FEATURE-BASED TRACKING USING 3D 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 tracking feature points in video sequences. In this method, the image intensity is represented by a $3D$ deformable surface model. Tracking is performed by exploiting a by-product of explicit surface deformation governing equations. The proposed method was compared with the well known KLT tracking algorithm, in terms of tracking accuracy and robustness. The obtained results show the superiority of the proposed method.

1. INTRODUCTION

Tracking rigid objects and object features in video sequences is a frequently encountered task in many video-based applications that include surveillance, video editing, virtual reality and computer animation, human-computer interaction and $3D$ scene reconstruction from uncalibrated video. It is obvious that building a tracking system is far from being a simple process due to varying lighting conditions, partial occlusions, clutter, unconstrained motion, etc. For a comprehensive review of different tracking methods the reader is referred to [1]-[3].

The feature based tracking approach proposed in this paper was motivated by the technique presented in [4] and [5], which aims at analyzing non-rigid object motion, with application to medical images. Nastar *et al.* [4] approximated the dynamic object surface deformations using a physically based deformable model. Based on the same principle, we assume that the image intensity of a video frame forms a deformable surface and use the generalized displacement vector which is the result of an intermediate step of the deformation process introduced in [6], for tracking feature points in $2D$ video frames.

The tracking procedure is based on measuring and matching the generalized displacement vector from frame to frame. Consequently, the tracking of $2D$ feature points in video sequences is transformed into tracking feature points in a vectorial space. The results indicate that this novel approach can offer reliable and robust tracking.

The proposed method can be used to track rigid and deformable objects. Our experiments involved tracking of human faces. It was assumed that the scenes contain a single object at most, occlusion is restricted and the initial position of the object of interest is known. It is, however, important to note that tracking can be performed in scenes with uncontrolled lighting conditions and a complex/moving background. The proposed algorithm was compared against the well-known and widely used Kanade-Lucas-Tomasi (KLT) tracker [7] using ground truth data and proved to exhibit better performance.

2. 3D PHYSICS-BASED DEFORMABLE SURFACE MODELING

Image intensity can be assumed to define a surface over the image domain that will be subsequently called intensity surface. Let $I(x,y)$ denote the intensity (grayscale value) of the pixel at position (x,y) on the image under study. By combining both the spatial (x,y) and grayscale $I(x,y)$ components of an image one can efficiently obtain a $3D$ surface representation $(x,y,I(x,y))$ of the image [8] (Figures 2a, b). An elastic $3D$ physics-based deformable model (Figure 1) [4] consisting of a mesh of $N = N_h N_w$ nodes, assumed to be equal to the image height and width, can be used to model this surface.

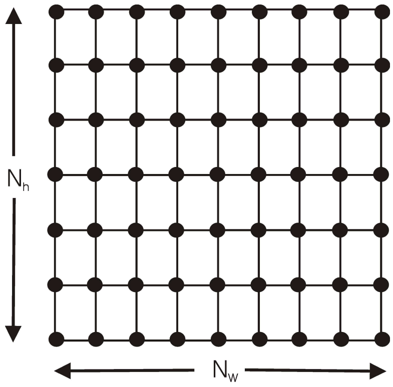

Figure 1: *The elastic* 3*D physics-based deformable model consisting of* $N_h \times N_w$ *nodes.*

The deformable surface model is ruled by Lagrangian dynamics [9]:

$$\mathbf{M}\ddot{\mathbf{u}}^{\tau} + \mathbf{C}\dot{\mathbf{u}}^{\tau} + \mathbf{K}\mathbf{u}^{\tau} = \mathbf{f}^{\tau}, \tag{1}$$

where $\mathbf{u}^{\tau}$ stores the displacements for spatial and grayscale values of the image and τ denotes the τ-th deformation time instance. $\mathbf{M}$, $\mathbf{C}$, and $\mathbf{K}$ [4, 6] are, respectively, the mass, damping, and stiffness matrices of the model and $\mathbf{f}^{\tau}$ is the external force vector, usually resulting from the attraction of the model by the image intensity and the pixel coordinates.

Instead of finding directly the equilibrium solution of (1), one can transform it by a basis change [6]:

$$\mathbf{u}^{\tau} = \mathbf{\Psi}\tilde{\mathbf{u}}^{\tau}, \tag{2}$$

where $\mathbf{\Psi}$ is a square nonsingular transformation matrix of order N to be determined and $\tilde{\mathbf{u}}^{\tau}$ is referred to as the *generalized displacements* vector. One effective way of choosing $\mathbf{\Psi}$ is setting it equal to matrix $\mathbf{\Phi}$ whose entries are the eigenvectors $\boldsymbol{\phi}_i$ (called vibration modes) of the generalized eigenproblem:

$$\mathbf{K}\boldsymbol{\phi}_i = \omega_i^2 \mathbf{M}\boldsymbol{\phi}_i, \tag{3}$$

$$\mathbf{u}^{\tau} = \mathbf{\Phi}\tilde{\mathbf{u}}^{\tau} = \sum_{i=1}^{N=N_h N_w} \tilde{u}_i^{\tau}\boldsymbol{\phi}_i. \tag{4}$$

Equation (4) is referred to as the *modal superposition equation*. $\tilde{u}_i^\tau$ is the amplitude of the i-th component of $\tilde{\mathbf{u}}^\tau$ and ω_i is the corresponding eigenvalue (also called *frequency*). If the matrix $\tilde{\mathbf{C}} = \mathbf{\Phi}^T\mathbf{C}\mathbf{\Phi}$ is diagonal (called standard Rayleigh hypothesis in [4]), then, the governing matrix-form equation is decoupled into N scalar equations in the modal space:

$$\ddot{\tilde{u}}_i^\tau + \tilde{c}_i\dot{\tilde{u}}_i^\tau + \omega_i^2\tilde{u}_i^\tau = \tilde{f}_i^\tau, \qquad i = 1,\ldots,N, \tag{5}$$

where $\tilde{c}_i$ is the i-th diagonal element of $\tilde{\mathbf{C}}$, $\tilde{f}_i^\tau$ is the i-th component of $\tilde{\mathbf{f}}^\tau$, where $\tilde{\mathbf{f}}^\tau = \mathbf{\Phi}^T\mathbf{f}^\tau$, $\mathbf{f}^\tau$ being the external force vector based on the Euclidean distance between a pixel of the image and the corresponding node coordinates. Solving these equations at iteration τ leads to $\tilde{u}_i^\tau$. When the iterative procedure converges, the displacement vector $\mathbf{u}^\tau$ of the model nodes is obtained by the modal superposition equation (4).

A significant advantage of the formulations described so far, is that the vibration modes (eigenvectors) $\boldsymbol{\phi}_i$ 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 [4]:

$$\omega^2(j,j') = \frac{4k}{m}\left(\sin^2\left(\frac{\pi j}{2N_h}\right) + \sin^2\left(\frac{\pi j'}{2N_w}\right)\right), \tag{6}$$

$$\boldsymbol{\phi}(j,j') = \left[\ldots, \cos\frac{\pi j(2n-1)}{N_h}\cos\frac{\pi j'(2n'-1)}{N_w}, \ldots\right]^T, \tag{7}$$

where $j \in \{0,1,\ldots,N_h-1\}$, $j' \in \{0,1,\ldots,N_w-1\}$, $n \in \{1,2,\ldots,N_h\}$, $n' \in \{1,2,\ldots,N_w\}$, $\omega^2(j,j') = \omega^2_{(j-1)N_w+j'}$ and $\boldsymbol{\phi}(j,j') = \boldsymbol{\phi}_{(j-1)N_w+j'}$.

In our case, where the initial and the final (desirable) deformable surface states, i.e. the initial model configuration and the image intensity surface, are known, it is assumed that a constant force load $\mathbf{f}$ is applied to the surface model. Thus, equation (1) is called the equilibrium governing equation and corresponds to the static problem:

$$\mathbf{K}\mathbf{u} = \mathbf{f}, \tag{8}$$

or in the modal space:

$$\tilde{\mathbf{K}}\tilde{\mathbf{u}} = \tilde{\mathbf{f}}, \tag{9}$$

where $\tilde{\mathbf{K}} = \mathbf{\Phi}^T\mathbf{K}\mathbf{\Phi}$.

In the new basis, equation (9) is simplified to $3N$ scalar equations:

$$\omega_i^2\tilde{u}_i = \tilde{f}_i. \tag{10}$$

Thus, instead of computing the displacements vector $\mathbf{u}$ from (8), $\mathbf{u}$ can be computed in terms of (10) and the vibration modes of the original surface model in (4). Thus, the final surface representation $\mathbf{v}$ (Figure 2c) is given by adding the deformations to the initial surface model $\mathbf{v}^{\tau_0}$:

$$\mathbf{v} = \mathbf{v}^{\tau_0} + \mathbf{u}. \tag{11}$$

3. FEATURE POINT TRACKING

The $2D$ feature point tracking problem is equivalent to finding the successive locations of the feature points in a temporal image sequence. Given an image sequence $\mathbf{I} = I_1, I_2, \ldots, I_T$ and a feature point $p_i^t = (x,y)$, $(t \in \{1,2,\ldots,T\})$ in the t-th image frame, the tracking problem can be formulated into finding a motion vector $d_i^t = (d_x^t, d_y^t)$, where (d_x^t, d_y^t) are the translation components of point p_i^t along each axis respectively, in order to locate its position $p_i^{t+1} = (x', y')$ in the next image frame:

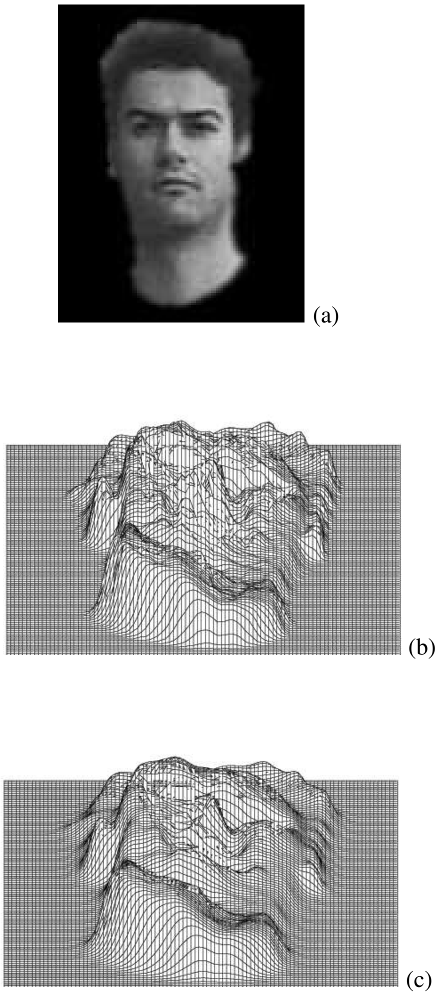

Figure 2: ***(a)** Facial image. **(b)** Surface representation of the image. **(c)** Deformed model.*

$$p_i^{t+1} = p_i^t + d_i^t. \tag{12}$$

The proposed tracking approach, computes for pixels (x,y) of an image I_t (or of an image region R_t) that correspond to feature points, the generalized displacement vector $\tilde{\mathbf{u}}^t$ of equation (9), on a small window around the pixel:

$$\tilde{\mathbf{u}}^t(x,y) = [\mathbf{l}^t_{1\,1}(x,y), \mathbf{l}^t_{1\,2}(x,y), \ldots, \mathbf{l}^t_{N_H N_W}(x,y)]^T, \tag{13}$$

where N_H and N_W are the height and width of the deformable surface model (being odd numbers) and ($\mathbf{l}^t_{i,j}(x,y) = [\tilde{u}^t_{x_{i,j}}, \tilde{u}^t_{y_{i,j}}, \tilde{u}^t_{z_{i,j}}]^T$) denote the displacements of the (i,j)-th node along x, y and z (intensity) axes respectively. We will call vector $\tilde{\mathbf{u}}^t(x,y)$ the *characteristic feature vector*.

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 $\mathbf{l}^t_{i,j}(x,y) = [\tilde{u}^t_{x_{i,j}}, \tilde{u}^t_{y_{i,j}}, \tilde{u}^t_{z_{i,j}}]^T$ of vector $\tilde{\mathbf{u}}^t(x,y)$ we have $\tilde{u}^t_{x_{i,j}} = \tilde{u}^t_{y_{i,j}} = 0$ and the characteristic feature vector is simplified to:

$$\tilde{\mathbf{u}}^t(x,y) = [\tilde{u}^t_{1\,1}(x,y), \ldots, \tilde{u}^t_{N_H N_W}(x,y)]^T, \tag{14}$$

where $\tilde{u}^t_{ij}(x,y) \triangleq \tilde{u}^t_{z_{ij}}(x,y)$. Using (4), (6) and (10), one can find that $\tilde{u}^t_{ij}(x,y)$ can be expressed as:

$$\tilde{u}^t_{ij}(x,y) \;=\; a_{ij}\sum_{k=0}^{N_H-1}\sum_{l=0}^{N_W-1}\mathbf{W}_{ij}(k,l)I_t\left(x-\frac{N_W-1}{2}+k,\; y-\frac{N_H-1}{2}+l\right), \quad (15)$$

where a_{ij} are constants ($a_{ij} < 1$) and $\mathbf{W}_{ij}$ are masks (matrices) of dimensions $N_H \times N_W$.

For the simplest case, where $N_H = N_W = 3$, the constants a_{ij} and masks $\mathbf{W}_{ij}$ that enter the evaluation of $\tilde{\mathbf{u}}^t(x,y)$, can be seen in Figure 3.

$a_{11} = 0.1111$ $\quad$ $a_{12} = 0.0962$ $\quad$ $a_{13} = 0.0278$

$$\mathbf{W}_{11} = \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix} \quad \mathbf{W}_{12} = \begin{bmatrix} 1 & 0 & -1 \\ 1 & 0 & -1 \\ 1 & 0 & -1 \end{bmatrix} \quad \mathbf{W}_{13} = \begin{bmatrix} 1 & -2 & 1 \\ 1 & -2 & 1 \\ 1 & -2 & 1 \end{bmatrix}$$

$a_{21} = 0.0962$ $\quad$ $a_{22} = 0.1111$ $\quad$ $a_{23} = 0.0385$

$$\mathbf{W}_{21} = \begin{bmatrix} 1 & 1 & 1 \\ 0 & 0 & 0 \\ -1 & -1 & -1 \end{bmatrix} \quad \mathbf{W}_{22} = \begin{bmatrix} 1 & 0 & -1 \\ 0 & 0 & 0 \\ -1 & 0 & 1 \end{bmatrix} \quad \mathbf{W}_{23} = \begin{bmatrix} 1 & -2 & 1 \\ 0 & 0 & 0 \\ -1 & 2 & -1 \end{bmatrix}$$

$a_{31} = 0.0278$ $\quad$ $a_{32} = 0.0385$ $\quad$ $a_{33} = 0.0159$

$$\mathbf{W}_{31} = \begin{bmatrix} 1 & 1 & 1 \\ -2 & -2 & -2 \\ 1 & 1 & 1 \end{bmatrix} \quad \mathbf{W}_{3\,2} = \begin{bmatrix} 1 & 0 & -1 \\ -2 & 0 & 2 \\ 1 & 0 & -1 \end{bmatrix} \quad \mathbf{W}_{3\,3} = \begin{bmatrix} 1 & -2 & 1 \\ -2 & 4 & -2 \\ 1 & -2 & 1 \end{bmatrix}$$

Figure 3: Constants (a_{ij}) and masks $\mathbf{W}_{ij}$ (matrices) of dimensions 3×3.

It can be seen that the masks in Figure 3 correspond to well known image processing operators, typically line and edge detectors. Masks $\mathbf{W}_{1\,2}$ and $\mathbf{W}_{2\,1}$ are the Prewitt operators [10] which look for edges in both vertical and horizontal directions. Additionally, masks $\mathbf{W}_{1\,3}$ and $\mathbf{W}_{3\,1}$ are detection masks for vertical and horizontal lines [10]. Moreover, masks $\mathbf{W}_{2\,3}$ and $\mathbf{W}_{3\,2}$ are edge detectors [11] and $\mathbf{W}_{3\,3}$ is the Laplacian line detection mask. In the case, where $N_H = N_W = 5$, the computed masks also correspond to line and edge detection operators.

To achieve tracking, the proposed approach computes for each feature point $p^t_i = (x,y)$ of the feature point set $\mathbf{p}^t$ in image frame I_t the characteristic feature vector $\tilde{\mathbf{u}}^t(x,y)$ over a window $N_H \times N_W$ centered around p^t_i and subsequently calculates $S^t_{x,y}$:

$$S^t_{x,y} = \sum^{N_H}_{(i,j)\neq(1,1)}\sum^{N_W}\left|\tilde{u}^t_{i,j}(x,y)\right|. \quad (16)$$

In order to find the position $p^{t+1}_i = (x',y')$ of the feature point i in the next image frame I_{t+1}, the algorithm computes the characteristic feature vector $\tilde{\mathbf{u}}^{t+1}(k,l)$ for each pixel of a search image region with height N_{Hreg} and width N_{Wreg} (being odd numbers), centered at coordinates (x,y) in image I_{t+1}. The new location of feature point i is given by:

$$p^{t+1}_i = (x',y') \longrightarrow \arg\min_{ij}(|S^t_{x,y} - S^t_{i,j}|), \quad (17)$$

where $i \in \{x - \frac{N_{Hreg}-1}{2}, \ldots, x, \ldots, x + \frac{N_{Hreg}-1}{2}\}$ and $j \in \{y - \frac{N_{Wreg}-1}{2}, \ldots, y, \ldots, y + \frac{N_{Wreg}-1}{2}\}$.

Since the motion characteristics of the object to be tracked might change over time, i.e. the object can speed up or slow down at certain frames, the algorithm uses a a search window R of variable size. For each frame the algorithm tries to locate the new position of a specific feature point using initially a small search window (e.g 7x7). However, if for the best candidate the error $|S^t_{x,y} - S^t_{i,j}|$ in (17) is above a certain threshold, the algorithm increases the search window size, trying to find a better match (a match corresponding to a matching error below the threshold) in a larger search area.If this is again not feasible, the size increase continues up to a certain maximum window size.

Tracking algorithms are usually preceded by an initialization step that aims at locating the target in the first frame, i.e., determining the initial feature point set $\mathbf{p}^1 = [p^1_1, p^1_2, \ldots, p^1_M]^T$. An initialization procedure based on the deformable surface models described in the previous Section has been also devised. According to this procedure, the initial feature point set is determined to be the M more salient feature points on the image, i.e. the M pixels that correspond to the M largest $S^t_{x,y}$ values in (16). We select the M maximum values because a large value of $S^t_{x,y}$ indicates that the $N_H \times N_W$ window around a pixel (x,y) contains edges, lines, corners or other characteristic features and thus the corresponding pixel is suitable for tracking. As it is expected, some of the selected M feature points are close to each other, since the image areas with edges, lines, corners etc. are not composed of a single pixel but of a set of pixels. If the selected feature points lie in a small neighborhood, problems in the subsequent tracking procedure can occur, e.g., in the case of partial occlusions where all feature points in an area might be lost. Thus, the M feature points $p_i, i \in \{1,2,\ldots,M\}$ that are finally selected, are the ones that have maximum $S^t(x,y)$ but at the same time maintain a certain Euclidean distance $\mathbb{D} = \|p^t_i - p^t_j\|$ from each other.

4. EXPERIMENTAL RESULTS

The proposed method was applied for face tracking on a set of studio test video sequences [12] as well as on outdoor video sequences. We compared our results with the ones produced by the well known feature-based Kanade-Lucas-Tomasi (KLT) tracking algorithm [7]. Our method was proven to be precise and robust in tracking the selected feature points, as will be seen subsequently.

The application of the two algorithms on a number of video sequences proves that the KLT algorithm ceases tracking feature points over time more frequently than the proposed method. This is illustrated in Table 1, that provides figures for the average life (in number of frames) of feature points for both algorithms and different sizes of the window R i.e. the model size for the proposed algorithm and the window considered by the KLT algorithm around each feature point. Lost feature points were detected by visual inspection.

Table 1: *The average life (in frames) of feature points for different model sizes.*

Model Size	Proposed tracker	KLT tracker
3×3	466.00	115.27
5×5	618.27	349.13
7×7	730.33	479.33

Furthermore, image intensity correlation for image regions around corresponding feature points was computed between the initial frame and the current one and the average correlation as well as the variance of the correlation over the entire video sequence for selected feature points (FP) was calculated. The average correlation can provide clues about the tracking performance of the algorithm since large average correlation indicates good tracking. The results are shown in Table 2. One can see that the average correlation over the entire video sequence for the proposed approach (AC-

PT) is much higher than the one achieved using KLT (AC-KLT). On the other hand, variance of correlation during the whole video sequence is smaller for the proposed approach (CV-PT) than for KLT (CV-KLT).

Table 2: *Average correlation and correlation variance for the proposed tracker and the KLT algorithm. (FP: feature point).*

	FP 1	FP 5	FP 10	FP 15	All FP
AC-PT	0.866	0.854	0.888	0.790	0.836
AC-KLT	0.596	0.642	0.786	0.694	0.633
CV-PT	0.007	0.007	0.005	0.015	0.012
CV-KLT	0.038	0.017	0.021	0.136	0.062

Finally, in order to evaluate tracking precision, we have manually produced ground truth data for a number of video sequences and compared it with the output of the two algorithms. The procedure that was used in this experiment was the following: we allowed KLT algorithm to select 9 feature points in the facial image region of the first frame of a number of video sequences. Afterwards, both algorithms were allowed to track the selected feature points for the rest of the video sequences. The positions which were produced by both algorithms were compared with the ground truth data, i.e. the manually selected positions of these feature points on each image. The Euclidean distance between the ground truth positions and the positions provided by the two algorithms, averaged over all feature points, was used for the comparison. Our algorithm was proven to be more precise in tracking as can be seen in Figure 4. The tracking error is constantly smaller for the proposed algorithm.

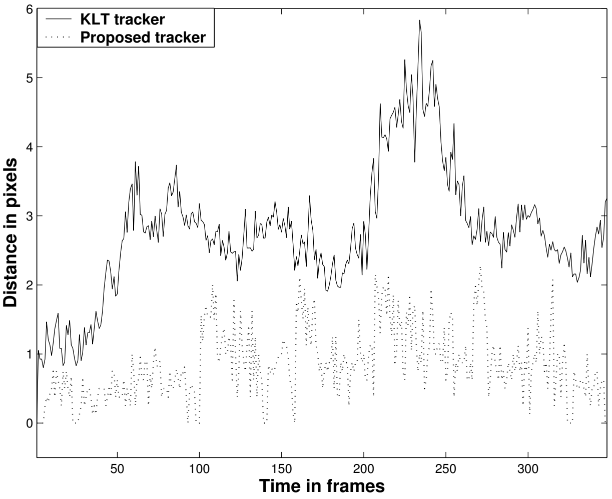

Figure 4: *Average Euclidean distance for all feature points between ground truth positions and tracked positions of the selected feature points for both algorithms.*

5. CONCLUSION

A novel 2*D* feature point tracking algorithm based on the use of a parameterized 3*D* physics-based deformable model was proposed in this paper. In this approach, the intensity surface of the image is represented by a 3*D* physics-based deformable model. We have shown how to tailor the deformation equations to track feature points in a video sequence. The presented tracking method was compared with the well known KLT algorithm. The results show that the proposed method produces superior tracking results, it provides better tracking accuracy and tracks feature points longer than KLT.

6. ACKNOWLEDGEMENT

The work presented was developed within VISNET, a European Network of Excellence (http://www.visnet-noe.org), funded under the European Commission IST FP6 programme.

REFERENCES

[1] T. B. Moeslund and E. Granum, “A survey of computer vision-based human motion capture,” *Computer Vision and Image Understanding*, vol. 81, pp. 231–268, 2001.

[2] J. J. Wang and S. Singh, “Video analysis of human dynamics - a survey,” *Real-Time Imaging*, vol. 9, no. 5, pp. 321–346, October 2003.

[3] G. Stamou, M. Krinidis, E. Loutas, N. Nikolaidis, and I. Pitas, “2D and 3D motion tracking in digital video,” in *Handbook of Image and Video Processing*, Alan C. Bovik, Ed. Academic Press, 2005.

[4] C. Nastar and 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.

[5] S. Krinidis, C. Nikou, and 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.

[6] A. Pentland and B. Horowitz, “Recovery of non-rigid motion and structure,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 13, no. 7, pp. 730–742, July 1991.

[7] C. Tomasi and T. Kanade, “Shape and motion from image streams: a factorization method,part 3, detection and tracking of point features,” Tech. Rep. CMU-CS-91-132, School of Computer Science Carnegie Mellon University Pittsburgh, 1991.

[8] B. Moghaddam, C. Nastar, and A. Pentland, “A bayesian similarity measure for direct image matching,” in *International Conference on Pattern Recognition (ICPR 1996)*, Vienna, Austria, August 1996, pp. 350–358.

[9] K. J. Bathe, *Finite Element Procedure*, Prentice Hall, Englewood Cliffs, New Jersey, 1996.

[10] R. Gonzalez and R. Woods, *Digital Image Processing*, Addison-Wesley Publishing Company, 1992.

[11] W. Frei and C.C. Chen, “Fast boundary detection: A generalization and a new algorithm,” *IEEE Transactions on Computers*, vol. C-26, no. 10, pp. 988–998, 1977.

[12] M. Krinidis, G. Stamou, H. Teutsch, S. Spors, N. Nikolaidis, R. Rabenstein, and I. Pitas, “An audio-visual database for evaluating person tracking algorithms,” in *Proc. of IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP 2005)*, Philadelphia, March 2005.