensors (or multiway arrays).

Given tensor $\mathcal{T} \in \mathbb{R}^{I_1\times I_2\times\ldots\times I_5}$, the decomposition of the fifth order tensor can be expressed as

$$\mathcal{T} = \sum_{J_1}\cdots\sum_{J_5} G_{J_1J_2\ldots J_5} U^{(1)}_{J_1} \otimes U^{(2)}_{J_2} \otimes \cdots \otimes U^{(5)}_{J_5} \quad (3)$$

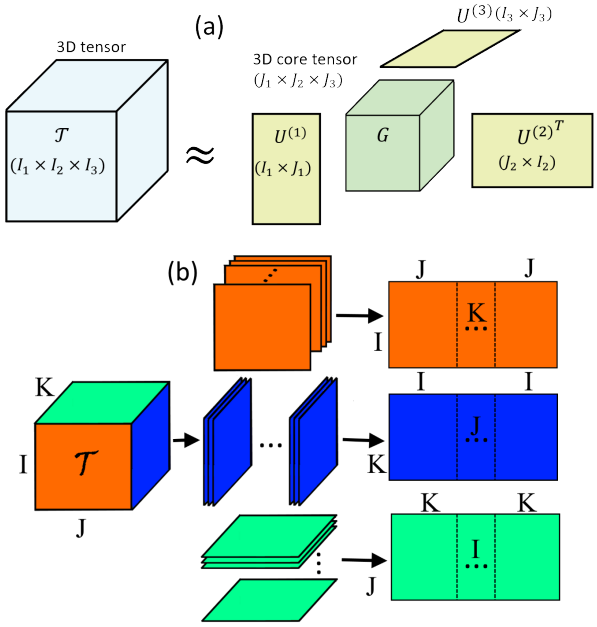

Figure 1. (a) Illustration of a 3D tensor decomposition. (b) Unfolding of the $(I \times J \times K)$-tensor $\mathcal{T}$ to the $(I \times JK)$-matrix, the $(J \times KI)$-matrix and the $(K \times IJ)$-matrix

or as a mode product [10]

$$\mathcal{T} = G \times_1 U^{(1)} \times_2 U^{(2)} \cdots \times_5 U^{(5)} \quad (4)$$

where $G \in \mathbb{R}^{J_1\times J_2\times\ldots\times J_5}$ and $U^{(n)} \in \mathbb{R}^{I_n\times J_n}$. The graphic representation of the Higher Order SVD (3D) is shown on the Figure 1(a);

The n-mode product of a tensor $G \in \mathbb{R}^{J_1\times J_2\times\ldots\times J_N}$ by a matrix $U \in \mathbb{R}^{I_n\times J_n}$ denoted by $G \times_n U$ is an $(J_1 \times J_2 \times \cdots \times J_{n-1} \times I_n \times J_{n+1} \times \cdots \times J_N)$-tensor of which the entries are given by

$$(G \times U)_{j_1j_2\cdots j_{n-1}i_nj_{n+1}\cdots j_N} = \sum_{j_n} g_{j_1j_2\cdots j_{n-1}j_nj_{n+1}\cdots j_N} u_{i_nj_n} \quad (5)$$

In HOSVD, all matrices $U^{(n)}$ can be calculated by performing a matrix SVD on the $I_n \times (I_1I_2\cdots I_{n-1}I_{n+1}\cdots I_N)$ matrix obtained by a flattering or unfolding of $\mathcal{T}$ [5, 10].

The n-mode matricization (or unfolding) of a tensor $\mathcal{T} \in \mathbb{R}^{I_1\times I_2\times\ldots\times I_N}$ is denoted by $\mathcal{T}_{(n)}$ and arranges the n-mode fibers to be column of the resulting matrix. Tensor element $(i_1, i_2, \ldots, i_N)$ maps to matrix element (i_n, j), where

$$j = 1 + \sum_{\substack{k=1\\k\neq n}}^{N} (i_k - 1)J_k; J_k = \prod_{\substack{m=1\\m\neq n}}^{k-1} I_m \quad (6)$$

An example of unfolding of the third order tensor $\mathcal{T}$ is shown in Figure 1(b)

Since $U^{(n)}$ matrices are orthogonal, G from equation (4) is easily calculated from (7) and it is called the core tensor which shows the interactions of $U^{(n)}$ matrices – factor matrices [10].

$$G = \mathcal{T} \times_1 {U^{(1)}}^T \times_2 {U^{(2)}}^T \cdots \times_5 {U^{(5)}}^T \quad (7)$$

4. Manifold modeling framework

The goal of pose estimation is to predict the relative orientation between the target subject and the viewer or camera, in terms of three factors: yaw, pitch and roll angles. Because data depends on several factors, like the identity of the subjects and the three angles, it is convenient to use tensors to hold the data. A powerful tool to analyze multi-factor data is multi-linear decomposition (known as tensor decomposition or HOSVD). It aims at separating variations produced by the different factors into individual subspaces, thus obtaining specific coefficients for each of the targeted factors. We will see that, for the subspaces corresponding to rotations, these coefficients approximately follow the structure of a closed curve, which can be modeled on terms of trigonometric functions.

4.1. Motivating example

We start by studying a motivating example using 2D images that capture out-of-plane rotations of simple objects along only one axis: the vertical axis. We consider the Columbia University Image Library (COIL-20) data-set [23], which consists of gray-scale images of multiple objects rotating continuously between 0 and 360 degrees.

We start by organizing the data in a tensor; since the objects rotate only along one axis, we have a 3-way tensor $\mathcal{T} \in \mathbb{R}^{I_1 \times I_2 \times I_3}$, where I_1 is the number of objects, I_2 is the number of views, I_3 is the number of pixels in an image. Then, we apply HOSVD (eq. 4) to the data tensor to obtain a representation that separates the different variation factors that compose the data: object identity, viewpoint, and appearance. The tensor $\mathcal{T}$ is decomposed as:

$$\mathcal{T} = G \times_1 U^{(id)} \times_2 U^{(view)} \times_3 U^{(pix)} \quad (8)$$

where G is the core tensor that governs the interaction between the modes; $U^{(id)}$ spans the identity subspace and contains row vector coefficients $u^{(id)}$ for each object; $U^{(view)}$ spans the viewpoint subspace and contains row vector coefficients $u^{(view)}$ for each rotation angle; and $U^{(pix)}$ spans the image/pixel space.

Let us focus on the rotation subspace. Figure 2 shows the values of the first three coefficients of vector $u^{(view)}$ for different rotation angles (e.g. the first three columns of matrix $U^{(view)}$). The figure also shows the corresponding images of one of the objects in the dataset to better illustrate the rotation angles. We can see that the values displayed in Figure 2 approximately describe a spiral curve, making apparent that the coefficients of the rotation subspace follow a uni-dimensional manifold structure. This is consistent with the fact that the variations captured by this subspace correspond to a single parameter: the yaw rotation angle.

In Figure 3 we plot the above three coefficients separately, against the rotation angle. It can be seen that the wave-forms obtained in this way resemble those from trigonometric functions (which seems reasonable to model rotation effects). It should be mentioned that, while we only display the first columns of matrix $U^{(view)}$, the remaining columns follow a similar pattern.

4.2. Head pose estimation framework

Based on the example from the previous section, we propose a 3D head pose estimation framework that combines multi-linear decomposition with an analytic formulation of each of the 3 rotation subspaces based on trigonometric functions.

To consider the general case of all possible rotations in 3D-space, we start by building a five-way tensor $\mathcal{T} \in \mathbb{R}^{I_1 \times I_2 \times \ldots \times I_5}$, where I_1 is the number of subjects in training set, I_2, I_3 and I_4 are the number of yaw, pitch and roll angles respectively, and I_5 is the dimension of the feature vector. This tensor is decomposed as:

$$\mathcal{T} = G \times_1 U^{(id)} \times_2 U^{(y)} \times_3 U^{(p)} \times_4 U^{(r)} \times_5 U^{(F)} \quad (9)$$

where G is the core tensor; $U^{(id)}$ corresponds to the identity subspace; $U^{(y)}$, $U^{(p)}$ and $U^{(r)}$ correspond to yaw, pitch and roll subspaces, and $U^{(F)}$ spans the feature subspace.

After obtaining the decomposition of the tensor $\mathcal{T}$ it is theoretically possible to use the second, third and fourth modes in eq. (9) to estimate the head rotation angles. Specifically, if $X \in \mathbb{R}^{I_5}$ is the feature vector of an input image with unknown pose, its rotation angles can be obtained by minimizing the reconstruction error [1, 32, 37]:

$$\underset{u^{(y)},u^{(p)},u^{(r)},u^{(id)}}{\operatorname{argmin}} \|X - \mathcal{W} \times u^{(y)} \times u^{(p)} \times u^{(r)} \times u^{(id)}\| \quad (10)$$

where $\mathcal{W} = G \times U^{(F)}$; $u^{(y)}, u^{(p)}, u^{(r)}$ and $u^{(id)}$ are the targeted vectors. The final head pose estimation is typically obtained by finding the nearest neighbors to the yaw, pitch and roll estimates from eq. (10) using the training samples from $U^{(y)}, U^{(p)}$ and $U^{(r)}$, respectively.

Unfortunately, it is difficult to guarantee accurate estimates based on eq. (10) given that the reconstruction error that is minimized depends on several factors. Indeed, the estimates obtained from eq. (10) are often not compliant with

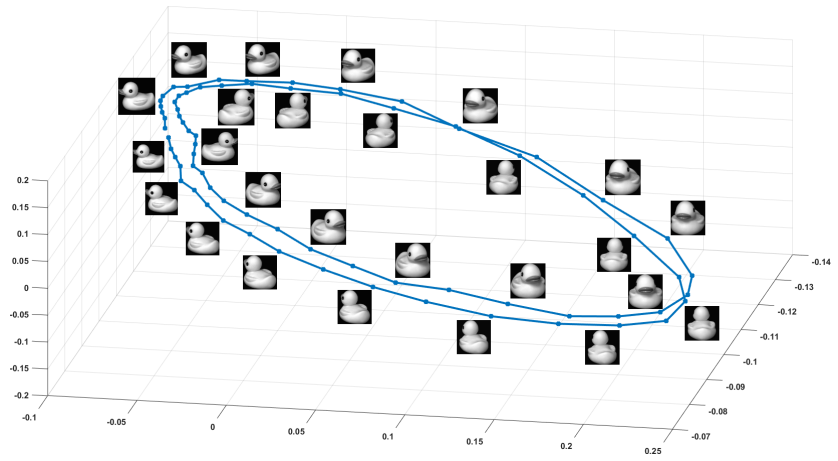

Figure 2. Visualization of the first three coefficients of the pose variation subspace.

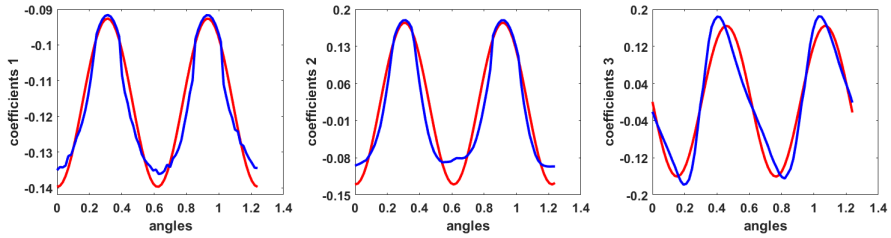

Figure 3. Values of the first three coefficients of the viewpoint subspace for the example of Section 4.1. The blue curves show the actual values of the first three columns of matrix $U^{(view)}$ and the red curves show their least-squares approximation with cosine functions

the manifold structure of the different subspaces, which is often enforced at a latter stage (e.g. nearest neighbor search [31]).

Thus, we propose to impose analytic constraints to the rotation subspaces by means of trigonometric functions. Concretely, for each of the targeted angles (generically ω) we model the coefficients in its corresponding subspace as:

$$f(\omega) = \alpha\, cos(\beta\omega + \gamma) + \varphi \tag{11}$$

where α, β, γ and φ are parameters learned by least squares approximation from the training set. Notice that there will be a different set of parameters for each dimension of the subspace, thus defining a spiral-like structure that analytically represents the underlying manifold.

Then, the head pose estimation in eq. (10) can be rewritten to include the trigonometric constraints as:

$$\underset{\phi,\theta,\psi}{\operatorname{argmin}} \|X - \mathcal{W} \times f(\phi) \times f(\theta) \times f(\psi) \times u^{(id)}\| \tag{12}$$

where f are the functions from eq. (11) that approximate each column of matrices $U^{(y)}, U^{(p)}$ and $U^{(r)}$ with cosine functions, $u^{(id)}$ is the vector of subject identity coefficients, and ϕ, θ, ψ are the targeted angles (yaw, pitch, roll respectively).

Notice that, for each angle, there are as many cosine functions as dimensions in the yaw, pitch and roll subspaces, but all the functions in a given subspace are governed by the same unique free variable – the rotation angle. Thus, with the above constraints, the minimization of the reconstruction error in eq. (12) directly yields a solution that is compliant with the unidimensional manifolds of the three rotation angles.

The different steps of the proposed framework for the training and test phases are summarized in Algorithms 1 and 2.

Algorithm 1: Training Phase

Input : X_k – feature vector for each of the subjects $k = 1..N$
ω – the set of angles

Output: $G, U^{(F)}$, set of parameters $\alpha, \beta, \gamma, \varphi$ for each of the matrices $U^{(y)}, U^{(p)}, U^{(r)}$

1 Build 5D tensor $\mathcal{T} \in \mathbb{R}^{I_1 \times I_2 \times \ldots \times I_5}$

2 Decompose $\mathcal{T}$ using HOSVD (eq.9)

3 $\mathcal{T} = G \times_1 U^{(id)} \times_2 U^{(y)} \times_3 U^{(p)} \times_4 U^{(r)} \times_5 U^{(F)}$

4 **foreach** *column* u_i *in matrices* $U^{(y)}, U^{(p)}, U^{(r)}$ **do**
$\operatorname{argmin}_{\alpha,\beta,\gamma,\varphi} \|u_i - \alpha\, cos(\beta\omega + \gamma) + \varphi\|$;

Algorithm 2: Test Phase

Input : X – feature vector of unknown subject
Output: estimated angles – ϕ, θ, ψ

Initialize :
  ϕ, θ, ψ with zeros;
  $u^{(id)}$ as vector with zeros;
Define :
  $f^{(y)}(\phi) = \alpha^{(y)} cos(\beta^{(y)}\phi + \gamma^{(y)}) + \varphi^{(y)}$
  $f^{(p)}(\theta) = \alpha^{(p)} cos(\beta^{(p)}\theta + \gamma^{(p)}) + \varphi^{(p)}$
  $f^{(r)}(\psi) = \alpha^{(r)} cos(\beta^{(r)}\psi + \gamma^{(r)}) + \varphi^{(r)}$
  $\mathcal{W} = G \times U^{(F)}$
Estimate angles ϕ, θ, ψ :
  $\text{argmin}_{\phi,\theta,\psi} \| X - \mathcal{W} \times f^{(y)}(\phi) \times f^{(p)}(\theta) \times f^{(r)}(\psi) \times u^{(id)} \|$

5. Experiments

Our framework focuses on analytically modeling the underlying structure of the pose manifold and can thus be applied to a wide variety of input features. Following the head pose estimation challenge organized on the SASE database [21], we base our experiments on the system proposed in [11], which was the winner of the challenge. We evaluate our framework in the same database of the aforementioned challenge (SASE [20]) and also in the widely used BIWI database [12], demonstrating state of the art performance in both cases.

5.1. Experiments using SASE database

The data in SASE has been acquired with Microsoft Kinect 2 camera and contains RGB and depth images in pairs. The entire database includes 50 subjects (32 male and 18 female) in the range of 7-35 years old, with more than 600 frames per subject. For each person, a wide range of yaw, pitch and roll variations are included. Specifically, yaw and pitch angles vary within ± 75 °, while roll angles vary within ± 45 ° [19].

The SASE database is distributed divided in two sets: *Training* (comprising 28 subjects with a total of $\sim$ 17K images) and *Validation* (12 subjects, $\sim$ 7K images)[11]. Thereby, we have used each of these sets for training and testing, respectively.

As mentioned before, we base our tests on the system described in [11], which is used as baseline. Specifically, for each input image, the head region is isolated using clustering and then a 3D mesh that contains the head and a variable part of the shoulders is reconstructed. Further, we use the Shape Regression with Incomplete Local Features (SRILF) algorithm [29] to automatically detect 12 prominent facial landmarks. An example of the 3D mesh of the face with the obtained landmarks is illustrated on Figure 4. Once the facial landmarks are available, we use their coordinates as input features to train and test our approach as described in Section 4.2. It is worth to mention that the use of SRILF to extract the input features provides robustness to both expression changes and missing parts. The latter is especially important in databases such as SASE and BIWI, because large pose variations induce self-occlusions that are likely to affect the visibility of some landmarks. SRILF deals with this problem by statistically inferring missing landmarks, thus providing a complete set of coordinates even under occlusions.

During the training phase, following Algorithm 1, we build a 5D tensor $\mathcal{T} \in \mathbb{R}^{28\times40\times40\times30\times36}$, that is: 28 subjects, 40 bins to discretize yaw and pitch angles, 30 bins for roll and 36-dimensional features (12 landmarks $\times$ 3 co-

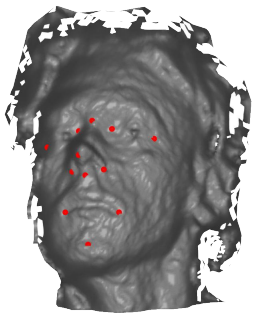

Figure 4. The example of the 3D mesh of the face with obtained landmarks

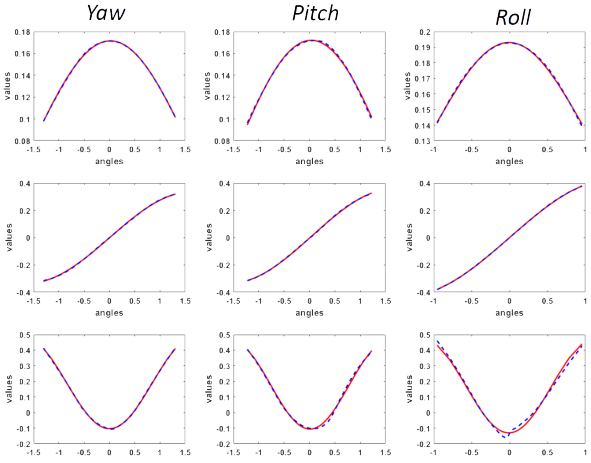

Figure 5. Curves defined by the coefficients in each of the subspaces corresponding to the head pose variation along one of the rotation axes. The first column corresponds to yaw rotation and shows the curves built from the coefficients of the first three columns of matrix $U^{(y)}$ (blue) and their approximation with a cosine function (red). The second and third columns correspond to pitch and roll angles, respectively

	Yaw (ϕ)	Pitch (θ)	Roll (ψ)
Without constraint	12.18	13.51	10.38
With constraint (proposed)	**6.50**	**7.07**	**6.06**

Table 1. Average pose estimation errors tested on the SASE database

ordinates). After the tensor is built, we decompose it using eq. (9), obtaining the core tensor $G \in \mathbb{R}^{28\times3\times3\times3\times10}$, $U^{(id)} \in \mathbb{R}^{28\times28}$ for the identity subspace; $U^{(y)} \in \mathbb{R}^{40\times3}$, $U^{(p)} \in \mathbb{R}^{40\times3}$ and $U^{(r)} \in \mathbb{R}^{30\times3}$ for yaw, pitch and roll subspaces, and $U^{(F)} \in \mathbb{R}^{36\times10}$ for the features subspace.

Next, we fit cosine functions to the pose coefficients (eq. 11) and obtain four parameters(α, β, γ and φ) for each of the coefficients of the three rotation subspaces (yaw, pitch and roll), thus achieving an analytic representation of the structure of the rotation manifolds. The results of the approximated coefficients for 3D pose variations are illustrated in Figure 5. For each of the rotation subspaces (yaw, pitch and roll), the first, second and third coefficients of all angle variations are plotted with two colors. The blue curves are the original values from the first three columns of the matrices $U^{(y)}, U^{(p)}$ and $U^{(r)}$ and the red curves are the approximated values obtained with cosine functions. It can be observed that the trigonometric approximation provides an excellent fit for the three rotation angles, with only minor deviations that could be easily attributed to noise in the data or in the extracted features.

After all function parameters were obtained, we used the estimation approach based on the minimization of the reconstruction error (eq. 12). For the test stage, $\sim$ 7K facial images from the Validation subset of SASE were used. We compared the obtained results of the proposed framework with respect to the approach based on minimizing the reconstruction error without any constraints. Table 1 summarizes the average pose estimation errors obtained by each approach. It can be seen that the approach based on the minimization without constraints obtained considerably higher estimation errors, confirming the usefulness of imposing manifold-compliant constraints.

To put our results in a wider context, we also compare them to other methods reporting head pose estimation error on the SASE database. Since this database is rather new, only a few papers have reported results on it. Table 2 summarizes the average pose estimation errors of the proposed framework and previous works on the SASE database. We can see that the proposed method performs well compared with state-of-the-art methods on the same dataset.

	Yaw (ϕ)	Pitch (θ)	Roll (ψ)
Lüsi *et al*. [19]	22	19	18
Derkach *et al*. [11]	6.51	7.49	6.52
Proposed	**6.50**	**7.07**	**6.06**

Table 2. Average pose estimation errors of the proposed framework and previous works on the SASE database

5.2. Experiments using BIWI database

The BIWI Database [12], acquired with a Kinect 1 sensor, contains 24 sequences of RGB-D images of subjects moving their heads over a range of roughly $\pm75°$ for yaw, $\pm60°$ for pitch and $\pm50°$ for roll. In total this database consists of around 17K images. Because there is no standard experimental protocol for this database, we perform our experiments under a leave-one-sequence-out strategy, so that no sequence is used for training and test at the same time. All other settings were kept as described in the previous section for the SASE database.

Table 3 summarizes our results, as well as those presented by previous works reporting pose estimation errors on this database. For each method, we show the average absolute error per angle together with the respective standard deviations (when provided by the authors). We also indicate the type of input data that is used (depth, RGB or both) and if pose estimations are done per-frame or using tracking.

The first thing we notice is that, despite our approach is the only one using just depth information without tracking, out results are quite competitive. Indeed, we clearly outperform other methods not doing tracking (except Papazov *et al*. [25] who report smaller averages but considerably higher standard deviations). Additionally, we achieve results that are comparable or better than four out the the seven tracking-based methods listed in Table 3, even though tracking-based algorithms benefit from the fact that test sequences start with nearly frontal head poses; thus, the accuracy of these algorithms to detect initial head poses other than frontal is not clear.

Another interesting aspect is that, among methods reporting standard deviations, our approach obtains the second-best results, only behind those from Padeleris *et al*. [24], who use tracking.

6. Conclusions

In this work we address 3D head pose estimation from depth data by proposing a novel approach to learn the manifold defined by 3D rotations. In particular, our method is able to explicitly model the underlying structure of the rotation manifold with an analytic form that takes into account the specific constraints imposed by orientation variations. For this purpose, we use multi-linear decomposition to split the pose variation factors into separate sub-spaces account-

Method	Tracking	Errors ± Std			Domain
		Yaw (ϕ)	Pitch (θ)	Roll (ψ)	
Wang [33]	No	8.8 ± 14.3	8.5 ± 11.1	7.4 ± 10.8	RGB + depth
Chen [8]	No	9.9 ± 12.4	12.8 ± 17.2	6.9 ± 9.8	RGB
Papazov [25]	No	2.5 ± 8.3	1.8 ± 4.3	2.9 ± 12.8	RGB + depth
	Yes	3.0 ± 9.6	2.5 ± 7.4	3.8 ± 16.0	
Padeleris [24]	Yes	2.4 ± 1.8	3.0 ± 2.16	2.8 ± 2.1	depth
Fanelli [13]	Yes	8.9 ± 13.0	8.5 ± 9.9	7.9 ± 8.3	depth
Meyer [22]	Yes	2.1	2.1	2.4	depth
Baltrusaitis [3]	Yes	6.3	5.1	11.3	RGB + depth
Li [17]	Yes	3.0	3.2	5.3	RGB + depth
Yu [36]	Yes	2.5	1.5	2.2	RGB + depth
Proposed	**No**	$\mathbf{3.6 \pm 4.6}$	$\mathbf{3.8 \pm 4.8}$	$\mathbf{5.2 \pm 5.8}$	**depth**

Table 3. Average pose estimation errors and standard deviations of the proposed frame-work and previous works on the BIWI database

ing for yaw, pitch and roll effects. We show that the coefficients within each of these subspaces define a continuous curve that can be modeled in terms of trigonometric functions, which are indeed the bases to explain rotation effects. We exploit this fact to introduce a minimization framework for pose estimation based on tensor decomposition constrained by trigonometric functions so that the obtained solutions are always compliant with the underlying manifold structure.

We show that the proposed modeling based on trigonometric functions can accurately capture the behaviour observed in the coefficients from the pose subspaces, by means of qualitative examples on 2D and 3D datasets. We also provide quantitative results of head pose estimation in two public database, which demonstrate the advantages introduced by the proposed constraints. Firstly, on the challenging SASE database, we show that directly applying existing multi-linear decomposition approaches yields poor pose estimation errors, which dramatically improve when introducing the proposed trigonometric constraints, reaching top state-of-the-art estimates. Later, we also report results on the widely used BIWI database, showing that the proposed framework is not only of theoretical interest but it can be translated into a practical system to produce competitive pose estimation results.

ACKNOWLEDGMENTS

This work is partly supported by the Spanish Ministry of Economy and Competitiveness under project grant TIN2017-90124-P, the Ramon y Cajal programme, and the Maria de Maeztu Units of Excellence Programme (MDM-2015-0502).

References

[1] A. Bakry and A. Elgammal. Untangling object-view manifold for multiview recognition and pose estimation. In *European Conference on Computer Vision*, pages 434–449. Springer, 2014.

[2] V. N. Balasubramanian, J. Ye, and S. Panchanathan. Biased manifold embedding: A framework for person-independent head pose estimation. In *Computer Vision and Pattern Recognition, 2007. CVPR'07. IEEE Conference on*, pages 1–7. IEEE, 2007.

[3] T. Baltrušaitis, P. Robinson, and L.-P. Morency. 3D constrained local model for rigid and non-rigid facial tracking. In *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*, pages 2610–2617. IEEE, 2012.

[4] C. BenAbdelkader. Robust head pose estimation using supervised manifold learning. In *European Conference on Computer Vision*, pages 518–531. Springer, 2010.

[5] G. Bergqvist and E. G. Larsson. The higher-order singular value decomposition: Theory and an application [lecture notes]. *IEEE Signal Processing Magazine*, 27(3):151–154, 2010.

[6] M. D. Breitenstein, D. Kuettel, T. Weise, L. Van Gool, and H. Pfister. Real-time face pose estimation from single range images. In *Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on*, pages 1–8. IEEE, 2008.

[7] L. M. Brown and Y.-L. Tian. Comparative study of coarse head pose estimation. In *Motion and Video Computing, 2002. Proceedings. Workshop on*, pages 125–130. IEEE, 2002.

[8] J. Chen, J. Wu, K. Richter, J. Konrad, and P. Ishwar. Estimating head pose orientation using extremely low resolution images. In *Image Analysis and Interpretation (SSIAI), 2016 IEEE Southwest Symposium on*, pages 65–68. IEEE, 2016.

[9] P. Comon. Tensors: a brief introduction. *IEEE Signal Processing Magazine*, 31(3):44–53, 2014.

[10] L. De Lathauwer, B. De Moor, and J. Vandewalle. A multilinear singular value decomposition. *SIAM journal on Matrix Analysis and Applications*, 21(4):1253–1278, 2000.

[11] D. Derkach, A. Ruiz, and F. M. Sukno. Head pose estimation based on 3-D facial landmarks localization and regres-

sion. In *Automatic Face & Gesture Recognition (FG 2017), 2017 12th IEEE International Conference on*, pages 820–827. IEEE, 2017.

[12] G. Fanelli, M. Dantone, J. Gall, A. Fossati, and L. Van Gool. Random forests for real time 3D face analysis. *International Journal of Computer Vision*, 101(3):437–458, 2013.

[13] G. Fanelli, T. Weise, J. Gall, and L. Van Gool. Real time head pose estimation from consumer depth cameras. In *Joint Pattern Recognition Symposium*, pages 101–110. Springer, 2011.

[14] Y. Fu and T. S. Huang. Graph embedded analysis for head pose estimation. In *Automatic Face and Gesture Recognition, 2006. FGR 2006. 7th International Conference on*, pages 6–pp. IEEE, 2006.

[15] T. G. Kolda and B. W. Bader. Tensor decompositions and applications. *SIAM review*, 51(3):455–500, 2009.

[16] D. Li and W. Pedrycz. A central profile-based 3D face pose estimation. *Pattern Recognition*, 47(2):525–534, 2014.

[17] S. Li, K. N. Ngan, R. Paramesran, and L. Sheng. Real-time head pose tracking with online face template reconstruction. *IEEE transactions on pattern analysis and machine intelligence*, 38(9):1922–1928, 2016.

[18] X. Liu, H. Lu, and W. Li. Multi-manifold modeling for head pose estimation. In *Image Processing (ICIP), 2010 17th IEEE International Conference on*, pages 3277–3280. IEEE, 2010.

[19] I. Lüsi, S. Escalera, and G. Anbarjafari. Human head pose estimation on SASE database using random hough regression forests. In *Video Analytics. Face and Facial Expression Recognition and Audience Measurement*, pages 137–150. Springer, 2016.

[20] I. Lüsi, S. Escarela, and G. Anbarjafari. SASE: RGB-depth database for human head pose estimation. In *Computer Vision–ECCV 2016 Workshops*, pages 325–336. Springer, 2016.

[21] I. Lüsi, J. C. S. Jacques Junior, J. Gorbova, X. Baró, S. Escalera, H. Demirel, J. Allik, C. Ozcinar, and G. Anbarjafari. Joint challenge on dominant and complementary emotion recognition using micro emotion features and head-pose estimation: Databases. In *Automatic Face and Gesture Recognition, 2017. Proceedings. 12th IEEE International Conference on*. IEEE, 2017.

[22] G. P. Meyer, S. Gupta, I. Frosio, D. Reddy, and J. Kautz. Robust model-based 3D head pose estimation. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 3649–3657, 2015.

[23] S. A. Nene, S. K. Nayar, H. Murase, et al. Columbia object image library (coil-20). 1996.

[24] P. Padeleris, X. Zabulis, and A. A. Argyros. Head pose estimation on depth data based on particle swarm optimization. In *Computer Vision and Pattern Recognition Workshops (CVPRW), 2012 IEEE Computer Society Conference on*, pages 42–49. IEEE, 2012.

[25] C. Papazov, T. K. Marks, and M. Jones. Real-time 3D head pose and facial landmark estimation from depth images using triangular surface patch features. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4722–4730, 2015.

[26] X. Peng, J. Huang, Q. Hu, S. Zhang, and D. N. Metaxas. Head pose estimation by instance parameterization. In *Pattern Recognition (ICPR), 2014 22nd International Conference on*, pages 1800–1805. IEEE, 2014.

[27] B. Raytchev, I. Yoda, and K. Sakaue. Head pose estimation by nonlinear manifold learning. In *Pattern Recognition, 2004. ICPR 2004. Proceedings of the 17th International Conference on*, volume 4, pages 462–466. IEEE, 2004.

[28] E. Seemann, K. Nickel, and R. Stiefelhagen. Head pose estimation using stereo vision for human-robot interaction. In *Automatic Face and Gesture Recognition, 2004. Proceedings. Sixth IEEE International Conference on*, pages 626–631. IEEE, 2004.

[29] F. M. Sukno, J. L. Waddington, and P. F. Whelan. 3-D facial landmark localization with asymmetry patterns and shape regression from incomplete local features. *IEEE transactions on cybernetics*, 45(9):1717–1730, 2015.

[30] Y. Sun and L. Yin. Automatic pose estimation of 3D facial models. In *Pattern Recognition, 2008. ICPR 2008. 19th International Conference on*, pages 1–4. IEEE, 2008.

[31] H. M. Takallou and S. Kasaei. Head pose estimation and face recognition using a non-linear tensor-based model. *IET Computer Vision*, 8(1):54–65, 2014.

[32] J. B. Tenenbaum and W. T. Freeman. Separating style and content. In *Advances in neural information processing systems*, pages 662–668, 1997.

[33] B. Wang, W. Liang, Y. Wang, and Y. Liang. Head pose estimation with combined 2d sift and 3d hog features. In *Image and Graphics (ICIG), 2013 Seventh International Conference on*, pages 650–655. IEEE, 2013.

[34] C. Wang, Y. Guo, and X. Song. Head pose estimation via manifold learning. In *Manifolds-Current Research Areas*. InTech, 2017.

[35] C. Wang and X. Song. Robust head pose estimation via supervised manifold learning. *Neural Networks*, 53:15–25, 2014.

[36] Y. Yu, K. A. F. Mora, and J.-M. Odobez. Robust and accurate 3D head pose estimation through 3dmm and online head model reconstruction. In *Automatic Face & Gesture Recognition (FG 2017), 2017 12th IEEE International Conference on*, pages 711–718. IEEE, 2017.

[37] H. Zhang, T. El-Gaaly, A. Elgammal, and Z. Jiang. Factorization of view-object manifolds for joint object recognition and pose estimation. *Computer Vision and Image Understanding*, 139:89–103, 2015.

[38] Y. Zhu, Z. Xue, and C. Li. Automatic head pose estimation with synchronized sub manifold embedding and random regression forests. *International Journal of Signal Processing, Image Processing and Pattern Recognition*, 7(3):123–134, 2014.