3D HEAD POSE ESTIMATION USING SUPPORT VECTOR MACHINES AND PHYSICS-BASED DEFORMABLE SURFACES

M. Krinidis, N. Nikolaidis and I. Pitas

Department of Informatics
Aristotle University of Thessaloniki
Box 451, 54124 Thessaloniki, GREECE
Email: $\{$*mkrinidi,nikolaid,pitas*$\}$*@aiia.csd.auth.gr*

ABSTRACT

This paper presents a novel approach for estimating $3D$ head pose in single-view video sequences. Following initialization by a face detector, a tracking technique that utilizes a $3D$ deformable surface model to approximate the image intensity is used to track the face in the video sequence. Head pose estimation is performed by using a feature vector which is a by-product of the equations that govern the deformation of the surface model used in the tracking. The afore-mentioned vector is used for training Support Vector Machines (SVM) in order to estimate the $3D$ head pose. The proposed method was applied to IDIAP head pose estimation database. The obtained results show that the proposed method can achieve an accuracy of 82% if angles are estimated in 10^o increments and 78% if angles are estimated in 5^o increments.

1. INTRODUCTION

Head pose estimation in video sequences is a frequently encountered task in many applications including intelligent surveillance and human-computer interaction, or as a preprocessing step in face detection, facial recognition and facial expression analysis, since face detection and recognition are very sensitive to even minor head rotations. Estimating head pose from a single camera is far from being a simple process due to image clutter, partial occlusions, unconstrained motion, varying lighting conditions, etc. A comparison of existing head pose estimation algorithms is given in [1, 2].

The single-view $3D$ head pose estimation approach proposed in this paper was motivated by the technique presented in [3] and [4], which aims at analyzing non-rigid object motion, with application to medical images. Based on a similar principle, we assume that the image intensity of a video frame forms a surface in $3D$ that can be approximated by a deformable surface. Then, we use the generalized displacement vector which is the result of an intermediate step of the deformation process, for both tracking the head and estimating its $3D$ pose in video sequences.

The tracking procedure is based on measuring and matching from frame to frame the generalized displacement vector of a deformable model placed on the face. The results indicate that this approach can offer reliable and robust tracking of the face. The generalized displacement vector is also used to train three SVM into estimating the pan, tilt and roll angles of the head. Angles are estimated in 10^o and 5^o increments. The proposed algorithm was tested on the IDIAP head pose database [2] which consists of different video sequences in natural environments with large rotations of the face. The database includes head pose ground truth information. The results show that the proposed algorithm can achieve an accuracy of 82%.

The remainder of the paper is organized as follows. The $3D$ physics-based deformable surface model is described in Section 2. Section 3 introduces the tracking algorithm and explains the derivation of the feature vector used for pose estimation. In Section 4 Support Vector Machines are reviewed and their use for pose estimation is explained. Performance evaluation of the proposed algorithm is provided in Section 5. Conclusions are drawn in Section 6.

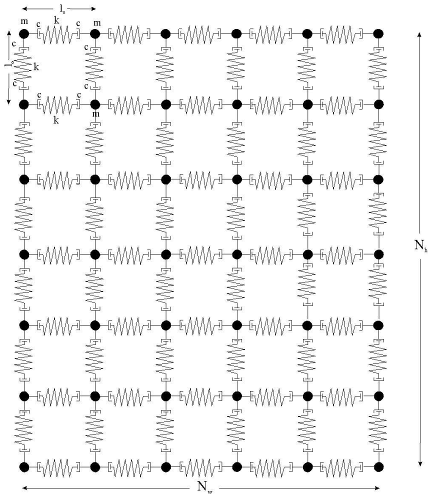

Fig. 1. *The elastic* $3D$ *physics-based deformable model consisting of* $N_h \times N_w$ *nodes.*

2. 3D PHYSICS-BASED DEFORMABLE SURFACE MODELING

Let $I(x, y)$ denote the intensity (grayscale value) of the pixel at position (x, y) on an image. By combining both the spatial (x, y) and grayscale $I(x, y)$ components of an image one can obtain a $3D$ surface representation $(x, y, I(x, y))$ of the image [5]. An elastic $3D$ physics-based deformable model [3] (Figure 1) consisting of a mesh of connected springs comprising of $N = N_h N_w$ nodes (assumed to be equal to the height and width of the image region of interest), can be used to approximate this surface through the application of forces that attract it towards the surface.

Nastar *et al.* [3] used deformable models to approximate the dynamic object surface deformations in time sequences of volume data. Modal analysis, which is a standard engineering technique was exploited to solve the deformation governing equations. In this paper, the deformable model formulation is used in a totally different application, i.e. that of face tracking and pose estimation.

The deformable model is used to approximate the intensity surface of the tracked face. In this case, the initial and the final (desirable) deformable surface states, i.e. the initial model configuration and the image intensity surface, are known and it can be assumed that a constant force load $\mathbf{f}$ is applied to the surface model. Thus, the equilibrium governing equation of the deformation procedure corresponds to the static problem:

$$\mathbf{K}\mathbf{u} = \mathbf{f}, \tag{1}$$

where $\mathbf{K}$ is the stiffness matrix, $\mathbf{u} = [\underline{\mathrm{u}}_1, \ldots, \underline{\mathrm{u}}_N]^T$ is defined as the vector comprising of the vectors of nodal displacements, and $\mathbf{f} = [\underline{\mathrm{f}}_1, \ldots, \underline{\mathrm{f}}_N]^T$ is the external force vector comprising of the external forces vectors applied to each node. The forces in this vector have zero x and y components whereas their z component is taken to be equal to the Euclidean distance between the point $(x, y, I(x, y))$ of the intensity surface and the corresponding node of the model in its initial configuration $(x, y, 0)$, i.e. equal to the intensity $I(x, y)$ of pixel (x, y): $f_z(x, y) = f_{(x-1)N_w+y,z} = I(x, y)$, where $f_{(x-1)N_w+y,z}$ is the z component of the $(x-1)N_w + y$-th element of vector $\mathbf{f}$.

Instead of solving directly the above equation for $\mathbf{u}$ one can use modal analysis and pursue a solution in the so-called modal space. Equation (1) is transformed in the modal space as follows:

$$\tilde{\mathbf{K}}\tilde{\mathbf{u}} = \tilde{\mathbf{f}}, \tag{2}$$

where $\tilde{\mathbf{K}} = \mathbf{\Phi}^T\mathbf{K}\mathbf{\Phi}$, $\mathbf{\Phi}$ is the matrix with the so-called vibration modes, $\tilde{\mathbf{f}} = \mathbf{\Phi}^T\mathbf{f}$ and $\tilde{\mathbf{u}}$ is the generalized displacement vector of the deformation process.

A significant advantage of the modal analysis described in [6], is that the vibration modes (eigenvectors) $\boldsymbol{\phi}_i$, i.e. the columns of $\mathbf{\Phi}$ and the frequencies (eigenvalues) ω_i of a plane topology do not have to be computed using eigen-decomposition techniques but have an explicit formulation [3]:

$$\begin{aligned} \omega^2(j, j') &= \omega^2_{(j-1)N_w+j'} = \\ &= \frac{4\underline{\mathrm{k}}}{m}\left(\sin^2\left(\frac{\pi j}{2N_h}\right) + \sin^2\left(\frac{\pi j'}{2N_w}\right)\right), \end{aligned} \tag{3}$$

where $j \in \{0, 1, \ldots, N_h - 1\}$, $j' \in \{0, 1, \ldots, N_w - 1\}$, $\underline{\mathrm{k}}$ is the stiffness of the springs, m is the mass of the nodes,

$$\begin{aligned} \boldsymbol{\phi}(j, j') &= \boldsymbol{\phi}_{(j-1)N_w+j'} = [\ldots, \\ &\cos\frac{\pi j(2n-1)}{N_h}\cos\frac{\pi j'(2n'-1)}{N_w}, \ldots]^T, \end{aligned} \tag{4}$$

where $n \in \{1, 2, \ldots, N_h\}$ and $n' \in \{1, 2, \ldots, N_w\}$.

We consider that no deformations occur along the x and y axes, i.e., deformations occur only along the intensity z axis, driven by the intensity (grayscale value) of the image under examination. Thus, for each component $[\tilde{u}_{x_i}, \tilde{u}_{y_i}, \tilde{u}_{z_i}]$ of vector $\tilde{\mathbf{u}}$ in (2), we have $\tilde{u}_{x_i} = \tilde{u}_{y_i} = 0$ and $\tilde{\mathbf{u}}$ is simplified to:

$$\tilde{\mathbf{u}} = [\tilde{u}_1, \ldots, \tilde{u}_{N_h N_w}]^T, \tag{5}$$

where $\tilde{u}_i \triangleq \tilde{u}_{z_i}$.

In the new basis, equation (1) is simplified to the following scalar equations:

$$\omega_i^2 \tilde{u}_i = \tilde{f}_i, \quad i = 1, \ldots, N, \tag{6}$$

where $\tilde{f}_i$ is the z component of the i-th elements $\tilde{\mathbf{f}}$.

Thus, instead of computing the displacements vector $\mathbf{u}$ from (1), one can firstly compute $\tilde{\mathbf{u}}$ in terms of (6), the frequencies (3) and the vibration modes (4) of the surface model. Once $\tilde{\mathbf{u}}$ has been computed, $\mathbf{u}$ can be calculated from the following equation:

$$\mathbf{u} = \mathbf{\Phi}\tilde{\mathbf{u}}. \tag{7}$$

In order to reduce the complexity of the problem, one can approximate nodal displacements by using only $N' < N_h N_w$ of the vibration modes $\boldsymbol{\phi}_i$ (those correspond to low frequency), or equivalently N' of the $\tilde{u}_i$. A value of N' which results in a compact but adequately accurate surface representation, is equal to 25% of the total number of the $\tilde{u}_i$.

3. FACE TRACKING AND DERIVATION OF THE POSE FEATURE VECTOR

Prior to the proposed tracking algorithm, a real-time frontal face detection algorithm [7] is applied to the first image of the video sequence. The face detection scheme is based on simple features that are reminiscent of Haar basis functions [8]. These features were extended in [7] to further reduce the number of false alarms. The output of the detection procedure is the center (x, y) (in pixels) of the face which usually corresponds to a point close to the nose.

Subsequently, a tracking approach similar to the one proposed in [6] is used to track the face center $\mathbf{p}^t = (x, y)$. This is done by applying the deformable model described in the previous section on a small window (e.g. one of dimensions 20×20 pixels) around this point and evaluating the generalized displacement vector $\tilde{\mathbf{u}}^t$ of equation (2) for this model:

$$\tilde{\mathbf{u}}^t(x, y) = [\tilde{u}_1^t(x, y), \tilde{u}_2^t(x, y), \ldots, \tilde{u}^t_{N_H N_W}(x, y)]^T, \tag{8}$$

where N_H and N_W are the height and width of the deformable surface model (equal to the dimensions of the window). We will call vector $\tilde{\mathbf{u}}^t(x, y)$ the *characteristic feature vector* (CFV).

In order to find the position $\mathbf{p}^{t+1} = (x', y')$ of the face center in the next frame I_{t+1}, the algorithm computes the CFV $\tilde{\mathbf{u}}^{t+1}(k, l)$ for each pixel of a search region R with height N_{Hreg} and width N_{Wreg}, centered at coordinates (x, y) in image I_{t+1}. The new location of the face center is found as the location (x', y') of the search region in the next frame whose CFV is closer to that of $\mathbf{p}^t$ in the current frame. More specifically:

$$\mathbf{p}^{t+1} = (x', y') \longrightarrow \arg\min_{kl}(|S^t_{x,y} - S^{t+1}_{k,l}|), \tag{9}$$

where $k \in \{x - \frac{N_{Hreg}-1}{2}, \ldots, x, \ldots, x + \frac{N_{Hreg}-1}{2}\}$ and $l \in \{y - \frac{N_{Wreg}-1}{2}, \ldots, y, \ldots, y + \frac{N_{Wreg}-1}{2}\}$ and $S^t_{x,y}$ is given by:

$$S^t_{x,y} = \sum_{i=1}^{N_H N_W} \left|\tilde{u}_i^t(x, y)\right|. \tag{10}$$

Since the motion characteristics of the face to be tracked might change over time, i.e. the face can speed up or slow down at certain frames, the algorithm uses a a search region R of variable size. For each frame the algorithm tries to locate the new center of the face using initially a small search region (e.g 7x7). However, if for the best candidate position the error $|S^t_{x,y} - S^{t+1}_{k,l}|$ in (9) is above a certain threshold, the algorithm increases the search region size, trying to find a better match (a match corresponding to a matching error below the threshold) in the larger search area. If this is again not feasible, the size increase continues up to a certain maximum region size.

In addition to its use for tracking, the CFV of the face center is used for deriving the head pose. The CFV contains information about the region around the center of the face, i.e. around the nose, since its elements are related to the displacements of the deformable surface model which approximates the intensity surface in this area. As the face/head changes orientation in the $3D$ space, its projection on the image ($2D$ space) changes. Thus, the fixed-size region centered at the nose includes the part of the face around the nose in different perspective views (Figure 2). Hence, this information can be used to derive the orientation of the face. The characteristics of the deformable surface model used in the experimental setup, were set so that the model was a rigid one. Thus, the final state of the deformable surface was a smoothed version of the face intensity surface, in order to be insensitive to clutter, dissimilarities of the faces between different persons and varying lighting conditions. By utilizing the truncated space of the modal analysis, one can reduce the size of the

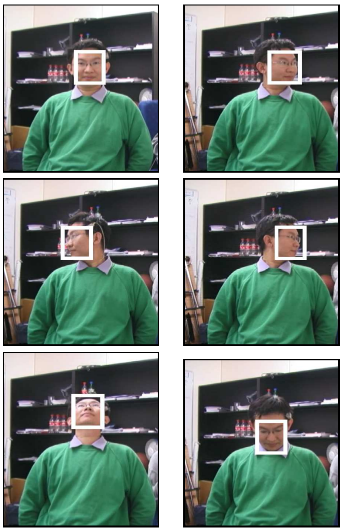

Fig. 2. *Different orientations of a face along with the region used for the evaluation of the CFV.*

CFV to 25% of its original size, without losing significant information. The information contained in the CFV was used along with appropriately trained SVM to derive pose information as will be described in the next section.

4. POSE ESTIMATION USING SUPPORT VECTOR MACHINES

Multiclass Support Vector Machines [9] (a generalization of the binary SVM) were used to classify the CFV $\tilde{\mathbf{u}}^t$ of frame I^t to one of the possible angle intervals for the three pose angles (pan, tilt, roll). More specifically, three SVM systems, each handling a different angle (pan, tilt, roll) of the face pose, were used. The value range of each of these parameters ($[-90\dots90]$ for pan, $[-60\dots60]$ for tilt and $[-30\dots30]$ for roll) was split into $M\in\{M_{pan},M_{tilt},M_{roll}\}$ intervals and each of the three M-class SVM systems was used to assign $\tilde{\mathbf{u}}^t$ to one of the corresponding M classes.

The main idea behind SVM is to construct hyperplanes that will separate the desired classes, in such a way that the margin (defined as the distance between the hyperplane and the nearest observation) is maximal. While training the SVM system, a set of CFVs $\tilde{\mathbf{u}}^t$ is used as an input, labelled properly with the true corresponding pose angles. To perform testing, an unlabelled feature vector $\tilde{\mathbf{u}}^{t'}$ is used as an input. The trained SVM system handling a certain pose angle, provides a label that classifies $\tilde{\mathbf{u}}^{t'}$ to one of the M possible intervals for this angle.

Through training the SVM creates a decision function $f(\tilde{\mathbf{u}}^t)$ which classifies a vector $\tilde{\mathbf{u}}^t$ into one of the M angle intervals, i.e. it provides a class label $l_j\in\{1\dots M\}$. To do so, the following equation should be minimized in the M class case:

$$\Theta(\mathbf{w},\xi)=1/2\sum_{m=1}^{M}(\mathbf{w}_m{}^T\cdot\mathbf{w}_m)+C\cdot\sum_{i=1}^{N}\sum_{m\neq l_i}\xi_i^m \tag{11}$$

with constraints

$$(\mathbf{w}_{l_i}{}^T\cdot\tilde{\mathbf{u}}_i)+b_{l_i}\geq(\mathbf{w}_m{}^T\cdot\tilde{\mathbf{u}}_i)+b_m+2-\xi_i^m \tag{12}$$

$$\xi_i^m\geq0,\quad i\in\{1,\dots,N\}\quad m\in\{1,...,M\}\backslash l_i, \tag{13}$$

where N is the number of the input vectors, $\mathbf{w}$ is the vector of hyperplane coefficients, $\boldsymbol{\xi}=[\xi_1,\dots,\xi_N]$ is the slack variable vector, $\mathbf{b}=[b_1,\dots,b_M]$ is the bias vector and C is the term that penalizes the training errors. The decision function derived from the minimization of (11) is of the general form:

$$f(\tilde{\mathbf{u}}^t)=\arg\max_n[(\mathbf{w}_n{}^T\cdot\tilde{\mathbf{u}}^t)+b_n],\quad n\in\{1,\dots M\} \tag{14}$$

The solution to this optimization problem in dual variables can be found by the saddle point of the Lagrangian

$$L(\mathbf{w},\mathbf{b},\boldsymbol{\xi},\boldsymbol{\alpha},\boldsymbol{\beta})=1/2\sum_{m=1}^{M}(\mathbf{w}_m{}^T\cdot\mathbf{w}_m)+C\sum_{i=1}^{N}\sum_{m=1}^{M}\xi_i^m$$
$$-\sum_{i=1}^{N}\sum_{m=1}^{M}\alpha_i^m[((\mathbf{w}_i-\mathbf{w}_m)^T\cdot\tilde{\mathbf{u}}_i)+b_{l_i}-b_m-2+\xi_i^m]$$
$$-\sum_{i=1}^{N}\sum_{m=1}^{M}\beta_i^m\xi_i^m \tag{15}$$

with the variables

$$\alpha_i^{l_i}=0,\quad\xi_i^{l_i}=2,\quad\beta_i^{l_i}=0,\quad i=\{1,\dots,N\} \tag{16}$$

and constraints

$$\alpha_i^m\geq0,\quad\beta_i^m=0,\quad\xi_i^m\geq0,\qquad i\in\{1,\dots,N\}\quad m\in\{1,...,M\}\backslash l_i \tag{17}$$

which has to be maximized with respect to $\boldsymbol{\alpha}$ and $\boldsymbol{\beta}$ (being the vectors of Lagragian multipliers) and be minimized with respect to $\mathbf{w}$ and $\boldsymbol{\xi}$.

By further processing [10] equation (14) is finally expressed as:

$$f(\tilde{\mathbf{u}}^t)=\arg\max_n[\sum_{i:l_i=n}A_i(\tilde{\mathbf{u}}_i^T\cdot\tilde{\mathbf{u}}^t)-\sum_{i:l_i\neq n}\alpha_i^n(\tilde{\mathbf{u}}_i^T\cdot\tilde{\mathbf{u}}^t)+b_n] \tag{18}$$

where A_i is defined as

$$A_i=\sum_{m=1}^{M}\alpha_i^m. \tag{19}$$

The previous analysis is used for linear decision surfaces. For the proposed method, nonlinear SVM were considered, i.e. a nonlinear mapping $Z(\tilde{\mathbf{u}}^t)$ to a high dimensional space was used. This mapping is defined by a positive kernel function, $k((\tilde{\mathbf{u}}^t)^T,\tilde{\mathbf{u}}^t)$, specifying an inner product in the feature space

$$Z((\tilde{\mathbf{u}}^t)^T)\cdot Z(\tilde{\mathbf{u}}^t)=k((\tilde{\mathbf{u}}^t)^T,\tilde{\mathbf{u}}^t). \tag{20}$$

The kernel used for the experiments was a d degree polynomial function, defined in general as

$$k((\tilde{\mathbf{u}}^t)^T,\tilde{\mathbf{u}}^t)=((\tilde{\mathbf{u}}^t)^T\cdot\tilde{\mathbf{u}}^t+1)^d. \tag{21}$$

In order to increase the performance of the head pose estimation system, a different variant where the input vectors of the SVMs consisted of a concatenation of the CFV $\tilde{\mathbf{u}}^t$ with the pose angles of the previous frame was also devised. More specifically, the SVMs were fed at time instant t with vectors of the form:

$$[\tilde{\mathbf{u}}^t|\theta_{t-1}]^T \tag{22}$$

where θ_{t-1} denotes the pan, tilt or roll angle in the previous frame. During training the ground truth pose angles were used whereas during testing, the estimates from the application of the system in the previous frame were inserted. This scheme can still be applied on-line, since for each frame only the angle estimates for the previous frame are necessary during testing.

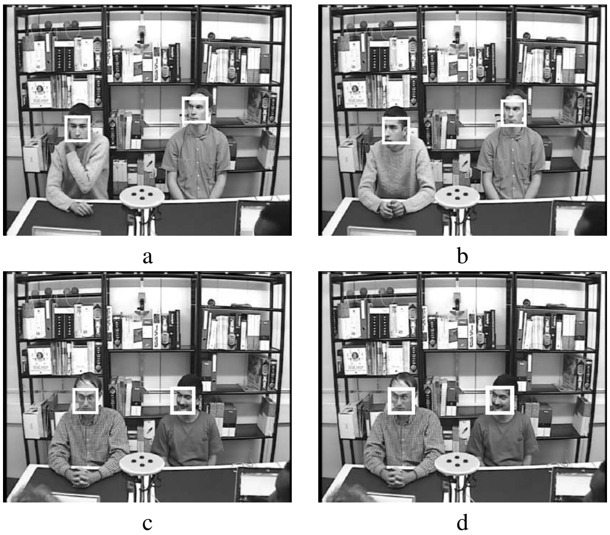

Fig. 3. *Tracking results for* 2 *different video sequences.*

5. EXPERIMENTAL RESULTS

The proposed method has been used to estimate the $3D$ head pose on parts of the IDIAP video database [2]. The database comprises of 23 video sequences involving people engaged in natural activities. In total, 16 different subjects participate in the video database. The database contains head pose ground truth in the form of pan, tilt and roll angles (i.e. Euler angles with respect to the camera coordinate system) for each frame of the video sequences. In all the experiments only 25% of the coefficients of the CFV were used for training and testing the SVM. The parameters of the deformable surface model were defined so as to give a smooth representation of the face intensity surface, i.e. a ratio $\frac{\underline{\mathrm{k}}}{m} = 10$ ($\underline{\mathrm{k}}$ being the stiffness of the springs and m the mass of the nodes) was used. Half of the video sequences were used for training the SVM and the rest for testing the system. The value ranges of the three pose angles were split into intervals of length 5^o and thus the pose angles were estimated in increments of 5^o. The accuracy of the system was measured as the percentage of frames where the ground truth pose angles were inside the same 5^o intervals as the estimated pose angles.

In the first set of experiments, the tracking algorithm was applied to the video sequences (see Figure 3) and the acquired CFVs were fed to the SVM. The results, in that case were not very satisfying. An average accuracy of 59% was achieved in this set of experiments. This is because video sequences were obtained in natural environment and the movements of the face are fast and sudden. Thus, tracking and subsequently pose estimation fail in certain cases.

In the next set of experiments, the above procedure was repeated, but this time, the variant where the pose angles for the previous frame were appended to the current CFV was used. The head pose estimation accuracy was significantly increased to 78%.

In the last set of experiments, the previous procedures were repeated but this time the pose angles were estimated in increments of 10^o. The average accuracy when SVM were fed only with CFVs was 64%. However, when the estimation of the pose angle in the previous frame was included in the input vector, the average accuracy of the proposed system was increased to 82%.

6. CONCLUSION

A novel $3D$ head pose estimation algorithm for single-view video sequences that utilizes a combination of $3D$ deformable surface models that approximate the image intensity surface, with SVM is introduced in this paper. An intermediate step of the deformation procedure, the so-called generalized displacement vector, is used for both tracking and pose estimation through appropriately trained SVM. The obtained results indicate that the introduced algorithm achieves an accuracy of 82% or 78% if angles are estimated in 10^o and 5^o increments respectively. Improvements of the method by using a two-pass approach are currently under investigation.

7. ACKNOWLEDGEMENT

This work has been conducted in conjunction with the ”SIMILAR” European Network of Excellence on Multimodal Interfaces of the IST Programme of the European Union ($http://www.similar.cc$).

8. REFERENCES

[1] L. M. Brown, Y.-L. Tian, Comparative Study of Coarse Head Pose Estimation, *IEEE Workshop on Motion and Video Computing*, pp. 125- 130, December, 2002.

[2] S. Ba, J.-M. Odobez, Evaluation of Multiple Cues Head Pose Estimation Algorithms in Natural Environments, *IEEE International Conference on Multimedia and Expo (ICME)*, Amsterdam, 2005.

[3] C. Nastar, N. Ayache, Frequency-based Nonrigid Motion Analysis: Application to four dimensional medical images, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 18, no. 11, pp. 1069-1079, 1996.

[4] S. Krinidis, C. Nikou, I. Pitas, ”Reconstruction of Serially Acquired Slices using Physics-based Modelling,” *IEEE Transactions on Information Technology in Biomedicine*, vol. 7, no. 4, pp. 394-403, December, 2003.

[5] B. Moghaddam, C. Nastar, A. Pentland, A Bayesian Similarity Measure For Direct Image Matching, *International Conference on Pattern Recognition (ICPR 1996)*, pp .350-358, Vienna, Austria, August, 1996.

[6] M. Krinidis, N. Nikolaidis, I. Pitas, Feature-based tracking using 3D physics-based deformable surfaces, *Proc. of 2005 EURASIP European Signal Processing Conference (EUSIPCO 2005)*, Antalya, Turkey, September, 2005.

[7] R. Lienhart, J. Maydt, An Extended Set of Haar-Like Features for Rapid Object Detection, *IEEE International Conference on Image Processing (ICIP02)*, pp. 900-903, Rochester, New York, USA, September,2002.

[8] P. Viola, M. J. Jones, Robust Real-time Object Detection, *Cambridge Research Laboratory*, Technical Report 2001/01, 2001.

[9] V. Vapnik, “The nature of statistical learning theory,” *Springer Verlag*, 1995.

[10] J. Weston, C. Watkins, Multi-class Support Vector Machines, Techical report CSD-TR-98-04, 2004.