# 2D Feature Point Selection and Tracking Using 3D Physics-Based Deformable Surfaces

Michail Krinidis†, Nikos Nikolaidis† and Ioannis Pitas†

***Abstract*— This paper presents a novel approach for selecting and tracking feature points in video sequences. In this approach, the image intensity is represented by a $3D$ deformable surface model. The proposed approach relies on selecting and tracking feature points by exploiting the so-called generalized displacement vector that appears in the explicit surface deformation governing equations. This vector is proven to be a combination of the output of various line and edge detection masks, thus leading to distinct, robust features. The proposed method was compared, in terms of tracking accuracy and robustness, with a well known tracking algorithm (KLT) and a tracking algorithm based on SIFT features. The proposed method was experimentally shown to be more precise and robust than both KLT and SIFT tracking. Moreover, the feature point selection scheme was tested against the SIFT and Harris feature points and it was demonstrated to provide superior results.**

***Index Terms*— Feature point selection, tracking, $3D$ deformable models, intensity surface, video analysis.**

[1]

## I. Introduction

Tracking objects in video sequences is a frequently encountered task in video-based applications, such as surveillance, hand gesture recognition, human-computer interaction, smart environments, motion capture for virtual reality and computer animation, video editing, medical and meteorological imaging and $3D$ scene reconstruction from uncalibrated video. Thus, in the last two decades intensive research has been carried out in this area. Building a tracking system is far from being a simple process due to varying lighting conditions, partial occlusions, clutter, unconstrained motion, etc. So far, various systems for person, face and object tracking have been presented in the literature. These systems can be broadly divided in four categories:

- color-based tracking,
- template-based tracking,
- contour tracking,
- feature-based tracking.

Additional information about the aforementioned tracking categories can be found in the excellent review publications that have appeared in the literature [1]-[5].


†Aristotle University of Thessaloniki, Department of Informatics, Box 451, 54124 Thessaloniki, Greece, *email:* {*mkrinidi, nikolaid, pitas*}@ *aiia.csd.auth.gr* Fax/Tel ++ 30 231 099 63 04, http://www.aiia.csd.auth.gr/.
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).

Color is a distinctive object feature and, therefore, is useful for object localization on static and video images. Color information produces satisfactory tracking results and allows fast processing, which is important for a tracking system that needs to run at a reasonable frame rate. Many approaches are based on color histograms while some others use global color reference models [6]-[7].

Template matching techniques are used by many researchers to perform object tracking by following the same principles with the template matching techniques used in object recognition [8]-[9]. Template-based tracking involves the use of multiple templates or template warping to accommodate changes in object pose. The process of determining correspondences between image and template pixels is computationally expensive but provides robust tracking results.

Tracking using outline contour information is easier than modeling and tracking the entire object area, e.g. when using color. Moreover, contour tracking is more robust than using simple corner or edge tracking, since it can be adapted to cope with partial occlusions. The active contour representation introduced by Kass *et al.* [10], is the most popular method for contour delineation and tracking.

Feature-based tracking is a frequently used approach, in which moving objects are represented by feature points detected prior to tracking or during tracking. Feature-based tracking, though prone to tracking errors, can be implemented very efficiently and it is important in many time-critical applications. The selection of the feature points depends on the algorithm and usually is based on specific features (local image properties) of these points. A significant number of feature tracking algorithms have been introduced trying, in general, to correlate image features from frame to frame. In [11], the feature points are stochastically selected based on the energy of their Gabor wavelet transform coefficients. The global placement of the feature points is determined by a 2-D mesh, using the area of the triangles formed by the feature points. This method uses a local feature vector containing Gabor wavelet transform coefficients and a global feature vector containing triangle areas. In order to find the corresponding features in the next frame, the 2-D golden section algorithm is employed. Middle level features (strokes) are used in [12], instead of low-level ones (edge points). Strokes are accomplished by organizing edge points through an edge linking operation. Two labels (valid/invalid) are considered for each stroke and a probability is assigned to each of them. In this way, all the strokes contribute to track the moving object but with different weights. In [13], multiple features were used in order to improve the accuracy and robustness

of a real-time tracker. More specifically, color histogram features combined with edge-gradient-based shape features were tracked over time under a Monte Carlo framework. A comparison of four feature point tracking algorithms is given in [14]. Many researchers, instead of trying to improve the tracking performance through the selection of "good" features, exploited the knowledge of how a tracker works and tried to impose several constraints so as to improve the tracking of feature points [15], [16], [17].

In most of the cases, an initialization step which depends on the tracking algorithm is applied prior to tracking and defines the area of points that will be tracked. In feature based algorithms, several feature point selection strategies can be used. The goal is to obtain distinctive feature points on the image that are appropriate for tracking. Many of these feature points are also used for image matching applications, e.g., for finding the correspondences between two views of the same scene. Lowe proposed the Scale Invariant Feature Transform (SIFT) feature points [18], which are scale, rotation and partially illumination invariant. The SIFT feature points were used for image matching and image retrieval. Harris *et al.* [19] proposed a combined edge and corner detector which provides feature points that exhibit high "cornerness" and thus, are suitable for tracking. In [20], the feature points are extracted based on the eigenvalues of an image gradient matrix constructed over a window around the candidate feature point. If the minimum eigenvalue of this matrix is larger than a user-defined threshold, then the feature point is considered to be good for tracking. The extracted feature points are optimal for the tracking algorithm presented in the same paper. Moreover, a scheme for the selection of discriminative tracking features was proposed in [21]. Given a set of features, the log likelihood ratios of class conditional sample densities from the objects of interest and the background were computed, to form a new set of candidate features tailored to the local object/background discrimination task. The two-class variance ratio is used to rank these new features according to how well they separate sample distributions of object and background pixels. This feature evaluation mechanism is embedded in a mean-shift tracking system that adaptively selects the top-ranked discriminative features for tracking.

A novel feature selection and tracking algorithm is proposed in this paper. The approach was motivated by the technique presented in [22]-[24], which aims at analyzing non-rigid object motion, with application to medical images. Nastar *et al.* [22] used deformable models to approximate the dynamic object surface deformations in time sequences of volume data (i.e. sequences of $3D$ data) and applied modal analysis techniques (a standard engineering technique that allows more effective computations and provides closed form solution of the deformation process) in order to describe and analyze the deformations. The framework proposed in this paper has been also exploited for the alignment of serially acquired slices [24], for multimodal brain image analysis [23] and segmentation of $2D$ objects [25]. In our case, the deformable model formulation is used in a totally different and novel application, i.e. that of feature point tracking. We assume that the image intensity in each video frame can be approximated by a deformable "intensity" surface, where we select and track characteristic feature points. The proposed technique exploits a byproduct of the explicit surface deformation governing equations, in order to select and subsequently track distinctive feature points. More specifically, the feature point selection process utilizes the so-called *generalized displacement vector* [22], which is shown to be a novel combination of the output of various line and edge detection masks and thus, produces feature points corresponding to local edges, lines, corners or other characteristic image features that are suitable for tracking. The connection between the deformable surface model and the line/edge detector operators is an important outcome of this work. The tracking procedure that follows the feature selection is based on measuring and matching the generalized displacement vector of the feature points from frame to frame.

In summary the novelty of the paper lies in the use of a deformable surface to approximate the image intensity surface and the consequent use of a term appearing in the deformation procedure to perform robust feature point selection and tracking. With respect to the deformable model and its modal analysis as introduced in [26], [22] and further used in [23], [24], the novelty lies in the use of the model for a different application (i.e. that of feature selection and tracking), the use of different external forces that attract the model towards the image intensity (as will be described in section II) and the use of an intermediate result (generalized displacement vector) of the deformation procedure instead of using the model per se.

Compared to existing feature selection and tracking algorithms, namely the KLT [27] and the SIFT algorithm [28], the proposed method achieves better performance in terms of tracking accuracy and robustness. The results show that the proposed method is robust against rotations, zooming, varying lighting conditions and hard shadows and can track the selected features for long time periods. Moreover, the feature point selection part of the algorithm, was compared to other feature selection algorithms, i.e. the SIFT [18] and Harris [19] feature point detectors and was shown to provide superior results in their subsequent tracking.

The remainder of the paper is organized as follows. In Section II, a brief description of the deformation procedure based on modal analysis is presented. The feature point selection procedure is introduced in Section III. The tracking algorithm is described in Section IV. The performance of the proposed technique, as well as a comparison between the proposed algorithm and the well known KLT feature-based tracking algorithm [27], tracking using SIFT feature points [18] and feature selection using SIFT and Harris [19] feature detectors are presented in Section V. Final conclusions are drawn in Section VI.

## II. Deformable Model Description

Image intensity $I(x, y)$ can be assumed to define a surface over the image domain $(x, y)$ that will be subsequently called *intensity surface* (Figure 2b). The proposed tracking approach focuses on parameterizing the $3D$ space defined by $(x, y, I(x, y))$ that is called the $XYI$ space [29]. A

$3D$ physics-based deformable surface model, introduced in [22], [23], [26], is used for this purpose. In this section, the methodology described in these papers will be briefly reviewed, so as to make this paper self-contained. For more details, including the assumptions that are involved, interested readers can consult the above-mentioned papers.

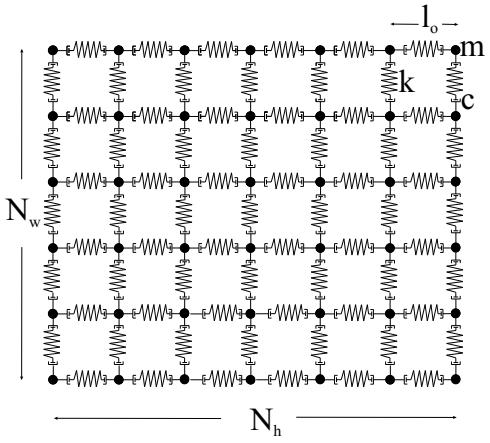



Fig. 1. *Quadrilateral surface (mesh) model.*

The deformable surface model consists of a uniform quadrilateral mesh of $N = N_h \times N_w$ nodes, as illustrated in Figure 1. In this section, we assume that $N_h$, $N_w$ are equal to the image height and width (in pixels) respectively, i.e. that each image pixel corresponds to one mesh node. The node coordinates of the model under examination are stacked in a vector:

$$\mathbf{v}^{(\tau)} = \left[\mathbf{r}_{1\,1}^{(\tau)}, \ldots, \mathbf{r}_{1\,N_w}^{(\tau)}, \mathbf{r}_{2\,1}^{(\tau)}, \ldots, \mathbf{r}_{j\,j'}^{(\tau)}, \ldots, \mathbf{r}_{N_h N_w}^{(\tau)}\right]^T = \left[\mathbf{r}_1^{(\tau)}, \ldots, \mathbf{r}_i^{(\tau)}, \ldots, \mathbf{r}_N^{(\tau)}\right]^T, \quad (1)$$

where $N = N_h \times N_w$, $j \in \{1, 2, \ldots, N_h\}$, $j' \in \{1, 2, \ldots, N_w\}$, $i = (j-1)N_w + j'$ and $i \in \{1, 2, \ldots, N\}$, $\mathbf{r}_i^{(\tau)} = [x_i^{(\tau)}, y_i^{(\tau)}, z_i^{(\tau)}]$. The coordinate $z_i^{(\tau)}$ corresponds to image intensity, namely $z_i^{(\tau)} = I(x_i^{(\tau)}, y_i^{(\tau)})$. $\tau$ denotes the $\tau$-th deformation time instance. Each model node is assumed to have a mass $m$ and is connected to its neighbors with perfect identical springs of stiffness $k$ having natural length $l_0$ and damping coefficient $c$. Under the effect of internal and external forces, the mass-spring system deforms to a $3D$ mesh representation of the image intensity surface, as can be seen in Figure 2d.

The model under study is a physics-based system governed by the fundamental dynamics equation:

$$f_{el}(\mathbf{r}_i^{(\tau)}) + f_d(\mathbf{r}_i^{(\tau)}) + f_{ext}(\mathbf{r}_i^{(\tau)}) = m_i \ddot{\mathbf{r}}_i^{(\tau)}, \quad i = 1, 2, \ldots, N, \quad (2)$$

where $\mathbf{r}_i^{(\tau)}$ is the $i$-th component of vector $\mathbf{v}^{(\tau)}$, i.e., the coordinates of the $i$-th node, $m_i$ its mass and $\ddot{\mathbf{r}}_i^{(\tau)}$ its acceleration under total force load. $f_d(\cdot)$ is a damping force proportional to node velocity $\dot{\mathbf{r}}_i^{(\tau)}$. $f_{ext}(\cdot)$ is the external force load on each node resulting from the attraction of the model by the image intensity, often based on the Euclidean distance between the intensity of an image pixel, whose representation in the $XYI$ space is $(x_{ij}, y_{ij}, I(x_{ij}))$ and the node coordinates [30], [31]. $f_{el}(\cdot)$ is the sum of elastic forces applied to the $i$-th node.

Under certain assumption that can be found in [22], the deformable surface model is ruled by Lagrangian dynamics [32]:

$$\mathbf{M}\ddot{\mathbf{u}}^{(\tau)} + \mathbf{C}\dot{\mathbf{u}}^{(\tau)} + \mathbf{K}\mathbf{u}^{(\tau)} = \mathbf{f}^{(\tau)}, \quad (3)$$

where $\mathbf{u}^{(\tau)}$ is the nodal displacement vector $\mathbf{u}^{(\tau)} = \mathbf{v}^{(\tau)} - \mathbf{v}^{(\tau_0)}$. $\mathbf{M}$, $\mathbf{C}$, and $\mathbf{K}$ [33] are, respectively, the $N \times N$ mass, damping, and stiffness matrices of the model and $\mathbf{f}^{(\tau)}$ is the external force vector.

If the initial and the final deformable surface states are known, it is assumed that a constant force load $\mathbf{f}$ is applied to the surface model [23]. This is the case of our problem formulation, where the initial state is the initial model configuration (Figure 2c) and the final, desirable state is the image intensity surface, shown in Figure 2b. Thus, equation (3) is transformed to the *equilibrium governing equation* that corresponds to the static problem:

$$\mathbf{K}\mathbf{u} = \mathbf{f}. \quad (4)$$

Instead of finding directly the equilibrium solution of (4), one can transform it by a basis change [34]:

$$\mathbf{u} = \mathbf{\Psi}\tilde{\mathbf{u}}, \quad (5)$$

where $\mathbf{\Psi}$ is a square nonsingular transformation matrix of order $N$ to be determined and $\tilde{\mathbf{u}}$ is referred to as the *generalized displacement* vector. One effective way of choosing $\mathbf{\Psi}$ is setting it equal to matrix $\mathbf{\Phi}$, whose columns are the eigenvectors $\boldsymbol{\phi}_i$ of the generalized eigenproblem:

$$\mathbf{K}\boldsymbol{\phi}_i = \omega_i^2 \mathbf{M}\boldsymbol{\phi}_i, \quad (6)$$

$$\mathbf{u}^{(\tau)} = \mathbf{\Phi}\tilde{\mathbf{u}}^{(\tau)} = \sum_{i=1}^{N=N_h N_w} \boldsymbol{\phi_i}\tilde{\mathbf{u}}_i^{(\tau)}. \quad (7)$$

The $i$-th eigenvector $\boldsymbol{\phi}_i$, i.e., the $i$-th column of $\mathbf{\Phi}$ is also called the $i$-th *vibration mode*. $\tilde{\mathbf{u}}_i$ is the $i$-th component of $\tilde{\mathbf{u}}$ and $\omega_i$ is the corresponding eigenvalue (also called *vibration frequency*). Equation (5) (and subsequently (7)) is known as *modal superposition equation*.

A significant advantage of this formulation, is that the vibration modes (eigenvectors) $\boldsymbol{\phi}_i$ and the frequencies (eigenvalues) $\omega_i$ of a plane topology have an explicit formulation [22] and they do not have to be computed using eigen-decomposition techniques:

$$\omega^2(j, j') = \frac{4k}{m}\left[\sin^2\left(\frac{\pi j}{2N_h}\right) + \sin^2\left(\frac{\pi j'}{2N_w}\right)\right], \quad (8)$$

$$\phi_{n,n'}(j, j') = \cos\frac{\pi j(2n-1)}{N_h}\cos\frac{\pi j'(2n'-1)}{N_w}, \quad (9)$$

where $j = 0, 1, \ldots, N_h - 1$, $j' = 0, 1, \ldots, N_w - 1$, $n = 1, 2, \ldots, N_h$, $n' = 1, 2, \ldots, N_w$, $\omega^2(j, j') = \omega_{j\,N_w+j'}^2$, $\phi_{n,n'}(j, j')$ is the $(n, n')$-th element of matrix $\boldsymbol{\phi}(j, j')$, where $\boldsymbol{\phi}(j, j') = \boldsymbol{\phi}_{j\,N_w+j'}$.

In the modal space, equation (4) can be written as:

$$\tilde{\mathbf{K}}\tilde{\mathbf{u}} = \tilde{\mathbf{f}}, \tag{10}$$

where $\tilde{\mathbf{K}} = \mathbf{\Phi}^T\mathbf{K}\mathbf{\Phi}$ and $\tilde{\mathbf{f}} = \mathbf{\Phi}^T\mathbf{f}$, $\mathbf{f}$ being the external force vector. Hence, by using (7), (8) and (9), equation (10) is simplified to $3N$ scalar equations:

$$\omega_i^2\tilde{u}_{i,j} = \tilde{f}_{i,j}. \tag{11}$$

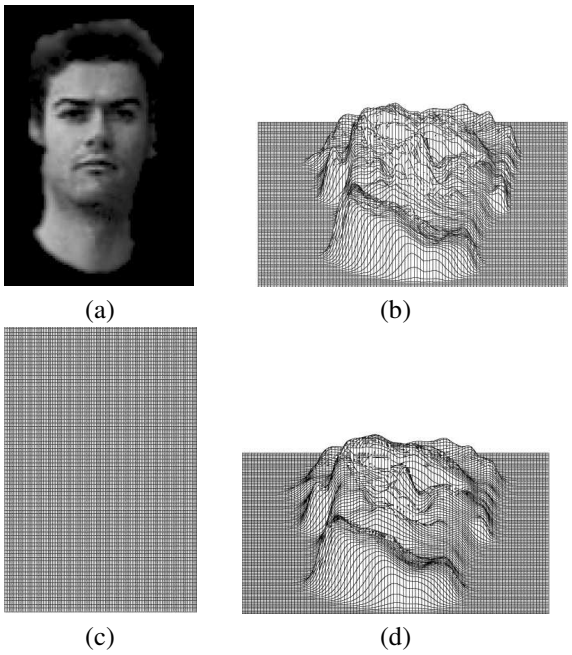


Fig. 2. ***(a)*** *Facial image,* ***(b)*** *intensity surface representation of the image,* ***(c)*** *the initial model configuration,* ***(d)*** *deformed model approximating the intensity surface.*

The components of the forces in $\mathbf{f}$ along the $x$ and $y$ axes are taken to be equal to zero, i.e. $f_{i,x} = f_{i,y} = 0$. The components of these forces along the $z$ (intensity) axis are taken to be proportional to the Euclidean distance between the point $(x, y, I(x, y))$ of the intensity surface and the corresponding model node position in its initial configuration $(x, y, 0)$, i.e., equal to the intensity $I(x, y)$ of pixel $(x, y)$: $f_{(x-1)N_w+y,z} = f(x, y) = I(x, y)$, where $f_{(x-1)N_w+y,z}$ is the component along $z$ axis of the $(x - 1)N_w + y$-th element of vector $\mathbf{f}$. Under such a condition, the model deforms only along the $z$ axis.

In our case, each frame of the video sequence is described in terms of vibrations of an initial model. Figure 2 illustrates the vibration mode parameterization of the $2D$ image of a human face shown in Figure 2a. The intensity surface representation of the image can be seen in Figure 2b. The size of the model (in nodes) that was used to parameterize the image surface was equal to the image size (in pixels). The quadrilateral mesh model is initialized (Figure 2c) and the elements $\tilde{u}_{ij}$ are explicitly computed:

$$\tilde{u}_{ij} = \frac{\sum_{n=1}^{N_h}\sum_{n'=1}^{N_w} I(n, n')\phi_{n,n'}(i, j)}{(1 + \omega^2(i, j))\sqrt{\sum_{n=1}^{N_h}\sum_{n'=1}^{N_w}\phi_{n,n'}^2(i, j)}}. \tag{12}$$

It should be noted that the deformable model achieves only an approximation of the intensity surface of the target image.

The generalized displacement vector $\tilde{\mathbf{u}}^{(t)}(x, y)$ of equation (10) is exploited, as will be shown in the following sections, in order to select and track feature points on $2D$ images. This vector will be called *characteristic feature vector* (CFV). A flow diagram of the proposed algorithm is shown in Figure 3. The details of the algorithm will be provided in the following sections.

## III. Image Feature Point Selection

In this section, we introduce the way in which $3D$ physics-based deformable surface models can be exploited to select characteristic feature points on an image. This feature point selection procedure can be applied on the first frame of a video sequence in order to initialize the tracking procedure that will be described in the next Section. The same procedure can be used to reinitialize the tracker in cases where such an action is required, e.g. when an object disappears (occlusion) or reappears (disocclusion). In order to select a pixel $(x, y)$ on the image $I_t$ at time $t$ as a characteristic feature point, we use the characteristic feature vector $\tilde{\mathbf{u}}^{(t)}(x, y)$ of (10) whose elements are given by (12).

For determining the CFV of a pixel $(x, y)$ of a video frame $I_t$, a deformable surface model of size $N_H \times N_W$ ($N_H \leq N_h$, $N_W \leq N_w$, $N_H$ and $N_W$ being odd numbers) is applied to an image window $\mathcal{D}_t$ of the same size ($N_H \times N_W$), centered at pixel $(x, y)$. The CFV $\tilde{\mathbf{u}}^{(t)}(x, y)$ for a specific pixel $(x, y)$ is:

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

where $\mathbf{k}_{i,j}^{(t)}(x, y) = [\tilde{u}_{x_{i,j}}^{(t)}(x, y), \tilde{u}_{y_{i,j}}^{(t)}(x, y), \tilde{u}_{z_{i,j}}^{(t)}(x, y)]^T$, is evaluated by applying the deformation procedure described in Section II, to the image window $D_t$. As already mentioned in the previous section, no deformations occur along the $x$ and $y$ axes, i.e., deformations occur only along the intensity axis ($z$ axis), driven by the image intensity under examination. Thus, $\tilde{u}_{x_{ij}}^{(t)}(x, y) = \tilde{u}_{y_{ij}}^{(t)}(x, y) = 0$ and the CFV is simplified to:

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

where $\tilde{u}_{ij}^{(t)}(x, y) \triangleq \tilde{u}_{z_{ij}}^{(t)}(x, y)$. Using (12), one can find that $\tilde{u}_{ij}^{(t)}(x, y)$ can be expressed as:

$$\tilde{u}_{ij}^{(t)}(x, y) = \sum_{k=0}^{N_H-1}\sum_{l=0}^{N_W-1}\Upsilon_{ij}(k, l) \cdot I_t\left(x - \frac{N_W - 1}{2} + k, y - \frac{N_H - 1}{2} + l\right), \tag{15}$$

where:

$$\Upsilon_{ij}(k, l) = \frac{\phi_{k,l}(i, j)}{(1 + \omega^2(i, j))\sqrt{\sum_{k=1}^{N_H}\sum_{l=1}^{N_W}\phi_{k,l}^2(i, j)}}, \quad 0 \leq k \leq N_H - 1, 0 \leq l \leq N_W - 1. \tag{16}$$

Evaluation of the matrices $\mathbf{\Upsilon}_{ij}$ (16) for typical values of $N_H$, $N_W$, (e.g. 3, 5) revealed that these matrices corresponds to

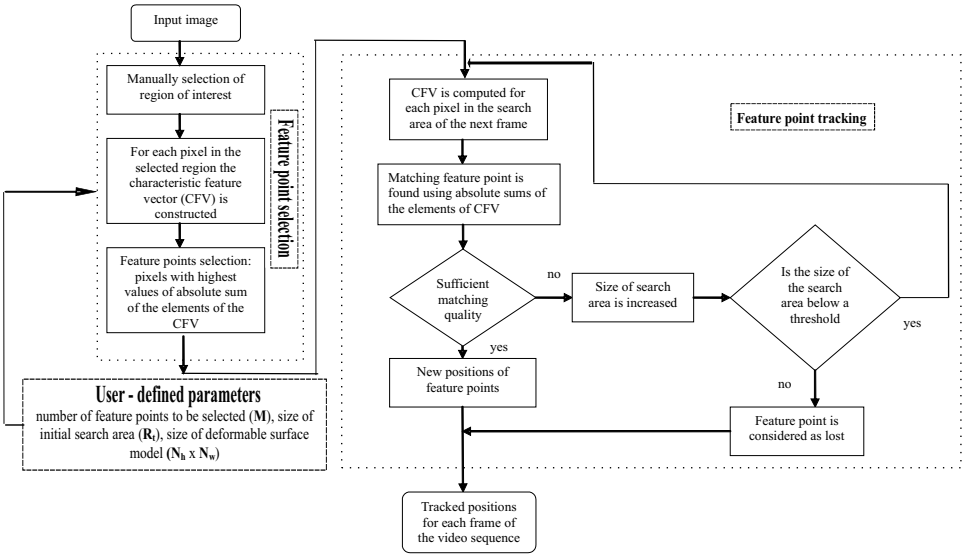



Fig. 3. *Flow diagram of the proposed feature point selection and tracking algorithm.*

well known image processing operator masks $\mathbf{W}_{ij}$, typically local line and edge detectors, scaled by a constant $a_{ij}$ whose value depends on the properties of the deformable surface, i.e. the stiffness of the springs and the mass of the nodes:

$$\mathbf{\Upsilon}_{ij}(k,l) = a_{ij}\mathbf{W}_{ij}(k,l). \tag{17}$$

For the simple case $N_H = N_W = 3$ and for a model with $k = 1$ and $m = 1$, the vector $\tilde{\mathbf{u}}^{(t)}(x,y)$ can be evaluated by applying the $a_{ij}$ and $\mathbf{W}_{ij}$ presented in Table I. It can be seen that indeed the masks in this table correspond to image processing operators. Masks $\mathbf{W}_{1\,2}$ and $\mathbf{W}_{2\,1}$ are the Prewitt edge detector operators [35], which detect edges along vertical and horizontal directions respectively. Additionally, masks $\mathbf{W}_{1\,3}$ and $\mathbf{W}_{3\,1}$ are vertical and horizontal line detection masks [35]. Moreover, masks $\mathbf{W}_{2\,3}$ and $\mathbf{W}_{3\,2}$ are edge detectors [36] and $\mathbf{W}_{3\,3}$ is the Laplacian line detection mask.

The masks and constants for the case $N_H = N_W = 5$ also correspond to local line and edge detection operators. Thus, the elements of the characteristic vector, evaluated using (15), are the outputs of line and edge detection operators, evaluated over the image window centered around pixel $(x,y)$ and weighted with appropriate coefficients. The connection between the characteristic feature vector (i.e. the generalized displacement vector of the deformable model) and the edge/line detection operators was one of the most interesting outcomes of this study. This connection can be attributed to way in which modal analysis operates. In essence, the modal superposition equation (5) performs a "frequency" decomposition of a shape, i.e., it decomposes it into a frequency-increasing set of modes (which are orthonormal basis functions corresponding to basic shapes) linearly combined by the modal amplitudes [22]. The masks related to the evaluation of the characteristic feature vector provide information for the frequency content of the image (edges, lines and mean intensity value or DC term) which enters into the frequency decomposition in the form of modal amplitudes $\tilde{\mathbf{u}}_i$ in (7).

In the proposed feature point selection approach, we compute the CFVs $\tilde{\mathbf{u}}^{(t)}(1,1)$, $\tilde{\mathbf{u}}^{(t)}(1,2)$, $\ldots$, $\tilde{\mathbf{u}}^{(t)}(N_{Hreg}, N_{Wreg})$, for each pixel $(x,y)$ of an image region $\mathrm{R}_t$ to be tracked at time instance $t$, where $N_{Hreg} \times N_{Wreg}$ is the image region size. Subsequently, the vector $\mathbf{S}^{(t)}$ whose elements $S^{(t)}_{x,y}$ are the sum of absolute values of the elements of CFVs $\tilde{\mathbf{u}}^{(t)}(x,y)$ is calculated:

$$\mathbf{S}^{(t)} = [S^{(t)}_{1\,1}, S^{(t)}_{1\,2}, \ldots, S^{(t)}_{N_{Hreg}N_{Wreg}}]^T, \tag{18}$$

$$S^{(t)}_{xy} \triangleq \sum^{N_H}\sum_{(k,l)\neq(1,1)}^{N_W} \left|\tilde{u}^{(t)}_{k,l}(x,y)\right|. \tag{19}$$

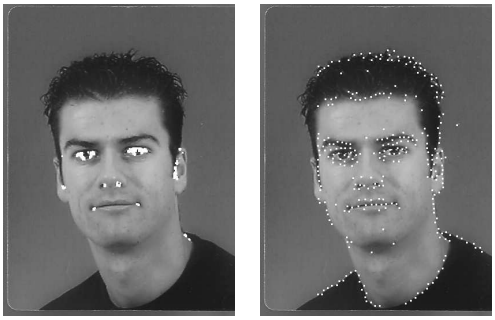
The element $\tilde{u}^{(t)}_{1,1}(x,y)$ is excluded from the calculations in (19), since this element corresponds to the mask $\mathbf{W}_{11}$, which simply calculates the local image average and therefore, bears no significant information. Thus, each pixel $(x,y)$ in the image (or the image region) under examination is assigned a scalar value. In order to select the $M$ most salient feature points (pixels) on the image, the $M$ pixels that correspond to the $M$ largest $S^{(t)}_{xy}$ values are selected, since a large value of $S^{(t)}_{xy}$ indicates that the $N_H \times N_W$ window around pixel $(x,y)$ contains edges, lines, corners or other characteristic formations and, thus, the corresponding pixel is suitable for tracking.

As expected, some of the $M$ selected feature points are located close to each other (Figure 4a), along image edges,

TABLE I

*Constants $a_{ij}$ and masks $\mathbf{W}_{ij}$ (matrices) of dimensions $3 \times 3$.*

| | | |
|---|---|---|
| $a_{11} = 0.1111$ $\mathbf{W}_{11} = \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}$ | $a_{12} = 0.0962$ $\mathbf{W}_{12} = \begin{bmatrix} 1 & 0 & -1 \\ 1 & 0 & -1 \\ 1 & 0 & -1 \end{bmatrix}$ | $a_{13} = 0.0278$ $\mathbf{W}_{13} = \begin{bmatrix} 1 & -2 & 1 \\ 1 & -2 & 1 \\ 1 & -2 & 1 \end{bmatrix}$ |
| $a_{21} = 0.0962$ $\mathbf{W}_{21} = \begin{bmatrix} 1 & 1 & 1 \\ 0 & 0 & 0 \\ -1 & -1 & -1 \end{bmatrix}$ | $a_{22} = 0.1111$ $\mathbf{W}_{22} = \begin{bmatrix} 1 & 0 & -1 \\ 0 & 0 & 0 \\ -1 & 0 & 1 \end{bmatrix}$ | $a_{23} = 0.0385$ $\mathbf{W}_{23} = \begin{bmatrix} 1 & -2 & 1 \\ 0 & 0 & 0 \\ -1 & 2 & -1 \end{bmatrix}$ |
| $a_{31} = 0.0278$ $\mathbf{W}_{31} = \begin{bmatrix} 1 & 1 & 1 \\ -2 & -2 & -2 \\ 1 & 1 & 1 \end{bmatrix}$ | $a_{32} = 0.0385$ $\mathbf{W}_{32} = \begin{bmatrix} 1 & 0 & -1 \\ -2 & 0 & 2 \\ 1 & 0 & -1 \end{bmatrix}$ | $a_{33} = 0.0159$ $\mathbf{W}_{33} = \begin{bmatrix} 1 & -2 & 1 \\ -2 & 4 & -2 \\ 1 & -2 & 1 \end{bmatrix}$ |



(a) (b)

Fig. 4. ***(a)*** *Positions of the feature points that correspond to the 300 largest values of $S_{xy}^{(t)}$ with model size $N_H = N_W = 7$,* ***(b)*** *positions of the feature points that correspond to the 300 largest values of $S_{xy}^{(t)}$ while being at least 7 pixels apart (in both horizontal and vertical directions). Only 25% of eigenvalues and eigenvectors are used during the deformation procedure.*

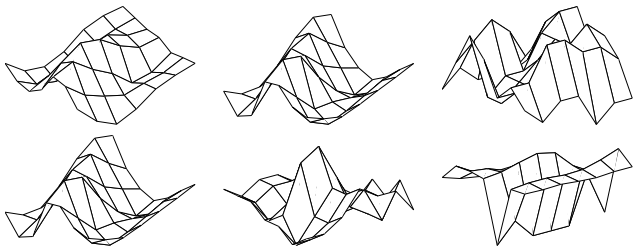
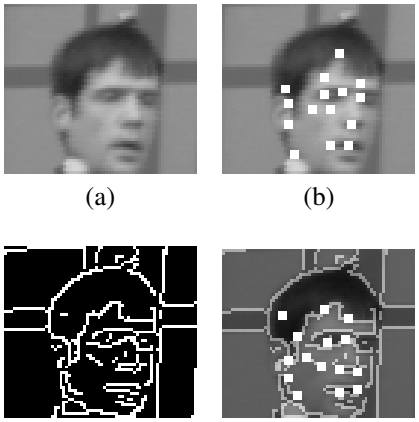


Fig. 5. *Local intensity surfaces of the original image for six of the selected feature points depicted in Figure 4 (model size $N_H = N_W = 7$, total number of selected feature points 300).*

lines and corners. If the selected feature points are concentrated on 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 $\underline{\mathbf{p}}_i^{(t)}$, $i \in \{1, 2, \ldots, M\}$ that are finally selected, are the ones that have maximum $S_{xy}^{(t)}$ but, at the same time, maintain a certain Euclidean distance $\mathbb{D} = \|\underline{\mathbf{p}}_i^{(t)} - \underline{\mathbf{p}}_j^{(t)}\| > D_{thres}$ from each other (Figure 4b). More specifically, we order the feature points with respect to $S_{xy}^{(t)}$ and select the one with the biggest $S_{xy}^{(t)}$ value. Subsequently, we choose as the second feature point the one with the largest value of $S_{xy}^{(t)}$ whose distance from the first is at least $D_{thres}$.



(a) (b) (c) (d)

Fig. 6. ***(a)*** *The initial image,* ***(b)*** *15 feature points on the initial image,* ***(c)*** *the binary output of Canny edge detector,* ***(d)*** *15 feature points selected on the output of the Canny edge detector.*

The procedure continues until the desired number of points is reached. The number $M$ of feature points that will be selected is a user-defined parameter of the algorithm. Figure 5 depicts the local intensity surfaces of the original image for some of the selected feature points shown in Figure 4b. The intensities values being along the vertical direction. It can be seen that the image intensity within the $7 \times 7$ neighborhood of these feature points has large variations. Thus, these points are expected to be suitable for tracking, as will be shown in the experiments.

The procedure of selecting feature points in an image is rather time consuming (although not considerably so), since, for each image pixel a deformable surface model of size $N_H \times N_W$ must be used. The computational complexity for each pixel is of the order $O(N^4)$ for a deformable surface model of size $N \times N$. This complexity figure can be derived from equation (12), since for each of the $N^2$ nodes of the deformable surface model, $2N^2 + 1$ additions and $4N^2 + 5$ multiplications are needed. To avoid applying the deformation procedure on each image pixel, in situations that require low computational complexity, one can incorporate in the proposed approach the well known Canny edge detector [37] as follows: the Canny edge detector is applied on the image and the selection of the feature point set (Figure 6b) is performed only among the image pixels where the corresponding output

of the Canny edge detector is sufficiently high (Figure 6c). This procedure offers a faster but suboptimal feature point selection. None of the experiments reported in this paper used this suboptimal version of the selection procedure, since complexity was of secondary concern.

## IV. Feature Point Tracking

The $2D$ feature point tracking problem is equivalent to finding the location of the feature points in successive frames of an image sequence. Given such an image sequence $\mathbf{I} = I_1, I_2, \ldots, I_T$ and a feature point $\underline{\mathbf{p}}_i^{(t)} = (x_i, y_i)$, $t \in \{1, 2, \ldots, T\}$ in the $t$-th image frame, the tracking problem can be formulated into finding a displacement vector $\underline{\mathbf{d}}_i^{(t)} = \left(d_i^{(t)}(x), d_i^{(t)}(y)\right)$, where $\left(d_i^{(t)}(x), d_i^{(t)}(y)\right)$ are the translation components of point $\underline{\mathbf{p}}_i^{(t)}$ along each axis respectively, in order to locate its position $\underline{\mathbf{p}}_i^{(t+1)} = (x', y')$ in the next image frame:

$$\underline{\mathbf{p}}_i^{(t+1)} = \underline{\mathbf{p}}_i^{(t)} + \underline{\mathbf{d}}_i^{(t)}. \tag{20}$$

To achieve tracking, the proposed approach computes for each feature point $\underline{\mathbf{p}}_i^{(t)} = (x_i, y_i)$ of the selected feature point set $\mathbf{p}^{(t)} = [\underline{\mathbf{p}}_1^{(t)}, \underline{\mathbf{p}}_2^{(t)}, \ldots, \underline{\mathbf{p}}_M^{(t)}]^T$ in image frame $I_t$ the CFV $\tilde{\mathbf{u}}^{(t)}(x, y)$ over a window $N_H \times N_W$ and subsequently calculates $S^{(t)}(x, y)$:

$$S_{x,y}^{(t)} = \sum_{(k,l)\neq(1,1)}^{N_H} \sum^{N_W} \left| \tilde{u}_{k,l}^{(t)}(x, y) \right|. \tag{21}$$

In order to find the position $\underline{\mathbf{p}}_i^{(t+1)} = (x_i', y_i')$ of the feature point $i$ in the next image frame $I_{t+1}$, the algorithm computes the CFV $\tilde{\mathbf{u}}^{(t+1)}(k, l)$ for each pixel of a $N_{S_H} \times N_{S_W}$ ($N_{S_H}$, $N_{S_W}$ being odd numbers) search region $R$, centered at coordinates $(x, y)$ in image $I_{t+1}$. The new location of feature point $i$ is given by:

$$\underline{\mathbf{p}}_i^{(t+1)} = (x_i', y_i') \longrightarrow \arg\min_{kl}(|S_{xy}^{(t)} - S_{kl}^{(t+1)}|), \tag{22}$$

where $k \in \{x - \frac{N_{S_H}-1}{2}, \ldots, x + \frac{N_{S_H}-1}{2}\}$ and $l \in \{y - \frac{N_{S_W}-1}{2}, \ldots, y + \frac{N_{S_W}-1}{2}\}$.

The use of the absolute difference (22) of $S_{x,y}^{(t)}$ (21) for judging the similarity/matching of features, during the tracking procedure, instead of other possible measures, such as the correlation between the CFVs, was decided after the experimental comparison of such metrics.

In order to identify feature points that lose their target and to cease tracking them, a mutual information based measure, expressing the goodness of tracking for feature points belonging to successive frames $I_t$ and $I_{t+1}$, was used. Let $C^{(t)}$, $C^{(t+1)}$ be two random variables with $P(c_i^{(t)}), P(c_i^{(t+1)})$ and $P(c_i^{(t)}, c_i^{(t+1)})$ their marginal and joint probability density functions. In our case, $c_i^{(t)} = I_t(\underline{\mathbf{p}}_i^{(t)})$, $c_i^{(t+1)} = I_{t+1}(\underline{\mathbf{p}}_i^{(t+1)})$ and $\underline{\mathbf{p}}_i^{(t)} \in \mathbf{p}^{(t)}$, $\underline{\mathbf{p}}_i^{(t+1)} \in \mathbf{p}^{(t+1)}$. In other words, the two random variables are the intensities of the same feature point on two successive frames. Thus, the marginal probability $P(c_i^{(t)})$ is estimated from the image histogram $H_{ist}(c_i^{(t)})$, i.e., $P(c_i^{(t)}) = \frac{H_{ist}(c_i^{(t)})}{N_h N_w}$. The joint probability is estimated by the $2D$ joint histogram of two frames, i.e., $P(c_i^{(t)}, c_j^{(t+1)}) = \frac{H_{ist}(c_i^{(t)}, c_j^{(t+1)})}{N_h N_w}$, where $H_{ist}(c_i^{(t)}, c_j^{(t+1)})$ is the number of corresponding pixels (pixels in the same spatial position) that have intensity $c_i^{(t)}$ in frame $I_t$ and $c_j^{(t+1)}$ in frame $I_{t+1}$. The mutual information of $C^{(t)}$, $C^{(t+1)}$ is defined as:

$$L(C^{(t)}, C^{(t+1)}) = \sum_{i=1}^{N_{max}} \sum_{j=1}^{N_{max}} P(c_i^{(t)}, c_j^{(t+1)}) \cdot \log_2 \frac{P(c_i^{(t)}, c_j^{(t+1)})}{P(c_i^{(t)}) P(c_j^{(t+1)})}, \tag{23}$$

where $N_{max}$ is the maximum number of the available image grayscales. In order to distinguish the lost feature points during the tracking process the following measure is defined:

$$E_m^{(t)} = \left| L(C^{t-1}, C^{(t)}) - L(C^{(t)}, C^{(t+1)}) \right|. \tag{24}$$

When $E_m^{(t)} \geq Th$ ($Th$ being a predefined threshold), the corresponding tracked feature point is labelled as lost and the algorithm stops tracking it. $Th$ was set equal to 1.6, because it was experimentally proven that this threshold can efficiently separate the well tracked feature points from the ones that lose the target. The experimental threshold evaluation was done by running the tracking algorithm for a number of feature points on various video sequences and evaluating the $E_m^{(t)}$ value for the frames where visual inspection of the results showed that a feature point has lost its target. The threshold $Th$ was set to be equal to the minimum of these values, since it was found that this value was considerably larger than the $E_m^{(t)}$ values corresponding to properly tracked feature points.

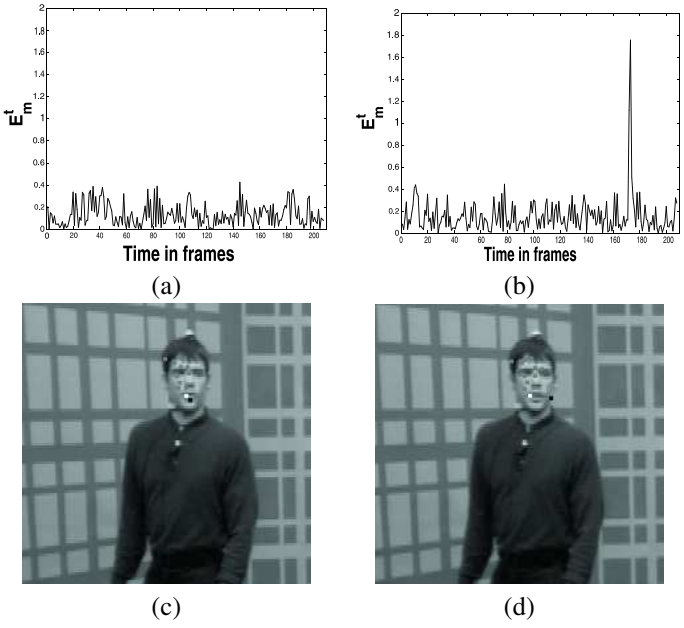



Fig. 7. ***(a)*** *$E_m^{(t)}$ versus time (in frames) for a tracked feature point (white dot).* ***(b)*** *$E_m^{(t)}$ versus time (in frames) for a feature point that ceases being properly tracking (black dot).* ***(c)*** *Tracking results for frame* 173. ***(d)*** *Tracking results for frame* 174*, where the black feature point loses the target and, therefore, is considered lost.*

An example of the previously described lost feature point detection procedure is shown in Figure 7. Fifteen feature points were selected in the initial frame of the video sequence and were tracked in the entire sequence. The time series of $E_m^{(t)}$ corresponding to two feature points is plotted in Figures 7a, 7b. Both feature points were successfully tracked till the frame 173 and $E_m^{(t)} \leq Th$ for $t \leq 173$. At frame 174 one feature point (black dot) lost its target (Figure 7d) and $E_m^{(t)}$ identified it as lost ($E_m^{(t)} \geq Th$).

Since it is difficult to know the proper size for the search window $R$ in (22) a priori, we propose using an adaptive window $R_t$, defined as the *best region* for estimating the new position of a feature point in the next video frame. Starting with a small window $R_t$ (e.g. of $9 \times 9$ pixels), the algorithm iteratively increases the window size up to a certain limit, until the normalized correlation between the CFV of the feature point in the previous frame and the CFV of the best match in the current frame, surpasses a threshold $E_o$. For evaluating a suitable value for $E_o$, normalized correlation was computed for different sizes of the deformable surface model for different video sequences and objects of interest. These experiments have demonstrated that one can achieve a good matching within the search area by setting the threshold $E_o$ equal to $0.7$. Similar methods have been used to find the best feature points correspondences in real and synthetic images in [38].

In time-critical applications, an exhaustive search in the region of the adaptive window $R_{t+1}$ at the next image frame can be avoided by employing the same technique which is used in the feature point selection procedure (Section III) and exploits the Canny edge detector output. Before determining the displacement $\underline{\mathbf{d}}_i^{(t)}$ of a feature point $\underline{\mathbf{p}}_i^{(t)}$ from frame $I_t$ to frame $I_{t+1}$, one can apply the Canny edge detector to the search region $R_{t+1}$ and search exhaustively only the subset of $R_{t+1}$ comprising of those pixels where the Canny edge detector output is above a certain threshold. The threshold is set so as the Canny edge detector output contains all significant edges of the region $R_{t+1}$ at image $I_{t+1}$. However, it should be noted, that this procedure results in inferior performance compared to that provided by the exhaustive search. This is the reason for using the exhaustive search in all the experiments reported in this paper.

## V. Experimental Results

To evaluate the proposed tracking method, which will be subsequently called *Modal Features* (MF) method, we applied it for face tracking on a set of test video sequences [39]. The material consists of 100 GB of raw video data in full PAL resolution (25 fps, $4:2:2$, $720 \times 576$, 24 bpp). The sequences contain studio scenes with one, two or more persons moving on a predefined or random trajectory, under either optimal (as defined by the studio technicians) or suboptimal lighting conditions created using studio lighting equipment, thereby introducing hard shadows and bright/dark areas in the recorded video sequences. Examples are shown in Figure 8. The proposed approach was also tested on other outdoors and indoors video sequences depicting persons performing different motions under various illumination conditions. In all the experiments, the stiffness $k$ and the mass $m$ of the nodes of the deformable surface model, were set equal to 1. It should be noted here that this paper does not aim at introducing a full-fledged tracking system with occlusion handling, lost feature points regeneration etc., but only at proposing a novel and efficient method for selecting and tracking feature points on video sequences, which can be integrated in any complex feature tracking system. Alternatively, occlusion/disocclusion handling mechanisms can be introduced in the proposed methodology to make it capable of coping with such situations. Thus, the performance of the MF method has been examined mainly in terms of tracking accuracy and robustness under different motions and lighting conditions, without taking into account occlusion cases since such cases are outside of the scope of the proposed methodology. The acquired results were compared with the ones produced by the well known feature-based Kanade-Lucas-Tomasi (KLT) tracking algorithm [27], [20] and a SIFT feature-based tracking system [28]. The selected feature points were also compared with feature points produced by SIFT [18] or Harris [19] operators. It should be noted that a small number of feature points has been used in all the experiments due to the small size of the tracked objects in certain cases, the painstaking procedure required in order to derive ground truth information for a large number of tracked points and the fact that a small number of feature points facilitates the visual inspection of the results.

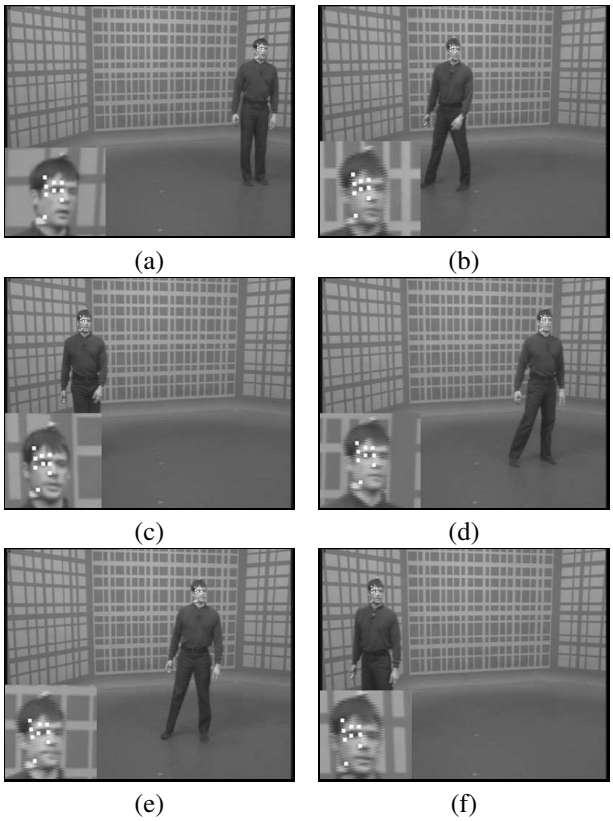



Fig. 8. *Tracking results of the MF algorithm for a* 600 *frame segment of an indoor video sequence. The shown sample frames are taken at* 120 *frame intervals.*

The first set of experiments dealt with the evaluation of the

accuracy of the MF algorithm when tracking individual feature points. The feature point selection method described in Section III was used to select a number of feature points ($M = 9$) on a specific image region, (i.e. a human face) which was manually outlined on the first frame of the video sequence under examination. The tracking procedure described in Section IV was then applied to all (600) video frames. Examples of the output are presented in Figure 8. The KLT algorithm was also applied on the same sequence in a similar manner. The KLT algorithm selected 9 feature points in the same manually selected area of the first frame using its own feature point selection algorithm described in [20] and tracked them over the rest of the video frames. The size of the deformable intensity surface model was set to be the same with the window $R$ used by the KLT algorithm around each pixel for feature point selection and tracking, i.e., equal to $7 \times 7$ pixels. Image intensity normalized correlation between corresponding feature point regions (image regions around each feature point) were computed between the initial frame and the current one and the average normalized correlation over the entire video sequence for all selected feature points was calculated. The average normalized correlation can provide clues about the tracking performance of the algorithm, since large values indicate good tracking. The results are shown in Table II. One can see that average normalized correlation over the entire video sequence for the proposed approach is much higher than the one achieved using KLT. Furthermore, the normalized correlation variance over the entire video sequence is much smaller for the MF approach than for KLT. This indicates that the proposed method has less fluctuations in tracking performance than KLT. Similar results were obtained for other video sequences that were used in the experiments.

TABLE II

*Average normalized correlation (NCA) in windows around selected feature points and normalized correlation variance (NCV) for the MF algorithm and the KLT algorithm.*

| Feature points | Point 1 | Point 3 | Point 6 | Point 9 | All Points |
|---|---|---|---|---|---|
| NCA (MF) | **0.8661** | **0.8546** | **0.8881** | **0.7900** | **0.8360** |
| NCA (KLT) | 0.5961 | 0.6428 | 0.7860 | 0.6946 | 0.6330 |
| NCV (MF) | **0.0070** | **0.0074** | **0.0050** | **0.0157** | **0.0123** |
| NCV (KLT) | 0.0380 | 0.0171 | 0.0211 | 0.1360 | 0.0621 |

The second set of experiments aimed at testing the ability of the proposed algorithm to correctly track feature points over a large number of frames. The MF and the KLT algorithms were applied on a number of video sequences. It was experimentally proven that the KLT algorithm ceases tracking feature points over time more frequently than the proposed method. This is illustrated in Table III that provides figures for the average tracking life (measured in number of frames) of feature points for both algorithms for different sizes of the window $R$, which is the model size for the MF algorithm and the window around each feature point used in KLT. Lost feature points were detected by visual inspection, i.e., by inspecting the tracking results in order to find where the tracked feature points deviated to a great extent from the target feature points. The superiority of the MF tracking method can also be verified by the plot in Figure 9, which depicts the number of tracked feature points (selected in the same region of the initial frame of the same image sequence) versus time (in frames) for both algorithms for a certain video sequence depicting a person moving towards the camera in a zig-zag fashion. The window size used was equal to $7 \times 7$ pixels. It is clearly seen that the MF method loses less feature points as tracking proceeds and continues tracking even when KLT breaks down completely at frame number 635, due to head rotation of the target person. Similar results were obtained in other video sequences.

TABLE III

*The average life (in frames) of feature point tracking for different model/window sizes.*

| Model/window Size | MF tracker | KLT tracker |
|---|---|---|
| 3×3 | **466.00** | 115.27 |
| 5×5 | **618.27** | 349.13 |
| 7×7 | **730.33** | 479.33 |

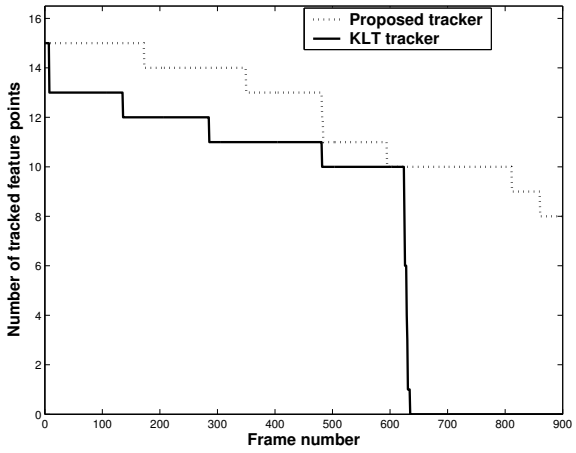



Fig. 9. *Number of successfully (not lost) tracked feature points at each frame of a video sequence for both algorithms.*

Furthermore, in order to evaluate tracking precision, we have manually produced tracking 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 both the KLT algorithm and the MF algorithm to select 9 feature points on the facial image region in the first frame of each sequence, each using its own feature selection procedure. Afterwards, both algorithms were allowed to track the selected feature points for the rest of the video sequences. The window size was set equal to $7 \times 7$ pixels. The feature point positions generated by the two algorithms were compared with the ground truth data, i.e. the manually tracked positions of the same feature points. The Euclidean distance between the ground truth positions and the positions provided by the two algorithms (i.e. the tracking position error) was used for the comparison. As can be seen in Figure 10 (for a sequence depicting a person moving parallel to the camera), our algorithm is more precise in tracking, i.e., the tracking error is constantly smaller for the MF algorithm. More specifically, the MF error is almost three times smaller than the one produced by KLT. Table IV presents the average

tracking position error and the variance of the position error over the entire video sequence for some feature points which were selected and tracked by the two algorithms. The position error variance is much smaller for the MF tracker than for the KLT one. These results show that the MF approach performs more accurate tracking than KLT.

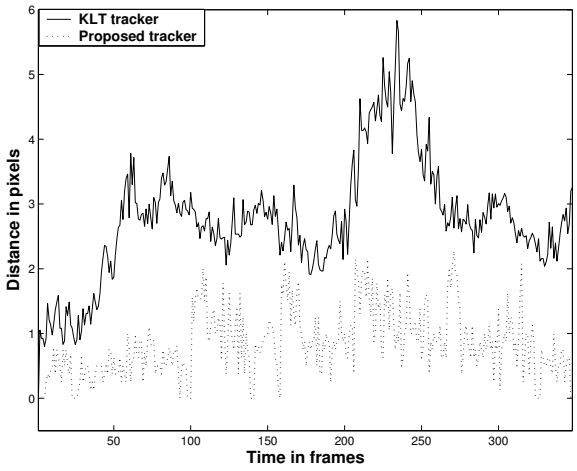



Fig. 10. *Euclidean distance (averaged over all feature points) between ground truth positions and the positions provided by the KLT and the MF algorithm.*

TABLE IV

*Mean (MPE) and Variance (VPE) of the Euclidean distance (error) in pixels between feature points tracked by the two algorithms and ground truth data. Each algorithm is initialized with its own feature points.*

| Feature points | Point 1 | Point 3 | Point 6 | Point 9 | All Points |
|---|---|---|---|---|---|
| MPE, (MF) | **0.6063** | **0.7564** | **0.9831** | **0.7445** | **0.8232** |
| MPE, (KLT) | 2.3390 | 2.4549 | 8.0851 | 2.0794 | 2.7638 |
| VPE, (MF) | **0.5433** | **0.4880** | **0.4442** | **0.6202** | **1.3731** |
| VPE, (KLT) | 1.1039 | 1.1748 | 0.6211 | 1.0728 | 2.1059 |

We have also repeated the same experiment but this time the feature point selection was based on the KLT algorithm. The aim of this experiment was to illustrate that the proposed tracking algorithm can offer satisfying results even if the feature point set has not been chosen by using the method introduced in Section III, i.e. that the success of the proposed tracking procedure is not only due to the feature point selection approach but also due to the feature vector used for tracking. In this experiment, both algorithms were initialized with the feature points selected by the KLT algorithm and were left to track these points over the entire video. Results were compared (using Euclidean distance) with the manually tracked ground truth data. Table V shows that the proposed tracking algorithm is not as precise as in the previous case but it is still more precise than the KLT algorithm.

In the next set of experiments, the proposed feature selection algorithm presented in Section III was compared to the well known SIFT feature points [18] which are appropriate for matching between different views of an object and to the feature points derived by the Harris feature point detector [19]. The proposed feature point selection algorithm, the SIFT

TABLE V

*Mean Euclidean distance (error) in pixels between feature points tracked by the two algorithms and ground truth data. Data in rows 2, 3 were derived by initializing both algorithms with the feature points selected by the KLT algorithm.*

| Feature points | Point 1 | Point 3 | Point 6 | Point 9 | Average |
|---|---|---|---|---|---|
| MF Tracker | **0.6063** | **0.7564** | **0.9831** | **0.7445** | **0.8232** |
| KLT - MF Tracker | 2.9111 | 0.9741 | 7.2948 | 1.5715 | 1.3731 |
| KLT Tracker | 2.3390 | 2.4549 | 8.0851 | 2.0794 | 2.7638 |

algorithm and the Harris detector were applied to the same manually selected area on the first image of a number of video sequences and feature points for each algorithm were selected. The selected feature points were tracked to the rest of the sequence with the proposed tracking algorithm. Normalized correlation between corresponding feature point regions (image regions around each feature point) was computed between the initial frame and the current one and the average normalized correlation over the entire video sequence for all selected feature points was calculated. The results for one of these sequences consisting of $600$ frames, are shown in Table VI and prove that the combination of the proposed feature selection scheme and tracking algorithm offers the best performance. In all the experiments of this set, the number of the feature points was defined by the output of the SIFT and Harris detectors.

TABLE VI

*Average normalized correlation (NCA) in windows around tracked feature points and normalized correlation variance (NCV) for the acquired results when SIFT, Harris and the proposed feature selection scheme were used as the initialization for the proposed tracking procedure (three feature points).*

| Feature points | Point 1 | Point 2 | Point 3 | All Points |
|---|---|---|---|---|
| NCA (MF) | 0.8794 | **0.9047** | **0.8989** | **0.8943** |
| NCA (SIFT) | **0.8994** | 0.7215 | 0.8180 | 0.8112 |
| NCA (Harris) | 0.7852 | 0.7845 | 0.5610 | 0.7120 |
| NCV (MF ) | **0.0055** | **0.0077** | 0.0102 | **0.0078** |
| NCV (SIFT ) | 0.0135 | 0.0271 | **0.0101** | 0.0169 |
| NCV (Harris) | 0.0162 | 0.0169 | 0.0257 | 0.0196 |

In another experiment, MF tracking performance was tested on a planar, rigid, textured object (a book cover) in order to illustrate the fact that the MF tracker can achieve very good results when tracking salient points. The tracked object remained fixed whereas the camera moved in a pattern that included in-plane translations, zoom-ins, zoom-outs and in- and out-of-plane rotations. Ground truth data, i.e. the manually tracked positions of the five feature points selected in the first frame by the proposed feature selection algorithm, was obtained for the video sequence. The Euclidean distance between the ground truth positions and the positions provided by the MF tracking (averaged over all tracked points) was used as performance criterion. As can be seen in Figure 11, MF algorithm obtains very good and highly stable tracking results, since the maximum average distance from the ground truth is $1.5$ pixels whereas the mean average distance is only $0.5$ pixels. Two frames of the video sequence are depicted in Figure 12.

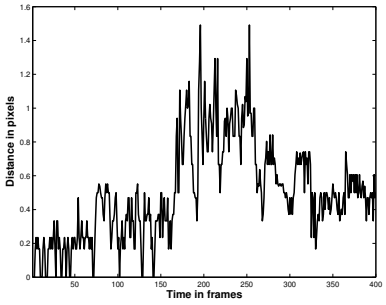



Fig. 11. *Euclidean distance (averaged over all five tracked points) between ground truth positions and the positions provided by the MF algorithm, , for the image sequence depicted in Figure 12.*

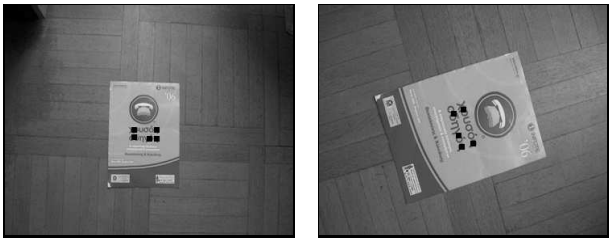


Fig. 12. *Two frames of a video sequence consisting of* $400$ *frames. The tracked object is a planar textured object (book cover) where the feature points (black points) are clearly visible.*

The following set of experiments aimed at providing results for the performance of the MF tracking algorithm when applied on motions frequently encountered in tracking situations, such as scaling and rotation of the object under examination, i.e. the human face. In this case, the performance of the algorithm was evaluated at the object level, i.e., on the basis of whether the entire object under examination was correctly tracked or not. True positives ($TP$), false positives ($FP$), false negatives ($FN$) were obtained using manually extracted ground truth data. The results where the bounding box of the tracked feature points contained only the object under examination, were considered as $TP$. When the bounding box included some background area, it was considered as contributing to $FP$. The situations where the tracking algorithm stops tracking the face (i.e. it loses the target), were considered contributing to $FN$. Based on these numbers, the well-known precision ($P = \frac{TP}{TP+FP}$) and recall measures ($R = \frac{TP}{TP+FN}$) were calculated for the tracking procedure. The results are summarized in Table VII for five sequences with different motion characteristics. In all sequences the tracked object was the head of the depicted person. One can see that the MF tracking algorithm is robust to such movements. Some tracking results of the MF algorithm are shown in Figures 13-17. In Figure 13, the subject tilts its head in various orientations. In Figure 14, the subject rotates its head up to $90$ degrees (out-of-plane rotation) and thus the characteristics of the facial features change dramatically. Only one feature point was selected on the nose of the subject, since nose is the only part of the face that remains visible in all frames. The MF tracker continues tracking the selected feature point throughout the sequence.

TABLE VII

*PRECISION AND RECALL OF THE TRACKING ALGORITHM FOR VIDEO SEQUENCES WITH DIFFERENT MOTIONS.*

| Motion | Frames | P | R |
|---|---|---|---|
| Free | 961 | 98.9 | 90.0 |
| Translation | 780 | 100 | 100 |
| Zoom | 639 | 100 | 100 |
| Tilt and pan | 575 | 100 | 90.6 |
| Roll | 341 | 79.3 | 100 |

Moreover, one can see in Figure 15 a subject rotating its head up to $45$ degrees, so as all the selected feature points remain visible. The MF tracking results are very good, since all the feature points are tracked properly. In Figure 16, the subject walks towards and away from the camera. As a result, the area of the face is dramatically enlarged/minimized and at the far end of the movement the facial features are almost invisible. However, the MF tracker tracks the selected feature points fairly well, especially when judging results at the object level. Finally, Figure 17 shows that the proposed algorithm is robust to illumination changes and shadows, such as those caused on the face by a person entering/leaving a brightly lit area of a room.


Fig. 13. *Tracking results of the MF algorithm for a* $400$ *frame segment of an indoor video sequence. The shown sample frames are taken at* $10$ *frame intervals. The subject tilts its head.*

TABLE VIII

*The average life (in frames) for the two variants of SIFT feature point tracking (SIFT-I and SIFT-II) and MF feature point tracking for various video sequences.*

| | **Average life in frames** | | |
|---|---|---|---|
| Number of frames | MF tracker | SIFT-I tracker | SIFT-II tracker |
| 550 | **550.00** | 46.04 | 290.27 |
| 500 | **412.60** | 62.00 | 303.90 |
| 650 | **543.40** | 78.09 | 320.12 |

In the final set of experiments, a recent SIFT feature based tracking algorithm [28], [18] was tested against the proposed

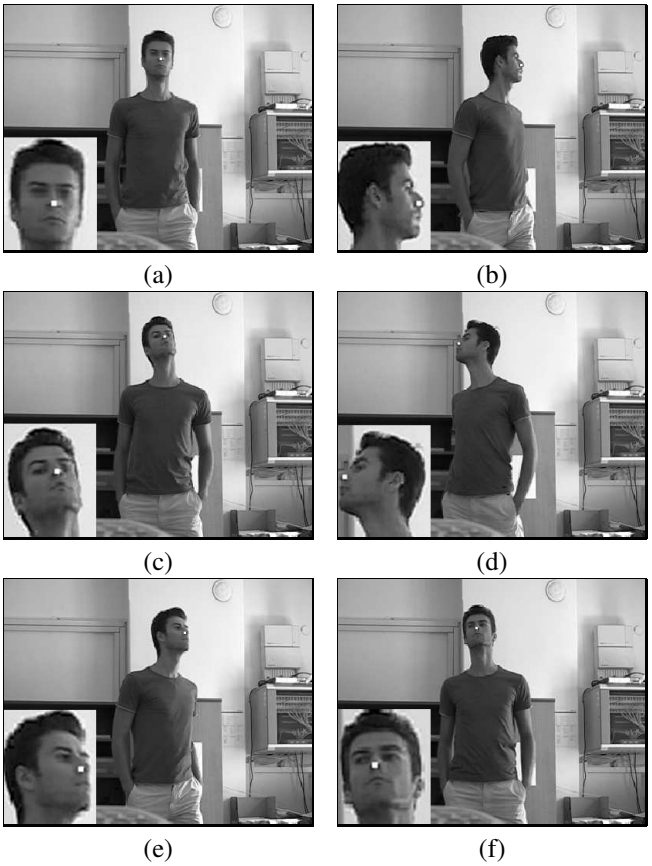


Fig. 14. *Tracking results of the MF algorithm for a* $330$ *frame segment of an indoor video sequence under varying lighting conditions. One feature point was selected on the nose, because the nose is the only part of the face which is constantly in the field of view during the time sequence. The shown sample frames are taken at* $60$ *frame intervals.*

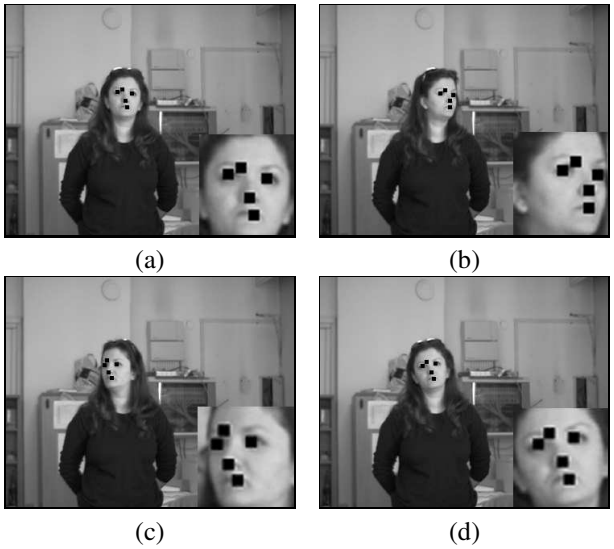


Fig. 15. *Tracking results of the MF algorithm for a* $300$ *frame segment of an indoor video sequence. The shown sample frames are taken at* $100$ *frame intervals. The subject rotates its head up to* $45$ *degrees.*

selection and tracking approach. In their paper, Gordon and Lowe [28] presented a complete system architecture for aug-

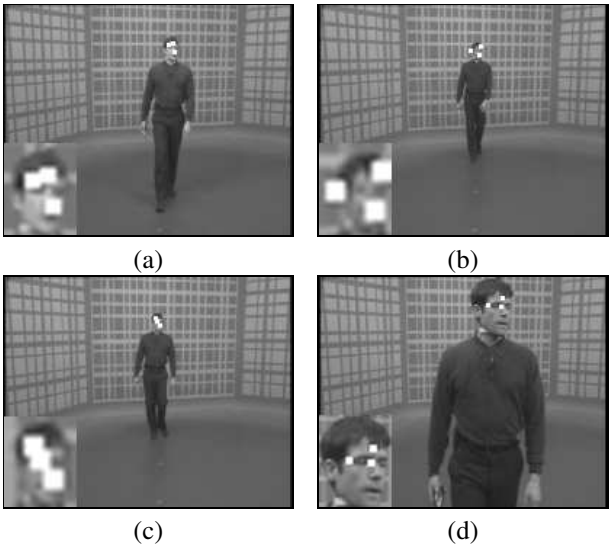


Fig. 16. *Tracking results of the MF algorithm for a* $450$ *frame segment of an indoor video sequence. The shown sample frames are taken at* $150$ *frame intervals. The subject moves towards and backwards to the camera in* $5$ *meter distance.*

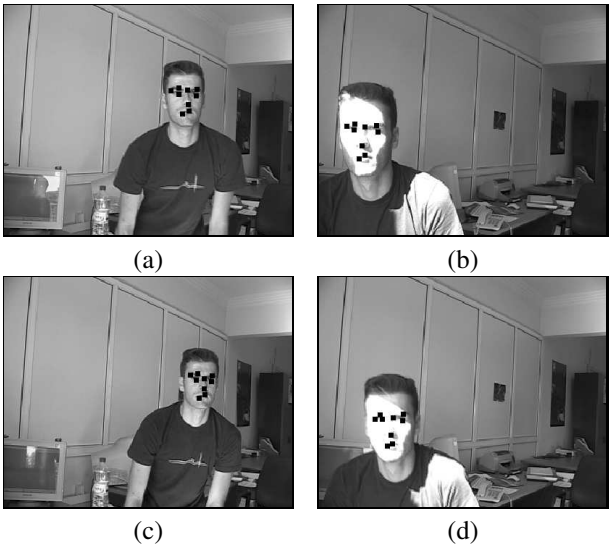


Fig. 17. *Tracking results of the MF algorithm for a* $450$ *frame segment of an indoor video sequence under varying lighting conditions. The existence of hard shadows on the face does not affect the tracking performance. The shown sample frames are taken at* $150$ *frame intervals.*

mented reality. The feature selection and tracking procedures of this system was implemented for comparison. Two different approaches were utilized. In the first approach (denoted as SIFT-I), which is directly comparable to the proposed method, SIFT feature points are found in the first and second frame and matched using the corresponding 128-dimensional feature vector through a search involving Euclidean distance. Then, those points on the second frame that have been matched in the previous step (involving the $1^{st}$ and $2^{nd}$ frame) and reside within the face, are matched with SIFT feature points detected on the third frame and so on. The procedure continues until all frames are examined. This procedure produced poor

results in terms of feature points lifetime. In the second approach (denoted as SIFT-II), SIFT points were detected in each pair of consecutive frames ($i$ and $i+1$, $i+1$ and $i+2$,) and the corresponding 128-dimensional feature vectors were used in order to find matches between them. This approach produced better results then the previous one, but is not directly comparable to the proposed tracking algorithm since it is not a feature tracking method (i.e. it does not track a feature point from the first frame onwards) but it just matches features from one frame to the next. Both approaches produced inferior results when compared with the proposed method. Results (average life in frames) for the two SIFT approaches and the MF algorithm are given in Table VIII.

## VI. Conclusion

A novel $2D$ feature point selection and tracking algorithm based on the use of a parameterized $3D$ physics-based deformable model was proposed in this paper. In this approach, the intensity surface of the image is represented by a $3D$ physics-based deformable model. We have shown how to tailor the model deformation equations to efficiently select and track feature points in a video sequence. It has been shown that these equations are a combination of the output of various line and edge detection masks. The presented tracking method was compared with the well known KLT algorithm and a SIFT feature-based tracking algorithm. The results show that the proposed method produces superior tracking results, it provides better tracking accuracy and tracks feature points for longer period of time than KLT and SIFT tracker. Moreover, the feature point selection mechanism was tested against SIFT and Harris feature points and it was shown to have better performance.

## References


[1] 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*, A. C. Bovik, Ed. Academic Press, 2005.

[2] 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] T. B. Moeslund, A. Hilton, and V. Krüger, "A survey of advances in vision-based human motion capture and analysis," *Computer Vision and Image Understanding*, vol. 104, no. 2-3, pp. 90–127, 2006.

[4] D. M. Gavrila, "The visual analysis of human movement: A survey," *Computer Vision and Image Understanding*, vol. 73, no. 1, pp. 82–98, 1999.

[5] J. K. Aggarwal and Q. Cai, "Human motion analysis: A review," *Computer Vision and Image Understanding*, vol. 73, no. 3, pp. 428–440, 1999.

[6] M.Vezhnevets, "Face and facial feature tracking for natural human-computer interface," *Proceedings in Graphicon*, pp. 86–90, September 2002.

[7] H. Sidenbladh and M. Black, "Learning the statistics of people in images and video," *International Journal of Computer Vision*, 2002.

[8] L. Tsap, D. Goldgof, and S. Sarkar, "Fusion of physically-based registration and deformation modeling for nonrigid motion analysis," *IEEE Transactions on Image Processing*, vol. 10, no. 11, pp. 1659–1669, November 2001.

[9] Y. Wang and S. Zhu, "Analysis and synthesis of textured motion: Particles and waves," *IEEE Transactions on Patern Analysis and Machine Intelligence*, vol. 26, no. 10, pp. 1348–1363, October 2004.

[10] M. Kass, A. Witkin, and D. Terzopoulos, "Snakes: Active contour models," *International Journal of Computer Vision*, vol. 1, no. 4, pp. 321–331, 1987.

[11] H. Chao, Y. F. Zheng, and S. C. Ahalt, "Object tracking using the gabor wavelet transform and the golden section algorithm," *IEEE Transactions on Multimedia*, vol. 4, no. 4, pp. 528–538, December 2002.

[12] J. C. Nascimento and J. S. Marques, "Robust shape tracking in the presence of cluttered background," *IEEE Transactions on Multimedia*, vol. 6, no. 6, pp. 852–861, December 2004.

[13] X. Tao and C. Debrunner, "Stochastic car tracking with line- and color-based features," *IEEE Transactions on Intelligent Transportation Systems*, vol. 5, no. 4, pp. 324–328, December 2004.

[14] J. Verestoy and D. Chetverikov, "Comprative performance evaluation of four feature point tracking techniques," in *Proceedings of 22nd Workshop of the Austrian Pattern Recognition Group*, Illmitz, Austria, 1998, pp. 255 – 263.

[15] T. Tomassini, A. Fusiello, M. Trucco, and V. Roberto, "Making good features track better," in *Proceedings of International Conference on Computer Vision and Pattern Recognition*, Santa Barbara, 1998, pp. 178 – 180.

[16] J. Bouguet, "Pyramidal implementation of the lucas kanade feature tracker," Intel Corporation, Microprocessor Research Labs, OpenCV Documents, Tech. Rep., 1999.

[17] J. Wieghardt, R. P. Wurtz, and C. Malsburg, "Gabor-based feature point tracking with automatically learned constraints," in *Dynamic Perception*, September 2002, pp. 121–126.

[18] D. Lowe, "Distinctive image features from scale-invariant keypoints," *International Journal of Computer Vision*, vol. 60, no. 2, pp. 91–110, November 2004.

[19] C. Harris and M. Stephens, "A combined corner and edge detector," in *Proceedings of the Fourth Alvey Vision Conference*, Manchester, March 1998, pp. 147–151.

[20] J. Shi and C. Tomasi, "Good features to track," in *IEEE International Conference on Computer Vision and Pattern Recognition (CVPR94)*, Seattle, United States, June 1994, pp. 593–600.

[21] R. T. Collins, L. Yanxi, and M. Leordeanu, "Online selection of discriminative tracking features," *IEEE Transactions on Patern Analysis and Machine Intelligence*, vol. 27, no. 10, pp. 1631–1643, October 2005.

[22] 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.

[23] C. Nikou, G. Bueno, F. Heitz, and J. Armspach, "A joint physics-based statistical deformable model for multimodal brain image analysis," *IEEE Transactions on Medical Imaging*, vol. 20, no. 10, pp. 1026–1037, 2001.

[24] 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.

[25] C. Nastar and N. Ayache, "Fast segmentation, tracking, and analysis of deformable objects," in *Proceedings of the Fourth International Conference on Computer Vision (ICCV'93)*, Berlin, Germany, May 1993, pp. 11–14.

[26] A. Pentland and S. Sclaroff, "Closed-form solutions for physically-based shape modeling and recognition," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 13, no. 7, pp. 715–729, July 1991.

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

[28] I. Gordon and D. G. Lowe, "Scene modelling, recognition and tracking with invariant image features," in *Third IEEE and ACM International Symposium on Mixed and Augmented Reality (ISMAR 2004)*, November 2004, pp. 110–119.

[29] 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.

[30] G. Borgefors, "On digital distance transforms in three dimensions," *Computer Vision and Image Understanding*, vol. 64, no. 3, pp. 368–376, 1996.

[31] P.-E. Danielsson, "Euclidean distance transform," *Computer Graphics and Image Processing*, vol. 14, pp. 227–28, 1980.

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

[33] C. Nastar, "Models physiques deformables et modes vibratoires pour l'analyse du mouvement non-rigide dans les images multidimensionnelles," Ph.D. Thesis, Ecole Nationale des Ponts et Chaussees, 1994.

[34] 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.
[35] R. Gonzalez and R. Woods, *Digital Image Processing*. Addison-Wesley Publishing Company, 1992.
[36] W. Frei and C. Chen, "Fast boundary detection: A generalization and a new algorithm," *IEEE Transactions on Computers*, vol. C-26, no. 10, pp. 988–998, 1977.
[37] J. Canny, "A computational approach to edge detection," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 8, pp. 679 – 698, 1986.
[38] T. Kanade, "A stereo matching algorithm with an adaptive window: Theory and experiment," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 16, no. 9, Sep 1994.
[39] 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 *Proceedings of IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP 2005)*, Philadelphia, March 2005.

**Michail Krinidis** received the B.S. degree from the Department of Informatics, Aristotle University of Thessaloniki, Thessaloniki, Greece, in 2002. He is currently pursuing a Ph.D. degree in the same department while also serving as a teaching assistant. His current research interests lie in the areas of 2D tracking, face detection and 3D head pose estimation in image sequences.

**Nikos Nikolaidis** received the Diploma of Electrical Engineering in 1991 and the PhD degree in Electrical Engineering in 1997, both from the Aristotle University of Thessaloniki, Greece. From 1992 to 1996 he served as teaching assistant in the Departments of Electrical Engineering and Informatics at the same University. From 1998 to 2002 he was postdoctoral researcher and teaching assistant at the Department of Informatics, Aristotle University of Thessaloniki. He is currently a Lecturer in the same Department. Dr. Nikolaidis is the co-author of the book "3-D Image Processing Algorithms" (J. Wiley, 2000). He has co-authored 6 book chapters, 22 journal papers and 82 conference papers. He currently serves as Associate Editor in the International Journal of Innovative Computing Information and Control and the EURASIP Journal on Image and Video Processing. His research interests include computer graphics, image and video processing and analysis, copyright protection of multimedia and 3-D image processing. Dr. Nikolaidis has been a scholar of the State Scholarship Foundation of Greece.

**Ioannis Pitas** received the Diploma of Electrical Engineering in 1980 and the PhD degree in Electrical Engineering in 1985 both from the Aristotle University of Thessaloniki, Greece. Since 1994, he has been a Professor at the Department of Informatics, Aristotle University of Thessaloniki. From 1980 to 1993 he served as Scientific Assistant, Lecturer, Assistant Professor, and Associate Professor in the Department of Electrical and Computer Engineering at the same University. He served as a Visiting Research Associate or Visiting Assistant Professor at several Universities. He has published 153 journal papers, 400 conference papers and contributed in 22 books in his areas of interest and edited or co-authored another 5. He has also been an invited speaker and/or member of the program committee of several scientific conferences and workshops. In the past he served as Associate Editor or co-Editor of four international journals and General or Technical Chair of three international conferences. His current interests are in the areas of digital image and video processing and analysis, multidimensional signal processing, watermarking and computer vision.