10

GIBBS MODEL BASED 3D MOTION AND STRUCTURE ESTIMATION FOR OBJECT-BASED VIDEO CODING APPLICATIONS

A. Aydın Alatan and Levent Onural

Electrical and Electronics Engineering Department

Bilkent University, Ankara 06533, Turkey

ABSTRACT

Motion analysis is essential for any video coding scheme. A moving object in a 3D environment can be analyzed better by a 3D motion model instead of 2D models, and better modeling might lead to improved coding efficiency. Gibbs formulated joint segmentation and estimation of 2D motion not only improves the performance of each stage, but also generates robust point correspondences which are necessary for rigid 3D motion estimation algorithms. Estimated rigid 3D motion parameters of a segmented object are used to find the 3D structure of those objects by minimizing another Gibbs energy. Such an approach achieves error immunity compared to linear algorithms. A more general (non-rigid) motion model can also be proposed using Gibbs formulation which permits local elastic interactions in contrast to ultimately tight rigidity between object points. Experimental results are promising for both rigid and non-rigid 3D motion models and put these models forward as strong candidates to be used in object-based coding algorithms.

1 INTRODUCTION

Motion analysis is vital in video processing. Motion information is used successfully in video coding as well as in some others areas such as robot navigation, obstacle avoidance, target tracking, traffic monitoring, motion of biological cells, cloud and weather systems tracking, etc. [1]. In video coding, the temporal redundancy in video sequences can be removed by making predictions between frames using motion information. Hence, the required

H. H. Li et al. (eds.), *Video Data Compression for Multimedia Computing*

number of bits to encode the whole frame or a moving object in one frame is considerably reduced. Such motion-compensation based video encoders achieve high compression which in turn enables them to use low capacity communication channels or require smaller storage.

All the application areas and methods mentioned above require a successful estimate of the motion field, while only 2D projections of the 3D scenes are being observed through some video frames. Currently, there is no method which estimates motion correctly in all complex real world environments; usually, simplifying assumptions for the motion are made. The degree and the nature of these simplifications affect the results significantly. Therefore, models which may be satisfactory for some applications may not be so for others. Obtaining better motion estimates that are applicable to a wider range of real world circumstances is the fundamental issue.

During video recording, many characteristics of a 3D scene are lost due to projection to 2D video frames. An observer may not be able to determine the structure of 3D scenes by only observing their 2D projections. Similarly, any motion in the 3D scene might also be perceived differently by various observes; for example, a rigid body motion in front of a stationary background can be easily observed as a non-rigid deformation of the overall scene. Hence, these inevitable ambiguities create serious problems when the aim is to estimate the "true" motion or structure. However, if the purpose is the best coding of video, then there is no need to find out the true 3D scene; rather, among different possible 3D scenes the one which gives the best coding performance may be chosen. This selection freedom, which is a serious drawback in computer vision applications, can be an advantage in video coding applications. All possible choices of 3D scenes which would give the observed 2D projections also include 2D screen-like objects which may be floating in 3D space. Therefore, a selection over all possible 3D scenes does not exclude the 2D motion. The saturation of the performance of video coding algorithms with 2D motion models hint that a 3D selection is quite likely to yield better results than 2D.

3D motion models can be applied to video coding. In the next sections, after a short survey on current motion estimation methods, a novel 3D object-based video coding method will be presented. Afterwards, a more general Gibbs formulated 3D model, which is applicable not only to rigid, but also to non-rigid motion, will be proposed. These methods will be supported by simulation results and all results will be presented at the end.

2 ESTIMATION OF MOTION IN VIDEO SEQUENCE

Any motion estimation method can be grouped into classes according to the dimension of the environment in which the motion is analyzed. 2D motion is analyzed on the 2D image plane and the analysis is performed only according to the information available on frames without taking into account the "real" motion of the moving objects. 2D motion models are simple, less realistic and are usually utilized for temporal prediction of intensities in video compression. The result of a 2D motion estimation method only reflects the intensity movements in 2D image plane, whereas there is strong evidence that 2D intensity movements and "true" pro jected 3D motion are usually different, except when some special conditions are satisfied [2]. However, for many practical applications, the optic flow and the "true" 2D motion are assumed as similar, without taking the above discussion into account.

3D motion estimation is a more difficult problem to solve since the observation data (i.e., 2D image frames) have one less dimension with respect to the unknown environment to be estimated. This situation leads to unavoidable ambiguities, such as scaling [3]. 3D motion analysis implicitly requires depth estimation for the scene. Modeling 3D motion is more complex compared to its 2D counterpart. 3D motion estimates are usually preferred in applications where the motion information must reflect the "true" motion in the environment, such as in robotics. Recently, 3D scene modeling has become popular in video coding [4, 5, 18].

In this section, 2D and 3D motion estimation methods will be examined briefly. A comparison between these methods will be presented at the end of the section.

2.1 2D Motion Estimation and Compensation

In the 70's, the first algorithms to calculate the motion of an object from television signals were proposed [7]. The estimates were calculated by using some simple frame difference operations. The segmentation of moving objects were also taken into account in some algorithms [8]. However, these methods are far from achieving successful motion estimates in a natural scene. Afterwards, the most important contribution for the estimation of 2D motion, came from Horn and Shunck [9], by the concept of *optic flow*, which relates the 2D motion vectors to spatio-temporal gradients of the image with the

assumption that intensity of a moving point does not change along its motion path. Ill-posedness of this problem has been overcome by imposing smoothness on 2D motion vectors. Later, motion estimation methods found direct applications in digital video compression [10].

The most well-known application of 2D motion estimation is in video coding. motion compensation and motion compensated interpolation are used in almost all sequence coding methods and standards, like H.261,3 and MPEGx [10]. The principal aim of these methods is to encode a frame with a small number of bits, by making good predictions between frames using motion information. In these methods, the "correct" motion vector is the one that minimizes the intensity difference between frames. In video coding algorithms, motion in formation is either transmitted to the receiver side or not. Hence, motion estimation methods which are utilized in video compression algorithms can be classified according to this criterion. In the latter case both the receiver and the transmitter estimate motion at the same time, hence motion is not necessarily sent as an overhead. *Pel-recursive* [11] algorithm is the most well-known example of this class. If motion is estimated only at the transmitter, as in the case of the *block-matching* [12] algorithm, which belongs to the former class, this information should be sent to the receiver side using some extra bits.

The basic idea in pel-recursive algorithms is to use the motion vectors of the causal neighbors for compensating the intensity of the current pixel and update the value of the motion vector of the current pixel by using the neighboring motion vectors and intensity values. Hence, this algorithm does not need to transmit motion information to the receiver. Intensity prediction (motion compensation) errors are transmitted, instead. In block-matching motion estimation algorithms, the intensity values of the current block is matched with intensities of another block inside a search window on the previous frame. For coding purposes this motion information must be transmitted as an overhead. It should be noted that in block-matching, 2D motion is modeled only as a translation. A hierarchical version of the block-matching method, which has distinct advantages over the former one, is also proposed [13]. There are also some generalized block-matching algorithms [14] which make matches between non-rectangular regions, and these methods are known as 2D active meshes [4]. The performance of video coding algorithms based on block-matching is definitely superior to methods which utilize pel-recursive methods. Hence, while block-matching methods currently exist in many standards [10], pel-recursive methods have lost their popularity.

A powerful 2D motion modeling is achieved using Markov Random Fields (MRF). These approaches model 2D motion in such a way that the 2D projection of 3D rigid or even non-rigid motions can be represented by the help of some local interactions between neighboring motion vectors. Block-matching can also be assumed as an MRF based motion estimation method with very tight local relations (all pixels move the same way) inside a rectangular region. However, by permitting looser interactions between neighbors, a wider class of motions can be modeled and estimated.

MRF modeling of 2D motion vector fields is first proposed as a minimization problem, and a solution to the problem is given as a VLSI implementation of analog binary resistive networks [15]. In fact, the minimized function is equal to the energy function of a Gibbs distribution which gives the probability distribution of random variables with Markovian properties [16]. Later, some extensions of this work were proposed by adding extra fields which make the segmentation of the scene easier [17]. The concept of line field [16], which is used to detect discontinuities during image restoration is also applied to motion fields [18, 19, 17]. There is also a high-level "boundary patterns" concept [20], which generalizes the local low-level line field concept by using more global interactions. Some researchers use another extra field instead of the line elements and call it the segmentation field [21]. This segmentation field basically divides the dense motion parameters into some regions, instead of defining contours. The occlusion (i.e., temporally unpredictable) areas, which are newly exposed or covered by the moving object, can also be detected together with motion estimation [22, 23, 21]. The recent results show that even if the image is corrupted with signal dependent [24] or Gaussian noise with some blur [25], Markov modeled motion estimation methods achieve good performances.

Apart from dedicated hardware [15], the minimization of a Gibbs energy function can be achieved by using some popular global optimization routines, such as Simulated Annealing (SA) [26], Iterated Conditional Modes (ICM) [27] or Multiscale Constrained Relaxation (MCR) [28]. SA guarantees to reach optimum with long iterations, whereas ICM gives good results with good initial estimates. On the other hand, MCR obtains comparable results with SA using ICM at each scale while propagating the minimization results between levels. Moreover, MCR does not need any initial estimate (as ICM) at its coarsest level.

Hence, the results of the previous MRF based algorithms show that Gibbs formulation gives successful results not only for 2D motion estimation, but also for segmentation, occlusion detection and noise immunity. However, the

computational burden of these algorithms in comparison to other approaches remains to be a problem to be solved. A more detailed survey on 2D motion estimation methods can be found in [29, 30, 31, 32, 10].

2.2 3D Motion Estimation Methods

Up to this point, the 2D motion of objects on the image plane is examined. Two basic assumptions in 2D motion models are the intensity matching between consecutive frames and the relations (strong or weak) between neighboring motion vectors. Since objects make their movements in the 3D world, 3D motion modeling might be more suitable in various situations. Even if the 2D modeling is successful, the description of this 2D motion field might be inefficient from the compression point of view (compared to the description of the 3D field).

In most of the current methods, 3D motion is usually modeled for rigid objects [33, 34, 35, 3]. There are also some approaches to model and estimate non-rigid motion; they are examined in detail in the next section [36, 37, 38, 39]. Rigidity can be physically explained as the equality of the 3D distance between object points before and after the motion. In dynamics, it is known that 3D motion of any *rigid* body can be expressed by a rotation and a translation, and the parameters of both rotation and translation are constant at each point of the rigid object [29].

Before going into details of the 3D motion estimation methods, the projection of the 3D world into the 2D image plane should be explained. The 3D environment is usually mapped into the image plane by one of the two projections, which are *orthographic* and *perspective* [40]. Perspective projection is more realistic, since it models a pin-hole camera (Fig. 1).

The methods which estimate 3D motion from consecutive monocular frames can be divided into two major classes, as *direct* and *correspondence based* methods. Direct methods [41, 42, 43] use spatio-temporal gradients in the image to find a solution to the 3D motion estimation problem. The rigid 3D motion relation (eq. (10.2)) is inserted into the famous optic flow equation [9] after perspective projection. Therefore, the unknown 3D motion parameters become related to the spatio-temporal image gradients. Currently, there is no general solution to direct methods; some simplifying assumptions are made about motion and/or structure [42]. Since the performance of the results depends on the accuracy of the gradients available on the discrete image, some

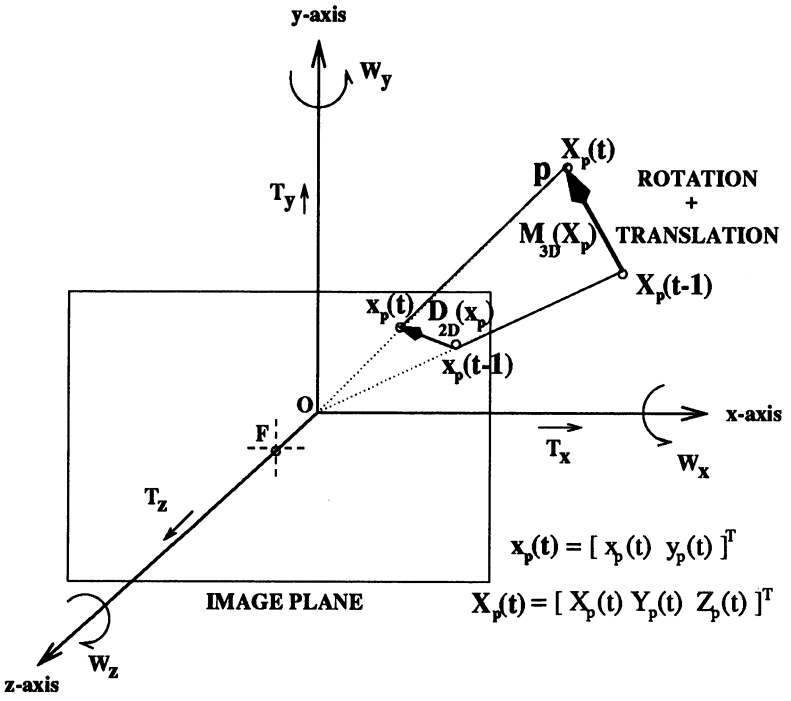

Figure 1 3D coordinate system.

improvements are proposed on how to estimate the differentials [42]. However, the difficulty with robust gradient calculation and their susceptibility to noise, make not only the direct methods, but also their 2D counterparts (pel-recursive algorithms [11]), less attractive compared to other approaches.

Any 3D motion estimation method which requires some dense or sparse set of 2D motion vectors for the estimation of 3D motion is said to be correspondence based. These methods require some matches between frames,

which are obtained by one of the previously explained 2D motion estimation methods. Since incorrect correspondence estimates may lead to unstable solutions, the performance of any correspondence based method depends on this initial matching step. Therefore, matches between some features, like corners or edges, are found in order to obtain immunity to errors [44]. There are also applications which estimate 3D motion vectors from line matches [45, 46]. Although most of the work in the literature is directed to finding the minimum number of correspondences to obtain a unique 3D motion and structure [33, 34, 35, 3], it is shown that even with infinite number of correspondences, some special hyperboloid surfaces will definitely have more than one solution as their motion [47].

Correspondence based methods with linear solutions have the advantage of yielding fast solutions compared to the nonlinear counterparts, but they are less immune to errors. The estimation of 3D motion for a planar patch [33] and any curved surface [3] can both be solved linearly. The concept of "pure" parameters, which relate the 2D coordinates of points on a rigid planar patch in two consecutive frames by eight parameters, is proposed in [33]. The pure parameters are very popular, since these parameters easily model the motion of a small planar patch and objects can be assumed to be made of small planar patches [48, 49, 50]. In [3], a solution to 3D motion estimation problem without any planarity constraint is proposed. By modeling the motion as in eq. (10.2), an "Essential" matrix, $\mathbf{E}$, which relates the 3D motion parameters with image plane coordinates linearly before and after the motion, is defined. This yields a least-squares solution of the 3D motion parameters. The *E-matrix* method is still one of the most popular 3D motion estimation methods. The noise susceptibility of this algorithm is later improved by a nonlinear robust version [51].

In recent years, 3D motion analysis has been used in video coding applications [49, 52, 18, 46]. Most of these methods have the assumption of rigidity, except those which have a generic wireframe model for human head [18]. The non-rigid facial actions are modeled in that wireframe-based method. However, methods based on rigidity are usually far from representing the general solutions. They either perform only object segmentation with 3D data [49],[53] or estimate the global (camera) motion and depth field using a dense depth field as an initial estimate [46]. In [49], the segmentation of the scene is achieved for the stationary and moving parts by the help of frame differences and pure parameters [33, 54], and different regions are coded in an appropriate way. For some other methods, no segmentation is performed and long sequences are used to estimate incremental 3D motion and sparse depth fields, which are interpolated afterwards [52]. Recently, a 3D object-based

motion and depth estimation method without any significant constraints, except rigidity, is proposed [55].

A more detailed survey on 3D motion and structure estimation methods can be found in [1, 56, 57, 40, 10].

2.3 Comparison of Motion Models

The 2D motion estimation methods are preferred in many video coding applications, since they are simple and their compression performance is sufficient for some bit-rates. However, it is also known that they do not represent the projection of a 3D object motion correctly, except when some special conditions are satisfied [2]. Even if the estimated 2D motion field is similar to the "true" 2D motion, the description and hence the compression of such a motion field is inefficient, when the real motion in the 3D scene is rigid with some amount of rotation. Therefore, the performance of the 2D motion estimation methods is definitely limited and currently it is observed to be saturated.

On the other hand, 3D motion estimation methods are difficult to cope with, but they have a promising future. For very-low bit rate coding, the performance of the existing 2D motion estimation based methods are saturated and some new approaches which try to model the 3D scene by some a priori knowledge has emerged. Although such 3D methods are not mature yet, they might lead to solutions that yield the lowest bit rate. Since 3D motion models provide the "simplest" way to describe any physical rigid motion with only 6 parameters, they are very efficient. Assuming that these 6 parameters can be encoded with few bits, the compression performance of methods with 3D rigid motion models strictly depends on the efficiency of the encoding of the depth field [58].

However, another approach might be the joint utilization of 2D and 3D motion models, rather than the use of only one of them. Since each one has advantages and disadvantages depending on the application, it will be better to model various regions in an image with different (2D or 3D) motion models according to their local performances. For example, rigid 3D motion models do not successfully represent any non-rigid or incorrectly articulated motions, whereas 2D motion models will still survive in these cases. On the other hand, 2D motion models can not describe any 3D motion more efficiently than a 3D

model. Hence, adaptive motion model selection may improve the performance of any system with some increase in the complexity of the algorithm.

3 AN OBJECT-BASED RIGID 3D MOTION ANALYSIS

The current trend in very low bit-rate coding applications is changing from motion compensated-DCT type algorithms, like MPEG1-2, to object-based methods [59]. 2D motion models are used in many of the current object-based algorithms, although such motion models have limited performance due to lack of representation of 3D world dynamics. Although rarely, 3D motion models are also used in video compression systems [49, 52, 18, 46]. Model-based methods [4],[5],[18] are some of the few applications in which the objects and their motions are represented with some highly constrained 3D wireframe models.

In this section, a novel 3D rigid object-based video coding method is presented. Some simulation results are also given at the end of the section.

3.1 Joint Segmentation and 2D Motion Estimation

In order to perform object-based analysis in a video sequence, motion based segmentation should be performed to determine the regions with intensity and motion coherence, i.e., the objects. It can be easily deduced that motion estimation is necessary for motion based segmentation, and vice versa. For example, in order not to have blurred motion boundaries, the motion discontinuities should be located, whereas to determine the correct boundaries, successful motion estimation results are needed at each side of these boundaries. Therefore, the methods which perform motion estimation and object segmentation steps individually have limited performance [50]. These two steps can be combined using Gibbs formulation.

The Gibbs energy function, $\mathcal{U}$, of 2D motion $\mathcal{D}$, *segmentation* $\mathcal{R}$ and *temporally unpredictable* $\mathcal{S}$ fields, which are all defined at each point of grid, Λ, can be written as [55]

$$\mathcal{U}(\mathcal{D},\mathcal{R},\mathcal{S} \mid \mathcal{I}_t,\mathcal{I}_{t-1}) = \mathcal{U}_n + \lambda_m \mathcal{U}_m + \lambda_R \mathcal{U}_R + \lambda_s \mathcal{U}_s \quad , \tag{10.1}$$

where

$$\mathcal{U}_n = \sum_{\mathbf{x}\in\Lambda} \left(I_t(\mathbf{x}) - I_{t-1}\left(\mathbf{x} - \mathbf{D}(\mathbf{x})\right)\right)^2 \left(1 - S(\mathbf{x})\right) + S(\mathbf{x})T_s \quad ,$$

$$\mathcal{U}_m = \sum_{\mathbf{x}\in\Lambda} \sum_{\mathbf{x}_c\in\eta_{\mathbf{X}}} ||\mathbf{D}(\mathbf{x}) - \mathbf{D}(\mathbf{x}_c)||^2 \quad \delta\left(R(\mathbf{x}) - R(\mathbf{x}_c)\right) \quad ,$$

$$\mathcal{U}_R = \sum_{\mathbf{x}\in\Lambda} \sum_{\mathbf{x}_c\in\eta_{\mathbf{X}}} \left[1 - \delta\left(R(\mathbf{x}) - R(\mathbf{x}_c)\right)\right] + \lambda_t \frac{\left[1 - \delta(R(\mathbf{x}) - R(\mathbf{x}_c))\right]}{1 + \left(I_t(\mathbf{x}) - I_t(\mathbf{x}_c)\right)^2} + \theta\left(R(\mathbf{x})\right) \quad ,$$

$$\mathcal{U}_s = \sum_{\mathbf{x}\in\Lambda} \sum_{\mathbf{x}_c\in\eta_{\mathbf{X}}} \left[1 - \delta\left(S(\mathbf{x}) - S(\mathbf{x}_c)\right)\right] \quad .$$

The intensities of each frame define an intensity field $\mathcal{I}$ on the 2D grid Λ. In the above equation, $I_t(\mathbf{x})$ defines an intensity value at $\mathbf{x}$ coordinate of frame $I_t \in \mathcal{I}$ at time t. $\mathbf{x}$ is an element of Λ and $\mathbf{x}_c$ is in the neighborhood of $\mathbf{x}$, denoted by $\eta_{\mathbf{x}}$. At each point on Λ, the unknown 2D motion vectors, $\mathbf{D}(\mathbf{x})$, are defined to constitute the 2D motion vector field $\mathcal{D}$, while similar relations are also valid between $\mathcal{R}$ and $R(\mathbf{x})$, and $\mathcal{S}$ and $S(\mathbf{x})$. $\mathbf{D}(\mathbf{x})$ shows the displacement from a corresponding point on frame I_{t-1} to $\mathbf{x}$ on I_t. If it is necessary, a subscript is used to denote that the vector field is 2D and such a vector, $\mathbf{D}_{2D}(\mathbf{x})$, can be found in Fig. 1. The $\mathcal{U}_n$ term in $\mathcal{U}$ supports intensity matching between frames using "true" 2D motion vectors and the $\mathcal{U}_m$ term tries to obtain similar values between neighboring motion vectors when they belong to the same object. The segmentation of objects in the scene are achieved using the $\mathcal{R}$ field which prevents $\mathcal{U}_m$ from getting a high penalty at motion boundaries. Since it is more expected to observe broad and connected regions which are the 2D projections of the object surfaces, the $\mathcal{U}_R$ term supports this reasoning by penalizing different neighboring regions. Textural coherence is also supported by giving a penalty to neighboring pixels with similar intensity values if they do not belong to the same region. $\theta(R(\mathbf{x}))$ term is designed to reject a few *taboo patterns*, such as single-point or small cross-shaped regions. Lastly, the $\mathcal{U}_s$ term supports the $\mathcal{S}$ field to consist of regions, instead of individual points and $\mathcal{S}$ is a binary field showing the *temporally unpredictable* regions, in which the motion compensation error is expected to be greater than a threshold, T_s. Such regions usually correspond to occluding or motion model failure regions. Similar energy functions can be found in [21],[60], [50], [55].

By minimizing the energy function $\mathcal{U}$, *MAP* estimates of the unknown 2D motion, segmentation fields and temporally unpredictable (TU) regions can be obtained at the same time. Hence, the scene is segmented into moving objects, while their 2D motion vectors are estimated.

3.2 Estimation of Rigid 3D Object Motion

As it is previously stated, there are many different approaches to the 3D motion and structure estimation problem [1]. Since direct methods have unavoidable drawbacks, correspondence based methods look as better candidates for the 3D motion estimation applications. The linear *E-matrix* approach [3] is the most popular among all due to its computational simplicity. In our algorithm, improved E-matrix approach [51], which is called *epipolar improvement*, is utilized to estimate the 3D motion parameters. This method is applied to each object individually, using the segmented 2D motion vectors which are obtained by the algorithm given in the previous section.

The required 2D correspondences are selected from dense 2D motion estimates which are obtained by minimizing eq. (10.1). The selection process is achieved by choosing object points which have low local energy defined by eq. (10.1) and high spatial gradient, like edges and corners. By simply thresholding of local energy and image gradient, some sparse set of "trusted" estimates are obtained from dense set of 2D motion vectors which contain many "outliers". It should be noted that in order to jointly estimate motion and segmentation fields, rather than choosing some feature points and making motion estimation only at these sparse points, motion vector selection after dense estimation is followed. Elimination of outliers is achieved by also discarding the temporally unpredictable points of the $\mathcal{S}$ field. However, using only a sparse set of vectors prevents one from finding a dense depth field using E-matrix method. A noise immune method to find the dense depth field using the estimated 3D motion parameters and input frames, is presented in the next section.

In order to define the 3D motion of an object point, first the object and its observable surface points should be defined. Let $\mathbf{P}$ define an object in the 3D object space and let $\mathbf{p} \in \mathbf{P}$ be an object point whose 3D coordinates at time t is given by $\mathbf{X_p}(t) = [X_\mathbf{p}(t)\ Y_\mathbf{p}(t)\ Z_\mathbf{p}(t)]^T$. After the perspective projection, the point $\mathbf{p}$ is observed on the image plane with the coordinates $\mathbf{x_p}(t) = [x_\mathbf{p}(t)\ y_\mathbf{p}(t)]^T$, and this is shown in Fig. 1. If the object $\mathbf{P}$ undergoes a rigid motion, then the relation between the 3D coordinates of point $\mathbf{p}$ before and after the motion is

$$\mathbf{X_p}(t-1) = \mathbf{R}\,\mathbf{X_p}(t) + \mathbf{T} \quad , \tag{10.2}$$

where $\mathbf{R}$ is a 3x3 rotation matrix and $\mathbf{T}$ is a 3x1 translation vector. For coding purposes, rather than using the "real" motion from $t-1$ to t, an "inverse" motion from time t to $t-1$ is chosen, and hence $\mathbf{R}$ and $\mathbf{T}$ represent this reverse motion. Without loss of generality, the focal length in the

perspective mapping is chosen to be the unit length and the relations between the displacements on the 2D image plane and 3D motion parameters are obtained as [3]

$$\begin{aligned} x_{\mathbf{p}}(t-1) &= \frac{r_{11}\cdot x_{\mathbf{p}}(t)+r_{12}\cdot y_{\mathbf{p}}(t)+r_{13}+\frac{T_x}{Z_{\mathbf{p}}(\mathbf{x}_{\mathbf{p}},t)}}{r_{31}\cdot x_{\mathbf{p}}(t)+r_{32}\cdot y_{\mathbf{p}}(t)+r_{33}+\frac{T_z}{Z_{\mathbf{p}}(\mathbf{x}_{\mathbf{p}},t)}} \quad , \\ y_{\mathbf{p}}(t-1) &= \frac{r_{21}\cdot x_{\mathbf{p}}(t)+r_{22}\cdot y_{\mathbf{p}}(t)+r_{23}+\frac{T_y}{Z_{\mathbf{p}}(\mathbf{x}_{\mathbf{p}},t)}}{r_{31}\cdot x_{\mathbf{p}}(t)+r_{32}\cdot y_{\mathbf{p}}(t)+r_{33}+\frac{T_z}{Z_{\mathbf{p}}(\mathbf{x}_{\mathbf{p}},t)}} \quad . \end{aligned} \tag{10.3}$$

$\mathbf{x}_{\mathbf{p}}(t-1) = [x_{\mathbf{p}}(t-1)\ \ y_{\mathbf{p}}(t-1)]^T$ is the projected 2D coordinates of the object point $\mathbf{p}$ at time $t-1$. $Z_{\mathbf{p}}(\mathbf{x}_{\mathbf{p}}, t)$ is the third component of the vector $\mathbf{X}_{\mathbf{p}}(t)$, simply called the "depth value". The set of depth values which are defined at each point on the 2D lattice Λ, is called the "depth field". After some manipulations and dropping depth field term from eq. (10.3) [3], the relation

$$\mathbf{U}'\mathbf{E}\mathbf{U} = 0 \tag{10.4}$$

is obtained. In this equation, the vector terms are equal to $\mathbf{U} = [x_{\mathbf{p}}(t)\ \ y_{\mathbf{p}}(t)\ \ 1]^T$, $\mathbf{U}' = [x_{\mathbf{p}}(t-1)\ \ y_{\mathbf{p}}(t-1)\ \ 1]$, and the unknown $\mathbf{E}$ matrix is equal to

$$\mathbf{E} = \begin{bmatrix} 0 & T_z & -T_y \\ -T_z & 0 & T_x \\ T_y & -T_x & 0 \end{bmatrix} \begin{bmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{32} & r_{33} \end{bmatrix} \quad , \tag{10.5}$$

where $T_{x,y,z}$'s are the elements of the translation vector $\mathbf{T}$ and r_{ij}'s are the elements of $\mathbf{R}$.

Using "trusted" 2D motion vectors, the $\mathbf{E}$ matrix in eq. (10.4) is solved in the least-squares sense. Then the estimated $\mathbf{E}$ matrix is decomposed into a rotation matrix $\mathbf{R}$ and a translation vector $\mathbf{T}$, analytically [3]. Since these estimates are very sensitive to noise and quantization errors of 2D motion vectors, a nonlinear function is used to improve the obtained 3D motion parameters which are obtained after the linear algorithm. The method which uses this nonlinear function is called *epipolar improvement* and the function to be minimized is written as [51]

$$\sum_i \frac{\mathbf{U}'_i\mathbf{E}\mathbf{U}_i}{\sigma_i^2} \quad , \tag{10.6}$$

where $\mathbf{U}_i$ is as in eq. (10.4) for ith trusted 2D motion vector coordinate. σ_i^2 is the variance of error for $\mathbf{U}'_i\mathbf{E}\mathbf{U}_i$ [51]. This function can be minimized using

Levenberg-Marquardt algorithm, and minimizing eq. (10.6) while using the initial estimates obtained from the linear algorithm, a substantial improvement against noise can be obtained with some increase in the complexity of the overall algorithm.

3.3 Noise Immune Depth Estimation

Using the estimated 3D and 2D motion values, a depth field can be estimated in the least-squares using eq. (10.3). However, this can be achieved only at points whose 2D correspondences are trustable. Hence, at the end of such an estimation, the obtained depth field will be sparse. Therefore, another strategy must be followed to find a dense depth field.

Since estimation of depth is usually achieved after 3D motion parameters are obtained [51, 42], all the a priori error and noise before this step will affect the depth estimation process. The susceptibility of 3D motion estimation algorithms, requires the depth estimation algorithm not only to produce dense values, but also to be immune to noise. Apart from these two requirements, estimation should be applied for each object separately rather than globally.

Hence, finding the depth field can be formulated as an estimation problem, as it is shown in Fig. 2. Using the "true" (error-free) intensity, $\mathcal{I}_{t,t-1}$, and 3D motion, $\mathcal{M}$, fields, it is possible to obtain the $\mathcal{Z}$ field, exactly. However, when these parameters are observed with some noise, there is no longer an exact relation. Hence, $\mathcal{Z}$ field should be estimated by taking the noise into account. Using the observed noise contaminated consecutive intensity fields, $\tilde{\mathcal{I}}_{t,t-1}$, and the observed 3D motion field, $\tilde{\mathcal{M}}$, which may also contain some error due to 2D and 3D motion estimation steps, the maximum a posteriori (MAP) estimate of the depth field can be found by maximizing the conditional probability distribution:

$$\max_{\mathcal{Z}}\{P(\mathcal{Z}|\tilde{\mathcal{M}},\tilde{\mathcal{R}},\tilde{\mathcal{I}}_t,\tilde{\mathcal{I}}_{t-1})\} \quad . \tag{10.7}$$

If this distribution can be written as a Gibbs distribution, the estimate will be equal to

$$\hat{\mathcal{Z}} = \arg\{\min_{\mathcal{Z}} \mathcal{U}(\mathcal{Z} \mid \tilde{\mathcal{M}},\tilde{\mathcal{R}},\tilde{\mathcal{I}}_t,\tilde{\mathcal{I}}_{t-1})\} \quad . \tag{10.8}$$

Using a priori information and the observations, the Gibbs energy function can be written as

$$\mathcal{U}(\mathcal{Z} \mid \tilde{\mathcal{M}},\tilde{\mathcal{R}},\tilde{\mathcal{I}}_t,\tilde{\mathcal{I}}_{t-1}) = \mathcal{U}_{\mathcal{Z}} + \lambda_{\mathcal{Z}}\mathcal{U}_{\mathcal{Z}} \quad , \tag{10.9}$$

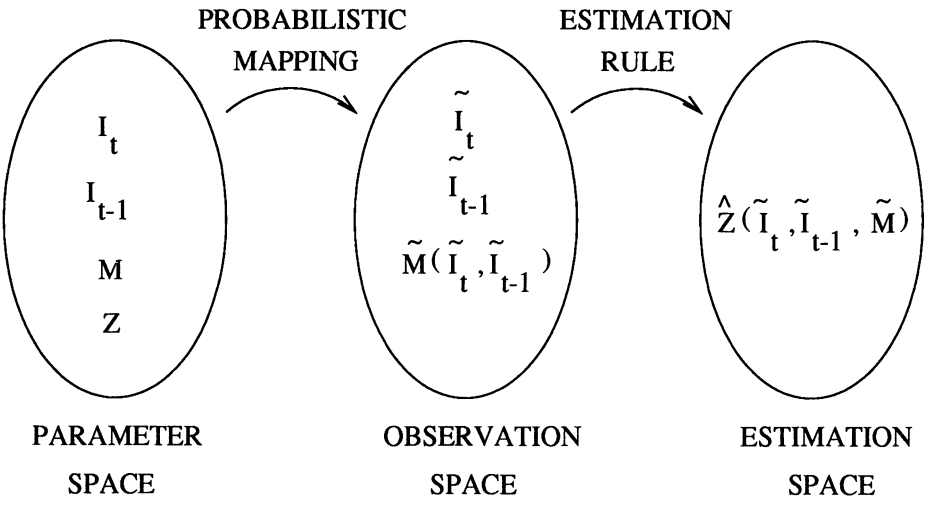

Figure 2 Depth estimation using MAP formulation.

where

$$\mathcal{U}_Z = \sum_{\mathbf{x_p} \in \Lambda} \left(\tilde{I}_t(\mathbf{x_p}(t)) - \tilde{I}_{t-1}(\mathbf{x_p}(t-1)) \right)^2 \quad ,(10.10)$$

$$\mathcal{U}_Z = \sum_{\mathbf{x_p} \in \Lambda} \sum_{\mathbf{x_{p,c}} \in \eta_{\mathbf{x_p}}} (Z_{\mathbf{p}}(\mathbf{x_p}, t) - Z_{\mathbf{p}}(\mathbf{x_{p,c}}, t))^2 \cdot \delta \left(\tilde{R}(\mathbf{x_p}) - \tilde{R}(\mathbf{x_{p,c}}) \right) \quad (10.11)$$

In the above equation, $\mathbf{x_p}(t-1)$ is the previous projected 2D coordinate of the object point, $\mathbf{p}$, at $\mathbf{x_p}(t) = (x_{\mathbf{p}}(t), y_{\mathbf{p}}(t))$ which are related by eq. (10.3). $\mathbf{x_{p,c}}(t)$ is the neighbor coordinate of $\mathbf{x_p}(t)$ defined in $\eta_{\mathbf{x_p}}$.

The $\mathcal{U}_Z$ term in eq. (10.11) is the a priori information about the depth field $\mathcal{Z}$. This function supports the experience that it is more likely to have neighboring points of an object to have similar depths. Obviously the smooth

variation of depth field is not valid along object boundaries which were segmented previously by the $\mathcal{R}$ field.

Using the estimated dense 2D motion vectors, $Z_{\mathbf{p}}(\mathbf{x_p}, t)$ values at each point can also be found linearly by solving eq. (10.3) independently at each location. However, such an attempt might result in degraded results since the performance of this estimation is susceptible to both 2D and 3D motion parameter errors and there might be some "untrustable" estimates among the dense 2D motion vectors.

The MAP estimate $\hat{\mathcal{Z}}$ is a dense depth field, consisting of $\hat{Z}(\mathbf{x_p}, t)$ defined at each point on the image. Hence, the intensity of all points can be motion compensated (i.e., predicted by the 3D motion parameters and the depth value at that point using eq. (10.3)) from the previous reconstructed frame at the receiver, if the 3D motion parameters and dense depth field are transmitted for each object. A 3D object-based motion and depth estimation method can be proposed using the methods explained up to this point and the corresponding flowchart is shown in Fig. 3.

Although, the encoding of the parameters are beyond the scope of this document, the efficient representation of 3D rigid motion parameters leaves only the depth field as an encoding problem to solve [58].

3.4 Simulation Results

Some simulations are carried out to test the validity of the 3D rigid object-based motion and depth estimation algorithm. Consecutive two frames (frame 100 and 103) from a standard sequence *Foreman* are used to determine the motion in between. These two frames which are QCIF sized (176×144) are shown in Fig. 4.

In these frames, apart from a very small camera movement, a head motion is observed. Since an estimate is necessary for the focal length of the camera which has recorded this sequence, it is assumed that this parameter is roughly equal to $250 units$ that corresponds to $50mm$ focal length of a $35mm$ camera. The results are still acceptable with this assumption. However, they will definitely be improved by using a calibrated camera. The center of projection is also assumed to pass through the center of each frame.

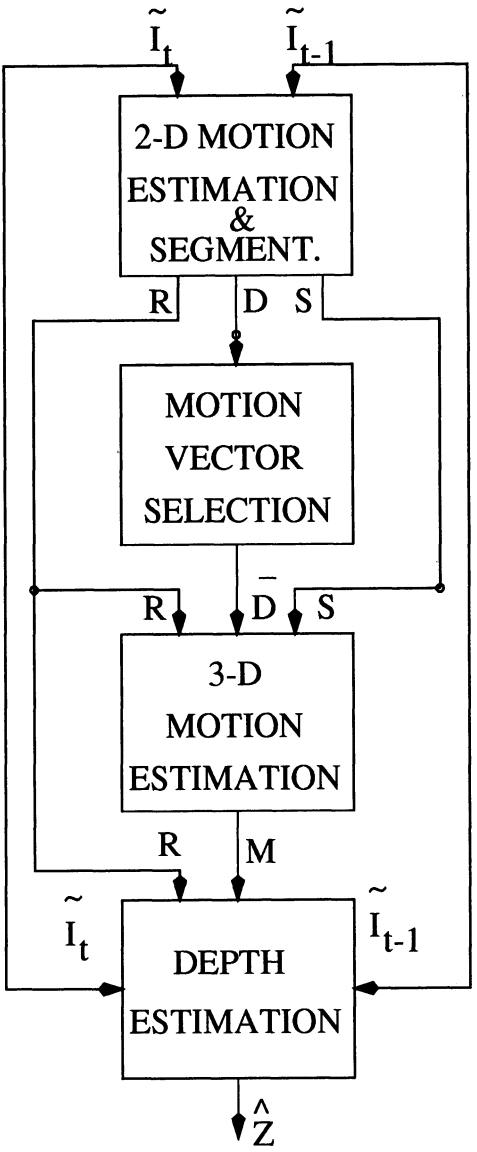

Figure 3 The proposed rigid 3D object-based motion and depth estimation scheme which can be used in object-based video coding.

The results of the 2D motion estimation is shown in Fig. 5. The minimization of eq. (10.1) is achieved using MCR algorithm [28] of 4 scales with 2 iterations

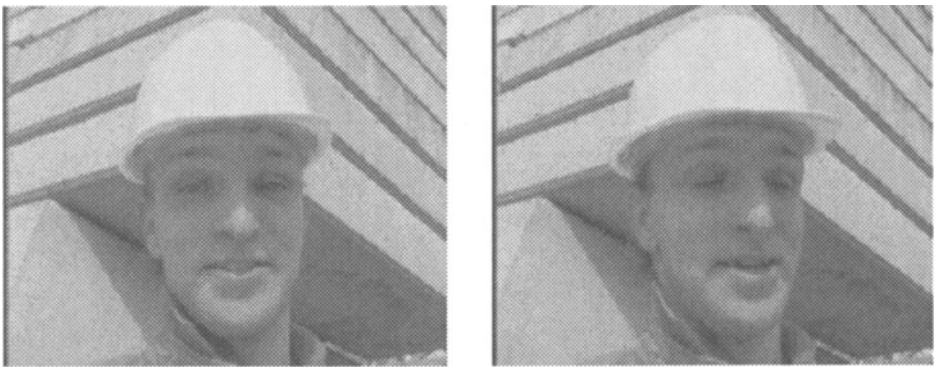

Figure 4 Original (a)100th and (b)103rd frames of "Foreman" sequence.

of ICM at each scale. The segmentation of the moving body is also shown in Fig. 5. A region based segmentation algorithm [61] is used to give an initial estimate for the segmentation field to the MCR algorithm in order to improve the segmentation performance. Although the hat of foreman is also moving with the head, the motion of this hat can not be determined because of its uniform surface and hence it is classified rather as part of the stationary background. The motion compensated frame has an SNR_{peak} equal to $38dB$.

The dense 2D motion vectors can not be trusted at each point and the "outliers" must be rejected by simply thresholding the local energies and image gradients. The motion vectors whose local energy is less than 5 and image gradient is greater that 250 is accepted as "trusted".

Using the trusted 2D correspondences (approximately 150 among 4000 object points), the E-matrix method is solved in least squares sense. After epipolar

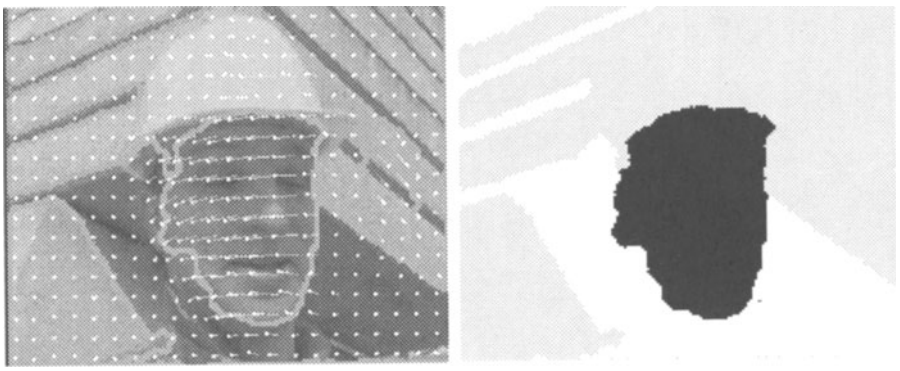

Figure 5 The experimental results of 2D motion analysis and segmentation for "Foreman" sequence; (a) The needlegram of the 2D motion estimates, (b) Segmentation field.

improvement, the rotation matrix and translation vector is found as

$$R = \begin{bmatrix} 0.9993 & 0.0242 & 0.0251 \\ -0.0242 & 0.9997 & 0.0003 \\ -0.0251 & -0.0003 & 0.9996 \end{bmatrix}, \quad T = \begin{bmatrix} -0.0117 \\ 0.5585 \\ 0.8293 \end{bmatrix} .$$

Although $\mathbf{T}$ is found as a unit vector because of the unavoidable *scaling ambiguity* [51] between the translation vector and the depth field, the estimated depth field still can be found using this unit vector. The eigenvalues of $E^T E$ matrix are [1.075 0.925 0], which theoretically should be [1.0 1.0 0] [57]. Hence, this result shows the existence of some error on 3D motion parameters, but the level of error is acceptable.

In order to find the *MAP* estimate of the depth field, eq. (10.9) is minimized by MCR method again, while $\mathbf{R}$ and $\mathbf{T}$ are used as input parameters. In Fig. 6, the reconstructed current frame, which is obtained using the estimated 3D motion parameters, previous frame and the estimated depth field, is shown for the arbitrarily chosen constant $\lambda_Z = 5$. The TU areas are

segmented by using eq. (10.1) ($\mathbf{D}$ is replaced with $\mathbf{D}_{2D}$). As it is observed from the depth field in Fig. 6, the location of the nose and the flatness of the rest of the face is found correctly.

4 A NON-RIGID GIBBS 3D MOTION MODEL

Most of the existing 3D motion estimation methods have some drawbacks. These drawbacks can be summarized as follows:

- Structural constraints, like rigidity, planarity [51, 42, 48, 62, 63];
- Errors due to discrete differentiation [42];
- Lack of segmentation of the scene [51, 42, 62, 63];
- Use of orthographic projection [62, 63];
- Susceptibility to noise [51]

In most of the motion analysis applications, 3D motion is usually assumed to be rigid. However, there are some situations where the object moves non-rigidly, such as movement of lips or heart. In such situations, the existing rigid 3D motion estimation methods are insufficient. In order to cope with such problems, non-rigid motion might be examined from a kinematic point of view and afterwards an appropriate model should be proposed.

There are some methods which track the non-rigid motion and estimate (update) the non-rigid shape. However, most of the methods assume that the initial shape is available [36, 37]. Some other approaches approximate the object surfaces with some constrained shape models, such as superquadratics [38, 39] or wireframes [4, 5], and these models are fitted to projected non-rigid objects on the image using some extra sensor data or hand. These models are then used to track both shape and motion.

A new approach which attempts to eliminate the drawbacks explained above is proposed in this paper. The basic idea is to formulate the problem in such

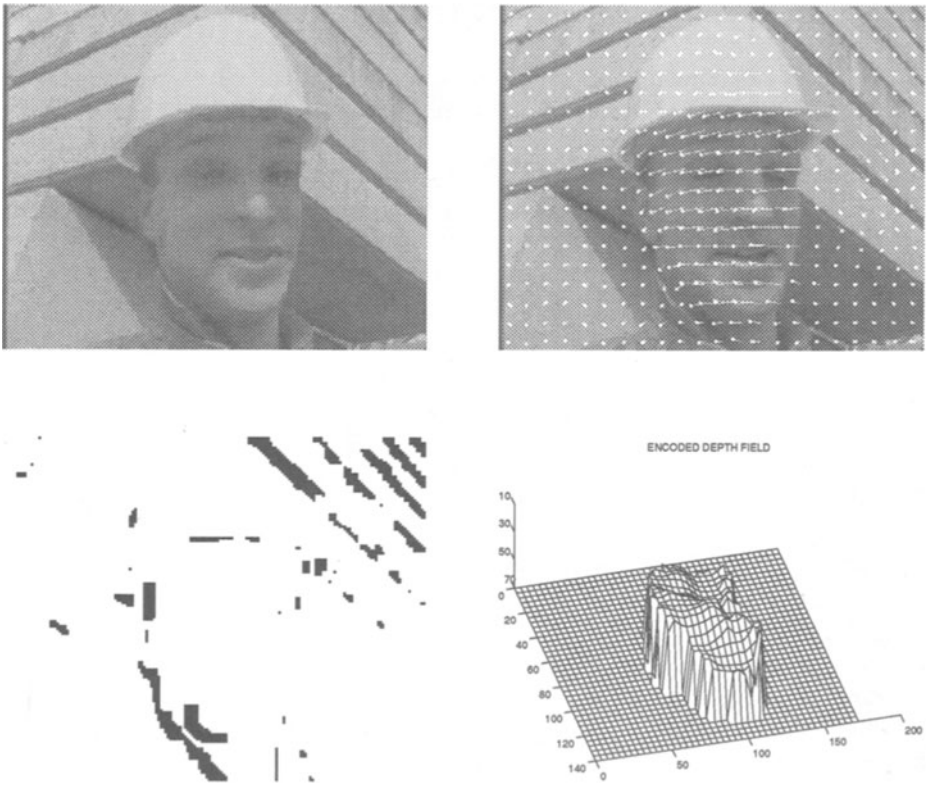

Figure 6 The results of 3D motion and depth estimation for "Foreman" sequence: (a) Motion compensated current frame using 3D motion parameters and encoded depth field; (b) The projection of 3D motion using "needlegram" representation; (c) Temporally Unpredictable areas (in black) for 3D motion and (d) Mesh representations of the depth field.

a way that, all the *a priori* information about motion can be inserted into a

cost function. This cost function is the energy function of a Gibbs probability density function for the 3D motion parameters.

A successful probabilistic model must be based on a thorough analysis of the non-rigid motion. Such an analysis is presented in the next section.

4.1 Formulation of Non-Rigid Motion

According to the fundamental theorem of kinematics the most general motion of a *sufficiently small* element of a deformable (i.e. non-rigid) body can be represented as the sum of [64]:

1. a translation,
2. a rotation,
3. an extension (contraction) in 3 mutually orthogonal directions.

In matrix form, the theorem above can be written as

$$\begin{bmatrix} X_{\mathbf{p}}(t-1) \\ Y_{\mathbf{p}}(t-1) \\ Z_{\mathbf{p}}(t-1) \end{bmatrix} = \begin{bmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{32} & r_{33} \end{bmatrix} \begin{bmatrix} X_{\mathbf{p}}(t) \\ Y_{\mathbf{p}}(t) \\ Z_{\mathbf{p}}(t) \end{bmatrix} + \begin{bmatrix} T_x \\ T_y \\ T_z \end{bmatrix} + \begin{bmatrix} s_{11} & s_{12} & s_{13} \\ s_{21} & s_{22} & s_{23} \\ s_{31} & s_{32} & s_{33} \end{bmatrix} \begin{bmatrix} X_{\mathbf{p}}(t) \\ Y_{\mathbf{p}}(t) \\ Z_{\mathbf{p}}(t) \end{bmatrix} . \tag{10.12}$$

In eq. (10.12), the matrix which consists of elements $r_{i,j}$ is a rotation matrix, orthogonal of 1st kind (i.e., determinant is equal to 1) with only 3 degrees of freedom. The matrix with elements $s_{i,j}$ is the linear deformation matrix, which is only symmetric. After a global rigid motion consisting of rotation and translation, and some local deformation, a point, $\mathbf{P}$, which has the coordinates $[X_{\mathbf{p}}(t)\ Y_{\mathbf{p}}(t)\ Z_{\mathbf{p}}(t)]^T$ moves to another location $[X_{\mathbf{p}}(t-1)\ Y_{\mathbf{p}}(t-1)\ Z_{\mathbf{p}}(t-1)]^T$. In fact, again this is a "reverse" motion in time from t to $t-1$ although the "real" motion is from time $t-1$ to t. For a general 3D motion, the rotation and translation parameters are constant at each point of the object, whereas the linear deformation matrix depends on local behavior of the non-rigid motion. Hence, the deformation matrix might be different at any object point. In human body dynamics, while the human body makes a global motion (e.g., rotation of the head), there might be some local deformations at different parts of it (e.g., lips and cheeks might move during rotation).

Using eq. (10.12), the following relation is valid for any object point (similar relations can also be written for $Y_{\mathbf{p}}(t-1)$ and $Z_{\mathbf{p}}(t-1)$):

$$X_{\mathbf{p}}(t-1) = (r_{11}+s_{11})X_{\mathbf{p}}(t)+(r_{12}+s_{12})Y_{\mathbf{p}}(t)+(r_{13}+s_{13})Z_{\mathbf{p}}(t)+T_x \quad . \quad (10.13)$$

The next coordinate of any point is determined by some global (the rigid rotation and translation parameters) and some local (the non-rigid deformation parameters at each point) motion. However, all the variables are unknown and to be determined. Although the deformation parameters are independent variables at each object point, because of some anatomical reasons (muscles, skin, bones) for human motion, the neighboring deformation parameters should be correlated with each other, i.e., they are expected to have similar values.

Taking the above ideas into consideration, a stochastic formulation which defines the motion parameters at each point as random variables and takes into account their interactions by a joint probability distribution, can be proposed. Observing two consecutive frames, the aim is to model and estimate the non-rigid motion between them.

4.2 Gibbs Model Based Non-Rigid Motion Estimation

The *a priori* information for the general motion model is the existence of some local correlation between neighboring motion parameters. The local interactions permit looser relations between neighboring parameters in contrast to the rigidity assumption which is ultimately tight. Therefore, this general approach may achieve motion estimation of non-rigid or weakly-rigid moving objects, as well as rigid body motion estimation.

From eq. (10.12), it can be observed that the number of independent unknown deformation variables for each point of the object is equal to 6 (due to the symmetry of the deformation matrix). There are also 6 unknown rotation and translation parameters, which are global and constant at every point of the object. Assuming the rotation angles between the frames are small, the rotation matrix can be written in terms of the rotation angles, W_x, W_y, W_z, around the x, y, z axes, respectively (Fig. 1),

$$R_{W_x,W_y,W_z} = \begin{bmatrix} 1 & W_z & -W_y \\ -W_z & 1 & W_x \\ W_y & -W_x & 1 \end{bmatrix} . \quad (10.14)$$

After defining some new variables as

$$w_x \doteq W_x + s_{23} = W_x + s_{32} , \quad t_x \doteq T_x + s_{11}X(t) + 2s_{13}Z(t) , \quad (10.15)$$
$$w_y \doteq W_y + s_{13} = W_y + s_{31} , \quad t_y \doteq T_y + s_{22}Y(t) + 2s_{21}X(t) , \quad (10.16)$$
$$w_z \doteq W_z + s_{12} = W_z + s_{21} , \quad t_z \doteq T_z + s_{33}Z(t) + 2s_{32}Y(t) . \quad (10.17)$$

Equation (10.13) can be rewritten using the rotation angles and the new variables and upon some manipulations as,

$$\begin{aligned} X_{\mathbf{p}}(t-1) &= X_{\mathbf{p}}(t) + w_z Y_{\mathbf{p}}(t) + -w_y Z_{\mathbf{p}}(t) + t_x , \\ Y_{\mathbf{p}}(t-1) &= -w_z X_{\mathbf{p}}(t) + Y_{\mathbf{p}}(t) + w_x Z_{\mathbf{p}}(t) + t_y , \\ Z_{\mathbf{p}}(t-1) &= w_y X_{\mathbf{p}}(t) + -w_x Y_{\mathbf{p}}(t) + Z_{\mathbf{p}}(t) + t_z . \end{aligned}$$

The 3D motion parameter vector $\theta = [w_x, w_y, w_z, t_x, t_y, t_z]$ is defined at each point of the object. Since only the surface points of the objects are observed on the image, the 3D parameter vector is only defined at each point on a 2D grid, Λ, on which the image intensities are also defined. Θ is defined as the set of motion parameters θ which are defined at each point on Λ.

Given a frame I_t and 3D motion parameters, Θ, the correct non-rigid motion parameters should find some intensity correspondences on the previous frame, I_{t-1}. After projecting the 3D coordinates perspectively, the displacements on the image plane are

$$\begin{aligned} x_{\mathbf{p}}(t-1) &= \frac{x_{\mathbf{p}}(t) + w_z y_{\mathbf{p}}(t) + (-w_y) + \frac{t_x}{Z_{\mathbf{p}}(\mathbf{x}_{\mathbf{p}},t)}}{w_y x_{\mathbf{p}}(t) + (-w_x) y_{\mathbf{p}}(t) + 1 + \frac{t_z}{Z_{\mathbf{p}}(\mathbf{x}_{\mathbf{p}},t)}} , \\ y_{\mathbf{p}}(t-1) &= \frac{(-w_z) x_{\mathbf{p}}(t) + y_{\mathbf{p}}(t) + w_x + \frac{t_y}{Z_{\mathbf{p}}(\mathbf{x}_{\mathbf{p}},t)}}{w_y x_{\mathbf{p}}(t) + (-w_x) y_{\mathbf{p}}(t) + 1 + \frac{t_z}{Z_{\mathbf{p}}(\mathbf{x}_{\mathbf{p}},t)}} . \end{aligned} \quad (10.18)$$

Using these displacements, when there is no noise, no occlusion and no illumination change in the environment, the *optic flow* will hold at each 2D image coordinate $\mathbf{x}_{\mathbf{p}}(t) \in \Lambda$:

$$I_t(\mathbf{x}_{\mathbf{p}}(t)) = I_{t-1}(\mathbf{x}_{\mathbf{p}}(t-1)) . \quad (10.19)$$

In order to find the MAP estimate of the 3D motion parameters between two consecutive frames, the energy function of Gibbs posterior distribution can be written as

$$\mathcal{U}(\Theta|\mathcal{I}_t, \mathcal{I}_{t-1}, \mathcal{Z}) = \mathcal{U}(\mathcal{I}_{t-1}|\mathcal{I}_t, \Theta, \mathcal{Z}) + \beta\mathcal{U}(\Theta|\mathcal{I}_t, \mathcal{Z}) \quad . \tag{10.20}$$

Minimizing the above equation with respect to Θ field, which consist of $\theta(\mathbf{x}(t))$, the parameter set at each point, we obtain the MAP estimate for the 3D motion parameter field.

From eq. (10.19) under the assumption that there is Gaussian noise in the environment, the first term on rhs of eq. (10.20) becomes

$$\mathcal{U}(I_{t-1}|I_t, \Theta, \mathcal{Z}) = \sum_{\mathbf{x_p} \in \Lambda} \left(I_t(\mathbf{x_p}(t)) - I_{t-1}(\mathbf{x_p}(t-1))\right)^2 \quad . \tag{10.21}$$

The second term on rhs of eq. (10.20) can be obtained using the *a priori* information on the 3D motion parameters, as

$$\mathcal{U}(\Theta|I_t, \mathcal{Z}) = \sum_{\mathbf{x_p} \in \Lambda} \sum_{\mathbf{x_{p,c}} \in \eta_{\mathbf{x_p}}} ||\theta(\mathbf{x_p}(t)) - \theta(\mathbf{x_{p,c}}(t))||^2 \quad , \tag{10.22}$$

where $\mathbf{x_{p,c}} \in \eta_{\mathbf{x_p}}$ is the neighbor of $\mathbf{x_p}$. This energy function favors similar values on neighboring parameters by assigning higher probabilities to such cases.

The formulation above assumes the availability of the depth field, $\mathcal{Z}$, a priori. The depth field can either be obtained from an extra sensor data or a stereo pair. If these data are not available, the depth field can be found, using linear techniques, from an estimated dense 2D motion vector field. By the help of eq. (10.18) and current 3D motion parameter estimates, the only unknown $Z_{\mathbf{p}}(\mathbf{x_p}, t)$ in eq. (10.18) can be found in the least-squares sense. This can be achieved at every step of the minimization of eq. (10.20).

Another approach is to define the depth field as a random field like motion parameters and to add a new term similar to eq. (10.11) into eq. (10.20). However, in such a situation, the minimization problem will become severely underconstrained due to scaling ambiguity between the depth field and the translation parameters and hence the convergence of the energy function will become extremely difficult.

Equation (10.20) can be improved by adding some new segmentation terms, which will also divide the scene into objects according to their 3D motion

coherence. Hence, this non-rigid analysis will also be applicable for object-based algorithms.

In summary, the above explained formulation,

- is valid for a general class of motion without structural constraints;
- does not contain errors due to discrete differentiation;
- can also segment the scene;
- uses more realistic projections;
- is more immune to noise as a result of the MAP estimation.

4.3 Weakening Rigidity Hierarchically

The energy function in eq. (10.20) is valid not only for non-rigid but also for rigid objects. The non-rigid motion formulation defines a global rigid motion with some local deformations "added" on top of them. Hence, it will be better first to find the global rigid motion and afterwards weaken this rigid result to obtain local interactions.

In order to implement these ideas while minimizing eq. (10.20), a multiscale approach is devised. In this approach, the 3D motion parameters are defined at each point on different grids for different resolutions, which are shown in Fig. 7. On the coarsest grid, the motion parameters are constant in a predefined rectangle and therefore the part of the object which is projected onto this rectangle is assumed to be rigid. These sparsely defined motion parameters are estimated by minimizing eq. (10.20) by one of the global optimization algorithms (e.g., SA, ICM). While the scale gets finer, the size of the rectangles in which the motion parameters are equal with each other, and consequently the rigid part of the object, gets smaller. Since the parameters are passed through scales from coarse to fine while minimization is achieved at each level, the global rigid motion still exist "under" local interactions. Hence, such an approach will estimate the motion of a rigid object without any convergence problems and is called "hierarchical rigidity" [65]. Similar minimization algorithms are independently proposed by [28], called Multiscale Constrained Relaxation, for general recovery problems.

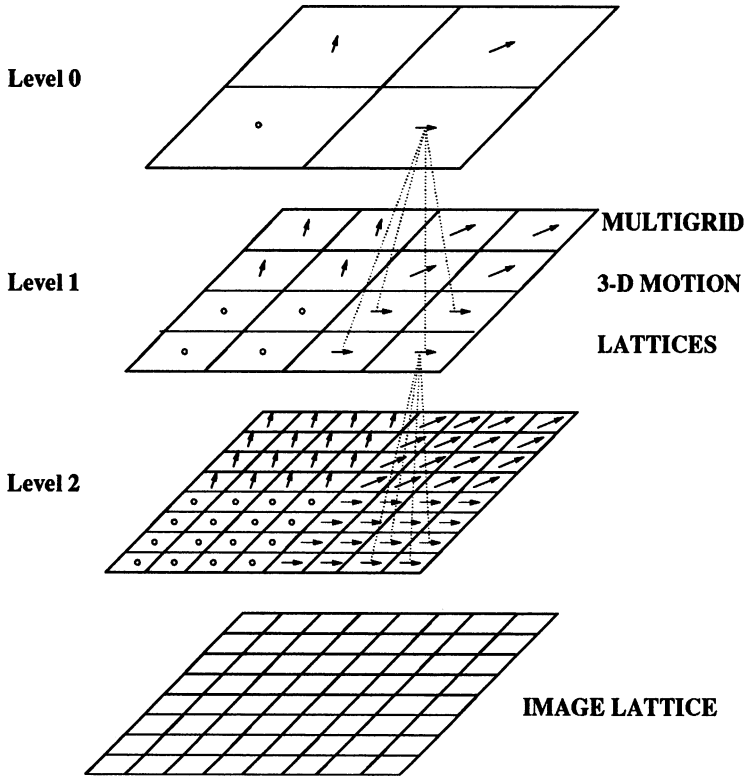

Figure 7 Different grids for motion vectors in "Hierarchical Rigidity" minimization. Estimation results are propagated from coarse to fine during ICM minimization at each scale.

4.4 Simulation Results

Although, the formulation is valid for non-rigid motion (which includes the rigid motion as a special case), all the simulations are carried on artificial sequences, which have rigid motion (Figs. 8 and 11). The experiments consist of three stages, as validation of hierarchical rigidity, segmentation for multiple moving objects and noise analysis.

In the first step, hierarchical rigidity is applied using ICM at each scale with 3 levels. A typical result is shown in Fig. 9. The estimated parameter w_z is shown by intensities over the image for coarsest and finest levels. The

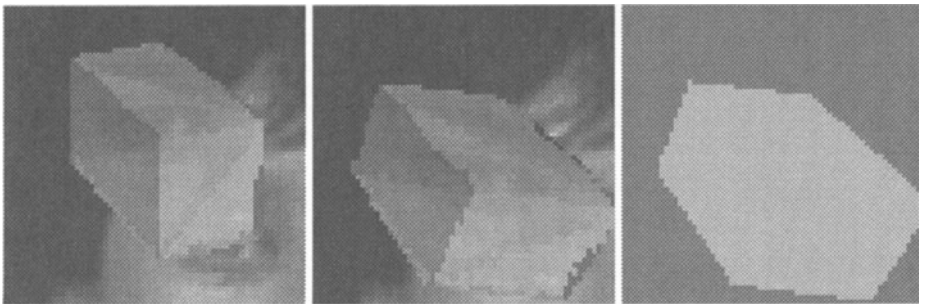

Figure 8 2 original frames (a),(b) of "Cube" sequence. (c) Ideal motion parameter value for w_z shown as an intensity representation.

estimates at the finest level are similar to ideal results at Fig. 8. In Fig. 9, the histogram representation of all 3D parameters are shown compared to the true values. The 2D projection of the estimated 3D motion vector parameters are shown in Fig. 10 by a "needlegram" on the reconstructed image, which is obtained by using the previous available frame, known depth values and 3D motion vectors. The estimated needlegram has comparable results with the true values inside the cube, except the occlusion areas.

In the second step, a scene with 2 cubes moving with different speeds are examined by hierarchical rigidity (Fig. 11). The results in Fig. 12 show that, the method can easily achieve segmentation of the cubes and assign their 3D values correctly.

As a final step, a noise analysis is performed on our proposed method. The images are corrupted by Gaussian noise, obtaining frames, having SNR_{peak} values as 28dB and 43dB. It is observed in Fig. 13 that, above approximately 30dB performance of the methods is acceptable.

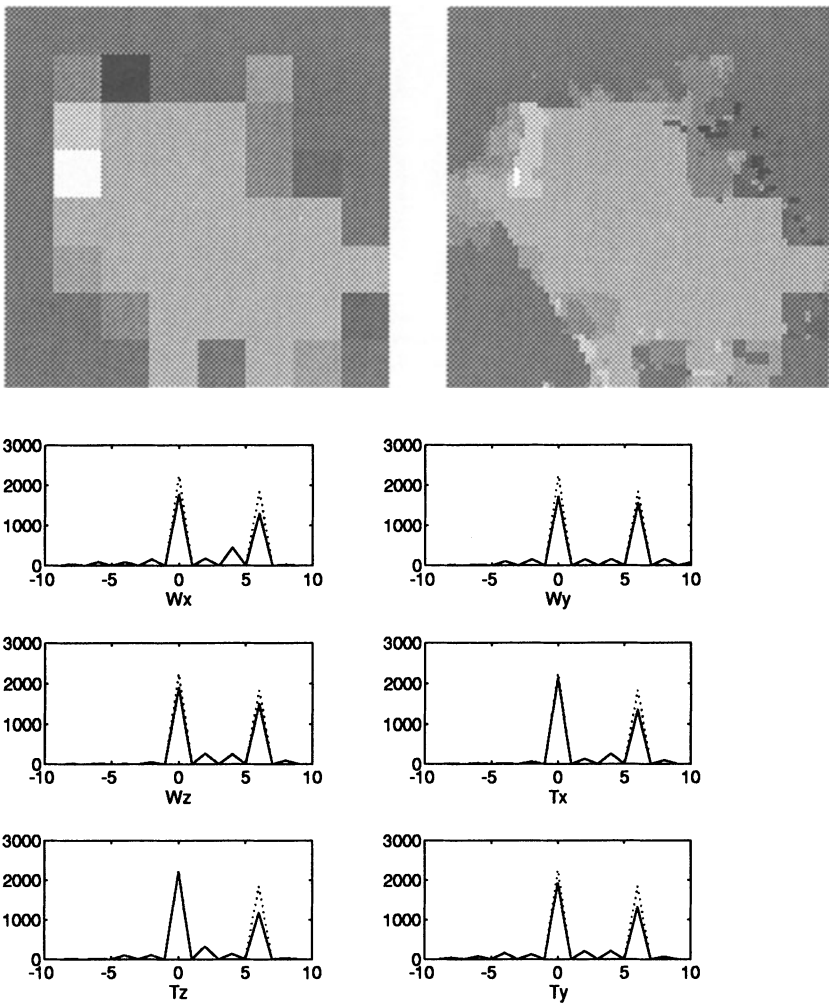

Figure 9 The intensity representation of w_z parameter for (a) 8x8 block size (coarsest level) and (b) 1x1 block size (finest level). (c) The histogram representation of $w_{x,y,z}$ and $t_{x,y,z}$ parameters. (Dotted lines ideal, solid lines estimated values.)

5 DISCUSSION AND CONCLUSSION

A novel rigid 3D object-based motion and structure estimation method is proposed. The simulation results show that this method can be easily inserted

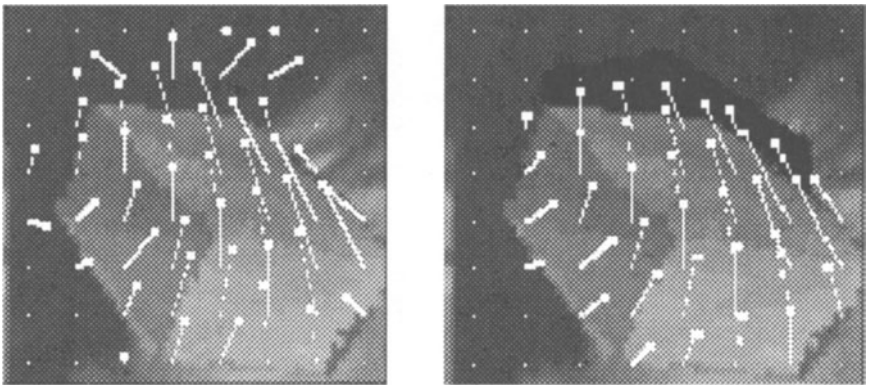

Figure 10 The (a) estimated and (b) true needlegrams of "Cube" on the reconstructed frames.

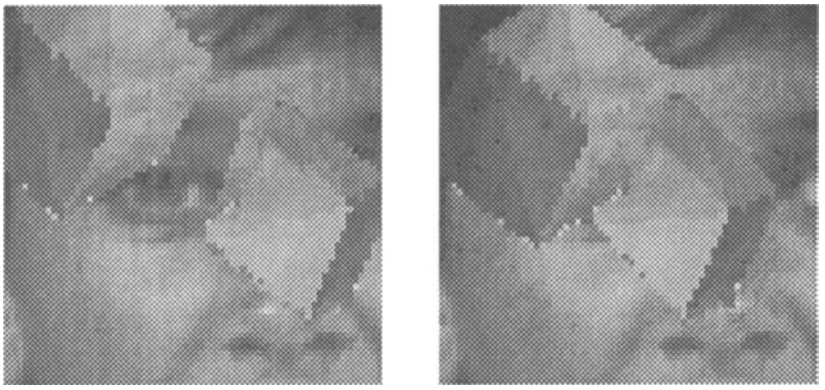

Figure 11 Two original frames of "Cubes" sequence.

into a video compression algorithm and the compression performance of such a coding algorithm depends on efficient encoding of the depth field, since

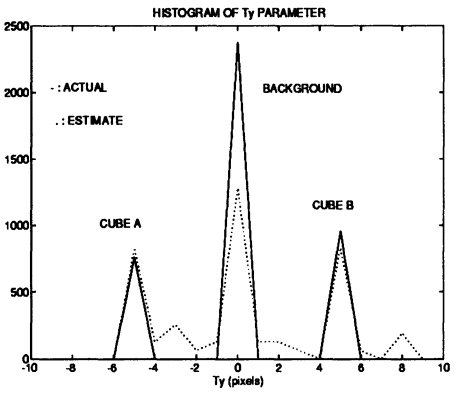

Figure 12 Histogram of t_y parameter for "Cubes". (True values with solid and estimates with dotted lines.)

motion description is very efficient. As it is observed during the experiments, the number of bits needed to encode the depth field might be quite high when only two video frames are considered. However, the temporal redundancy in the structure of a rigid object is ultimately high even for a long sequence of frames; therefore, the expected improvement in coding efficiency is significant. As it is shown by the experimental results, it is possible to represent the intensity movement of thousands of object points by using 6 parameters and a depth field with ultimately high temporal redundancy. Hence, 3D object-based motion analysis is a strong candidate for improving the saturated compression performance of current coding standards.

Although better feature detection and matching algorithms exist, Gibbs model based 2D matching is chosen to achieve joint motion estimation and object segmentation. The experimental results of 2D motion estimation and segmentation show how these steps are successful when Gibbs formulation is being utilized. The face of the foreman is segmented accurately, although the hat is specified as stationary because of its uniform surface.

By rejecting the outliers of the dense motion field, the overall performance of the 3D motion estimation algorithm, which uses 2D motion matches, improves considerably compared to the case which uses all the available 2D

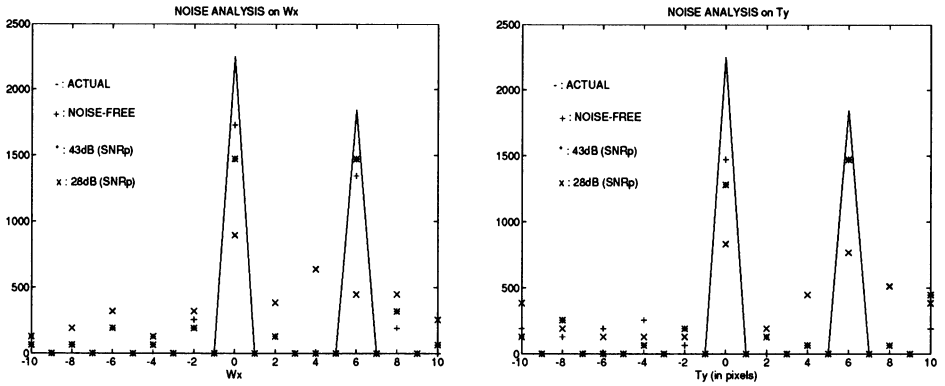

Figure 13 The estimation results of motion parameters (a) w_x and (b) t_y for input frames with different SNR_{peak} values.

motion vectors. Hence, rejection of outliers is a necessary step in the overall algorithm.

The most dominant drawback of the E-matrix method is its susceptibility to input noise and errors. For small images (e.g., QCIF size), the unavoidable quantization of 2D motion vectors at pel accuracy, degrades results considerably. Without having the nonlinear epipolar improvement step in the algorithm, the 3D motion parameter estimates will be more susceptible to computational noise. In the computer simulations, as the obtained eigenvalues indicate, the estimated E-matrix is valid, implicitly containing a rotation matrix and a translation vector.

During the simulations real frames from a video recorder are used. In addition to the natural noise on them, there is also computational noise which assures that the input to the depth estimation algorithm is noisy. Estimating the depth field by taking the noise into account using MAP criterion should be superior to any method which does not consider such erroneous inputs. The depth estimation algorithm proposed in the previous sections takes into account the input noise by proper selection of the energy terms. The

simulation results show that a successful dense depth field is obtained even in noisy cases.

One of the drawbacks of the proposed 3D object-based motion analysis method is the assumption of rigidity. Whenever the object is in a non-rigid or incorrectly segmented articulated motion, the proposed rigid algorithm would be unsatisfactory. A solution to this problem is to use 2D motion models, which are more flexible to any kind of motion and to adaptively select between rigid 3D and 2D models according to their performance. Another solution is to model non-rigid motion by taking into account local motions with some Markovian interactions. A novel Gibbs formulated non-rigid motion estimation algorithm, which has a promising performance for the analysis of elastic motions, is also proposed. The kinematic relations between neighboring motion elements of a non-rigid body can be successfully modeled by a Gibbs distribution. Such a formulation, not only models elastic motion with loose local relations, but also models rigid motion with ultimately tight neighbor relations, and this formulation can be solved by the help of hierarchical rigidity. The experimental results of this method on rigid bodies give promising results.

However, the description complexity of non-rigid motions makes them unattractive for compression purposes, except for long sequences with high temporal redundancy. Hence, non-rigid motion analysis methods are more suitable for applications in which motion tracking or structure estimation is more important compared to coding.

REFERENCES

[1] J.K. Aggarwal and N. Nandhakumar, "On the computation of motion from image sequences-a review," *IEEE Proc.*, vol. 76, pp. 917-935, Aug. 1988.

[2] A. Verri and T. Poggio, "Motion field and optical flow: Qualitative properties," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 11, pp. 490-498, May 1989.

[3] R.Y. Tsai and T.S. Huang, "Uniqueness and estimation of three-dimensional motion parameters of rigid objects with curved surfaces," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 6, pp. 13-27, Jan. 1984.

[4] K. Aizawa and T.S. Huang, "Model-based image coding: Advanced video coding techniques for very low bit-rate applications," *IEEE Proc.*, vol. 83, pp. 259-271, Feb. 1995.

[5] G. Bozdağı, *Three Dimensional Facial Motion and Structure Estimation in Video Coding.* PhD thesis, Bilkent University, Jan. 1994.

[6] H. Li, P. Roivainen and R. Forchheimer, "3-d motion estimation in model-based facial image coding," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 15, pp. 545-555, June 1993.

[7] J.O. Limb and J.A. Murphy, "Measuring the speed of moving objects from television signals," *IEEE Trans. Commn.*, vol. , pp. 474-478, April 1975.

[8] C. Cafforio and F. Rocca, "Methods of measuring small displacement of television images," *IEEE Trans. Inform. Theo.*, vol. 22, pp. 573-579, Sep. 1976.

[9] B.K.P. Horn and B.G. Shunck, "Determining optical flow," *Artificial Intell.*, vol. 17, pp. 185-203, 1981.

[10] A.M. Tekalp, *Digital Video Processing.* Prentice Hall, 1995.

[11] A.N. Netravali and J.D. Robbins, "Motion-compensated television coding, part-1," *AT & T Techn. Jour.*, vol. 58, pp. 629-668, March 1979.

[12] J.R. Jain and A.K. Jain, "Displacement measurement and its application in interframe image coding," *IEEE Trans. Commun.*, vol. 29, pp. 1799-1808, Dec. 1981.

[13] M. Bierling, "Displacement estimation by hierarchical blockmatching," in *Proc. of SPIE Visual Commun. and Im. Proc. 88*, pp. 942-951, 1988.

[14] V. Seferidis and M. Ghanbari, "General approach to block-matching motion estimation," *Optical Engineering*, vol. 32, pp. 1664-1474, July 1993.

[15] J. Hutchison, C. Koch, J. Luo and C. Mead, "Computing motion using analog and binary resistive networks," *IEEE Trans. Computer*, vol. , pp. 52-63, March 1988.

[16] S. Geman and D. Geman, "Stochastic relaxation, Gibbs distributions, and the Bayesian restoration of images," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 6, pp. 721–741, Nov. 1984.

[17] J. Konrad and E. Dubois, "Bayesian estimation of motion vector fields," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 14, pp. 910-927, Sep. 1992.

[18] F. Heitz and P. Bouthemy, "Multimodal motion estimation and segmentation using Markov random fields," in *Proc. Int. Conf. on Pat. Recog. 90*, pp. 378-383, 1990.

[19] F. Heitz and P. Bouthmey, "Multimodal estimation of discontinuous optical flow using Markov random fields," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 15, pp. 1217-1232, Dec. 1993.

[20] R. Buschmann, "Joint multigrid estimation of 2d motion and object boundaries using boundary patterns," in *SPIE Visual Commun. and Image Proc. 93*, pp. 106-118, 1993.

[21] S. Iu, "Robust estimation of motion vector fields with discontinuity and occlusion using local outliers rejection," in *SPIE Visual Commun. and Image Proc. 93*, pp. 588-599, 1993.

[22] E. Dubois and J. Konrad, "Motion estimation and motion-compensated filtering of video signals," in *Proc. of IEEE ICASSP 93*, pp. 95-98, 1993.

[23] R. Depommier and E. Dubois, "Motion estimation with detection of occlusion areas," in *Proc. of IEEE ICASSP 92*, pp. 269-272, 1992.

[24] C.L. Chan, J.C. Brailean, A.K. Katsaggelos and A.V. Sahakin, "Maximum a posteriori displacement field estimation in quantum-limited image sequences," in *SPIE Visual Commun. and Image Proc. 93*, pp. 396-407, 1993.

[25] J.C. Brailean and A.K. Kataaggelos, "Recursive map displacement estimation and restoration of noisy-blurred image sequences," in *SPIE Visual Commun. and Image Proc. 93*, pp. 384-395, 1993.

[26] S. Kirkpatrick, C.D. Gelatt and M.P. Vecchi, "Optimization by simulated annealing," *Science*, vol. 220, pp. 661-680, 1983.

[27] J. Besag, "On the statistical analysis of dirty pictures," *J.R. Statist. Soc.*, vol. 48, pp. 259-302, 1986.

[28] F. Heitz, P. Perez and P. Bouthemy, "Multiscale minimization of global energy functions in some visual recovery problems," *CVGIP: Image Understanding*, vol. 59, pp. 125-134, Jan. 1994.

[29] W.N. Martin and J.K. Aggarwal, *Motion Understanding, Robot and Human Vision*, pp. 329-352. Kluwer Academic Publishers, Boston, 1988.

[30] J.F. Vega-Riveros and K. Jabbour, "Review of motion analysis techniques," *IEE Proc.*, vol. 136, pp. 397-404, Dec. 1989.

[31] J.N. Driessen, *Motion Estimation for Digital Video.* PhD thesis, Delft University of Technology, Sep. 1992.

[32] M.I. Sezan and R. L. Lagendijk, ed. *Motion Analysis and Image Sequence Processing.* Kluwer Academic Publishers, 1993.

[33] R.Y. Tsai and T.S. Huang, "Estimating three-dimensional motion parameters of a rigid planar patch," *IEEE Trans. Sig. Proc.*, vol. 29, pp. 1147-1152, Dec. 1981.

[34] X. Hu and N. Ahuja, "Sufficient conditions for double and unique solution of motion and structure," *CVGIP: Image Understanding*, vol. 58, pp. 161-176, Sep. 1993.

[35] J.W. Roach and J.K. Aggarwal, "Determining the movement of objects from a sequence of images," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 2, pp. 554-562, Nov. 1980.

[36] S. UIlman, "Maximizing rigidity: The incremental recovery of 3d structure from rigid and nonrigid motion," *Perception*, vol. 13, pp. 255-274, 1984.

[37] A. Pentland and B. Horowitz, "Recovery of nonrigid motion and structure," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 13, pp. 730-742, July 1991.

[38] D. Terzopoulos and D. Metaxas, "Dynamic 3d models with local and global deformations : Deformable superquadrics," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 13, pp. 703-714, July 1991.

[39] C. W. Chen , T. S. Huang and M. Arrott, "Modelling, analysis, and visualization of left ventricle shape and motion by hierarchical decomposition," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 16, pp. 342-356, April 1994.

[40] J. Weng, T.S. Huang and N. Ahuja, *Motion and Structure from Image Sequences.* Springer-Verlag, 1993.

[41] A.N. Netravali and J. Salz, "Algorithms for estimation of three-dimensional motion," *AT & T Techn. Jour.*, vol. 64, pp. 335-346, 1985.

[42] B.K.P. Horn and E.J. Weldon Jr., "Direct methods for recovering motion," *Int. Jour. of Computer Vision*, vol. 2, pp. 51-76, 1988.

[43] S. Peleg and H. Rom, "Motion based segmentation," in *Proc. Int. Conf. on Pat. Recog. 90*, pp. 109-111, 1990.

[44] J. Weng, N. Ahuja and T. S. Huang, "Matching two perspective views," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 14, pp. 806-825, Aug. 1992.

[45] J. Weng and T.S. Huang, "Estimating motion and structure from line matches: Performance obtained and beyond," in *Proc. Int. Conf on Pat. Recog. 90*, pp. 168-172, 1990.

[46] A. Zakhor and F. Lari, "Edge-Based 3-D Camera Motion Estimation with Applications to Video Coding," *IEEE Trans. Image Proc.*, vol. 2, pp. 481-498, Oct. 1993.

[47] S. Negahdaripour, "Multiple interpretations of the shape and motion of objects from two perspective images," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 12, pp. 1025-1039, Nov. 1990.

[48] G. Adiv, "Determining three-dimensional motion and structure from optical flow generated by several moving objects," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 7, pp. 384-402, July 1985.

[49] M. Hotter and R. Thoma, "Image segmentation based on object oriented mapping parameter estimation," *Sig. Proc.*, vol. 15, pp. 315-334, 1988.

[50] M. Chang, M.I. Sezan and A.M. Tekalp, "A Bayesian framework for combined motion estimation and scene segmentation in image sequences," in *Proc. of IEEE ICASSP 94*, pp. 221-224, 1994.

[51] J. Weng, N. Ahuja and T.S. Huang, "Optimal motion and structure estimation," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 15, pp. 864-884, Sep. 1993.

[52] H. Morikawa and H. Harashima, "3d structure extraction coding of image sequences," *Jour. Visual Commun. Image Repr.*, vol. 2, pp. 332-344, Dec. 1991.

[53] N. Diehl, "Object-oriented motion estimation and segmentation in image sequences," *Signal Proc.: Image Commun.*, vol. 3, pp. 23-56, 1991.

[54] R.Y. Tsai, T.S. Huang and W. Zhu, "Estimating three-dimensional motion parameters of a rigid planar patch, ii: Singular value decomposition," *IEEE Trans. Sig. Proc.*, vol. 30, pp. 525-534, Aug. 1982.

[55] A.A. Alatan and L. Onural, "Object-based 3-d motion and structure estimation," in *Proc. IEEE Int. Conf. on Im. Proc. '95, Washington D.C., Oct.*, pp. I. 390-393, 1995.

[56] D.W. Murray and B.F. Buxton, *Experiments in the Machine Interpretation of Visual Motion.* MIT Press, 1990.

[57] S. Maybank, *Theory of Reconstruction from Image Motion.* Springer-Verlag, 1993.

[58] A.A. Alatan and L. Onural, "Optimal depth encoding for 3-D object-based video coding," *submitted to IEEE Trans. Image Proc.*.

[59] H.G. Mussmann, M. Hotter and J. Ostermann, "Object-oriented analysis-synthesis coding of moving images," *Signal Proc.: Image Commun.*, vol. 1, pp. 117-138, Oct. 1989.

[60] J. Zhang and J. Hanauer, "The mean field theory for image motion estimation," in *Proc. IEEE ICASSP 93*, pp. 197-200, 1993.

[61] M.J. Biggar, O.J. Morris and A.G. Constantinides, "Segmented-image coding: Performance comparison with the discrete cosine transform," *IEE Proc.*, vol. 135, pp. 121-132, April 1988.

[62] T.S Huang and H. Lee, "Motion and structure from orthographic projection," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 11, pp. 536-540, May 1989.

[63] X. Hu and N. Ahuja, "Motion estimation under orthographic projection," *IEEE Trans. Pattern Anal. Machine Intell.*, Dec. 1991.

[64] A. Sommerfeld, *Mechanics of Deformable Bodies.* Academic Press, 1950.

[65] A.A. Alatan and L. Onural, "Gibbs random field model based 3-d motion estimation by weakened rigidity," in *Proc, IEEE Int. Conf. Im. Proc. '94, Austin, Nov.*, pp. II.790-794, 1994.

A. Aydin Alatan was born in Ankara, Turkey in 1968. He received the B.S. degree from Middle East Technical University, Ankara, Turkey in 1990 and the M.S. and DIC degrees from Imperial College of Science, Medicine and Technology, London , UK in 1992, all in Electrical Engineering. He was a British Council scholar between 1991 and 1992. He is currently an assistant in Bilkent University and pursuing his Ph.D. degree. His research interests are Motion estimation, 3-D motion models, non-rigid motion analysis, Gibbs Random Field based models, object-based coding and very low bit rate video coding. A. Aydin Alatan is a student member of IEEE.

Levent Onural was born in İzmir, Turkey in 1957. He received the B.S. and M.S. degrees in electrical engineering from Middle East Technical University, Ankara, Turkey, in 1979 and 1981, respectively, and the Ph.D. degree in electrical and computer engineering from State University of New York at Buffalo in 1985. He was a Fulbright scholar between 1981 and 1985. After a Research Assistant Professor position at the Electrical and Computer Engineering Department of State University of New York at Buffalo, he joined the Electrical and Electronics Engineering Department of Bilkent University, Ankara, Turkey, where he is a full Professor at present. He visited the Electrical and Computer Engineering department of University of Toronto on a sabbatical leave between September 1994 - February 1995. His current research interests are in the area of image and video processing, with emphasis on very low bit rate video coding, texture modeling, non-linear filtering, holographic TV and signal processing aspects of optical wave propagation. Dr. Onural is a senior member of IEEE and a member of SPIE. He was the organizer and the first chairman of IEEE Turkey Section; he served as the chairman of the IEEE Circuits and Systems Chapter in Turkey between 1994 - 1996. He is now the chairman of IEEE Region 8 Student Activities Committee. In 1995, Dr. Onural received the Young Investigator Award from TÜBİTAK.