Available online at www.sciencedirect.com
SCIENCE DIRECT®
Signal Processing: *Image Communication* 18 (2003) 185–202
SIGNAL PROCESSING: IMAGE COMMUNICATION
www.elsevier.com/locate/image

Rigid and non-rigid 3D motion estimation from multiview image sequences ☆

N. Ploskas[a], D. Simitopoulos[a], D. Tzovaras[b], G.A. Triantafyllidis[a,*], M.G. Strintzis[a,b,1]

[a] *Information Processing Laboratory, Electrical and Computer Engineering Department, Aristotle University of Thessaloniki, 54124 Greece*

[b] *Informatics and Telematics Institute, 1st Km Thermi-Panorama Road, Thermi-Thessaloniki 57001, Greece*

Received 23 November 2001; received in revised form 1 April 2002; accepted 30 October 2002

Abstract

Multiview image sequence processing has been the focus of considerable attention in recent literature. This paper presents an efficient technique for object-based rigid and non-rigid 3D motion estimation, applicable to problems occurring in multiview image sequence coding applications. More specifically, a neural network is formed for the estimation of the rigid 3D motion of each object in the scene, using initially estimated 2D motion vectors corresponding to each camera view. Non-linear error minimization techniques are adopted for neural network weight update. Furthermore, a novel technique is also proposed for the estimation of the local non-rigid deformations, based on the multiview camera geometry. Experimental results using both stereoscopic and trinocular camera setups illustrate and evaluate the proposed scheme.

Keywords: Motion estimation; Rigid/non-rigid; Multiview

1. Introduction

Depth understanding is an important element of enhanced perception and tele-presence in image communication [10,29,3]. A direct way of inferring the depth information is provided by stereo and multi-ocular vision [4,22]. Stereoscopic, or in general multiview video, can provide more vivid and accurate information about the scene structure than simple video. Therefore, multiview video processing has been the focus of considerable attention in recent literature [8,14,21,24,25]. In a multiview image sequence, each different view is recorded with a difference in the observation angle, creating an enhanced 3D feeling to the observer, and increased "tele-presence" e.g. in teleconferencing.

Model-based coding has long attracted considerable attention as a promising alternative to block-based encoding for the analysis and coding

☆This work was supported by the EU project IST "HI-SCORE".

*Corresponding author. Informatics and Telematics Institute, 1st Km Thermi-Panorama Road, Thermi-Thessaloniki, 57001 Greece.

E-mail addresses: ploskas@dion.ee.auth.gr (N. Ploskas), dsim@olympus.ee.auth.gr (D. Simitopoulos), dimitrios.tzovaras@iti.gr (D. Tzovaras), gatrian@iti.gr (G.A. Triantafyllidis), strintzi@eng.auth.gr (M.G. Strintzis).

[1] Also for correspondence.

0923-5965/03/$ - see front matter
doi:10.1016/S0923-5965(02)00131-5

of stereo and multiview image sequences, achieving excellent performance, and producing fewer blocking artifacts than those commonly in block-based hybrid DCT coders at moderate and low bit rates [15,30]. The derivation of 3D models directly from images, usually requires estimation of dense disparity fields, post-processing to remove erroneous estimates and fitting of a surface model to the calculated depth map. Current model-based image coding schemes may be divided into two broad categories. The first category [2,9,11,31] is knowledge-based and uses human head models for coding primarily video-conferencing scenes. The second analysis-by-synthesis group of methods [5,12,15] is suitable for the coding of more general classes of images. The ability of model-based coding techniques to describe a scene in a structural way, in contrast to traditional waveform-based coding techniques, opens new areas of applications [1]. Video production, realistic computer graphics, multimedia interfaces and medical visualization are some of the applications that may benefit by exploiting the potential of model-based schemes.

In [14] an algorithm was presented which optimally models each scene using a hierarchical structure derived directly from intensity images. The wireframe model consists of adjacent triangles that may be split into smaller ones, over areas that need to be represented in higher detail. In [13,26] the 3D model is initialized by adapting a 2D wireframe to the foreground object. Using depth and multiview camera geometry the 2D wireframe is reprojected in the 3D space, forming a consistent wireframe for all views.

In all model- and object-based monoscopic image sequence coding schemes, motion estimation and motion compensated prediction are used to reduce temporal redundancy. Similarly, coding of stereo and multiview images may be based on disparity compensation or the best of motion and disparity compensation [14,25].

An efficient approach for 3D motion estimation between two or more consecutive time frames using neural networks was presented in [6,7]. In this approach, the initial 3D correspondence was first found, using a feature extraction procedure and matching of the corresponding feature points by a Hopfield neural network. Following this, a neural network was designed to estimate the rigid 3D motion parameters of the moving object, based on this initial 3D correspondence. The non-rigid 3D motion was also estimated based on the initial 3D correspondence by the set of neural networks described in [7]. However, the establishment of an initial 3D correspondence is not an easy task in real scenes [16] and thus this algorithm cannot be used successfully in image sequence coding applications.

In [29] a neural network approach was introduced for estimating the 3D motion parameters of the rigid 3D scene objects, from monoscopic image sequences using initial 2D motion vectors on the camera image plane. The authors adapted the results of [6] for the solution of coding-oriented motion estimation problems, where only 2D motion vectors rather than 3D correspondences are initially available. The initial 2D motion field was obtained by a simple block matching motion estimation between the consecutive frames. The technique in [29] was seen to improve the 3D motion estimates of [6,7], even in cases where 3D correspondences were known with accuracy.

In the present paper, we extend this technique so as to make it applicable for multiview image sequences. In this case, initial 2D vectors are available at the projections of the 3D nodes on all the image planes of a multiview camera geometry. This is seen to improve significantly the results in [29], in all examined cases, even in the presence of measurement noise. The rigid 3D motion of each articulated object in the scene, is estimated using a neural network based on the available 2D motion information on the image planes of the multiview camera geometry. The weights of the neural network are updated using non-linear error minimization techniques. The technique in [29] is also extended by developing a novel approach for flexible 3D motion estimation of each node of the object model. The performance of the rigid and non-rigid 3D motion estimation techniques is evaluated experimentally on both synthetic and real 3D object motion, assuming stereo and trinocular camera setups. The basic approach of the paper may easily be extended to any multiview system using arbitrary number and arrangements of cameras.

The paper is organized as follows. In Section 2 the camera geometry of the system is briefly described. The rigid 3D motion estimation procedure for each articulated 3D object is discussed in Section 3, while in Section 4 a robust rigid 3D motion estimation via outlier removal is presented. The non-rigid 3D motion estimation procedure for each node of the objects is elaborated in Section 5. Experimental results given in Section 6 demonstrate the performance of the proposed methods. Finally, conclusions are drawn in Section 7.

2. Camera model

A camera model describes the projection of 3D points onto a camera target. The model used here is the CAHV model introduced in [32] (Fig. 1). This model describes extrinsic camera parameters such as position and orientation and intrinsic camera parameters such as focal length and intersection between optical axis and image plane.

The experimental results in the present paper were obtained using a stereo setup ($c =$ left, right) and a trinocular camera setup ($c =$ left, top, right). The latter is increasingly being used in teleconference applications whenever higher tele-presence is desired [28]. For each camera c the model contains the following parameters: (a) position of the camera $\mathbf{C}_c$, (b) optical axis $\mathbf{A}_c$, i.e. the viewing direction of the camera (unit vector), (c) horizontal camera target vector $\mathbf{H}_c$ (x-axis of the camera target), (d) vertical camera target vector $\mathbf{V}_c$ (y-axis of the camera target) and sx, sy the pixel size.

In the aforementioned camera model we shall assume that the camera parameters are estimated at an initial calibration stage using the techniques in [17]. We also assume that the radial distortion is compensated prior to any other operation, at an initialization stage, following camera calibration. According to this model, the projection of a 3D point $\mathbf{P}$, with coordinates relative to world coordinate system, onto the image plane $(\mathbf{X}'_c, \mathbf{Y}'_c)$ is [32]

$$X'_c = \frac{(\mathbf{P}-\mathbf{C}_c)\cdot\mathbf{H}_c}{(\mathbf{P}-\mathbf{C}_c)\cdot\mathbf{A}_c}, \quad Y'_c = \frac{(\mathbf{P}-\mathbf{C}_c)\cdot\mathbf{V}_c}{(\mathbf{P}-\mathbf{C}_c)\cdot\mathbf{A}_c}. \tag{1}$$

The coordinates $(\mathbf{X}'_c, \mathbf{Y}'_c)$ are camera centered (image plane coordinate system) with the unit pel. The origin of the coordinate system is the center point of the camera. The coordinates of a point relative to the picture coordinate system $(\mathbf{X}_c, \mathbf{Y}_c)$ are given by $(X_c, Y_c) = (X'_c + O_{x,c}, Y'_c +$

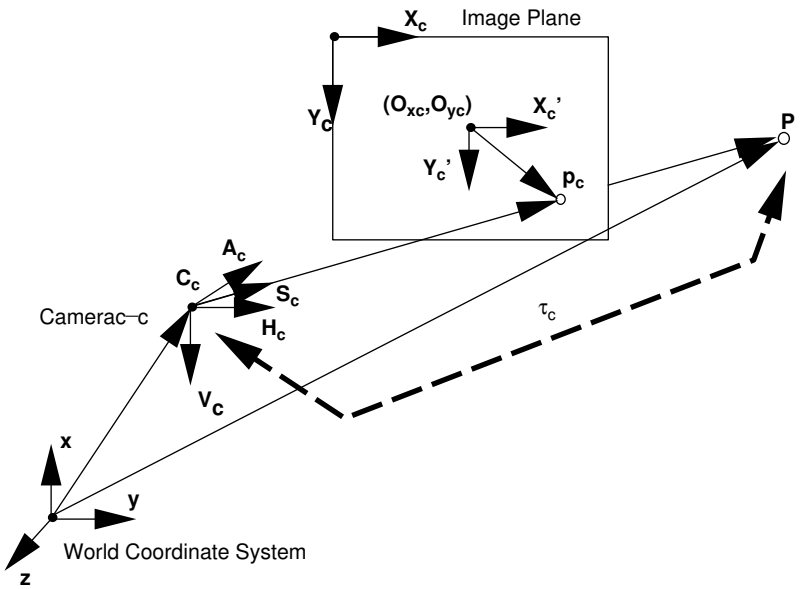


Fig. 1. The CAHV camera model.

$O_{y,c}$), where $(O_{x,c}, O_{y,c})$ is the center of the image plane in the picture coordinate system.

Conversely, given its position $(\mathbf{X}_c, \mathbf{Y}_c)$ on the camera plane, the 3D position of a point can be determined by

$$\mathbf{P} = \mathbf{C}_c + \tau_c \cdot \mathbf{S}_c(X_c, Y_c), \tag{2}$$

where $\mathbf{S}_c(X_c, Y_c)$ is the unit vector pointing from the camera c to the point in the direction of the optical axis and τ_c is the distance between the 3D point and the center of camera c.

3. Rigid 3D motion estimation from trinocular and stereo image sequences

Let us assume the availability of a 3D model of the scene, along with a valid articulation at time instant t. The following motion model is used for each object in the scene:

$$\mathbf{p}(t+1) = \mathbf{R} \cdot \mathbf{p}(t) + \mathbf{T}, \tag{3}$$

or equivalently, $$\mathbf{P}(t+1) = \mathbf{M} \cdot \mathbf{P}(t), \tag{4}$$

where $\mathbf{M}$ is the homogeneous matrix of the form

$$\mathbf{M} = \begin{bmatrix} \mathbf{R} & \mathbf{T} \\ 0 & 1 \end{bmatrix}, \text{ or equivalently,}$$

$$\mathbf{M} = \begin{bmatrix} \mathbf{W}_1 \\ \mathbf{W}_2 \\ \mathbf{W}_3 \\ \mathbf{W}_4 \end{bmatrix} = \begin{bmatrix} w_{11} & w_{12} & w_{13} & w_{14} \\ w_{21} & w_{22} & w_{23} & w_{24} \\ w_{31} & w_{32} & w_{33} & w_{34} \\ 0 & 0 & 0 & 1 \end{bmatrix}, \tag{5}$$

and $\mathbf{P}(t) = [\mathbf{p}(t)\ 1]^{\mathrm{T}}$ is the corresponding homogeneous 3D point. The formulation of Eq. (4) may represent non-rigid as well as rigid motion. In the specific case of rigid motion,

$$\mathbf{M} = \begin{bmatrix} k_x^2(1-\cos(\theta)) + \cos(\theta) & k_x k_y(1-\cos(\theta)) - k_z \sin(\theta) & k_x k_z(1-\cos(\theta)) + k_y \sin(\theta) & T_x \\ k_x k_y(1-\cos(\theta)) + k_z \sin(\theta) & k_y^2(1-\cos(\theta)) + \cos(\theta) & k_y k_z(1-\cos(\theta)) - k_x \sin(\theta) & T_y \\ k_x k_z(1-\cos(\theta)) - k_y \sin(\theta) & k_y k_z(1-\cos(\theta)) + k_x \sin(\theta) & k_z^2(1-\cos(\theta)) + \cos(\theta) & T_z \\ 0 & 0 & 0 & 1 \end{bmatrix}, \tag{6}$$

where $[T_x, T_y, T_z]^{\mathrm{T}}$ is the translation vector, $\mathbf{k} = [k_x, k_y, k_z]^{\mathrm{T}}$ is the axis and θ is the angle of rotation. One method for the extraction of vector $\mathbf{k}$ from the w terms of $\mathbf{M}$, which was experimentally found to be robust and very accurate is the following [30,6]:

$$k_x = \mathrm{sgn}(w_{32} - w_{23})\sqrt{\frac{w_{11} - \cos(\theta)}{1 - \cos(\theta)}},$$

$$k_y = \mathrm{sgn}(w_{13} - w_{31})\sqrt{\frac{w_{22} - \cos(\theta)}{1 - \cos(\theta)}},$$

$$k_z = \mathrm{sgn}(w_{21} - w_{12})\sqrt{\frac{w_{33} - \cos(\theta)}{1 - \cos(\theta)}}, \tag{7}$$

where

$$\mathrm{sgn}(x) = \begin{cases} + & \text{if } x \geqslant 0, \\ - & \text{if } x \leqslant 0 \end{cases} \tag{8}$$

and

$$\cos(\theta) = \frac{1}{2}(w_{11} + w_{22} + w_{33} - 1). \tag{9}$$

Observing also that two of (k_x, k_y, k_z) suffice to describe the rotation axis vector $\mathbf{k}$ (since $\mathbf{k}$ is a unit vector, hence $k_x^2 + k_y^2 + k_z^2 = 1$), we conclude that in the case of rigid motion only six parameters suffice to characterize $\mathbf{M}$.

Let us also assume that 2D motion vector measures are available at the projections of the nodes of the 3D model on the image planes of the left, top and right camera (in the case of trinocular vision), or of the left and right camera (in case of stereo vision) found by means of an initial 2D motion estimation procedure, applied on the three image planes (two, in the case of stereo). This information, consisting of 2D vectors $[\hat{d}^{(c)}_{x(ik)}, \hat{d}^{(c)}_{y(ik)}]$, will be used for the estimation of the rigid 3D motion parameters of the object, by minimizing the following error measures for the projection of the ith vertex $\mathbf{P}_i^{(k)}(t)$, $1 \leqslant i \leqslant 3$, of the kth triangle of the object on the image plane of camera c (where c = l, t, r, left, top and right camera, respectively, in the case of trinocular vision),

$$e^{(c)}_{x(ik)} = (d^{(c)}_{x(ik)} - \hat{d}^{(c)}_{x(ik)})^2$$

and

$$e^{(c)}_{y(ik)} = (d^{(c)}_{y(ik)} - \hat{d}^{(c)}_{y(ik)})^2, \tag{10}$$

where

$$\hat{d}^{(c)}_{x(ik)} = \frac{(\mathbf{MP}^{(k)}_i(t) - \mathbf{C}_c)\cdot \mathbf{H}_c}{(\mathbf{MP}^{(k)}_i(t) - \mathbf{C}_c)\cdot \mathbf{A}_c} - \frac{(\mathbf{P}^{(k)}_i(t) - \mathbf{C}_c)\cdot \mathbf{H}_c}{(\mathbf{P}^{(k)}_i(t) - \mathbf{C}_c)\cdot \mathbf{A}_c}, \tag{11}$$

$$\hat{d}^{(c)}_{y(ik)} = \frac{(\mathbf{MP}^{(k)}_i(t) - \mathbf{C}_c)\cdot \mathbf{V}_c}{(\mathbf{MP}^{(k)}_i(t) - \mathbf{C}_c)\cdot \mathbf{A}_c} - \frac{(\mathbf{P}^{(k)}_i(t) - \mathbf{C}_c)\cdot \mathbf{V}_c}{(\mathbf{P}^{(k)}_i(t) - \mathbf{C}_c)\cdot \mathbf{A}_c}, \tag{12}$$

and $d^{(c)}_{x(ik)}$ and $d^{(c)}_{y(ik)}$ are the x- and y- components of the initially estimated 2D motion vectors on the image plane of camera c. Note that $1 \leqslant k \leqslant M$ assuming that the wireframe of the object consists of M triangles.

Other error measures may be determined if it is assumed that the structure of the 3D model of the rigid object remains unchanged and thus the distances between the vertices of each triangle remain constant with time. Let $A^{(k)}_{ij}$ be the distance between vertices i and j of the working triangle, at time t and $\hat{A}^{(k)}_{ij}$ the distance between the same vertices at time $t+1$. If Eq. (4) is used, the following error measures are defined:

$$\begin{aligned}
e^{(k)}_1 &= (|\mathbf{P}^{(k)}_1(t) - \mathbf{P}^{(k)}_2(t)| - |\mathbf{M}(\mathbf{P}^{(k)}_1(t) - \mathbf{P}^{(k)}_2(t))|)^2 \\
&= [A^{(k)}_{12} - \hat{A}^{(k)}_{12}]^2, \\
e^{(k)}_2 &= (|\mathbf{P}^{(k)}_1(t) - \mathbf{P}^{(k)}_3(t)| - |\mathbf{M}(\mathbf{P}^{(k)}_1(t) - \mathbf{P}^{(k)}_3(t))|)^2 \\
&= [A^{(k)}_{13} - \hat{A}^{(k)}_{13}]^2, \\
e^{(k)}_3 &= (|\mathbf{P}^{(k)}_2(t) - \mathbf{P}^{(k)}_3(t)| - |\mathbf{M}(\mathbf{P}^{(k)}_2(t) - \mathbf{P}^{(k)}_3(t))|)^2 \\
&= [A^{(k)}_{23} - \hat{A}^{(k)}_{23}]^2.
\end{aligned} \tag{13}$$

If $\mathbf{C}(t)$ is the centroid of the rigid object, three more error measures may be determined:

$$\begin{aligned}
e^{(k)}_4 &= |\mathbf{P}^{(k)}_1(\mathrm{t}) - \mathbf{C}^{(\mathrm{t})}| - |\mathbf{M}(\mathbf{P}^{(k)}_1(t) - \mathbf{C}^{(\mathrm{t})})| \\
&= [A^{(k)}_{1c} - \hat{A}^{(k)}_{1c}], \\
e^{(k)}_5 &= |\mathbf{P}^{(k)}_2(\mathrm{t}) - \mathbf{C}^{(\mathrm{t})}| - |\mathbf{M}(\mathbf{P}^{(k)}_2(\mathrm{t}) - \mathbf{C}^{(\mathrm{t})})| \\
&= [A^{(k)}_{2c} - \hat{A}^{(k)}_{2c}], \\
e^{(k)}_6 &= |\mathbf{P}^{(k)}_3(t) - \mathbf{C}^{(\mathrm{t})}| - |\mathbf{M}(\mathbf{P}^{(k)}_3(t) - \mathbf{C}^{(\mathrm{t})})| \\
&= [A^{(k)}_{3c} - \hat{A}^{(k)}_{3c}].
\end{aligned} \tag{14}$$

Thus, the following 18×1 error vector is formed for a stereoscopic camera setup:

$$\begin{aligned}
\mathbf{E}_k = [&e^{(k)}_1, \ldots, e^{(k)}_6, e^{(\mathrm{l})}_{x(1k)}, e^{(\mathrm{l})}_{y(1k)}, e^{(\mathrm{r})}_{x(1k)}, e^{(\mathrm{r})}_{y(1k)}, \\
&\ldots, e^{(\mathrm{l})}_{x(3k)}, e^{(\mathrm{l})}_{y(3k)}, e^{(\mathrm{r})}_{x(3k)}, e^{(\mathrm{r})}_{y(3k)}]^{\mathrm{T}},
\end{aligned} \tag{15}$$

and for a trinocular camera setup the 24×1 error vector is formed as follows:

$$\begin{aligned}
\mathbf{E}_k = [&e^{(k)}_1, \ldots, e^{(k)}_6, e^{(\mathrm{l})}_{x(1k)}, e^{(\mathrm{l})}_{y(1k)}, e^{(\mathrm{t})}_{x(1k)}, e^{(\mathrm{t})}_{y(1k)}, \\
&e^{(\mathrm{r})}_{x(1k)}, e^{(\mathrm{r})}_{y(1k)}, \ldots, e^{(\mathrm{l})}_{x(3k)}, e^{(\mathrm{l})}_{y(3k)}, e^{(\mathrm{t})}_{x(3k)}, e^{(\mathrm{t})}_{y(3k)}, \\
&e^{(\mathrm{r})}_{x(3k)}, e^{(\mathrm{r})}_{y(3k)}]^{\mathrm{T}}.
\end{aligned} \tag{16}$$

The motion parameter vector $\mathbf{W}$ defined by

$$\begin{aligned}
\mathbf{W} = [&w_{11}, w_{12}, w_{13}, w_{14}, w_{21}, \\
&w_{22}, w_{23}, w_{24}, w_{31}, w_{32}, w_{33}, w_{34}]^{\mathrm{T}}
\end{aligned} \tag{17}$$

is determined at iteration $n+1$ using the Newton–Raphson procedure

$$\mathbf{W}^{(n+1)} = \mathbf{W}^{(n)} + \mathbf{DW}^{(n)}, \tag{18}$$

where

$$\mathbf{DW}^{(n)} = \sum_{k=1}^{M} \left(-\mu[\mathbf{J}^{\mathrm{T}}_k \mathbf{J}_k]^{-1} \mathbf{J}^{\mathrm{T}}_k \mathbf{E}_k\right) \tag{19}$$

and $\mathbf{J}_k$ is the Jacobian matrix corresponding to the error vector $\mathbf{E}_{(k)}$ and μ is the learning rate. This procedure may be represented by the neural network shown in Fig. 2 with weights adapted as prescribed by Eq. (18).

The proposed neural network is composed of three layers and minimizes the above error terms using supervised modification of the weights between processing elements between the first and the second layer. The components of the point vectors of the three vertices of the working triangle are input to the first layer of the network. The weights of the connections between the neurons of the first and the second layer are the components of the motion parameter vector $\mathbf{W}$ given by Eq. (17). The outputs of the second layer are the components of the estimated point vectors of the vertices of the working triangle at time instant $t+1$. Obviously, these correspond to the motion parameter vector $\mathbf{W}$ of each iteration of the training procedure. The weights between the second and the third layer are constant and are used to implement the functions of the point

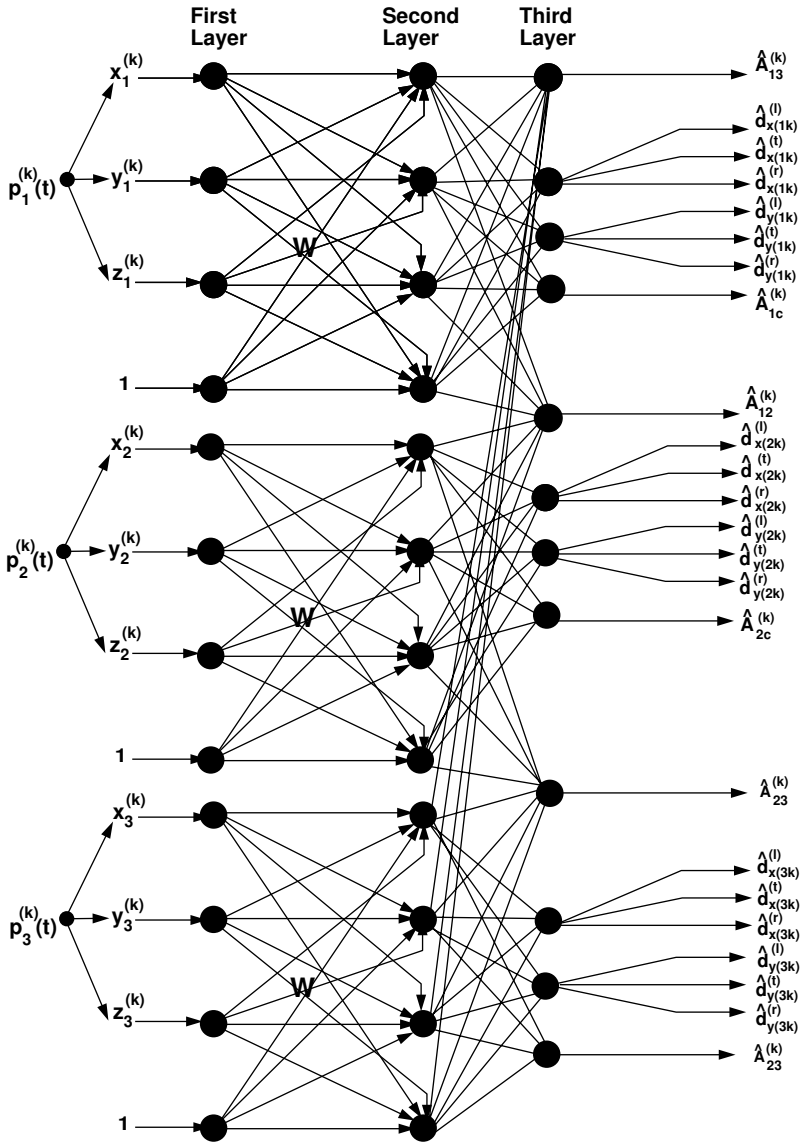


Fig. 2. Neural network estimating the rigid 3D motion parameters for a trinocular camera setup.

vectors at time instant $t+1$ that produce the outputs of the neural network. Learning is based on the minimization of the error measures which are defined by the deviation of the output of the third layer from the desirable one. This minimization is carried out by modifying the weights between the first and the second layer according to Eq. (18). The desirable output of the third layer is composed of the initially estimated 2D motion vectors and the distances between the vertices of the working triangle along with the distances between each vertex and the centroid of the object.

4. Robust rigid 3D motion estimation via outlier removal

The input 2D motion vector field is obtained by block matching techniques and therefore is not always composed of reliable measurements. In

particular, errors may occur within homogeneous areas in the interior or exterior of the objects. Thus an initial outlier removal procedure is necessary for the successful implementation of the algorithm.

The model parameters are then estimated using an iterative estimation method based on the moving least median of squares approximation algorithm (MLMS) [23] which minimizes the error of the motion model. The MLMS algorithm is based on median filtering and is optimal in suppressing noise consisting of a large number of outliers. In such situations, conventional least squares techniques are likely to fail [23].

At each iteration of the procedure, a number of N_p triangles are randomly selected from the predetermined set of M triangles composing the object. The neural network is then trained based on this subset of triangles and a corresponding parameter vector $\mathbf{W}$ is computed at iteration L. The parameter vector is calculated by minimizing the LMS (least median of squares) objective function, over all selected N_p triangles.

At each iteration, the algorithm discards the triangles that do not fit the motion model (“outlier” triangles). More specifically, a triangle is characterized as outlier and rejected, if it is very frequently a member of the random sets that lead to high model fit errors. The new set of more reliable triangles is inserted to the neural network and the whole procedure is iterated until convergence to the optimal parameter vector $\mathbf{W}$. This procedure appears to be very computationally intensive since $C_{N_p}^{M}$ (combinations M of N_p) subsets of random triangles have to be chosen. However, in actual practice only a limited number of iterations of the algorithm are needed before convergence.

5. Non-rigid 3D motion estimation from stereo and trinocular image sequences

Let us assume again the availability of a 3D model of the scene at time instant t. The following motion model is used for each node:

$$\mathbf{p}_j(t+1) = \mathbf{p}_j(t) + \mathbf{D}_j, \tag{20}$$

where $\mathbf{p}_j(t)$ and $\mathbf{p}_j(t+1)$ are vectors in 3D space which represent the positions of the jth node at time instants t and $t+1$, respectively, and $\mathbf{D}_j$ is the translation in 3D space of the specific node between those two time instants. Note that $1 \leqslant j \leqslant N$, where N is the total number of the nodes of the 3D model of the scene.

Let us also assume again that 2D motion vector measures are available at the projections of the nodes of the 3D model on the images from the left, top and right camera, in the case of trinocular vision, found by means of an initial 2D motion estimation procedure on the three image planes. This information will be used for the estimation of the displacements in 3D space of the nodes of the available 3D model, which will be assumed to estimate the non-rigid 3D motion parameters of the objects in the scene. For a trinocular camera setup the initial motion information for each node consists of three 2D vectors $[\hat{d}_{x(j)}^{(c)}, \hat{d}_{y(j)}^{(c)}]\,(c = l, t, r)$, at the projections of the node on the image planes of the left, top and right camera, respectively. The information of the available 2D motion vectors on the three images is used for the estimation of $\mathbf{D}_j$ by minimizing the following error measures for the projection of the working node on the image plane of camera c. For a stereo camera setup only the measurements corresponding to cameras c ($c = \mathrm{l}, r$). For either setup,

$$e_{x(j)}^{(c)} = (d_{x(j)}^{(c)} - \hat{d}_{x(j)}^{(c)})^2$$

and

$$e_{y(j)}^{(c)} = (d_{y(j)}^{(c)} - \hat{d}_{y(j)}^{(c)})^2, \tag{21}$$

where

$$\hat{d}_{x(j)}^{(c)} = \frac{(\mathbf{p}_j(t) + \mathbf{D}_j - \mathbf{C}_c)\cdot \mathbf{H}_c}{(\mathbf{p}_j(t) + \mathbf{D}_j - \mathbf{C}_c)\cdot \mathbf{A}_c} - \frac{(\mathbf{p}_j(t) - \mathbf{C}_c)\cdot \mathbf{H}_c}{(\mathbf{p}_j(t) - \mathbf{C}_c)\cdot \mathbf{A}_c}, \tag{22}$$

$$\hat{d}_{y(j)}^{(c)} = \frac{(\mathbf{p}_j(t) + \mathbf{D}_j - \mathbf{C}_c)\cdot \mathbf{V}_c}{(\mathbf{p}_j(t) + \mathbf{D}_j - \mathbf{C}_c)\cdot \mathbf{A}_c} - \frac{(\mathbf{p}_j(t) - \mathbf{C}_c)\cdot \mathbf{V}_c}{(\mathbf{p}_j(t) - \mathbf{C}_c)\cdot \mathbf{A}_c}, \tag{23}$$

and $d_{x(j)}^{(c)}$ and $d_{y(j)}^{(c)}$ are the x- and y-components of the initially estimated 2D motion vectors on the image plane of camera c. Thus, the following 4×1 error vector is formed for a stereoscopic

camera setup:

$$\mathbf{E}_j = [e^{(l)}_{x(j)}, e^{(l)}_{y(j)}, e^{(r)}_{x(j)}, e^{(r)}_{y(j)}]^T, \quad (24)$$

while for a trinocular camera setup the 6×1 error vector is formed,

$$\mathbf{E}_j = [e^{(l)}_{x(j)}, e^{(l)}_{y(j)}, e^{(t)}_{x(j)}, e^{(t)}_{y(j)}, e^{(r)}_{x(j)}, e^{(r)}_{y(j)}]^T, \quad (25)$$

The motion parameter vector $\mathbf{D}_j$, defined by

$$\mathbf{D}_j = [d^{(j)}_x, d^{(j)}_y, d^{(j)}_z]^T, \quad (26)$$

is determined using the Newton–Raphson procedure

$$\mathbf{D}^{(n+1)}_j = \mathbf{D}^{(n)}_j - [\mathbf{J}^T_j \mathbf{J}_j]^{-1} \mathbf{J}^T_j \mathbf{E}_j, \quad (27)$$

where $\mathbf{J}$ is the Jacobian matrix.

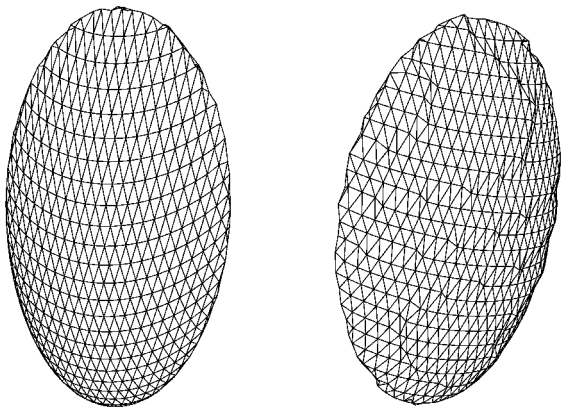


Fig. 3. (a), (b) Synthetic data model for time instants 1 and 2, respectively.

6. Experimental results

The proposed 3D motion estimation algorithms were evaluated on both synthetically created and real image sequences obtained with the camera setups described in Section 2.

6.1. Experimental results in synthetic image sequences

In order to test the performance of the proposed algorithm in 3D motion estimation, a synthetic 3D model of about 300 triangles was created. A hemispherical structure was chosen, since this shape approximates several natural volumes including the human face. The wireframe of the 3D model is shown in Fig. 3(a).

The 3D data of the available model were subjected to a 3D transformation consisting of a rigid and a non-rigid part. Each node of the wireframe $\mathbf{p}_j(t=1)$ at time instant $t=1$ was moved to point $\mathbf{p}_j(t=2)$ given by

$$\mathbf{p}_j(t=2) = \mathbf{R} \cdot \mathbf{p}_j(t=1) + \mathbf{T} + \mathbf{D}_j, \quad (28)$$

where the rotation matrix $\mathbf{R}$ and the translation vector $\mathbf{T}$ define a global rigid 3D motion and $\mathbf{D}_j$ denotes the non-rigid translation vector of each node. The rigid 3D motion parameters corresponding to $\mathbf{R}$ and $\mathbf{T}$, denoted as ideal, are given in Table 1. The three components of $\mathbf{D}_j$ are chosen to be independent Gaussian random variables with zero mean and standard deviation $\sigma_D = 2$ mm. The resulting synthetic 3D data model for time instant $t=2$ is shown in Fig. 3(b). The 2D motion vectors were then computed by projecting the 3D points at time instants $t=1$ and 2 on the three image planes of a trinocular camera arrangement and computing their difference. In order to

Table 1
Test results on synthetic images

Motion parameters	$\mathbf{k}$	θ	$\mathbf{T}$	$E[d_{err}]$
Ideal results	$[0.577, 0.577, 0.577]^T$	15.0°	$[-5.0, 15.0, 5.0]^T$	0
Monoscopic	$[0.570, 0577, 0.585]^T$	14.1°	$[-4.7, 11.9, 12.1]^T$	8.4%
Stereo	$[0.564, 0.583, 0.585]^T$	14.6°	$[-4.8, 14.4, 6.8]^T$	2.7%
Trinocular	$[0.573, 0.577, 0.581]^T$	14.8°	$[-4.7, 14.6, 5.9]^T$	1.3%
Method in [6,7]	$[0.512, 0.476, 0.458]^T$	12.7°	$[-4.5, 15.9, 5.7]^T$	11.2%

simulate the effect of the non-accuracy of 2D motion estimation, the available vectors were assumed to be corrupted by additive white Gaussian noise. A signal-to-noise ratio of 20 dB was chosen. The displacement vectors served then as input to the neural network for the estimation of the global rigid 3D motion, as discussed in Section 3.

The learning rate of the network μ relates to the speed of convergence (i.e. required number of iterations) and stability issues concerning the motion estimation procedures. In general, the selection of a large learning rate value incurs fast convergence. However, setting this value too high can also lead to instability and result in computational oscillations. It was also observed that as the number of the 3D triangles increased, a smaller learning rate was required for the rigid-3D-motion estimator. The learning rate of the network μ in Eq. (18) was selected equal to 0.01. The non-rigid motion of each node was estimated next using the method described in Section 5. It was observed that the proposed technique outperformed the 3D motion estimators described in [6,7] offering an improvement of approximately 50% in terms of speed of convergence.

The method was compared with the approach described in [6,7]. The effect of increasing the number of camera views from one to three was also investigated. Note that for a monoscopic camera setup only the global rigid 3D motion was estimated, since the proposed technique for non-rigid motion estimation cannot be applied using measurements from only one camera. The algorithms were tested in terms of the accuracy of the estimated rigid 3D motion parameters (rigid translation vector $\mathbf{T}$, axis of rotation $\mathbf{k}$ and angle of rotation θ) and the mean 3D displacement prediction error $E[d_{\mathrm{err}}]$ after the rigid and non-rigid motion compensation,

$$E[d_{\mathrm{err}}] = E[|\mathbf{p}_j(t=2) - \hat{\mathbf{p}}_j(t=2)|], \tag{29}$$

where $\hat{\mathbf{p}}_j(t=2)$ denotes the estimated position in 3D space at time instant $t=2$, of the point $\mathbf{p}_j(t=1)$. Results are shown in Table 1 where $E[d_{\mathrm{err}}]$ is expressed as a percentage of $E[|\mathbf{p}_j(t=2) - \mathbf{p}_j(t=1)|]$.

By observing Table 1, it can easily be seen that the performance in terms of suppression of measurement noise, is improving with the number of camera views used. For a monoscopic camera setup, the prediction accuracy is rather poor when compared with the stereo and trinocular camera arrangements. The presence of non-rigid local deformations also proves to deteriorate considerably the 3D correspondence established by the Hopfield neural network in [6] since the constraints imposed are based on the rigidity of the object.

6.2. Experimental results for real image sequences

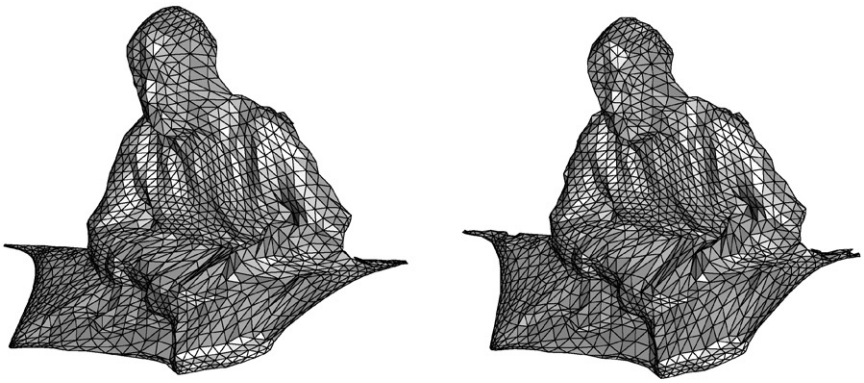
The proposed object-based coding was also evaluated for the 3D motion estimation from real image sequences. The interlaced multiview videoconference sequences "Ludo" and "Chantal" were used for the tests. All experiments were performed at the top field of the interlaced sequence, of dimension 360×288.

A 3D model of 2000 triangles was used to approximate the shape of the foreground object at each time instance for the sequence Ludo. Fig. 4 shows the 3D model adapted to the first and the second frame of the image sequence Ludo. The 3D model used in the experiments was produced using the shape initialization module of the EC PANORAMA project [18–20]. This method is based on back-projection of initially estimated depth information followed by triangulation using Discrete Smooth Interpolation [32]. The foreground object was subsequently subdivided into two sub-objects (i.e. head and body in case of Ludo) using the algorithm described in [27]. The proposed algorithm described in Section 3 was then used to estimate the rigid 3D motion parameters of each sub-object.

After the estimation of the rigid 3D motion and its compensation on the three image planes, the algorithm described in Section 5 was used for the estimation of the non-rigid 3D motion of the objects. It should be noted that no further object segmentation was applied in order to improve the results of rigid motion extraction through non-rigid motion estimation and compensation.

The initial 2D motion field was first obtained by a block-matching motion estimation between the

original first and second frames of Ludo. The 2D motion correspondence was established on the image planes of the three cameras of the multiview geometry. Inputs to the proposed neural network are the nodes of the wireframe at time instant $t = 1$ and the 2D motion vectors at the projections of the above nodes. Note that the methods in [6,7] require accurate knowledge of the 3D description of the objects for two consecutive time instances. The NN described in Section 3 was used for the estimation of the rigid 3D motion of the head of Ludo. Figs. 5 and 6 show the first and second multiview frame of Ludo while Figs. 8 and 12 show the rigid motion compensated estimates of frame 2 based on the 3D motion parameters of the head estimated by the NN using a stereo and a



(a) (b)

Fig. 4. (a), (b) Real data model for the image sequence Ludo for time instants 1 and 2, respectively.

(a) (b) (c)

Fig. 5. (a), (b), (c) Original camera images of frame 1 (left, top, right views, respectively).

(a) (b) (c)

Fig. 6. (a), (b), (c) Original camera images of frame 2 (left, top, right views, respectively).

trinocular camera setup, respectively. The learning rate of the network μ was selected to be equal to 0.01 and the number of iterations required for convergence was approximately 50.

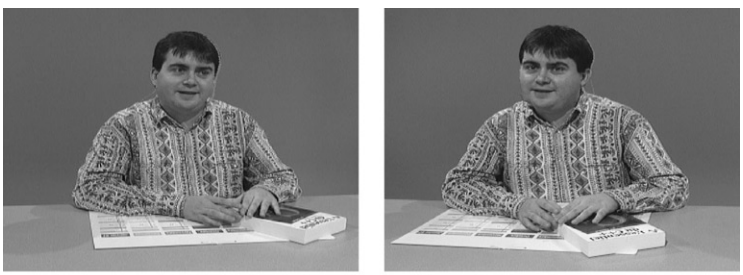
The rigid motion estimate of frame 2 was then used for the estimation of the non-rigid motion of the objects in the scene. A 2D motion correspondence was established between the above frame prediction of time instant $t = 2$ and the original frame 2. The NN described in Section 5 was used for the estimation of the non-rigid 3D motion of the head of Ludo. The number of iterations required for convergence was less than 10 for the non-rigid 3D motion estimation of each node. Figs. 10 and 14, show the rigid and non-rigid motion compensated estimates of frame 2 based on the 3D motion parameters of the head estimated by the proposed NNs, in case of stereo and trinocular setup, respectively. The computational time required for the rigid and non-rigid motion estimation of the 3D scene objects was a few seconds in a R4400 INDIGO II SGI machine.

Figs. 5 and 6 show a rotation of the head of Ludo from left to right along with some non-rigid motion of the eyes and the mouth which move independently. The frame difference between frames 1 and 2 is shown in Fig. 7, zoomed in the head area (where the 3D motion occurs), while the corresponding zoomed rigid motion compensated displaced frame difference, is shown in Fig. 9 and 13, where a stereoscopic and a trinocular camera setup has been used, respectively. The improved rigid and non-rigid motion compensated displaced frame differences are shown in Figs. 11 and 15. The corresponding PSNR measurements for both camera arrangements are presented in Tables 2 and 3 (Figs. 17–19).

The proposed methods were also evaluated for the estimation of the rigid and non-rigid 3D motion between frames 1 and 2 of the image sequence ''Chantal''. This image sequence exhibits a significantly higher amount of non-rigid local deformations when compared to Ludo. More accurate 3D models of about 8000 triangles were

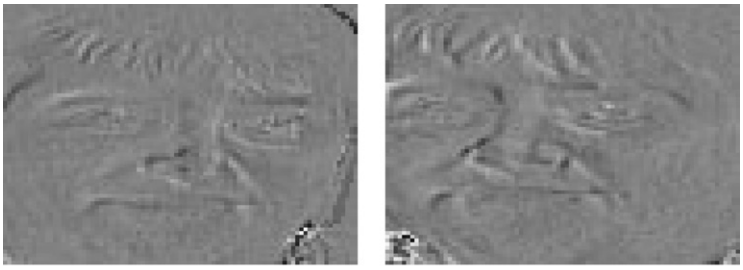
(a) (b) (c)

Fig. 7. (a), (b), (c) Zoom in the difference between original frames 2 and 1 (left, top, right views, respectively).



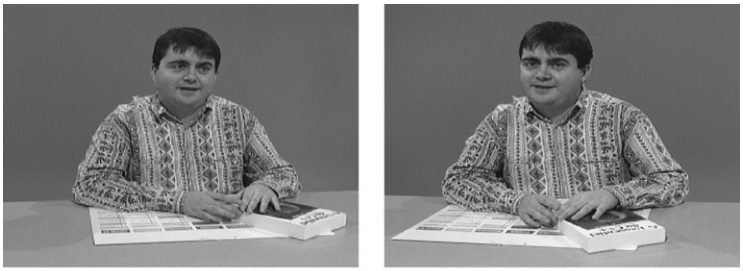
(a) (b)

Fig. 8. (a), (b) Rigid motion compensated estimate of frame 2 for a stereo camera setup (left, right views, respectively).



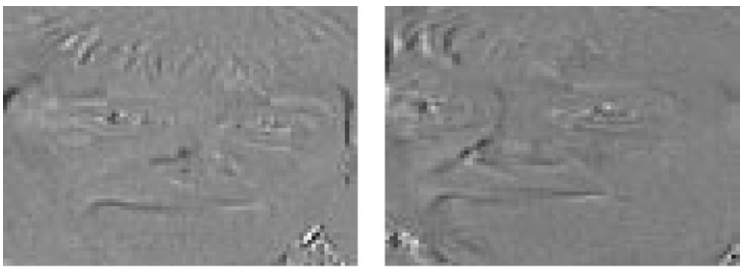
(a) (b)

Fig. 9. (a), (b) Zoom in the displaced frame difference between original frame 2 and its rigid motion compensated estimate for a stereo camera setup (left, right views, respectively).



(a) (b)

Fig. 10. (a), (b) Rigid and nonrigid motion compensated estimate of frame 2 for a stereo camera setup (left, right views, respectively).



(a) (b)

Fig. 11. (a), (b) Zoom in the displaced frame difference between original frame 2 and its rigid and nonrigid motion compensated estimate for a stereo camera setup (left, right views, respectively).

(a) (b) (c)

Fig. 12. (a), (b), (c) Rigid motion compensated estimate of frame 2 for a trinocular camera setup (left, top, right views, respectively).

(a) (b) (c)

Fig. 13. (a), (b), (c) Zoom in the displaced frame difference between original frame 2 and its rigid motion compensated estimate for a trinocular camera setup (left, top, right views, respectively).

(a) (b) (c)

Fig. 14. (a), (b), (c) Rigid and nonrigid motion compensated estimate of frame 2 for a trinocular camera setup (left, top, right views, respectively).

(a) (b) (c)

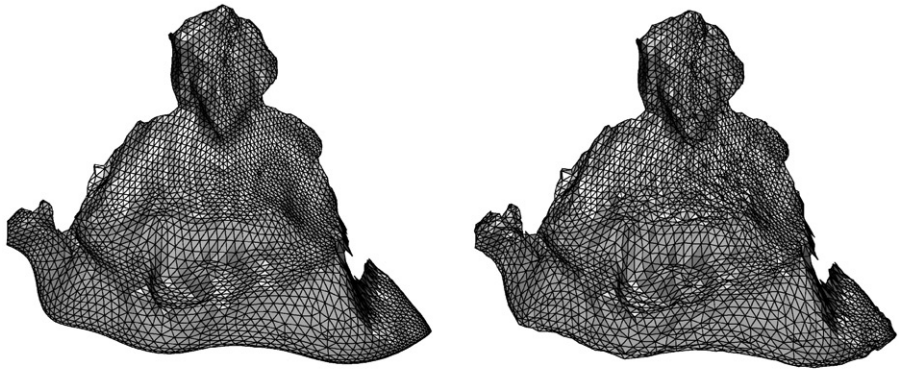
Fig. 15. (a), (b), (c) Zoom in the displaced frame difference between original frame 2 and its rigid and nonrigid motion compensated estimate for a trinocular camera setup (left, top, right views, respectively).

Table 2
Test results on real trinocular images in terms of PSNR for the head area of image sequence "Ludo"

Camera view	Difference between original frames 2 and 1	Difference between frame 2 and its rigid motion estimate	Difference between frame 2 and its aggregate motion estimate	Difference between frame 2 and its estimate with method in [6,7]
Left	22.99 dB	27.36 dB	28.22 dB	23.97 dB
Top	23.95 dB	29.31 dB	30.08 dB	24.89 dB
Right	22.24 dB	27.21 dB	28.44 dB	24.22 dB

Table 3
Test results on real stereo images in terms of PSNR for the head area of image sequence "Ludo"

Camera view	Difference between original frames 2 and 1	Difference between frame 2 and its rigid motion estimate	Difference between frame 2 and its aggregate motion estimate	Difference between frame 2 and its estimate with method in [6,7]
Left	22.99 dB	27.54 dB	27.74 dB	23.97 dB
Right	22.24 dB	26.32 dB	28.43 dB	24.22 dB



(a) (b)

Fig. 16. (a), (b) Real data model for the image sequence "Chantal" for time instants 1 and 2, respectively.

(a) (b) (c)

Fig. 17. (a), (b), (c) Original camera images of frame 1 (left, top, right views, respectively).

(a) (b) (c)

Fig. 18. (a), (b), (c) Original camera images of frame 2 (left, top, right views, respectively).

(a) (b) (c)

Fig. 19. (a), (b), (c) Difference between original frames 2 and 1 (left, top, right views, respectively).

(a) (b) (c)

Fig. 20. (a), (b), (c) Rigid motion compensated estimate of frame 2 for a trinocular camera setup (left, top, right views, respectively).

(a) (b) (c)

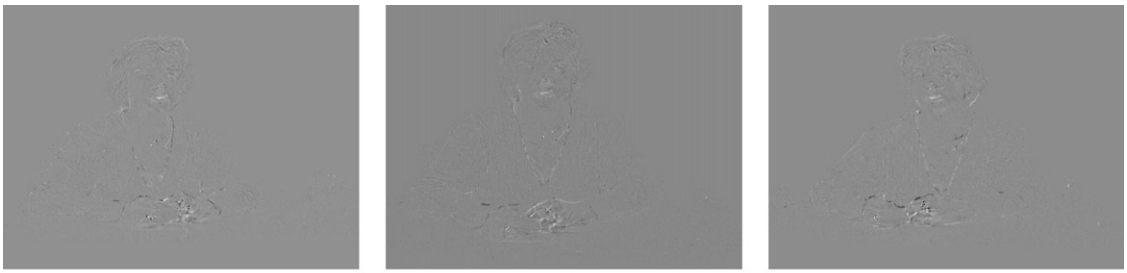
Fig. 21. (a), (b), (c) Displaced frame difference between original frame 2 and its rigid motion compensated estimate for a trinocular camera setup (left, top, right views, respectively).

used to approximate the foreground object at the two consecutive time instants, as shown in Fig. 16. The 3D structure was subdivided into three sub-objects (i.e. head, left arm and body in case of Chantal) and an equivalent 3D motion estimation procedure was followed. Results are shown in Figs. 20–23. The performance of the proposed techniques in terms of PSNR is evaluated by observing Tables 4 and 5.

The performance of the method was compared with the approach described in [6,7]. The performance of these techniques depends on the quality of the output of the Hopfield network that establishes the initial 3D correspondence. In case of realistic image sequence coding experiments, such as coding of the Ludo and Chantal sequences, the Hopfield network converges very slowly to an inaccurate 3D correspondence, affecting

(a) (b) (c)

Fig. 22. (a), (b), (c) Rigid and nonrigid motion compensated estimate of frame 2 for a trinocular camera setup (left, top, right views, respectively).



(a) (b) (c)

Fig. 23. (a), (b), (c) Displaced frame difference between original frame 2 and its rigid and nonrigid motion compensated estimate for a trinocular camera setup (left, top, right views, respectively).

Table 4
Test results on real trinocular images in terms of PSNR for image sequence "Chantal"

Camera view	Difference between original frames 2 and 1	Difference between frame 2 and its rigid motion estimate	Difference between frame 2 and its aggregate motion estimate	Difference between frame 2 and its estimate with method in [6,7]
Left	33.48 dB	34.02 dB	36.74 dB	33.82 dB
Top	33.10 dB	33.76 dB	36.17 dB	33.68 dB
Right	34.76 dB	35.27 dB	37.53 dB	35.01 dB

Table 5
Test results on real stereo images in terms of PSNR for image sequence "Chantal"

Camera view	Difference between original frames 2 and 1	Difference between frame 2 and its rigid motion estimate	Difference between frame 2 and its aggregate motion estimate	Difference between frame 2 and its estimate with method in [6,7]
Left	33.48 dB	33.98 dB	36.02 dB	33.82 dB
Right	34.76 dB	35.15 dB	36.89 dB	35.01 dB

considerably the accuracy of the rigid 3D motion estimation. As shown by the experimental results presented in Tables 2–5 the present scheme, produces considerably more accurate 3D motion estimation.

7. Conclusions

The present paper extended the efficient technique for object-based rigid 3D motion estimation from monoscopic image sequences, described in [29], so as to make it applicable to rigid and non-rigid 3D motion estimation problems in multiview image sequence coding applications. More specifically, a neural network was formed for the estimation of the rigid 3D motion of each object, using initially estimated 2D motion vectors corresponding to each camera view. A technique was further proposed for the estimation of the local non-rigid deformations. Experimental results using stereoscopic and trinocular camera setups have shown that the performance in terms of suppression of measurement noise is improving with the number of camera views used. The proposed technique was compared to the techniques in [6,7] and its performance was found to be significantly better in application on both synthetic and real image sequences.

References

[1] K. Aizawa, H. Harashima, T. Saito, Model-based analysis-synthesis image coding (MBASIC) system for a persons face, Signal Processing: Image Communication 1 (October 1989) 139–152.

[2] K. Aizawa, T.S. Huang, Model-based image coding: advanced video coding techniques for very low bit-rate applications, Proc. IEEE 83 (2) (February 1995) 259–271.

[3] S. Barnard, W. Tompson, Disparity analysis of images, IEEE Trans. Pattern Anal. Machine Intell. 2 (July 1980) 333–340.

[4] K.L. Boyer, A.C. Kak, Structural stereopsis for 3-D vision, IEEE Trans. Pattern Anal. Machine Intell. 10 (March 1988) 144–166.

[5] H. Busch, Subdividing nonrigid 3D objects into quasi rigid parts, in: Proceedings of the IEE Third Internat. Conference on Image Processing Applications, Warwick, UK, 1989.

[6] T. Chen, W.-C. Lin, C.T. Chen, Artificial neural networks for 3D motion analysis – Part-I: rigid motion, IEEE Trans. Neural Networks 6 (6) (November 1995) 1386–1393.

[7] T. Chen, W.-C. Lin, C.T. Chen, Artificial neural networks for 3D motion analysis – Part-II: nonrigid motion, IEEE Trans. Neural Networks 6 (6) (November 1995) 1394–1401.

[8] N. Grammalidis, S. Malassiotis, D. Tzovaras, M.G. Strintzis, Stereo image sequence coding based on 3D motion estimation and compensation, Signal Processing: Image Communication 7 (August 1995) 129–145.

[9] L. Haibo, P. Roivanen, R. Forcheimer, 3D motion estimation in model-based facial image coding, IEEE Trans. Pattern Anal. Machine Intell. 15 (June 1993) 545–555.

[10] B.K.P. Horn, B. Shunck, Robot Vision, MIT Press, Cambridge, MA, 1986.

[11] J. Jaou, N. Duffy, A texture mapping approach to 3D facial image synthesis, Comput. Graphics Forum 7 (1988) 129–134.

[12] F. Kappei, C.E. Liedtke, 3D motion estimation in model-based facial image coding Dept. Elec. Eng. Rep. LiTH-ISY-I-1278 October 1991.

[13] I. Kompatsiaris, D. Tzovaras, M.G. Strintzis, Flexible 3D motion estimation and tracking for multiview image sequence coding, Signal Processing: Image Communication (Special Issue on 3D Video Technology) 14 (1–2) (1998) 95–110.

[14] S. Malassiotis, M.G. Strintzis, Model-based joint motion and structure estimation from stereo images, Comput. Vision Image Understanding 65 (1) (January 1997) 79–94.

[15] H.G. Mussman, M. Hotter, J. Ostermann, Object-oriented analysis-synthesis coding of moving images, Signal Processing: Image Communication 1 (2) (October 1989) 117–138.

[16] N.M. Nasrabadi, C.Y. Choo, Hopfield network for stereo vision correspondence, IEEE Trans. Neural Networks 3 (1) (January 1992) 5–11.

[17] F. Pedersini, A. Sarti, S. Tubaro, Accurate feature detection and matching for the tracking of calibration parameters in multi-camera acquisition systems, Internat. Conference on Image Processing, ICIP-98, 4–7 October 1998, Chicago, IL, USA.

[18] T. Riegel, 3-D shape initialisation, AC092/SIE/DS/R011/b1, EC ACTS PANORAMA Project's Deliverable, August 1997.

[19] Th. Riegel, A. Kaup, Shape initialisation of 3-D objects in videoconference scenes, in: Proceedings of the Stereoscopic Displays and Virtual Reality Systems IV, SPIE, Vol. 3012 San Jose, 11–14 February 1997, pp. 116–124.

[20] T. Riegel, R. Manzotti, F. Pedersini, 3-D shape approximation for objects in multiview image sequences, in: Proceedings of the International Workshop on Synthetic-Natural Hybrid Coding and 3D Imaging (IWSNHC3DI '97), Rhodes, 5–9 September 1997.

[21] L. Robert, R. Deriche, Dense depth map reconstruction using a multiscale regularization approach with discontinuities preserving, in: M.G. Strintzis et al. (Eds.),

Proceedings of the Internat Workshop on Stereoscopic and 3D Imaging, Santorini, Greece, September 1995, pp. 32–39.

[22] R.Y.C. Shah, R.B. Mahani, A new technique to extract range information from stereo images, IEEE Trans. Pattern Anal. Machine Intell. 11 (July 1989) 768–773.

[23] S.S. Sinha, B.G. Schunck, A two-stage algorithm for discontinuity-preserving surface reconstruction, IEEE Trans. PAMI 14 (January 1992).

[24] A. Tamtaoui, C. Labit, Constrained disparity and motion estimators 3DTV image sequence coding, Signal Processing: Image Communication 4 (November 1991) 45–54.

[25] D. Tzovaras, N. Grammalidis, M.G. Strintzis, Object-based coding of stereo image sequences using joint 3D motion/disparity compensation, IEEE Trans. Circuits Systems Video Technol. 7 (2) (April 1997) 312–328.

[26] D. Tzovaras, N. Grammalidis, M.G. Strintzis, S. Malassiotis, Coding for the storage and communication of 3D medical data, Signal Processing: Image Communication 13 (January 1998) 65–87.

[27] D. Tzovaras, I. Kompatsiaris, M.G. Strintzis, 3D object articulation and motion estimation for efficient multiview image sequence coding, in: IEEE ICIP-97, Santa Barbara, CA, USA, October 1997.

[28] D. Tzovaras, I. Kompatsiaris, M.G. Strintzis, 3D object articulation and motion estimation in model-based stereoscopic video-conference image sequence analysis and coding, Signal Processing: Image Communication 14 (4) (1999) 817–840.

[29] D. Tzovaras, N. Ploskas, M.G. Strintzis, Rigid 3D motion estimation using neural networks and initially estimated 2D motion data, IEEE Trans. Circuits Systems Video Technol. 10 (1) (February 2000) 158–166.

[30] D. Tzovaras, S. Vachtsevanos, M.G. Strintzis, Optimization of quadtree segmentation and hybrid 2D and 3D motion estimation in a rate-distortion framework, IEEE Trans. Selected Areas Comm. (Special Issue on Very Low Bit Rate Coding) 15 (9) (December 1997) 1726–1738.

[31] B. Welsh, Model-based coding of images, Ph.D. Dissertation, British Telecom Research Laboratory, January 1991.

[32] Y. Yakimovski, R. Cunningham, A system for extracting 3D measurements from a stereo pair of TV cameras, CVGIP 7 (1978) 195–210.