

OBJECT-BASED 3-D MOTION AND STRUCTURE
ANALYSIS FOR VIDEO CODING APPLICATIONS

A DISSERTATION
SUBMITTED TO THE DEPARTMENT OF ELECTRICAL AND
ELECTRONICS ENGINEERING
AND THE INSTITUTE OF ENGINEERING AND SCIENCES
OF BILKENT UNIVERSITY
IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR THE DEGREE OF
DOCTOR OF PHILOSOPHY

By

A. Aydin Alatan

24 February 1997

THESIS
7A
1637
A43
1997

OBJECT-BASED 3-D MOTION AND STRUCTURE
ANALYSIS FOR VIDEO CODING APPLICATIONS

A DISSERTATION

SUBMITTED TO THE DEPARTMENT OF ELECTRICAL AND ELECTRONICS ENGINEERING

AND THE INSTITUTE OF ENGINEERING AND SCIENCE

OF BİLKENT UNIVERSITY

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS

FOR THE DEGREE OF

DOCTOR OF PHILOSOPHY

By

A. Aydın Alatan

24 February 1997

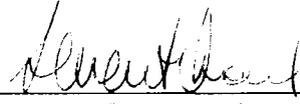
A. Aydın Alatan.

tarafından bağışlanmıştır

7H
1637
A43
1937

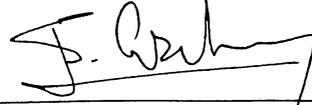
BC30919

I certify that I have read this thesis and that in my opinion it is fully adequate, in scope and in quality, as a dissertation for the degree of Doctor of Philosophy.



Levent Onural, Ph. D. (Supervisor)

I certify that I have read this thesis and that in my opinion it is fully adequate, in scope and in quality, as a dissertation for the degree of Doctor of Philosophy.



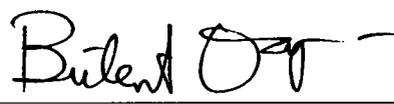
Erdal Arıkan, Ph. D.

I certify that I have read this thesis and that in my opinion it is fully adequate, in scope and in quality, as a dissertation for the degree of Doctor of Philosophy.



Orhan Arıkan, Ph. D.

I certify that I have read this thesis and that in my opinion it is fully adequate, in scope and in quality, as a dissertation for the degree of Doctor of Philosophy.



Bülent Özgüç, Ph. D.

I certify that I have read this thesis and that in my opinion it is fully adequate, in scope and in quality, as a dissertation for the degree of Doctor of Philosophy.



Aydan Ekmen, Ph. D.

Approved for the Institute of Engineering and Science:



Mehmet Baray, Ph. D.
Director of Institute of Engineering and Science

*this thesis is dedicated to
my father who has never had an opportunity to be an engineer,
my mother whose love has enabled me to smile during all my life,
my wife Lale who has suffered a lot due to this thesis ...*

Abstract

OBJECT-BASED 3-D MOTION AND STRUCTURE ANALYSIS FOR VIDEO CODING APPLICATIONS

A. Aydın Alatan

Ph. D. in Electrical and Electronics Engineering

Supervisor:

Prof. Levent Onural

24 February 1997

Novel 3-D motion analysis tools, which can be used in object-based video codecs, are proposed. In these tools, the movements of the objects, which are observed through 2-D video frames, are modeled in 3-D space. Segmentation of 2-D frames into objects and 2-D dense motion vectors for each object are necessary as inputs for the proposed 3-D analysis. 2-D motion-based object segmentation is obtained by Gibbs formulation; the initialization is achieved by using a fast graph-theory based region segmentation algorithm which is further improved to utilize the motion information. Moreover, the same Gibbs formulation gives the needed dense 2-D motion vector field. The formulations for the 3-D motion models are given for both rigid and non-rigid moving objects. Deformable motion is modeled by a Markov random field which permits elastic relations between neighbors, whereas, rigid 3-D motion parameters are estimated using the E-matrix method. Some improvements on the E-matrix method are proposed to make this algorithm more robust to gross errors like the consequence of incorrect segmentation of 2-D correspondences between frames. Two algorithms are proposed to obtain dense depth estimates, which are robust to input errors and suitable for encoding, respectively. While the former of these two algorithms gives simply a *MAP* estimate, the latter uses rate-distortion theory. Finally, 3-D motion models are further utilized for occlusion detection and motion compensated temporal interpolation, and it is observed that for both applications 3-D motion models have superiority over their 2-D counterparts. Simulation results on artificial and real data show the advantages of the 3-D motion models in object-based video coding algorithms.

Keywords: Very low bit-rate video compression, object-based coding, 3-D motion estimation, 3-D structure estimation, Markov random fields, segmentation, 2-D motion estimation, *MAP* estimation, rate distortion theory, temporal interpolation, occlusion detection.

Özet

VIDEO KODLAMA UYGULAMLARI İÇİN NESNEYE DAYALI ÜÇ BOYUTLU HAREKET VE DERİNLİK ANALİZİ

A. Aydın Alatan

Elektrik ve Elektronik Mühendisliği Doktora

Tez Yöneticisi:

Prof. Levent Onural

28 Ocak 1997

Nesneye dayalı kodlama uygulamaları için 2-B görüntü kareleri ile izlenen nesnelerin yer değiştirmeleri 3-B uzayda modellenmektedir. Önerilen 3-B hareket analizi için 2-B hareket vektörlerinin kestirimi ile birlikte imge karelerinin nesnelere bölütlenmesi de gereklidir. Bölütleme için Gibbs formülasyonu kullanılmaktadır. Çizge kuramına dayalı bir bölütleme metodunun hareket bilgisi kullanabilecek biçimde geliştirilmiş bir uyarlaması ise Gibbs formülasyonu kullanan metoda ön kestirimleri sağlamaktadır. Ayrıca Gibbs formülasyonu 3-B hareket kestirimi için kullanılan 2-B hareket vektörlerinin elde edilmesine de olanak tanımaktadır. 3-B hareket modeli formülasyonu ise hem katı hem de katı olmayan nesnelere için ayrı ayrı yapılmıştır. Biçim değiştirebilir nesnelere için modelleme komşulukları arasında esnek ilişkilere izin veren Markov rasgele alanları yardımıyla yapılırken, katı 3-B hareket E-matrisi yöntemi kullanılarak bulunmaktadır. E-matrisi metodu, yanlış bölütlemenin sebep olduğu nesneye ait olmayan hareket vektörlerine bağlı hatalara karşı gürbüzleştirilmiştir. Derinlik kestirimi için iki ayrı metod önerilmiştir. Bu metodlarda derinlik alanlarının gürültüye dayanıklı ve verimli kodlanmaya elverişli kestirimi sırasıyla MAP ve hız bozulma kuramı kullanılarak başarılmıştır. 3-B hareket modellerinin video kodlama uygulamalarında diğer kullanım alanları olarak zamanda hareket dayalı aradeğerleme ve açılan-örtülen bölge tespiti de sayılabilir. Yapay ve gerçek veriler ile yapılan deneylerde önerilen tüm metodların 2-B harekete dayalı benzer modellere karşı üstünlük sağladığı gözlenmiştir.

Anahtar Çok düşük bit hızlarında video sıkıştırma, nesneye dayalı kodlama, 3-B hareket
Sözcükler: kestirimi, 3-B derinlik kestirimi, Markov rasgele alanları, bölütleme, 2-B hareket kestirimi, hız-bozulma kuramı, zamanda aradeğerleme, açılan-örtülen bölge tespiti

Acknowledgment

I would like to express my deepest gratitude to Prof. Levent Onural for his supervision and encouragement in all steps of the development of this work. He was enthusiastic to help whenever I needed.

My special thanks go to Dr. Tanju Erdem for inspiring discussions on many topics in this thesis during his one year visit to Bilkent University.

I like to acknowledge the financial supports of Bilkent University, TÜBİTAK through COST 211ter Project, and BAYG BDBP Programme, and IEEE Turkey Section for making the presentations of this work in the national and international conferences, possible.

I am also indebted to the “lunch-time-gang”; Noyan, Tunç, Fatih, Uğur, Kerem, Ayhan, Güçlü, Ertem, Dr. Erzin and Dr. Akar for sharing many pleasant moments with me.

My sincere thanks go to my family for their love, patience and continuous moral support throughout my graduate study. It is their unhesitating self-sacrifice which has enabled me to achieve my goals in my life.

Finally, Lale, who has made my life much easier and colorful during my PhD study, deserves the most among all these acknowledgements.

Contents

Acknowledgment	iii
Contents	iv
List of Figures	vii
List of Tables	xii
1 Introduction	1
1.1 Motion Analysis using Video Sequences	2
1.2 Object-based Video Coding	4
1.3 Structure of the Dissertation	6
2 Object-based 2-D Motion Analysis	8
2.1 Segmentation of Moving Objects	9
2.2 Object Segmentation using Recursive Shortest Spanning Tree (RSST) . .	11
2.2.1 RSST based Image Segmentation	11
2.2.2 Improved-RSST for Object Segmentation from Video	12

2.3	Gibbs Formulated Object Segmentation	13
2.3.1	Formulation of Gibbs Energy	14
2.3.2	Minimization of Gibbs Energy	16
2.4	A Hybrid Object Segmentation Method	17
2.4.1	The Algorithm	18
2.4.2	Simulations	19
2.4.3	Discussion on Object-based Motion Analysis	22
3	3-D Motion Estimation	24
3.1	Current Methods on Estimating 3-D Motion	25
3.1.1	Rigid Motion	26
3.1.2	Non-rigid Motion	30
3.2	Proposed Object-based Rigid 3-D Motion Estimation Method	33
3.2.1	Description of the Algorithm	34
3.2.2	Simulations	37
3.3	Proposed Object-based Non-rigid Motion Estimation Method	42
3.3.1	Gibbs Model based Non-rigid Motion Estimation	44
3.3.2	Simulations	48
3.4	Discussion on the Motion Models	52
4	Depth Analysis in 3-D Motion Models	55
4.1	Noise Immune Depth Estimation	56

4.1.1	Formulation	57
4.1.2	Simulations	59
4.2	Optimal Depth Estimation and Encoding	67
4.2.1	Theoretical Limits of Depth Encoding	68
4.2.2	Selection of Encoding Criteria	69
4.2.3	Simulations	75
4.3	Discussion on Depth Estimation and Encoding	82
5	Utilization of 3-D Motion for Occlusions and Temporal Interpolation	84
5.1	Detection of Occlusion Areas using 3-D Motion Models	84
5.1.1	Improved Detection of Occlusion using 3-D Motion	86
5.1.2	Simulations	88
5.1.3	Discussion	88
5.2	Motion Compensated Temporal Interpolation	90
5.2.1	Temporal Interpolation using 3-D Motion Models	93
5.2.2	Simulations	95
5.2.3	Discussion	95
6	Conclusions	97
6.1	Contributions	97
6.2	Possible Future Topics	99
Vita		116

List of Figures

2.1	3-D coordinate system	16
2.2	Different levels and propagation of minimization for Multiscale Constrained Relaxation algorithm, when it is applied to 2-D motion estimation problem.	18
2.3	Original (a)10th and (b)16th frames of Salesman sequence. (c) The difference image	20
2.4	RSST-based segmentation using (a) only intensity (b) only motion (c) both intensity and motion information	21
2.5	The results for Hybrid Method : (a) 2-D motion estimation, (b) object segmentation, (c) reconstruction of frame 16 (Temporally Unpredictable regions are shown with white regions) using motion data.	21
2.6	Original (a)100th and (b)103th frames of Foreman sequence. (c) The segmentation result using hybrid algorithm.	22
2.7	Original (a)38th and (b)41th frames of Mother and Daughter sequence. (c) The segmentation result using hybrid algorithm.	22
3.1	(a), (b) Consecutive original frames of <i>Cube</i> sequence. (c) Ideal motion parameter value for w_z shown as an intensity representation.	48

3.2	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 are true, where as solid lines are the estimated values.	49
3.3	(a) The estimated and (b) true needlegrams of “Cube” on the reconstructed frames.	50
3.4	Original (a) first and (b) second frames of <i>Cubes</i> sequence	50
3.5	Histogram of t_y parameter for “Cubes”. True values are shown using solid, whereas the estimates with dotted lines.	51
3.6	The estimation results of motion parameters (a) w_x and (b) t_y for input frames with different SNR_{peak} values.	51
4.1	Depth estimation using MAP formulation.	57
4.2	The proposed rigid 3-D object-based motion and depth estimation scheme which can be used in object-based video coding.	59
4.3	Original (a)first and (b)second frames of <i>Salecube</i> sequence. (c) The ideal segmentation result.	61
4.4	Second frame from <i>Salecube</i> sequence. The results after noise injection :(a) 35 dB, (b) 25 dB, (c) 15 dB.	61
4.5	The needlegram representation of the motion between first and second frames of <i>Salecube</i> sequence. The results are obtained for (a) true, (b) noise-free, (c) 15 dB cases.	62
4.6	The mesh representations of the depth fields for the second frame of <i>Salecube</i> sequence. The results are obtained for (a) true, (b) 15 dB E-matrix, (c) 15 dB proposed algorithm, cases.	64

4.7	The needlegram representations of the 2-D motion field which is obtained after projecting the estimated 3-D motion and depth field of the second frame of <i>Salecube</i> sequence. The results are obtained for (a) true, (b) 15 dB E-matrix, (c) 15 dB proposed algorithm, cases.	64
4.8	The reconstructed frame of the second frame of <i>Salecube</i> sequence which is obtained using the the projected 2-D motion field of the estimated 3-D motion and depth field. The results are obtained using MAP-based method for (a) noise-free, (b) 45 dB, (c) 25 dB, cases.	65
4.9	The depth maps of the 41th frame of the <i>Mother and Daughter</i> sequence. The results are obtained using the E-matrix method for (a) noise-free, (b) 15 dB cases and also MAP-based method for (c) noise-free, (d) 15 dB, cases	66
4.10	The reconstructed noise-free 41th frame of <i>Mother and Daughter</i> sequence which is obtained using the the projected 2-D motion field of the estimated 3-D motion and depth field. The results are obtained for noise-free cases. (a) Original (b) using E-matrix method (c) MAP-based method.	67
4.11	Rate (\mathcal{B}) versus Distortion (Δ)	68
4.12	Bit-rate vs. distortion curve for computed and experimental bit-rate values for “Cube” sequence for different values of λ (tabulated in Table 4.5).	77
4.13	The mesh representations of the (a) true and (b) encoded depth fields of the current frame of the “Cube” sequence.(c) Depth field with intensity description (color-bar shows the depth levels with respect to intensities). Note that the assigned depth values for the background is dummy since it can not be determined by any means.	78

4.14	The experimental results for “Cube” sequence; (a) Segmentation, (b) Reconstructed frame using encoded depth with TU areas detected, (c) The projection of 3-D motion as a “needlegram” ($D_{2D}(\hat{Z}(\mathbf{x}, t))$) of Equation 4.9 is represented by the vector whose direction is from the thicker end to the thinner end of the pin where the thinner end shows $\mathbf{x}(t)$, (d) TU areas (white).	79
4.15	For different values of λ , corresponding rate-distortion pairs;	80
4.16	For the segmented head, (a) Encoded depth field and (b) reconstructed frame using the encoded depth field and motion parameters, for $\lambda = 5$. .	81
4.17	The results of 3-D motion and depth estimation for <i>Salesman</i> sequence;(a) Motion compensated current frame using 3-D motion parameters and encoded depth field (TU areas are segmented) (b) Needlegram of 2-D projection of 3-D motion; Encoded depth field with (c) mesh and (d) intensity representations.	82
5.1	The epipolar constraint.	87
5.2	The occlusion regions for the second frame of the <i>Salecube</i> sequence. The results are obtained for 2D and 3-D motion models; (a) Temporally Unpredictable regions using 2-D motion, (b) Reconstructed frame by the help of 2-D motion, (c) Temporally Unpredictable regions using 3-D motion, (d) Reconstructed frame by the help of 3-D motion and structure.	89
5.3	The occlusion regions for the 41th frame of the <i>Mother and Daughter</i> sequence. The results are obtained for 2D and 3-D motion models; (a) Temporally Unpredictable regions using 2-D motion, (b) Reconstructed frame by the help of 2-D motion, (c) Temporally Unpredictable regions using 3-D motion, (d) Reconstructed frame by the help of 3-D motion and structure.	90

5.4	The occlusion regions for the 16 th frame of the <i>Salesman</i> sequence. The results are obtained for 2D and 3-D motion models; (a) Temporally Unpredictable regions using 2-D motion, (b) Reconstructed frame by the help of 2-D motion, (c) Temporally Unpredictable regions using 3-D motion, (d) Reconstructed frame by the help of 3-D motion and structure.	91
5.5	Motion compensated temporal interpolation with the corresponding motion trajectories of 2-D and 3-D models and occlusion areas.	92
5.6	The original first 3 frames of <i>Salecube</i> sequence. (a) First, (b) second and (c) third frame.	95
5.7	The error between the original second and the interpolated frame using (a) 2-D motion and (c) 3-D motion models. The reconstructed second frame using temporal interpolation by the help of (b) 2-D motion and (d) 3-D motion model.	96

List of Tables

3.1	Simulations on E-matrix method using artificial data	39
3.2	Simulations on 3-D motion parameter estimation using the conventional E-matrix method using 10th and 16th frames of <i>Salesman</i> sequence . . .	40
3.3	Simulations on 3-D motion parameter estimation using the proposed method using 10th and 16th frames of <i>Salesman</i> sequence	41
3.4	Simulations on 3-D motion parameter estimation using the proposed and conventional E-matrix method using 100th and 103th frames of <i>Foreman</i> sequence	42
3.5	Simulations on 3-D motion parameter estimation using the proposed and conventional E-matrix method using 38th and 41th frames of <i>Mother and Daughter</i> sequence	42
4.1	Noise analysis of 2-D motion estimation step for <i>Salecube</i> sequence. Two quality parameters, $Q_{1,2}$ and the SNR_{peak} of the reconstructed second frame are tabulated for different noise levels.	62
4.2	The results of the noise analysis of depth fields for <i>Salecube</i> sequence using E-matrix method.	63
4.3	The results of the noise analysis of depth fields for <i>Salecube</i> sequence using <i>MAP</i> -based formulation	63

4.4	Noise analysis of 3-D motion estimation step for 38th and 41th frames of <i>Mother and Daughter</i> sequence. Five test parameters, $T_{1,2,3,4,5}$ and the performance indicator, P , are tabulated for noise-free and noisy cases. . .	65
4.5	The experimental results for <i>Cube</i> sequence. For different values of λ , Equation 4.15 is minimized to obtain Δ and \mathcal{B} (with $k = 0.5$) values. Bit-rate of the depth field is obtained after lossless encoding of the prediction error field.	76
4.6	For different values of λ , Equation 4.6 is minimized to obtain Δ and \mathcal{B} (with arbitrary $k = 0.5$) values. Bit-rate is obtained after encoding of the prediction error.	80
4.7	The experimental results for “Salesman” sequence. For each object and different values of λ , Equation 4.15 is minimized to obtain the corresponding Δ and bit-rate values.	81

Chapter 1

Introduction

Over the last 20 years, many scientists from image processing and computer vision community have been trying to match image points, i.e., pixels, correctly between consecutive video frames for different reasons. In this dissertation, a different motion model is examined for pixel matching. Moreover, the results are used in a different application area, called object-based video compression.

Lossy video compression is a process similar to “orange juice extraction”; squeeze the orange and take the most necessary part out of it. However, there is more about this analogy : after taking the glass of orange juice from the kitchen to the customer’s table, we also have to obtain the orange back at the table from the juice itself! During the last decade, a significant amount of research was devoted to extracting the juice in the most efficient way, so that with minimum amount of juice, the “best” orange can be obtained back in the table. In practice, by sending only the most necessary information about a frame sequence from the transmitter and reconstructing the frames at the receiver side with minimum distortion, significant amount of gain is obtained for both transmission and storage of video data.

Ongoing research on video compression has proven that most of the redundancy in video data is in between frames, i.e., in the temporal domain. Hence, the common trick

is analyzing the motion of pixels between frames and predicting the intensities of the encoded frame from the previous available frames using the obtained motion information, i.e., the “juice”.

As a preliminary step and for motivation, some other terms and concepts associated with this dissertation are explained in Sections 1.1 and 1.2

1.1 Motion Analysis using Video Sequences

Motion analysis can be divided into two main classes: motion estimation and utilization of the estimated motion. This dissertation is concerned with both estimation and utilization. Motion estimation is a process which can be simply explained as “image frames in, motion information out”. More formally, the determination of the movement of image pixels by observing two or more consecutive frames is called motion estimation. The movement of not only 2-D image pixels, but also the 3-D object points, which generate the corresponding image points after projection from 3-D world, are also within the scope of motion analysis. After the analysis step, the obtained motion information can be utilized in video coding as well as in some others areas, such as robot navigation, obstacle avoidance, target tracking, traffic monitoring, motion of biological cells and weather systems (cloud) tracking [1].

All the application areas mentioned above require a successful estimate of the motion field which is indeed difficult to obtain in general. Currently, there is no method which estimates the motion correctly from visual data in a complex scene without making assumptions. Moreover, it can be predicted that it will be very difficult to obtain an “ideal” motion estimator in the near future, realizing the difficulty even for an intelligent human (who might be confused with the motion of the “barbers pole” in the scene!). The complexity of the problem results from a number of reasons. First of all, the visual data can be perceived different compared to the “real” motion, as in the case of barbers pole. Moreover, by projecting a 3-D scene onto a 2-D image plane, some information is

also lost. As a simple example, the 3-D motion of an object will look exactly the same in the image plane, when a second object moves twice as fast as the first object at a distance twice the distance of the first. Obviously, this is true when the projection of the 3-D world into the image plane is perspective. Without knowing anything about the environment and the object, such a problem is impossible to solve. In order to observe and estimate motion, the moving object should have some discriminatory features, such as texture or simply a spatial gradient information on itself. Motion of a mat white ball, which is rotating around an axis or translating in front of a background with the same mat texture while there is constant illumination in the environment, can not be detected. The estimation of such a motion using only visual data is impossible. Finally, noise, which is always present in the real world, makes motion estimation methods fragile in many cases. Apart from these problems, there might be many other difficulties when the problem is tried to be solved by using a computer.

Realizing that most of the problems occur due to observation of the scene through video frames, the insistence on using video as the observation data for motion analysis can be quite questionable. Although, lasers or acoustic sensors can be more precise and successful for analyzing the motion, when the aim (or utilization) is video coding, the observation data becomes completely determined. Even stereo video frames, which can be quite helpful to tackle problems, can not be utilized due to the same reason. It is obvious that many difficult problems, which are tried to be solved in this thesis, can be handled very easily if the stereo views of the scene are available. Hence this dissertation is devoted to analyze motion which is always observed through monocular video frames.

Analysis is possible by the help of a model. In motion analysis, motion models can be examined in two classes. A method, which tries to find the 3-D dynamics and structure of the objects in a scene by the help of video data, should utilize a 3-D motion model. The rest of the methods is assumed to be 2-D motion-based methods. All the 2-D motion models in the literature can be simply summarized with the assumptions of smoothness between neighboring 2-D motion vectors and intensity matching between frames. On the other hand, the models for 3-D motions are obtained using the theorems of kinematics.

However, for video coding purposes 2-D motion models have always been more popular, due to their simplicity in both modeling and computation. On the other hand, 3-D motion models have been more utilized in computer and robotic vision applications.

In this dissertation, 3-D motion models will be examined from a video coding point of view. Since the performance of the coding algorithms utilizing 2-D motion models has been almost saturated, new approaches should be explored. The strong theory behind 3-D models, huge amount of related work by computer vision researchers and the description (i.e. encoding) simplicity of 3-D motion put this approach as a strong candidate for an alternative to the current motion models. Due to some specific requirements in video coding, the previous research on 3-D motion analysis needs some adjustments and as well as some improvements which are achieved in the third, fourth and fifth chapters of this dissertation.

1.2 Object-based Video Coding

In video coding applications, it is sometimes necessary to divide the observed scene into a number of regions with semantic meanings. Such regions are called objects in video coding. In order to define (segment) objects, a possible approach is to utilize motion and intensity information in the image frames. In such an approach, a 2-D region with intensity and motion coherence defines an object. Although this region is supposed to be a projection of a moving body in 3-D world, this situation can not be guaranteed in each case. Moreover, if recognition of the objects is not the principal aim, then it is also not strictly necessary to obtain one region for the projection of the 3-D object. Since the main purpose of video coding is efficient compression, a region with maximum texture and intensity coherence should be preferable. While the motion coherence is expected to be more effective for locating the objects in the image, intensity information is usually necessary for obtaining finer boundaries. In order to understand the reason for defining objects in video coding applications, the history of video coding must be examined.

First compression algorithms were aimed to encode still images rather than sequences. Discrete Cosine Transform (DCT) has become the winner of still image coding problem and this transform has initiated a still image coding standard, called Joint Photographic Expert Group (JPEG) standard. Extensions of DCT-based algorithms were developed for video sequence compression by compensating the motion information between frames and these approaches turned into a number of standards (e.g., Moving Picture Expert Group (MPEG) 1,2 or ITU's H.261, H.263) for video coding applications, such as teleconferencing or videophony [2]. Afterwards, the blurring effect of DCT has initiated a new approach in still image coding, called *second generation image compression* which is basically a region-based coding [3]. Region-based coders usually work with the principal of encoding the boundaries and texture of the regions within those boundaries separately, and hence, they eliminate the blurring effect of DCT considerably. Second generation approaches have been able to reach similar (for some bit-rates better, especially subjectively) performances compared with DCT, but there were no significant improvements, especially when computational complexity of the algorithms are also taken into account [4]. Hence a still image coding standard, which uses region-based algorithms, was not achieved. In the second half of 1980's, when the performance of DCT-based video compression algorithms were found out to be saturated for low bit-rates, a new generation has emerged for video coding algorithms [5]. Similar to region-based coding, moving objects were defined, segmented and encoded separately, in order to achieve maximum coding efficiency for each separate object. Some demands from the market for object-based video transmission and storage have also supported the ongoing research for this type of approach and currently the standardization issues continue (expected to end up around 1998) for MPEG-4 which will be an object-based video encoding standard.

Obviously, it will not be fair to compare object and DCT-based algorithms, since the research on object-based methods currently continues, while DCT-based algorithms are very mature. In the early days of object-based coding, the DCT-based approaches had a superiority over these immature algorithms, even at low bit-rates. In the preliminary subjective tests of MPEG-4, the anchor algorithm H.263, which is DCT-based, has

surpassed all the other proposed object-based algorithms. However, currently new object-based algorithms are challenging the H.263 with better results. This result might be expected, because in very low bit-rates, object-based algorithms do not have blurring effects in contrast to their DCT-based counterparts. By handling each motion and intensity coherent entity separately, the coding efficiency improves. However, some increase in computational complexity is inevitable for object-based approaches and this problem is expected to be solved by the increased processing power.

In this dissertation, a promising, market-demanded and “hot” object-based approach is selected to search for new horizons in video compression. However, the aim of this dissertation is to create some tools which can be utilized in some different parts of a full object-based video coder, rather than constructing this full codec. It should be noted that currently there are only a few full object-based coders which are the outcome of joint research of some video coding groups around the world and the research on these codecs still continue. Hence, obtaining a full object-based codec is beyond the scope of this dissertation.

Another important concept worth to mention is the relation between objects and 3-D motion models. Obviously, it is not suitable to apply 3-D motion models in a block-based manner as it is usually done in many video coding algorithms. Since 3-D motion belongs to an object, the analysis should be conducted on the projection of this object in the image plane. If the image is segmented (correctly) into some regions, which represent the projections of 3-D objects, the 3-D motion analysis will be more effective and meaningful. Hence, compared to block-based schemes, 3-D motion analysis is more suitable for region-based approaches.

1.3 Structure of the Dissertation

After this preliminary introductory chapter, the next chapters explain some new tools that can be utilized in a 3-D motion modeled object-based video coder.

The second chapter of this dissertation is devoted to object-based segmentation. After a brief overview on current segmentation approaches, three new algorithms are explained. All three methods take two video frames (the data) as inputs and generate the segmentation masks and 2-D motion information as outputs. Simulations show that the algorithms have different performances, and the best object segmentation method is chosen among three according to the simulation results.

3-D motion is estimated in the third chapter, using the segmentation masks and 2-D correspondences which are obtained in the second chapter. Initially, some 3-D motion estimation methods in computer vision literature are examined and compared in order to choose the one which is more appropriate for video coding applications. Rigid and non-rigid motions are analyzed separately and an algorithm for each is proposed. After some simulations, the performances of the algorithms are examined and discussed.

The fourth chapter examines the structure of the 3-D objects. Since noise immune estimation and efficient encoding is strictly necessary for a successful video coder, the depth analysis, which tries to find the 3-D structure of the moving objects in the scene, is one of the most dominating factors in the performance of a video coder with a 3-D motion model. The estimated 3-D motion information in the previous chapter is used in two different algorithms : one of the algorithms yields a robust depth field estimate and the other one encodes this field. Some similarities in the formulation of these two algorithms and some simulation results are discussed at the end of fourth chapter.

The fifth chapter is devoted to further utilization of 3-D motion in video coding applications, apart from motion compensated prediction. Two main subjects, occlusion detection and motion compensated interpolation, are examined together with the available 2-D motion-based methods. New methods are proposed to solve these problems. Some simulation results are given in order to compare both methods with the current 2-D motion modeled approaches.

The last chapter concludes this dissertation by summarizing the contributions and giving some possible future research topics.

Chapter 2

Object-based 2-D Motion Analysis

Object-based motion analysis deals with both segmentation of the scene into the objects and estimation of the motion of these objects by the help of the input frames. Before this analysis, a short look on current motion estimation methods can be quite helpful. All the methods mentioned below estimate 2-D motion and they can be applied for object-based motion analysis.

In the 70's, the first algorithms to calculate the motion of an object from television signals were proposed [6]. Later, the segmentation of moving objects were also taken into account in some algorithms [7]. However, these methods are far from achieving successful motion estimates in natural complex scenes. Afterwards, an important contribution for the estimation of 2-D motion came from Horn and Shunck [8] as the concept of *optic flow* which relates the 2-D motion vectors to spatio-temporal gradients of the image with the assumption that intensity of a moving point does not change along its motion trajectory. Since the ill-posedness of this problem has been overcome by imposing smoothness on 2-D motion vectors, the obtained motion boundaries are usually blurred. Afterwards, based on optic flow concept many different algorithms were devised [9]. Later, motion estimation methods began to find direct applications in the field of video compression. In these methods, the “correct” motion vector is the one which minimizes the intensity difference between frames and these methods can be classified

into two classes according to the transmission of motion information. In the first class both receiver and transmitter estimate motion. Hence the motion is not sent as an extra information and *pel-recursive* [10] algorithm is the most well-known example of this class. If the motion is estimated only at the transmitter, as in *block-matching* [11] algorithm, which belongs to the second class, this motion information has to be sent to the receiver side as an overhead. The experimental results show that block-matching type of algorithms have better performance compared to their pel-recursive counterparts. However both types have limited performance, since their motion estimates do not represent the “true” projected 3-D motion which is very difficult to estimate without good models [12].

Powerful 2-D motion modeling is achieved by using *Markov Random Fields* (MRF). These approaches model 2-D motion in such a way that the 2-D projection of 3-D rigid, or even non-rigid, motion can be represented by the help of some local interactions between neighboring motion vectors [13, 14, 15, 16]. MRF-based methods have high performance in modeling and estimating the 2-D motion, but they have also high computational complexity. There are also *2-D parametric motion models* which try to fit usually a quadratic function to the motion field of each object [17, 18, 19]. In order for these models to be valid, such methods make some assumptions for the motion and structure of the moving objects. Their performances usually degrade for large displacements.

After this short overview on current 2-D motion estimation methods, a closer look is presented for the segmentation problem in the next section.

2.1 Segmentation of Moving Objects

Current object segmentation approaches can be divided into three classes as *direct intensity based*, *motion vector based* and *simultaneous motion estimation and segmentation* methods [2]. The direct intensity based methods use spatio-temporal intensity information, instead of motion estimates and they usually find a change detection mask

which separate moving and stationary regions as a segmentation output [20]. In case of noise and illumination changes, these methods need powerful post-processing algorithms to eliminate the small irrelevant regions. Moreover, the intersected moving multiple objects can not be detected, either. Motion vector based segmentation is similar to image segmentation, except motion information is being used instead of intensities. Given the motion vectors, the scene can be segmented into a pre-determined number of regions using K-means algorithm, modified Hough transform [21] or Bayesian segmentation [22]. There are also some methods based on simultaneous estimation of both motion and segmentation fields from the intensity information of the consecutive frames [23]. Such methods usually utilize MRF to model both motion and segmentation fields together.

The principal difficulty of object segmentation can be explained as follows : In order to segment objects, successful motion estimates are necessary, especially at the motion/object borders. Since most of the motion estimation methods use smoothing (regularization) functions, it is difficult to obtain sharp boundaries using such algorithms. On the hand, the object borders, i.e., successful segmentation, are required for obtaining sharp motion boundaries and good motion estimates. Ironically, both segmentation and motion estimation need a successful estimate of each other to obtain good results. Therefore, among three object segmentation classes, methods based on the first two classes have limited performance, whereas simultaneous estimation and segmentation method looks as the only possible solution for better results. Hence, in this dissertation, an algorithm, which simultaneously estimates 2-D motion and segmentation fields, is proposed in Section 2.3. Some drawbacks of this algorithm is also tried to be minimized by an improved version in Section 2.4.

Before examining the proposed segmentation methods, the reason for choosing 2-D motion models in the segmentation step can be explained as follows : In order to analyze the 3-D motion of objects, the initial step is to segment the moving objects in the scene using a motion model. On the other hand, rigid 3-D motion estimation algorithms usually (and also in this dissertation) require 2-D correspondences between

consecutive image frames. Hence, the best way is to make segmentation using a 2-D motion model, since the required 2-D correspondences can also be obtained at the same time. Furthermore, better pixel correspondences for the segmented objects will be achieved by simultaneously segmenting the scene and estimating 2-D motion of the objects.

In the next sections, three object segmentation algorithms will be examined. The first method belongs to the class of motion vector based segmentation and it is an extension of a powerful image segmentation algorithm. Second algorithm simultaneously segments and estimates motion and it is based on Gibbs formulation. The last one is a hybrid method which utilizes both of the first two algorithms in an appropriate way.

2.2 Object Segmentation using Recursive Shortest Spanning Tree (RSST)

As it is discussed previously, the simultaneous approaches are expected to achieve better results for segmentation and motion estimation with a significant increase in computation time. However, it might be necessary to obtain fast and acceptable estimates for segmentation and 2-D motion in some applications. Moreover, such results can also be utilized as initial estimates to any computationally demanding algorithm. In the following sections, a novel approach to obtain such estimates is explained.

2.2.1 RSST based Image Segmentation

Graph theory can be applied to image segmentation by Recursive Shortest Spanning Tree (RSST) method [24]. This method is also used in still-image compression [4]. The RSST algorithm maps the original image into a graph so each node (region) initially contains only one pixel. Sorted link weights, which are associated with the links between neighboring regions in the image, are used to decide which link should be eliminated

and therefore which regions are merged. The link weights are usually chosen as the difference between neighboring region intensities. After each merge, the link weights are recalculated and resorted. Thus, the number of regions is progressively reduced from $N \times M$ (for an image size N by M) down to, if desired just one [4]. The removed links define a spanning tree of the original graph [24]. By noting the order in which the links are eliminated, the image can be segmented into K regions by using the last removed $K - 1$ links.

RSST has the advantage of not imposing any external constraints on the image. Some other methods, such as split-merge algorithm, which requires segments consisting of nodes of a quadtree, can produce artificial region boundaries. Furthermore, RSST segmentation permits simple control over the number of regions and therefore amount of detail in the segmentation image. The simulation results on still images support the superior image segmentation performance of this method [24, 4].

2.2.2 Improved-RSST for Object Segmentation from Video

Object segmentation has similar properties to image segmentation and these similarities can be used to develop new algorithms. While only intensity information is used for image segmentation, object segmentation should be achieved by using both motion and intensity information. Hence, a pre-determined motion data is needed for object segmentation.

Since the aim is to devise a fast algorithm among different 2-D motion estimation methods, which are shortly examined in the previous section, block-based algorithms are most preferable due to their lower computational load. The hierarchical application of the block-based algorithms also give better results for large displacements, as in the method called Hierarchical Block Matching (HBM) [25]. It should also be noted that in order to obtain a dense 2-D motion field with a block-based motion estimation algorithm, the locations between block-motion vector positions are interpolated bilinearly.

The object segmentation can be achieved by using RSST with a proper selection

of “link weights” between regions, i.e., objects. Since every point on the image has a corresponding motion vector as well as an intensity, the new link weights can be selected as the norm of a difference vector between objects. This difference vector will consist of three elements which are the intensity, and, the horizontal and vertical components of the 2-D motion vector at that region. However, there should be a “weighting parameter” which adjusts the relation between intensity and motion information. This weighting parameter and the number of regions to segment are important factors which determine the performance of such an algorithm. Since there is no quick way to find a weighting parameter if no extra constraint or information is available, an ad-hoc, but feasible solution is to give equal weights to intensity and motion, after a proper normalization is achieved for motion information. There is also no drawback to select the number of objects higher than the true value if there is a global object-merge mechanism afterwards.

The method described above is expected to be fast, but not optimal due to the consecutive application of motion estimation and motion segmentation steps. Since the motion estimation step is achieved without any segmentation information, motion boundaries are expected to be inexact. Although, the intensity term in the link weight vector might compensate for this loss by obtaining better object contours, this can not be guaranteed for each case. The simultaneous approach in the next section is expected to overcome all these drawbacks if sufficient amount of computation can be devoted.

2.3 Gibbs Formulated Object Segmentation

Over the last 20 years, many researchers have been using Markov Random Field (MRF) models, i.e. Gibbs energies, for developing robust algorithms to solve different image processing problems. The basic definitions about MRF and Gibbs modeling can be found in [26]. There are various applications of Markov Random Field modeling, such as image restoration [27], stereo vision and disparity measurement [28, 29], modeling and segmentation of textured and noisy still images [30], texture generation [31], sequence restoration [32, 33], object recognition [34], shape from texture [35], scene segmentation

using motion data [22], partial shape completion [36], image deblurring [37], edge modeling [38] and finally 2-D motion estimation [13, 14, 39, 40, 41, 15, 42, 23, 43, 16]. The advantage of using MRF models comes from developing systematic algorithms based on mathematical principles. A simple cost function (Gibbs energy) might take all the a-priori knowledge into account as constraints and model the problem successfully as a maximum entropy problem. The Gibbs formulation for 2-D motion estimation has an important contribution in 2-D motion modeling by imposing smoothness among neighboring motion vectors and intensity matching between intensities.

Since Gibbs formulation allows incorporating prior contextual information or constraints into the problem easily, object segmentation could also be inserted into Gibbs formulated 2-D motion estimation by the help some extra variables. *Line* [27, 14, 15] and *region* [44, 23] fields are used to segment objects after these variables are appropriately inserted into the original Gibbs energy which is used to estimate 2-D motion. While line field only detects the motion discontinuities, region field gives an object tag to every motion vector accordingly. Hence, the region field is more appropriate for object-based motion analysis. Apart from object segmentation, detection of *temporally unpredictable* (TU) areas, which are newly exposed or covered by the moving object, is also possible using the MRF models [45, 43, 44]. More emphasis is given for TU areas in Chapter 5. However, all these fields modify the original Gibbs energy in such a way that the resulting function becomes non-convex and difficult to minimize.

The aim is to generate a sophisticated Gibbs energy, which not only estimates 2-D motion, but also segments objects both using motion and intensity information, and detects TU areas. 2-D motion estimation, segmentation and TU detection are all necessary to obtain the robust 3-D motion estimates for the objects.

2.3.1 Formulation of Gibbs Energy

The Gibbs energy function \mathcal{U} , which is the negative exponent of the exponential joint probability density function of 2-D motion \mathcal{D} , *segmentation* \mathcal{R} and *temporally*

unpredictable \mathcal{S} fields, can be written as follows

$$\mathcal{U}(\mathcal{D}, \mathcal{R}, \mathcal{S} | \mathcal{I}_t, \mathcal{I}_{t-1}) = \mathcal{U}_n + \lambda_m \mathcal{U}_m + \lambda_R \mathcal{U}_R + \lambda_s \mathcal{U}_s \quad (2.1)$$

in which

$$\begin{aligned} \mathcal{U}_n &= \sum_{\mathbf{x} \in \Lambda} (I_t(\mathbf{x}) - I_{t-1}(\mathbf{x} - \mathbf{D}(\mathbf{x})))^2 (1 - S(\mathbf{x})) + S(\mathbf{x})T_s \\ \mathcal{U}_m &= \sum_{\mathbf{x} \in \Lambda} \sum_{\mathbf{x}_c \in \eta_{\mathbf{x}}} \|\mathbf{D}(\mathbf{x}) - \mathbf{D}(\mathbf{x}_c)\|^2 \delta(R(\mathbf{x}) - R(\mathbf{x}_c)) \\ \mathcal{U}_R &= \sum_{\mathbf{x} \in \Lambda} \sum_{\mathbf{x}_c \in \eta_{\mathbf{x}}} [1 - \delta(R(\mathbf{x}) - R(\mathbf{x}_c))] + \lambda_t \frac{[1 - \delta(R(\mathbf{x}) - R(\mathbf{x}_c))]}{1 + (I_t(\mathbf{x}) - I_t(\mathbf{x}_c))^2} + \theta(R(\mathbf{x})) \\ \mathcal{U}_s &= \sum_{\mathbf{x} \in \Lambda} \sum_{\mathbf{x}_c \in \eta_{\mathbf{x}}} [1 - \delta(S(\mathbf{x}) - S(\mathbf{x}_c))] \end{aligned}$$

In the above equation, Λ is a 2-D grid on which the intensity fields, \mathcal{I} are defined. $I_t(\mathbf{x})$ is an intensity value of the frame $I_t \in \mathcal{I}$ at time t for the location $\mathbf{x} \in \Lambda$, where \mathbf{x}_c is a neighbor of \mathbf{x} . $\eta_{\mathbf{x}}$ is the neighborhood of \mathbf{x} , defined on Λ . \mathcal{D} is the unknown 2-D motion vector field, which consists of $\mathbf{D}(\mathbf{x})$ vectors which are also defined at each point on Λ (similar relations are also valid between \mathcal{R} and $R(\mathbf{x})$, and \mathcal{S} and $S(\mathbf{x})$). $\mathbf{D}(\mathbf{x})$ is defined for each \mathbf{x} on frame I_t and it shows the displacement from the corresponding point on frame I_{t-1} to \mathbf{x} on I_t . If needed, a subscript as in $\mathbf{D}_{2D}(\mathbf{x})$ is used to denote that the vector field is 2-D ($\mathbf{D}_{2D}(\mathbf{x})$ vector is shown in Figure 2.1). The true 2-D motion vectors are expected to match intensities between I_t and I_{t-1} (\mathcal{U}_n term in \mathcal{U}) and have similar values between neighbors except at object boundaries (\mathcal{U}_m term in \mathcal{U}). \mathcal{R} field is used to segment objects in the scene and prevents \mathcal{U}_m getting a high penalty at motion boundaries. \mathcal{U}_R term supports objects which have projected broad regions on 2-D image plane with textural coherence. Textural coherence is supported by giving a penalty to neighboring pixels with similar intensity values if they do not belong to the same region. Additionally some *taboo patterns*, such as single-point or cross-shaped patterns which are defined on an 8-neighborhood system are rejected by giving a high penalty, using $\theta(R(\mathbf{x}))$ term. \mathcal{S} is a binary field and shows the *temporally unpredictable* regions, in which the motion compensation error is expected to be greater than a threshold, T_s . Lastly, \mathcal{U}_s term supports \mathcal{S} field to consist of regions, instead of individual points. Similar energy functions can be found in [44, 46, 23, 47].

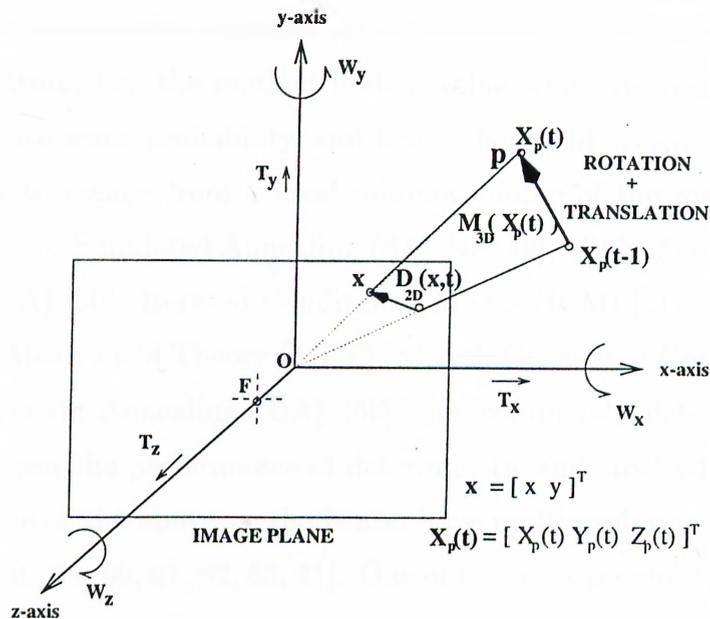


Figure 2.1: 3-D coordinate system

By minimizing the energy function, \mathcal{U} , a *Maximum A Posteriori (MAP)* estimate of the unknown 2-D motion field, segmentation field and temporally unpredictable (TU) regions can be obtained simultaneously. Hence, the scene is segmented into moving objects whose 2-D motion vectors are determined. There are different approaches for the minimization of this non-convex energy function. The algorithm which gives the best result within the shortest time should be selected.

2.3.2 Minimization of Gibbs Energy

The number of unknowns in a Markov-modeled 2-D estimation, segmentation and occlusion detection problem is extensively high : 4 unknowns per pixel. Moreover, the energy function is non-convex due to segmentation and TU fields. Therefore, the minimization of the energy function turns out to be much more difficult. The approaches, which try to minimize such energy functions, can be divided into two groups as *deterministic* and *stochastic*. Deterministic approaches assign the values in the minimization process in a “hard decision” nature. They find and apply a value

which always decreases the cost function. On the other hand, stochastic approaches have “soft decision” nature, i.e. the method finds a value which decreases or increases the energy function with some probability, and hence there will always be some possibility for the algorithm to escape from a local minima. Some of the methods, which have stochastic nature, are Simulated Annealing (SA) [48, 49], Gibbs Sampler (GS) [27] and Tree Annealing (TA) [50]. Iterated Conditional Modes (ICM) [51], Highest Confidence First (HCF) [52], Mean Field Theory (MFT) [53, 54], Graduated Non Convexity (GNC) [55] and Deterministic Annealing (DA) [56], are completely deterministic methods. Comparisons between the performance of deterministic and stochastic methods can be found in [57]. Some of the above methods also have multiresolution versions for better convergence rate [58, 59, 60, 61, 62, 63, 41]. One of the most powerful and fast algorithm of this kind is the Multiscale Constrained Relaxation (MCR) [64] method.

MCR method uses ICM at each level and the unknown variables are defined for different lattices at each scale (Figure 2.2). While the input data is defined at the finest level of lattice and does not change between resolutions, the minimization is propagated from the coarsest to the finest scale.

All the deterministic methods suffer from being trapped into local minima. Although they are computationally more efficient than stochastic methods, the experimental results on GS and ICM showed that the stochastic methods performs better from a convergence point of view [57]. On the other hand, the deterministic MCR algorithm obtains good convergence, comparable to stochastic methods, if acceptable initial estimates are used as inputs [64]. Hence, the utilization of MCR with some initial estimates looks as the most feasible choice if good and fast convergence are both necessary.

2.4 A Hybrid Object Segmentation Method

Two different algorithms are presented in the previous sections. The first method, which basically segments the motion field, has the obvious drawback of separate segmentation

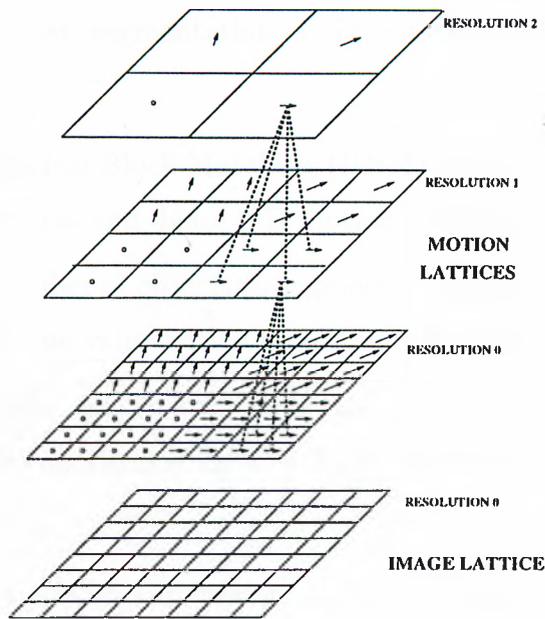


Figure 2.2: Different levels and propagation of minimization for Multiscale Constrained Relaxation algorithm, when it is applied to 2-D motion estimation problem.

and motion estimation phases. While the first method is considerably fast, the MRF-based algorithm should have better results for both motion estimation and segmentation with more computation. Obviously, both of these methods have drawbacks.

The drawbacks of these two methods can be partially eliminated by properly utilizing them in one algorithm. Since it is possible to decrease the computation time in MRF-based methods by using “greedy” minimization algorithms, the overall performance can also be conserved by using good initial estimates which can be obtained from RSST-based object segmentation algorithm. On the other hand, the RSST-based algorithm can also be improved by using the motion estimates of MRF-based approach as inputs. Taking these ideas into account, an improved algorithm is proposed in the next section.

2.4.1 The Algorithm

The hybrid algorithm can be summarized as :

1. Find coarse and fast segmentation and motion estimates using RSST-based segmentation.
 - (a) Apply Hierarchical Block Matching (HBM) algorithm and bilinear interpolation to find coarse and dense 2-D motion estimates.
 - (b) Choose the number of objects to segment. This parameter can be higher than the unknown true value without causing any significant problems.
 - (c) Segment motion and intensity by the help of RSST-based method with less emphasize on the untrustable motion information by adjusting weights appropriately.
2. Using the obtained motion and segmentation results as initial estimates, minimize Equation 2.1 using the deterministic Multiscale Constrained Relaxation (MCR) algorithm.
3. Refine segmentation :
 - (a) Use RSST-based method again, this time with more emphasize on motion data, which is obtained in the MRF-based algorithm at the previous step,
 - (b) Choose the number of objects much smaller than the previous step. If available, use some a priori knowledge about the number of moving objects.
4. Minimize Equation 2.1 again using ICM (single scale MCR) algorithm by the help of improved segmentation estimates of the previous step.

2.4.2 Simulations

In order to test the performance of the algorithms, a number of simulations are conducted. Consecutive frames from some standard video sequences are arbitrarily selected in order to perform these experiments.

In the first phase of experiments, RSST-based object segmentation algorithm is examined. Two frames (Frame 10 and 16), which are shown in Figure 2.3 (a)-(b), from

the sequence *Salesman* are selected. Significant amount of textural detail and motion are present in these frames. The amount of motion can be observed using the difference image (for better visualization, the intensity differences are augmented by tangent inverse function which is followed by a normalization) in Figure 2.3 (c).

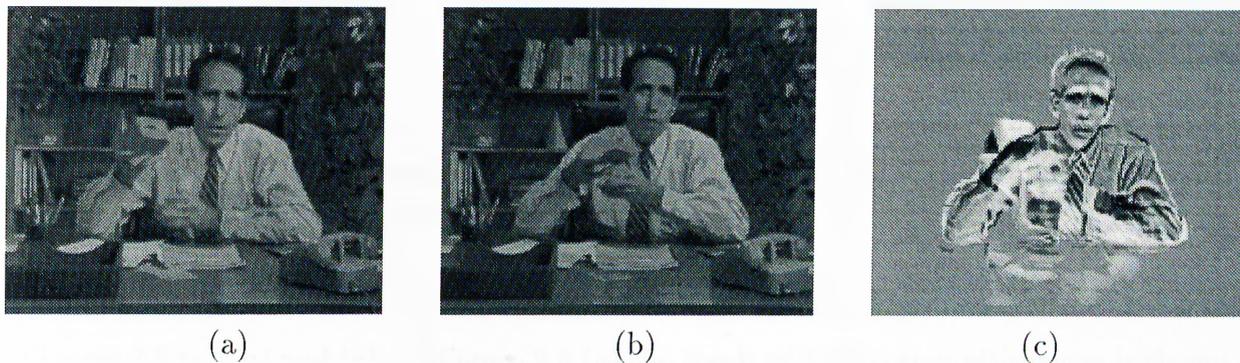


Figure 2.3: Original (a)10th and (b)16th frames of Salesman sequence. (c) The difference image

If the algorithm explained in Section 2.2.2 is applied to these frames, the results shown in Figure 2.4 are obtained for different weighting parameters. For each case, the segmentation is fixed to six regions. Figure 2.4 (a) shows the result for intensity segmentation. In Figure 2.4 (b), the regions are obtained by utilizing only motion information which is estimated using the HBM algorithm. Figure 2.4 (c) shows the result of the segmentation in which both motion and intensity information (equally weighted) are used. The experimental data shows the improved performance of using both motion and intensity over the other two, but the overall performance is still not quite acceptable.

In the second phase of the algorithms, the hybrid method, which is explained in Section 2.4.1, will be applied to the frames above as well as to some other inputs. In the hybrid algorithm, for part 1.c, the weighting parameter is chosen such that intensity information is three times more dominant than that of motion. The number of regions in part 1.b is selected as 250. On the other hand, for part 3.a, the weighting parameter is selected to support motion information with the same ratio as before. The final number of regions are selected as six in order to make comparison with the previous method. The MCR method is applied in three scales, for which only two iterations of

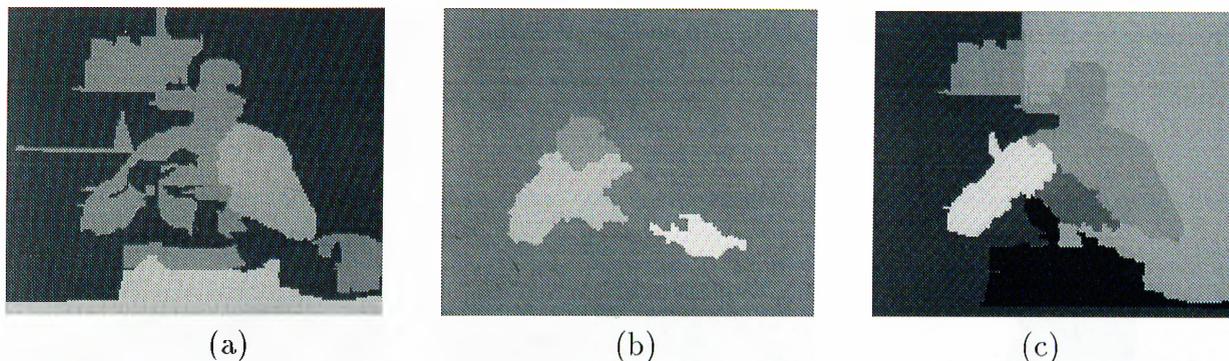


Figure 2.4: RSST-based segmentation using (a) only intensity (b) only motion (c) both intensity and motion information

ICM is used. The results, which are obtained after minimizing Equation 2.1, are shown in Figures 2.5 (a),(b) and (c). In Figure 2.5 (a) the result of 2-D motion estimation is shown by using “needlegram” representation. Figure 2.5 (b) shows the final segmentation which contains six objects : one of the regions is the stationary background, objects 2 and 5 are the occlusion regions and objects 1, 3 and 4 are the moving bodies in the scene. Frame 16 is reconstructed in Figure 2.5 (c) using the estimated motion data, while Temporally Unpredictable (TU) regions are also detected as a result of minimizing Equation 2.1. The reconstructed image has the SNR_p of 33.2 dB excluding the TU regions. As it can be clearly observed, compared to the segmentation estimates in Figure 2.4, the results of hybrid method are better and acceptable from a semantic point of view.

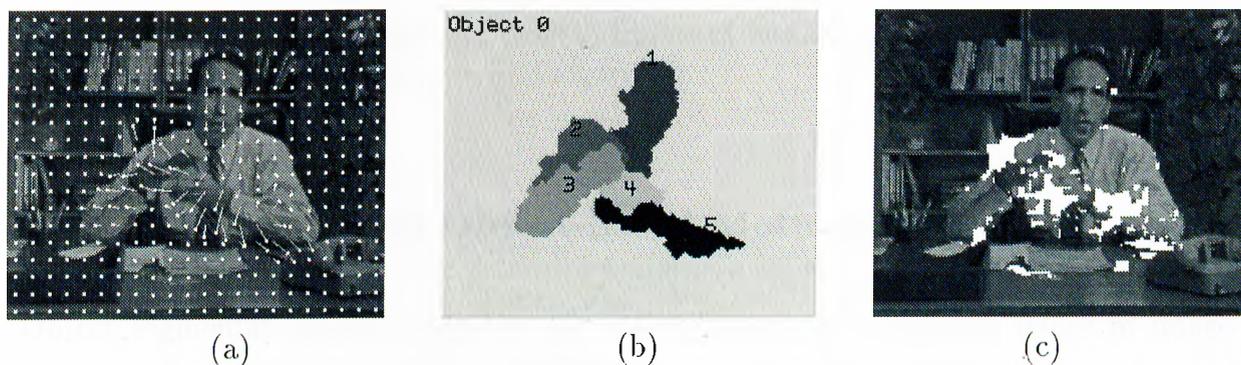


Figure 2.5: The results for Hybrid Method : (a) 2-D motion estimation, (b) object segmentation, (c) reconstruction of frame 16 (Temporally Unpredictable regions are shown with white regions) using motion data.

The results of the hybrid method are shown in Figures 2.6 and 2.7 for the sequences *Foreman* and *Mother and Daughter*, respectively. The moving heads are successfully located in both frame pairs.

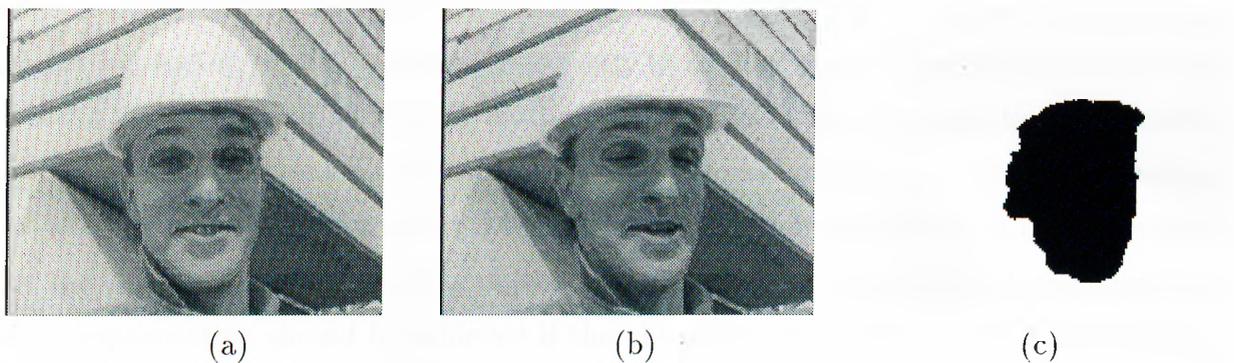


Figure 2.6: Original (a)100th and (b)103th frames of Foreman sequence. (c) The segmentation result using hybrid algorithm.

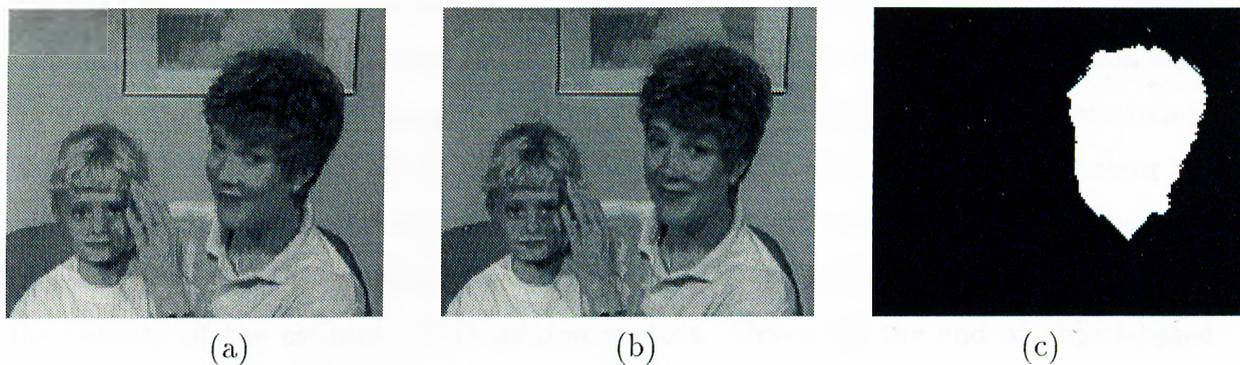


Figure 2.7: Original (a)38th and (b)41th frames of Mother and Daughter sequence. (c) The segmentation result using hybrid algorithm.

2.4.3 Discussion on Object-based Motion Analysis

Object segmentation from video is still one of the most challenging issues in image processing. There is currently no powerful segmentation routine which can handle any arbitrary complex scene. Even in the standardization issues of MPEG-4, the research still continues in order to find a good segmentation method which help to analyze and encode frames for object-based applications.

In most of the motion estimation and segmentation algorithms, the estimated motion data is used for intensity prediction between frames. Hence they are not necessarily to represent the projected true motion. If the estimated 2-D motion is used for the analysis of 3-D motion of the objects, then the segmentation performance becomes very critical. A motion vector, which is assigned incorrectly to another object, causes serious problems during the estimation of the 3-D motion of this object. Moreover, motion boundaries are also very important since a misclassified background motion vector near the moving object boundary with a value equal to zero might lead to inconsistency in the estimation of the 3-D motion of the moving object. In summary, simultaneous motion estimation and segmentation should be achieved if the obtained data is to be used in 3-D motion analysis.

The hybrid method has the best performance among the proposed three algorithms since it simply utilizes the advantages of the first two. If there is no time constraint while minimizing the energy function, the second algorithm based on Gibbs formulation leads to acceptably good results using a stochastic optimization algorithm. Hence, the hybrid algorithm will not be beneficial in such a situation. The experimental results show the superior performance of the hybrid algorithm for correct segmentation of the moving objects. Moreover, the needlegrams and SNR_p 's of the reconstructed frames support the validity of the estimated 2-D motion vectors. Hence, at the end of object-based motion analysis, it can be stated that the obtained results can be used at the next step of the dissertation to find the 3-D motion estimates of the objects.

Chapter 3

3-D Motion Estimation

3-D motion estimation refers to finding the actual motion of an object in a 3-D scene which is observed through consecutive 2-D video frames. Some applications of 3-D motion analysis are robotic vision, passive navigation, surveillance imaging, intelligent vehicle highway systems, harbor traffic control and object-based video compression [2]. This dissertation is only concerned with the latter application.

The 2-D projection of the actual motion of an object not only depends on the 3-D motion parameters, but also the object *depth* information (structure) which is simply defined as the distance of the object surface points from the camera. Hence for some applications (e.g. motion compensated prediction in video compression) in which 2-D projections are utilized, the depth information should also be estimated as well as 3-D motion. However, there are different ways to approach 3-D motion and structure estimation problems. In contrast to motion estimation and segmentation, in this problem the estimation process can be divided into two stages without any complications. Although, there are some methods, which estimate the depth first [65], the usual approach is to find 3-D motion parameters of the object before estimating the depth information [66, 67, 68].

In this chapter, 3-D motion estimation using consecutive video frames is examined

and depth analysis is left to the next chapter. In the first section, a brief overview of the current 3-D motion estimation methods is given. Afterwards, two different (rigid and non-rigid) motion estimation methods are proposed and supported by some simulations. At the end of this chapter, the advantages and also drawbacks of the proposed methods and 3-D motion analysis in general are discussed.

3.1 Current Methods on Estimating 3-D Motion

Before scanning through the current methods on 3-D motion estimation, the projection between 3-D world to 2-D image plane should be defined. 3-D environment can be projected into the image plane by many different mappings; the most popular projections are *orthographic* and *perspective* [2]. Orthographic (parallel) projection is the mapping of the 3-D coordinates, (X, Y, Z) , onto the image plane coordinates, (x, y) , simply by the relations $x = X$ and $y = Y$. On the other hand, the relation between 3-D and 2-D coordinates for perspective projection is given by

$$x = F \frac{X}{Z(X, Y)} \quad , \quad y = F \frac{Y}{Z(X, Y)} \quad (3.1)$$

where F is the focal length of the image sensor (Figure 2.1).

The selection between these two projections depends on the view angle of the recording device. Since the recording device is usually modeled as a pin-hole camera, the perspective projection becomes the most appropriate model to be utilized. However, if the view angle of the scene is narrow, which corresponds to a high focal length, then the perspective projection can be simplified in order to obtain orthographic projection. Nevertheless, a typical videophone scene is usually mapped onto the image plane more realistically by using the perspective projection.

According to kinematics, 3-D motions of objects can be classified into two main groups as *rigid* and *non-rigid* motions [69]. The motion of an object with a non-deforming surface in time is accepted as rigid. Any change in the structure of the moving object makes the analysis difficult and this kind of motion is called non-rigid. Moreover, for some

cases, a non-rigid motion consists of a number of rigid motions of some connected parts and this situation is known as *articulated* motion. In order to clarify these definitions, some examples can be given. While the rotation of an human head is rigid, the lips of the talking person make non-rigid motion. On the other hand, opening or closing of fingers is an example of an articulated motion; while the overall motion of the hand is non rigid, each bone segment of the fingers makes rigid motion.

After these basic definitions, a brief overview of current rigid and non-rigid 3-D motion estimation methods is given in the next two subsections.

3.1.1 Rigid Motion

In most of the 3-D motion estimation methods in the literature, 3-D motion is usually modeled for rigid objects [70, 67, 68, 66, 71]. In kinematics, according to *Chasles Theorem*, 3-D motion of a *rigid* body can be expressed in a linear form using a rotation and a translation as [72],

$$\mathbf{X}(t + \Delta t) = \mathbf{R} \mathbf{X}(t) + \mathbf{T} \quad (3.2)$$

where \mathbf{R} is a 3x3 rotation matrix, \mathbf{T} is a 3x1 translation vector and \mathbf{X} is the vector showing the 3-D coordinates at time instants t and $t + \Delta t$, before and after the 3-D motion, respectively. A rotation of an object can be defined using different 3x3 rotation matrix representations. There are two popular representations for the rotation matrices : rotations around (x, y, z) coordinate axes and rotation around an axis passing through the origin [73]. Any motion in real world can be analyzed using one of these representations and conversion from one to other is also possible [73]. For rotations around (x, y, z) coordinate axes, the corresponding rotation matrix can be written as

$$\mathbf{R}_{x,y,z} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos(w_x) & \sin(w_x) \\ 0 & -\sin(w_x) & \cos(w_x) \end{bmatrix} \cdot \begin{bmatrix} \cos(w_y) & 0 & -\sin(w_y) \\ 0 & 1 & 0 \\ \sin(w_y) & 0 & \cos(w_y) \end{bmatrix} \cdot \begin{bmatrix} \cos(w_z) & \sin(w_z) & 0 \\ -\sin(w_z) & \cos(w_z) & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (3.3)$$

where w_x , w_y and w_z are the rotation angles around x , y and z axes, respectively. While this representation does not commute, for rotations with small angles, the corresponding matrix commutes and it can be simplified into

$$\mathbf{R}_{small} = \begin{bmatrix} 1 & w_z & -w_y \\ -w_z & 1 & w_x \\ w_y & -w_x & 1 \end{bmatrix} \quad (3.4)$$

For the other representation, the movement is modeled as a rotation around an arbitrary axis passing through the origin whose direction is given by the unit vector (n_x, n_y, n_z) with angle of w_n and the corresponding rotation matrix can be written as

$$\mathbf{R}_n = \begin{bmatrix} (n_x^2 - 1)c + 1 & n_x n_y c - n_z s & n_x n_z c + n_y s \\ n_x n_y c + n_z s & (n_y^2 - 1)c + 1 & n_y n_z c - n_x s \\ n_x n_z c - n_y s & n_y n_z c - n_x s & (n_z^2 - 1)c + 1 \end{bmatrix} \quad (3.5)$$

where $c \doteq 1 - \cos(w_n)$ and $s \doteq \sin(w_n)$. The advantage of this model is the simple conversion from 9 dependent rotation matrix parameters to 3 independent rotation motion parameters. However, as it is stated previously, any of the above rotation representations can be used for any application without any complications.

The coordinates of a 3-D object can also be represented using homogeneous coordinates [74]. In this representation, rigid 3-D motion is formulated using only one matrix, rather than one rotation matrix and one translation vector. This matrix is usually called as a transformation [75]. It is very easy to represent consecutive motions by transformations. Another advantage of homogeneous coordinates is their ability to model scalings or zooming of the camera [2].

It should be stressed that while estimating the 3-D motion parameters by observing monocular frame sequences using perspective projection, the depth field and the translation vector can only be determined up to a scale constant. In other words, it is impossible to determine the *absolute* depth (translation), but rather a *relative* depth (translation) can be found. The reason of this phenomena is tried to be explained in the first chapter by the help of an example in which the motion and the distance of a moving

object can not be estimated without additional information. Mathematically, the same situation can also be observed in Equation 3.8 which gives the 2-D projection of 3-D motion and structure. In that equation, the depth and the translation parameters are always found as pairs, dividing each other. Hence, multiplication of both by using an arbitrary constant will not change the projected 2-D motion which is the observed data.

The methods which estimate 3-D rigid motion from consecutive monocular frames can be divided into two major classes, as *direct* and *correspondence based* methods. Direct methods use spatio-temporal gradients in the image to find a solution to the 3-D motion estimation problem [76, 77, 78, 79]. The rigid 3-D motion relation (Equation 3.2) is inserted into the famous *optic flow equation* [8] after perspective projection. Therefore, the unknown 3-D motion parameters become related to the spatio-temporal image gradients. Currently, there is no general solution to direct methods; only by making some simplifying assumptions about motion and/or structure, a solution can be obtained [80]. Since the performance of the results depends on the accuracy of the gradients on the discrete image, some improvements are also proposed on how to estimate the differentials [77]. However, the difficulty with robust gradient calculation, the requirement of small displacements for better linearization and susceptibility to noise, make not only the direct methods, but also their 2-D counterparts (any algorithm that use optic flow concept with spatio-temporal gradients, e.g. pel-recursive algorithms [10]), less attractive compared to other approaches.

On the other hand, any 3-D motion estimation method which requires some dense or sparse set of 2-D motion vectors for finding the 3-D motion parameters is said to be correspondence based. These methods require some point matches, which can be obtained by one of 2-D motion estimation methods and these methods are explained in the preceding chapter. Since incorrect matches may lead to unstable solutions, the performance of any correspondence based method mainly depends on this initial matching step. Therefore, in order to obtain immunity to errors, matches between some features, like corners or edges, are usually preferred [81]. There are also some other approaches which estimate 3-D motion vectors from line matches [82, 83]. Although most

of the work in the literature is directed to find the minimum number of correspondences to obtain a unique 3-D motion and structure [70, 67, 68, 66, 71], it is shown that even with infinite number of correspondences, the motion of some special hyperboloid surfaces will definitely have more than one solution for their motion [84]. It should also be noted that 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 and noise.

Estimation of 3-D motion for a planar patch [70] and any curved surface [66] can be both solved linearly. The concept of “pure” parameters, which relates the 2-D coordinates of the points on a rigid planar patch in two consecutive frames by eight parameters, is proposed in [70]. The pure parameters are very popular since these parameters can easily model the motion of a small planar patch and objects can be assumed to be made of small planar patches [21, 20, 23].

A solution to 3-D motion estimation problem without any planarity constraint is proposed in [66]. By modeling the motion as in Equation 3.2, an “Essential” matrix, \mathbf{E} , which relates the 3-D motion parameters with 2-D image plane coordinates before and after the motion linearly, is defined. This formulation yields a least-squares solution of the 3-D motion parameters. The *E-matrix* method is still one of the most popular 3-D motion estimation method. Afterwards, the noise susceptibility of this algorithm is tried to be improved by nonlinear robust versions [85, 86, 87, 88].

More detailed surveys on rigid 3-D motion and structure estimation methods and comparisons in between can be found in [89, 1, 90, 91, 92, 93, 65, 94, 2].

During the last decade, some 3-D motion tools have been used in video coding applications [20, 95, 96, 83, 94]. Most of these methods have the assumption of rigidity, except those which have a generic wireframe model for human head [96]. The non-rigid facial actions are also 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 3-D motion data [20, 17] or estimate the global (camera)

motion and depth field using an initial estimate of the dense depth field [83]. In [20], the segmentation of the scene is achieved for the stationary and moving parts by the help of frame differences and pure parameters [70, 97], and different regions are coded in an appropriate way. For some other methods [95], no segmentation is performed and long sequences are used to estimate incremental 3-D motion and sparse depth fields which are interpolated afterwards. Recently, a 3-D object-based motion and depth estimation method without any significant constraints, except rigidity, is also proposed [47].

3.1.2 Non-rigid Motion

Although joint motion-and-structure analysis of a deformable object is quite difficult, non-rigid motion can still be examined from a kinematic point of view. According to the fundamental theorem of kinematics [69], the most general motion of a *sufficiently small* element of a continuously deformable body can be approximated as the sum of a translation, a rotation and an extension (contraction) in 3 mutually orthogonal directions. In matrix form, the theorem above can be written as [69]

$$\begin{bmatrix} X_{\mathbf{p}}(t-1) \\ Y_{\mathbf{p}}(t-1) \\ Z_{\mathbf{p}}(t-1) \end{bmatrix} = \begin{bmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{32} & r_{33} \end{bmatrix} \begin{bmatrix} X_{\mathbf{p}}(t) \\ Y_{\mathbf{p}}(t) \\ Z_{\mathbf{p}}(t) \end{bmatrix} + \begin{bmatrix} T_x \\ T_y \\ T_z \end{bmatrix} + \begin{bmatrix} s_{11} & s_{12} & s_{13} \\ s_{21} & s_{22} & s_{23} \\ s_{31} & s_{32} & s_{33} \end{bmatrix} \begin{bmatrix} X_{\mathbf{p}}(t) \\ Y_{\mathbf{p}}(t) \\ Z_{\mathbf{p}}(t) \end{bmatrix}. \quad (3.6)$$

In Equation 3.6, the matrix which consists of elements $r_{i,j}$ is a rotation matrix, orthonormal with only 3 degrees of freedom. The matrix with elements $s_{i,j}$ is the linear deformation matrix which is symmetric [69]. 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 $[X_{\mathbf{p}}(t-1) \ Y_{\mathbf{p}}(t-1) \ Z_{\mathbf{p}}(t-1)]^T$ after a global rigid motion consisting of rotation and translation and a deformation, which is tried to be approximated by a global deformation matrix. In fact, this is a “reverse” motion in time from t to $t-1$ although the “real” motion is from time $t-1$ to t . It should also be noted that Equation 3.6 is simply the first few terms of the Taylor series expansion of the 3-D coordinate vector $\mathbf{X}_{\mathbf{p}}(t-1)$ around $\mathbf{X}_{\mathbf{p}}(t)$. The higher order terms of the expansion are neglected for simplicity. Although this assumption is valid when $r_{i,j}$

and s_{ij} terms are larger compared to higher order terms, Equation 3.6 is still a highly constrained model for locally deformed motions.

Three classes of non-rigid bodies are usually of major interest : articulated objects, elastic bodies and fluids. The application areas of non-rigid motion analysis are medical imaging (heart-wall motion tracking), fluid dynamics (interpretation of complex fluid motions), computer graphics and video coding (face analysis/synthesis, wireframes) [98]. In the earlier research on deformable motion analysis, non-rigidness, which is simply elasticity in that case, is modeled by springs which are assumed to exist between object points [99]. When the initial object structure and correspondences between views are given, the new structure is tried to be obtained by consistency to correspondence data and by maximizing the rigidity. The structure after motion should be as similar as possible to the initial one and the deviations between two structures should be obtained according to the tensions on the springs. Afterwards, two new approaches emerged to analyze non-rigid motion; one of these methods tries to find a basis for deformation modes to analyze and even recognize deformed bodies [100, 101], whereas the other uses physical phenomena behind the deformable motion [102, 103]. In the former method, the non-rigid motion is parameterized in terms of the eigenvalues of the finite-element model of the deformed object using a known or measured (by sensors) geometry of the undeformed object. However, in the latter one, some physical features, such as damping and mass, are included in the structural models in order to simulate the dynamics of deformable objects in response to intrinsic and extrinsic forces [2]. This latter method has the advantage of utilizing dynamics of motion compared to the former one [104]. This method relates the structural and motion parameters with mass, damping and stiffness matrices. Inertial and external forces are also taken into account in its formulation [104]. Simulation results show that physics-based modeling of non-rigid objects gives the best results among all non-rigid methods [102, 103, 104].

In most of these methods, the structural models are usually chosen from superquadrics [105, 106], hyperquadrics [107] and 'deformable balloon models [108]. However, in all of these approaches, the initial (undeformed) shape is either known

or obtained from sensor data. There are some application specific approaches which are also used in analyzing the non-rigid motion. In [109], only the extraction and contraction of bodies are examined. In an other application, the heart-wall motion is analyzed using hierarchical motion decomposition [110] and afterwards, it is solved using artificial neural nets [111]. It should be noted that almost all the methods require an initial structure to be able to track the non-rigid deformations and 3-D (sometimes 2-D [112]) motion. There is currently no generic method which estimates 3-D motion and structure of a deformable body by only observing video sequences.

The non-rigid 3-D motion analysis methods, which are explained in the previous paragraph, do not have applications in video compression. However, in *knowledge-based coding* there are some other approaches which use non-rigid motion for video compression purposes. In knowledge-based coding, it is assumed that a generic wireframe for some objects (usually head and shoulders of humans) is available at both transmitter and receiver. This wireframe might make some local deformations as well as some global 3-D motion. There are some generic wireframes, which are constructed by different laboratories [113, 96, 114]. Apart from some different approaches for estimating the motion on triangular patches of the wireframe [115, 116, 117, 94], all these methods have comparable performances. In addition to global head motion of the wireframe, the local movements of the triangular patches can model the non-rigid deformations on the face. However, the performance of these highly constrained wireframe models considerably degrades whenever the speaker has an unexpected simple feature, like a hat or eyeglass, which is not included in the model. Apart from this, the initial reconstruction of the wireframe on an irregularly oriented face also generates problems.

3.2 Proposed Object-based Rigid 3-D Motion Estimation Method

In this section, an object-based 3-D motion estimation method, which takes the previously estimated 2-D motion vectors and segmentation mask between two consecutive frames as input and creates the rigid 3-D motion parameters for the output, is proposed. The 3-D motion estimation is carried out for each object, as defined by the segmentation mask, separately. The formulation of the method is as follows :

Let \mathbf{P} define an object in the 3-D object space and let $\mathbf{p} \in \mathbf{P}$ be an object point whose 3-D coordinates at time t are given by $\mathbf{X}_{\mathbf{p}}(t) = [X_{\mathbf{p}}(t) Y_{\mathbf{p}}(t) Z_{\mathbf{p}}(t)]^T$. The perspective projection of $\mathbf{X}_{\mathbf{p}}(t)$ onto the image plane, which is shown in Figure 2.1, is written as $\mathbf{x}_{\mathbf{p}}(t) = [x_{\mathbf{p}}(t) y_{\mathbf{p}}(t)]^T$. For any rigid motion from time $t - 1$ to t , the 3-D coordinates of object point \mathbf{p} at time $t - 1$ can be written in terms of $\mathbf{X}_{\mathbf{p}}(t)$ as

$$\mathbf{X}_{\mathbf{p}}(t - 1) = \mathbf{R} \mathbf{X}_{\mathbf{p}}(t) + \mathbf{T} \quad (3.7)$$

where \mathbf{R} is a 3x3 rotation matrix, \mathbf{T} is a 3x1 translation vector. It should be noted that \mathbf{R} and \mathbf{T} do not reflect the “real” motion from time $t - 1$ to t , but rather an “inverse” motion from time t to $t - 1$. After perspective projection of the 3-D object points onto 2-D image plane, the equations below are obtained [66]

$$\begin{aligned} x_{\mathbf{p}}(t - 1) &= f \cdot \frac{r_{11} \cdot x_{\mathbf{p}}(t) + r_{12} \cdot y_{\mathbf{p}}(t) + r_{13} \cdot f + \frac{T_x \cdot f}{Z_{\mathbf{p}}(\mathbf{x}_{\mathbf{p}}, t)}}{r_{31} \cdot x_{\mathbf{p}}(t) + r_{32} \cdot y_{\mathbf{p}}(t) + r_{33} \cdot f + \frac{T_z \cdot f}{Z_{\mathbf{p}}(\mathbf{x}_{\mathbf{p}}, t)}} \\ y_{\mathbf{p}}(t - 1) &= f \cdot \frac{r_{21} \cdot x_{\mathbf{p}}(t) + r_{22} \cdot y_{\mathbf{p}}(t) + r_{23} \cdot f + \frac{T_y \cdot f}{Z_{\mathbf{p}}(\mathbf{x}_{\mathbf{p}}, t)}}{r_{31} \cdot x_{\mathbf{p}}(t) + r_{32} \cdot y_{\mathbf{p}}(t) + r_{33} \cdot f + \frac{T_z \cdot f}{Z_{\mathbf{p}}(\mathbf{x}_{\mathbf{p}}, t)}} \end{aligned} \quad (3.8)$$

where f is the focal length of the camera, r_{ij} is an element of the rotation matrix and (T_x, T_y, T_z) are the elements of translation vector. $\mathbf{x}_{\mathbf{p}}(t - 1) = [x_{\mathbf{p}}(t - 1) y_{\mathbf{p}}(t - 1)]^T$ are the projected 2-D coordinates of the object point \mathbf{p} at time $t - 1$. Notice that $Z_{\mathbf{p}}(\mathbf{x}_{\mathbf{p}}, t)$ is the third component of the vector $\mathbf{X}_{\mathbf{p}}(t)$ whose perspective projection gives $\mathbf{x}_{\mathbf{p}}(t)$ and simply called as “depth value”. However, it should be noted that “depth field” term is being used as the set of depth values defined on the 2-D lattice, Λ . Hence the depth field

reflect only the Z values of the projected 3-D object points. After some manipulations [66], the relation

$$\mathbf{U}'\mathbf{E}\mathbf{U} = 0 \quad (3.9)$$

is obtained for the \mathbf{E} matrix where $\mathbf{U} = [x_{\mathbf{p}}(t) \ y_{\mathbf{p}}(t) \ 1]^T$, $\mathbf{U}' = [x_{\mathbf{p}}(t-1) \ y_{\mathbf{p}}(t-1) \ 1]$. For notational simplicity, \mathbf{p} subscript will not be used to label the object point coordinates in the rest of this section. The unknown \mathbf{E} matrix is equal to

$$\mathbf{E} = \begin{bmatrix} 0 & T_z & -T_y \\ -T_z & 0 & T_x \\ T_y & -T_x & 0 \end{bmatrix} \begin{bmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{32} & r_{33} \end{bmatrix} \quad (3.10)$$

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

By using all (at least 8) the correspondences, Equation 3.9 can be solved in the least squares sense and afterwards \mathbf{E} -matrix can be decomposed into \mathbf{R} and \mathbf{T} analytically [66]. A depth field for each correspondence point can also be found by utilizing the estimated 3-D motion parameters, the correspondence vector and Equation 3.8.

3.2.1 Description of the Algorithm

Before application of the \mathbf{E} -matrix method to 3-D object-based coding, a few points should be emphasized. It should be first remembered that only 8 feature correspondences are sufficient to estimate the matrix \mathbf{E} in Equation 3.9 and, consequently, the 3-D motion parameters of an object. In order to define the objects, a segmentation step is necessary prior to 3-D motion estimation. Usually, the segmentation is based on 2-D dense motion vectors. This dense set of 2-D motion vectors for each object usually contains correct 2-D correspondences as well as some outliers, i.e., incorrect matches. Moreover, some irrelevant motion vectors of the neighboring objects might also be included in the object motion vector set due to incorrect segmentation. Hence, rather than finding, using another algorithm, robust 2-D correspondences between frames, which are needed for the \mathbf{E} -matrix computation, a robust selection mechanism over the existing 2-D dense motion vectors is proposed. Although there are some statistical methods, which might

find and reject the outliers [44], an iterative selection mechanism, in which a performance indicator is tried to be maximized through the iterations, is proposed. On the other hand, if the E-matrix method is solved directly in least-squares sense using all the data, even if there are excessive number of correspondences, the performance of the algorithm might degrade considerably as a consequence of a few incorrect segmented motion vectors or wrong matches.

Since there are some error sources which affects the performance of the E-matrix method, some parameters can be defined to test whether the estimated 3-D motion parameters are valid or not. First of all, the error between the original (input) and projected 2-D motion vectors must approach to zero for a correct 3-D motion estimate set. The projected motion vectors are obtained utilizing the estimated 3-D motion parameters and depth field. It is also shown that the eigenvectors of $E^T E$ must be equal to $[1 \ 1 \ 0]$, in order to have the matrix E implicitly contain a valid rotation (orthonormal of 1st kind) matrix and a translation vector [92]. Hence, the eigenvalues of $E^T E$ matrix contain valuable information to measure the performance of the 3-D motion estimation. Moreover, using the E-matrix method, a depth value for each point can be determined by the help of the 2-D motion vector at the corresponding point and the estimated 3-D motion parameters [93]. The validity of this estimated depth, consequently the 3-D motion parameters, can also be tested as follows : since all the objects are assumed to move in front of the image plane, they should all have positive depths [118]. Taking the above ideas into account, five parameters are defined to test the accuracy of 3-D motion estimates :

- $T_1 \doteq \frac{\sum_{\mathbf{x} \in R_i} |D_{2D}^x(\mathbf{x}) - D^x(\mathbf{x})|}{\sum_{\mathbf{x} \in R_i} |D^x(\mathbf{x})|}$, where D_{2D}^x is the horizontal component for the 2-D projection of the estimated 3-D motion and D^x is the horizontal component for the input 2-D motion estimates for the Object i .
- $T_2 \doteq \frac{\sum_{\mathbf{x} \in R_i} |D_{2D}^y(\mathbf{x}) - D^y(\mathbf{x})|}{\sum_{\mathbf{x} \in R_i} |D^y(\mathbf{x})|}$, where D_{2D}^y is the vertical component for the 2-D projection of the estimated 3-D motion and D^y is the vertical component for the input 2-D motion estimates for the Object i .

- $T_3 \doteq \lambda_{E3}$, where λ_{E3} is the smallest eigenvalue (usually have zero value) of $E^T E$ matrix.
- $T_4 \doteq \frac{|\lambda_{E1} - \lambda_{E2}|}{\sqrt{\lambda_{E1}^2 + \lambda_{E2}^2}}$ where λ_{E1} and λ_{E2} are the non-zero eigenvalues of $E^T E$ matrix.
- $T_5 \doteq \frac{n_t}{N} \cdot \frac{n_{t-1}}{N}$, where n_t and n_{t-1} are the number of negative depth values for N tested points at time t and $t - 1$ respectively.

It should be noted that there are also different error criteria [9] for comparing 2-D motion vectors other than $T_{1,2}$. However, the proposed test variables are computationally less complex compared to the others. Ideally, all the parameters above should be equal to zero for the correct rotation and translation estimates. Consequently, the sum of these five test parameters, which can be denoted as T_{sum} , should also approach to value zero for valid motion parameters. A new parameter, P , is defined to be $\frac{1}{1+T_{sum}}$ and it should be equal to 1 for the correct 3-D motion parameters. P is chosen as the *performance indicator*; according to the value of P , the estimation results can be “trusted”, or not.

Random Sample Consensus (RANSAC) is a paradigm for fitting a model to experimental data [119]. It is capable of interpreting and smoothing data containing a significant percentage of gross errors. Classical techniques for parameter estimation, such as least squares, optimize (according to a specified objective function) the fit of a functional description, i.e., model, to all of the presented data. Since these methods do not have a detection and rejection mechanism, they are susceptible to gross errors in many cases. In RANSAC, assuming that a given procedure requires at least n data points to determine the model parameters and there are N points in the initial experimental data set ($N > n$), a random subset of the data points with n points is selected to construct a model. For the initial N -point data set, the percentage of the points, which fits to the constructed model within an error tolerance, is checked using a threshold. If the percentage is less than this threshold, then a new random subset is used until the error tolerance is satisfied or the maximum number of iterations is reached.

Application of RANSAC to E-matrix method is straightforward. The E-matrix method requires at least 8 correspondences to estimate the E-matrix and there are dense

2-D motion fields for each object. Additionally, the error tolerance can be determined using the test parameters T_1 and T_2 , which test the validity of the model to the data, as it is suggested by the original RANSAC approach. However, RANSAC can further be improved by using the performance parameter P , instead. While the test parameters $T_{1,2,5}$ check the fit of the model to all the input data, $T_{3,4}$ is necessary to understand whether the randomly selected subset is a “good” choice or not.

The overall algorithm can be summarized as below :

```

For each moving object do :
  1. Select a random subset from 2-D motion vectors
  2. Find E-matrix, 3-D motion and corresponding P
  3. If ( P > Threshold_for_P )
      exit with current 3-D motion
  else
      If ( P > maximum_P_so_far )
          save P and 3-D motion
      If maximum_#_of_iterations has reached
          exit with saved 3-D motion
      goto step 1

```

Hence, using the algorithm above, a rotation matrix and a translation vector are found for each object in a robust manner.

3.2.2 Simulations

THE EFFECTS OF ERROR SOURCES TO THE CONVENTIONAL E-MATRIX METHOD

The simulations on 3-D motion estimation are conducted in three phases. In the first phase, artificial data is used to test the performance of the conventional E-matrix method [86] by varying the image resolution, matching errors and focal length error. In this phase,

RANSAC is not utilized. Such an experiment gives an idea about the upper limit of the performance of this algorithm for natural data. 100 points are selected on an artificial image and these points are moved using known 3-D motion and structure parameters. The projection of these 3-D points onto the image plane are achieved after

1. quantizing the new 2-D coordinates according to the image size to simulate errors due to resolution,
2. adding some Gaussian noise to simulate matching errors,
3. distorting the true focal length to simulate focal length error,
4. applying the above three procedures together.

For all four steps of the first phase, test parameters are obtained to measure the performance of the algorithm. Step 1 is achieved for two different image resolutions as 176×144 (Step 1.a), which is a standard *QCIF* image size, and 1760×1440 (Step 1.b). In the second step, Gaussian noise, whose variance is 2 pixel^2 , is added to each component of the projected 2-D motion vectors for *QCIF* resolution. 3-D motion parameters and depth values for 100 points are selected so that the average motion of the artificial data is approximately 20 *pixels* in both directions. In the third step, the correct focal length of imaging system is distorted about 10 percent before projecting the 3-D objects points back to the image plane after motion. This step is necessary, since in many of the standard video sequences focal length information is not available. Last step, as it is denoted above, is a combination of the first three phases (with *QCIF* image size). The results of the first phase are given in Table 3.1.

The results of Step 1 show that for natural sequences it is expected to obtain some amount of error due to quantization noise. On the other hand, as it is denoted by the test parameters of Step 1.b, the usage of frames with high spatial resolutions might improve the performance of the algorithm. However, in video coding applications it is preferable to use frames with small resolutions for better compression and real time communication. As it can be clearly observed from the test parameters of Step 2, the

Step	T_1	T_2	T_3	T_4	T_5	P
Ideal	0.0000	0.0000	0.0000	0.0000	0.00	1.0000
1.a	0.0639	0.0415	0.0252	0.0000	0.18	0.7630
1.b	0.0291	0.0108	0.0192	0.0000	0.17	0.8136
2	0.1388	0.1581	0.1844	0.0000	0.25	0.5776
3	0.1401	0.3605	0.0146	0.0000	0.19	0.5864
4	0.1591	0.3933	0.2214	0.0000	0.35	0.4708

Table 3.1: Simulations on E-matrix method using artificial data

E-matrix method is very susceptible to correspondence errors. The focal length should also be selected or estimated carefully, since the distortion on focal length might also degrade the results considerably. As expected, the combined effect of these three error sources is much severe and such a situation is also highly expected in natural sequences.

THE PERFORMANCE OF THE CONVENTIONAL E-MATRIX METHOD WITH REAL DATA

In the second phase of the experiments, the conventional E-matrix method [86] is applied to real data. The object segmentation and 2-D motion estimation results of Section 2.4.2 are used as inputs to 3-D motion estimation step. For each object, all the 2-D motion vectors are used to estimate the E-matrix in the least squares sense. The estimated 3-D motion parameters are not tabulated since they can not be compared with a (true) motion parameter set, but rather test parameters, which give an idea about the performance of the results, are presented. Moreover, in order to illustrate the performance of 3-D motion estimation step, some reconstructed frames can also be presented. However, the depth field is necessary to achieve this goal, since the projected 2-D motion depends on both 3-D motion and depth. The next chapter is completely devoted to depth analysis and in the simulation results (Section 4.1.2) of Chapter 4, the reconstructed frames, which are obtained using the 3-D motion parameters of this section, are presented. Hence, reconstructed frames are not presented here in the simulations of Chapter 3.

In the second phase, the results shown in Figure 2.5 for the 10th and 16th frames

of *Salesman* sequence are used in order to find the 3-D motion parameters for each object. The size of the frames are *QCIF* (176×144) and it is assumed that the unknown focal length of the camera is equal to 250 pixels (This approximately corresponds to 50 mm focal length of a 35 mm camera). Although this assumption is rough, it still gives acceptable results. However, to improve the overall performance, a camera with a known focal length might be used or the focal length can be estimated [120]. Similar to Figure 2.1, the optical (z -axis) axis is assumed to pass through the center of these images. The test parameters of the conventional E-matrix method are given in Table 3.2.

Object	T_1	T_2	T_3	T_4	T_5	P
0	-	-	-	-	-	-
1	0.285	0.955	0.050	0.000	0.730	0.331
2	0.630	0.605	0.100	0.000	0.784	0.321
3	0.243	0.819	0.301	0.000	0.629	0.334
4	0.519	0.241	0.686	0.000	0.673	0.321
5	0.148	0.429	0.682	0.000	0.632	0.346

Table 3.2: Simulations on 3-D motion parameter estimation using the conventional E-matrix method using 10th and 16th frames of *Salesman* sequence

THE PERFORMANCE OF THE PROPOSED METHOD WITH REAL DATA

In the third phase of the simulations, the method proposed in Section 3.2.1 is utilized in order to find the 3-D motion parameters from the previously estimated 2-D motion vectors for each object. Using the same 2-D motion vectors and segmentation results of Section 2.4.2 as inputs, the obtained test parameters and performance indicators are tabulated in Table 3.3. The maximum number of iterations is chosen as 50, whereas the threshold for P , which determines the acceptability of the obtained motion parameters, is selected as 0.5. Hence, as it is realized from Table 3.3, Objects 1 and 5 required 50 iterations to converge, whereas the other three needed much less.

As it can be observed from the last columns of the Tables 3.2 and 3.3, the proposed scheme has superiority over the conventional method for each object. It should be noted

Object	T_1	T_2	T_3	T_4	T_5	P
0	-	-	-	-	-	-
1	0.640	0.042	0.935	0.000	0.141	0.363
2	0.124	0.375	0.000	0.000	0.191	0.592
3	0.177	0.057	0.484	0.000	0.233	0.512
4	0.017	0.515	0.004	0.000	0.025	0.502
5	0.920	0.300	0.000	0.000	0.082	0.434

Table 3.3: Simulations on 3-D motion parameter estimation using the proposed method using 10th and 16th frames of Salesman sequence

that Object 0 is the stationary background, thus motion estimation is not achieved for this object. Objects 2 and 5 belong to the occlusion regions and it is expected to have small P values, since there is no rigid motion in these regions. However, while Object 5 obtains an expected small P value, the simulation results show that it is also possible to find a trustable a 3-D motion parameter set for Object 2. Hence, even in occlusion regions, if the obtained 2-D motion estimates for such regions are smoothly varying, then it is also possible to find some 3-D motion representation for such regions. The reason for obtaining an untrustable P value for Object 1 is due to the small head motion in the horizontal direction. This situation usually leads to some instabilities in the E-matrix method while estimating a 3-D motion parameter set. The small amount of motion in the horizontal direction is very susceptible to matching errors and also quantization noise for low spatial resolution [86]. Object 3 and 4 are both moving objects and the performance indicator of Object 4 is higher than that of Object 3, possibly due to better segmentation and 2-D motion estimation.

Similar experiments are conducted to test the performances of two algorithms (conventional E-matrix versus the proposed method) on different frame pairs, which are shown in Figures 2.6 and 2.7. In Tables 3.4 and 3.5, the simulation results are tabulated for these input data, respectively. For both frame pairs, 3-D motions of the segmented heads are tried to be estimated.

In these experiments, compared to the conventional E-matrix method, the proposed

Method	T_1	T_2	T_3	T_4	T_5	P
Proposed	0.121	0.612	0.000	0.000	0.567	0.435
E-matrix	0.513	0.812	0.924	0.000	0.482	0.268

Table 3.4: Simulations on 3-D motion parameter estimation using the proposed and conventional E-matrix method using 100th and 103th frames of *Foreman* sequence

Method	T_1	T_2	T_3	T_4	T_5	P
Proposed	0.171	0.400	0.000	0.000	0.331	0.526
E-matrix	0.456	0.890	0.863	0.000	0.426	0.275

Table 3.5: Simulations on 3-D motion parameter estimation using the proposed and conventional E-matrix method using 38th and 41th frames of *Mother and Daughter* sequence

method has also superior values for the test parameters. However, it should be noted that while the performance of the 3-D motion estimation step improves with this new scheme, the overall computation time also increases according to the number of iterations. Nevertheless, the linear E-matrix method is not a time consuming algorithm by itself (less than 1 second at a Sun Sparc 10 workstation) and the total execution time of the proposed algorithm is still acceptable.

3.3 Proposed Object-based Non-rigid Motion Estimation Method

Most of the existing rigid and non-rigid 3-D motion estimation methods have some drawbacks. Some of the methods [86, 77, 21, 121, 122, 96, 113] have some structural constraints, like rigidity, planarity or wireframes. A group of approaches [80, 77, 79] uses spatio-temporal gradients during 3-D motion estimation and errors occur as a result of discrete differentiation. Lack of segmentation of the scene is another problem which exists in some of the algorithms [86, 77, 121, 122, 83]. Moreover, the use

of orthographic projection approximation in many methods [123, 124, 125, 121, 122] limits the performance of such algorithms in some scenes which seem to be projected perspectively. Finally, some methods [70, 97, 66], especially the ones which are solved by linear approaches, have susceptibility to noise. The aim of this section is to propose a novel method which handles all these drawbacks.

There are some methods which track the non-rigid motion and estimate (update) the non-rigid shape. These methods are explained briefly in Section 3.1.2. A new approach, which attempts to eliminate the drawbacks explained in the first paragraph of this section, is proposed in Section 3.3.1. The basic idea is to formulate the problem in such a way that all the *a priori* information about non-rigid motion can be inserted into a cost function. This cost function can be selected as the energy function of a Gibbs probability density function for the non-rigid 3-D motion parameters.

According to Equation 3.6, the deformation matrix is constant over the whole object, hence the corresponding motion is still highly constrained. Such a modeling allows only globally linear elastic deformations of the overall object rather than local non-rigid behavior. If all the higher order terms, which are neglected from Equation 3.6, are taken into account, then any local deformable motion can also be modeled. Rather than taking into account all the higher order terms, another approach is to define the deformation motion parameters at each point for a more general motion modeling. Due to the 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.

Another important observation concerning Equation 3.6 is expressing the next coordinate of any point by using some global (the rigid rotation and translation parameters) as well as some deformable (the non-rigid deformation parameters) motion. This means that every non-rigid motion has an implicit rigid behavior. If the deformation parameters are defined at each point, such an observation might help while estimating these parameters.

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, is proposed in the next section. Observing two consecutive frames, the aim is to model and estimate the non-rigid motion between them.

3.3.1 Gibbs Model based Non-rigid Motion Estimation

The *a priori* information for a general non-rigid 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. Assuming the rotation angles between the frames are small (as in Equation 3.4), for each point on the object, the relation between two coordinates (before and after the non-rigid motion) can be written as

$$\begin{bmatrix} X_p(t-1) \\ Y_p(t-1) \\ Z_p(t-1) \end{bmatrix} = \begin{bmatrix} 1 & w_z & -w_y \\ -w_z & 1 & w_x \\ w_y & -w_x & 1 \end{bmatrix} \begin{bmatrix} X_p(t) \\ Y_p(t) \\ Z_p(t) \end{bmatrix} + \begin{bmatrix} t_x \\ t_y \\ t_z \end{bmatrix}. \quad (3.11)$$

The 3-D motion parameter vector $\theta = [w_x, w_y, w_z, t_x, t_y, t_z]^T$ is defined at each point on the object. Only the surface points of the objects are observed on the image and the 3-D parameter vector is defined at each point on a 2-D 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 . It should be noted that the proposed non-rigid motion definition can also be achieved using other rotation matrix representations in Section 3.1.1 without any problems. In such a case, the new parameter vector will be equal to $\theta = [w_n, n_x, n_y, n_z, t_x, t_y, t_z]^T$ which corresponds to the rotation matrix in Equation 3.5.

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

projecting the 3-D coordinates perspectively using Equation 3.1, the displacements on the image plane can be written as

$$\begin{aligned} x_{\mathbf{p}}(t-1) &= \frac{x_{\mathbf{p}}(t) + w_z y_{\mathbf{p}}(t) + (-w_y) + \frac{t_x}{Z_{\mathbf{p}}(\mathbf{x}_{\mathbf{p}},t)}}{w_y x_{\mathbf{p}}(t) + (-w_x) y_{\mathbf{p}}(t) + 1 + \frac{t_z}{Z_{\mathbf{p}}(\mathbf{x}_{\mathbf{p}},t)}} , \\ y_{\mathbf{p}}(t-1) &= \frac{(-w_z) x_{\mathbf{p}}(t) + y_{\mathbf{p}}(t) + w_x + \frac{t_y}{Z_{\mathbf{p}}(\mathbf{x}_{\mathbf{p}},t)}}{w_y x_{\mathbf{p}}(t) + (-w_x) y_{\mathbf{p}}(t) + 1 + \frac{t_z}{Z_{\mathbf{p}}(\mathbf{x}_{\mathbf{p}},t)}} . \end{aligned} \quad (3.12)$$

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

$$I_t(\mathbf{x}_{\mathbf{p}}(t)) = I_{t-1}(\mathbf{x}_{\mathbf{p}}(t-1)) . \quad (3.13)$$

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

$$\mathcal{U}(\Theta | \mathcal{I}_t, \mathcal{I}_{t-1}, \mathcal{Z}) = \mathcal{U}(\mathcal{I}_{t-1} | \mathcal{I}_t, \Theta, \mathcal{Z}) + \beta \mathcal{U}(\Theta | \mathcal{I}_t, \mathcal{Z}) . \quad (3.14)$$

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 3-D motion parameter field.

From Equation 3.13 under the assumption that there is Gaussian noise in the environment, the first term on rhs of Equation 3.14 becomes

$$\mathcal{U}(\mathcal{I}_{t-1} | \mathcal{I}_t, \Theta, \mathcal{Z}) = \sum_{\mathbf{x}_{\mathbf{p}} \in \Lambda} (I_t(\mathbf{x}_{\mathbf{p}}(t)) - I_{t-1}(\mathbf{x}_{\mathbf{p}}(t-1)))^2 . \quad (3.15)$$

The second term on rhs of Equation 3.14 can be obtained using the *a priori* information on the non-rigid 3-D motion parameters, as

$$\mathcal{U}(\Theta | \mathcal{I}_t, \mathcal{Z}) = \sum_{\mathbf{x}_{\mathbf{p}} \in \Lambda} \sum_{\mathbf{x}_{\mathbf{p},c} \in \eta_{\mathbf{x}_{\mathbf{p}}}} \|\theta(\mathbf{x}_{\mathbf{p}}(t)) - \theta(\mathbf{x}_{\mathbf{p},c}(t))\|^2 , \quad (3.16)$$

where $\mathbf{x}_{\mathbf{p},c} \in \eta_{\mathbf{x}_{\mathbf{p}}}$ is the neighbor of $\mathbf{x}_{\mathbf{p}}$. $\eta_{\mathbf{x}}$ is the neighborhood of \mathbf{x} , defined on Λ . This energy function favors similar values on neighboring parameters by assigning higher

probabilities to such cases. Such a relation is highly expected in most of the non-rigid motions.

The formulation above assumes the availability of the depth field, \mathcal{Z} , a priori, similar to all non-rigid motion estimation algorithms in the literature that are known by the author. The depth field can either be obtained from an extra sensor data or a stereo pair [28, 29]. Another approach might be obtained after defining the depth field as a new random field like motion parameters and to add a new term into Equation 3.14. This term should support a priori knowledge which can only be the smoothness of the surface. However, in such a situation, the minimization problem will become severely under constrained 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 3.14 can be improved by adding some new segmentation terms (similar to Equation 2.1), which segment the scene into objects according to their 3-D motion coherence. Hence, this non-rigid analysis will also become applicable for object-based algorithms after making some appropriate adjustments in Equation 3.14.

CONCEPT OF HIERARCHICAL RIGIDITY

Before minimizing Equation 3.14 in order to obtain non-rigid motion parameters, there is an important property to mention. As it is stated previously, the energy function in Equation 3.14 is valid not only for non-rigid but also for rigid motions, which is an extreme case where all the parameters are equal to each other. On the other hand, a general non-rigid motion formulation also 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. This goal can be achieved during the minimization of Equation 3.14.

In order to implement these ideas while minimizing Equation 3.14, a multiscale optimization approach is devised [126]. In this approach, the 3-D motion parameters

are defined at each point on different grids, similar to the grids shown in Figure 2.2, for different resolutions. On the coarsest grid, all the motion parameters are equal to each other in a predefined rectangle and therefore the part of the object which is projected onto this rectangle is assumed to be rigid. In other words, the resolutions determine the size of the rectangle in which the motion parameters are constant. 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. At the finest scale, the rectangle contains only one 3-D motion parameter vector, θ , as it is initially proposed. 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 also estimate the motion of a rigid object without any convergence problems. The proposed method in Section 3.3.1 with such a minimization approach is called “hierarchical rigidity” [126]. Similar minimization algorithms are independently proposed by [64], called Multiscale Constrained Relaxation (MCR), for general recovery problems.

The motion parameters can be estimated by minimizing Equation 3.14 by one of the global optimization algorithms explained in Section 2.3.2. However, the neighborhood definitions between motion vectors change for different scales. In Equation 3.16, the motion vectors are accepted as neighbors according to a new neighborhood, $\eta_{\mathbf{x}}^n$, which is defined on a grid, Λ^n , for the resolution n . In this way, the elastic relations between 3-D motion parameters continue at each scale.

In summary, the proposed method is valid for a general class of 3-D motions without structural constraints, like planarity or rigidity. Using hierarchical rigidity, both rigid and non-rigid motions can be estimated. This method also does not contain errors due to discrete differentiation and can be easily upgraded to achieve segmentation of the scene. Moreover, it is more immune to noise as a result of the *MAP* estimation with a more realistic projection being used. The most dominant drawback is the necessity of an initial depth field. The requirement of six parameters per pixel for the description of motion is also another disadvantage, especially from computational complexity.

3.3.2 Simulations

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 motions (Figures 3.1 and 3.4). Nevertheless, these simulations support the validity of the non-rigid motion estimation algorithm, at least for the rigid motion case.

The experiments consist of three stages, as validation of hierarchical rigidity, handling multiple moving objects and noise analysis.

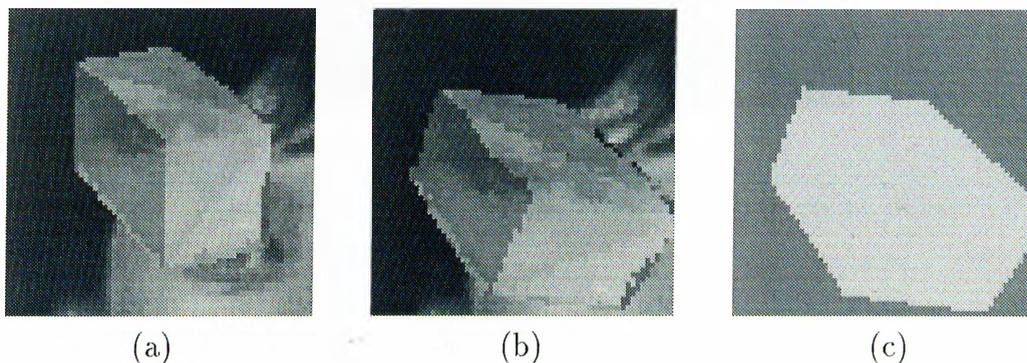


Figure 3.1: (a), (b) Consecutive original frames of *Cube* sequence. (c) Ideal motion parameter value for w_z shown as an intensity representation.

In the first step, hierarchical rigidity is tested. Minimization of Equation 3.14 is achieved using ICM for 3 scales with only 5 iterations at each scale. While all the 3-D motion parameters are estimated, a typical result is shown in Figure 3.2. The estimated parameter w_z is shown by intensities over the image for coarsest and finest levels in Figure 3.2 (a) and (b), respectively. The block sizes of the coarsest level can be observed from this figure. The estimates at the finest level are similar to ideal results at Figure 3.1(c). In Figure 3.2 (c), the histogram representation of all 3-D parameters are shown compared to the true values. The distribution of the true motion values, which are shown by the dotted lines, are similar to that of the estimated 3-D motion parameters in this histogram. 2-D projection of the estimated 3-D motion parameters is also shown in Figure 3.3 by a “needlegram” on the reconstructed image. The reconstructed second frame of *Cube* sequence is obtained by using the previous available frame, known depth

values and 3-D motion parameters. The obtained needlegram has similar results with the true 2-D motion vectors, except for the occlusion regions. Hence, it can be concluded that the hierarchical rigidity approach gives satisfactory results for rigid motion estimation.

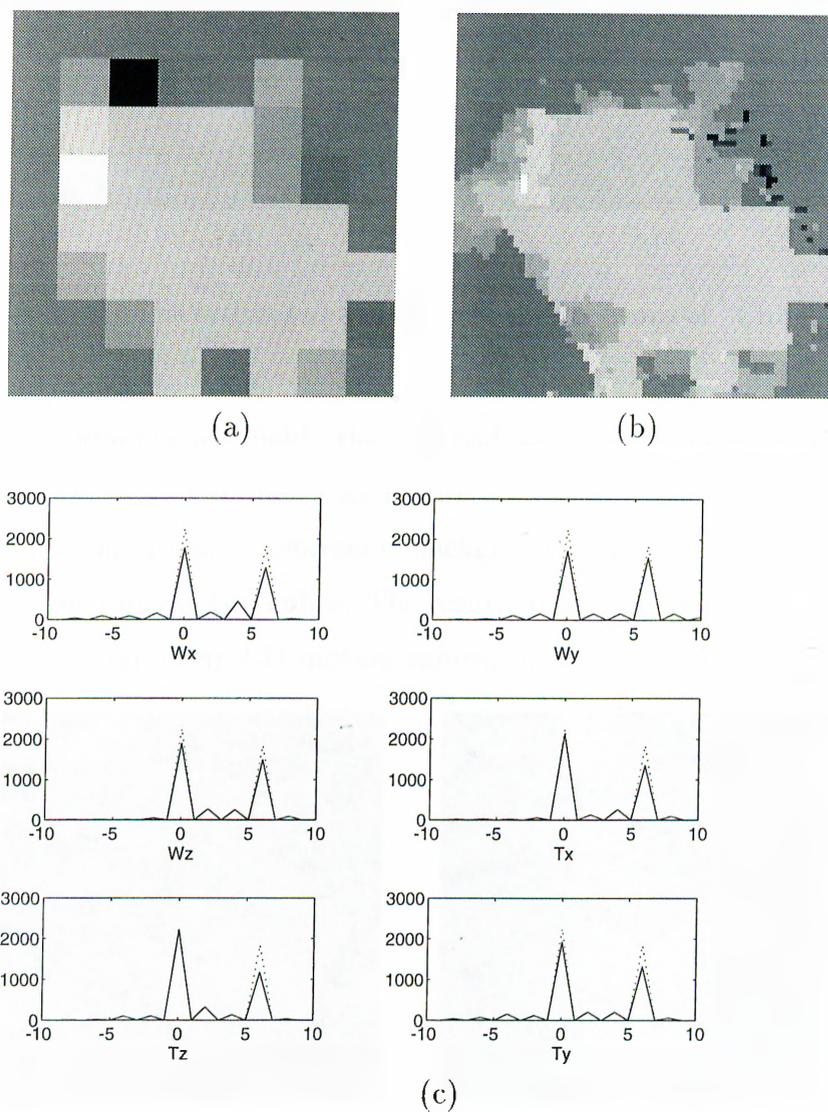


Figure 3.2: The intensity representation of w_z parameter for (a) 8x8 block size (coarsest level) and (b) 1x1 block size (finest level). (c) The histogram representation of $w_{x,y,z}$ and $T_{x,y,z}$ parameters. Dotted lines are true, whereas solid lines are the estimated values.

In the second step, two consecutive frames (Figure 3.4) of a scene with two cubes (*Cubes* sequence) moving with different speeds are examined using hierarchical rigidity. After minimization of the energy function, the results in Figure 3.5 show that without

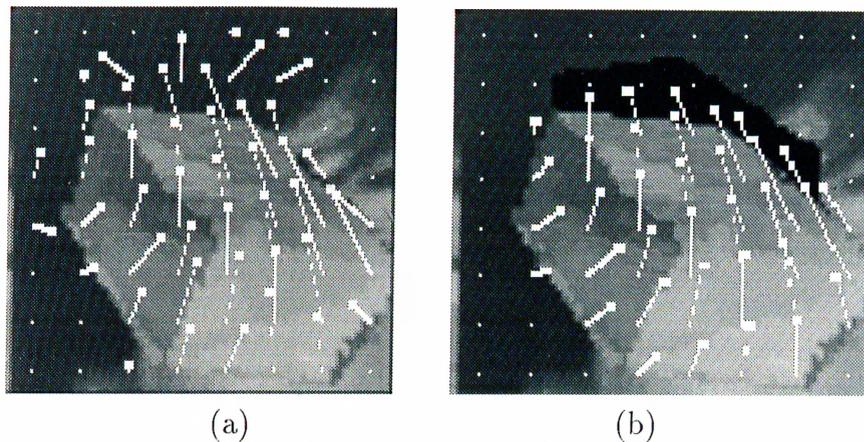


Figure 3.3: (a) The estimated and (b) true needlegrams of “Cube” on the reconstructed frames.

an initial segmentation field, the method achieves good results. In the histogram representation of t_y parameter in Figure 3.5, while the middle peak, which is 0 *pixel* translation along y -axis, represents background, the two peaks at positive and negative sides show the moving two cubes. The results show that this method can handle multiple objects and assign their 3-D motion values, correctly for this input frame pair.

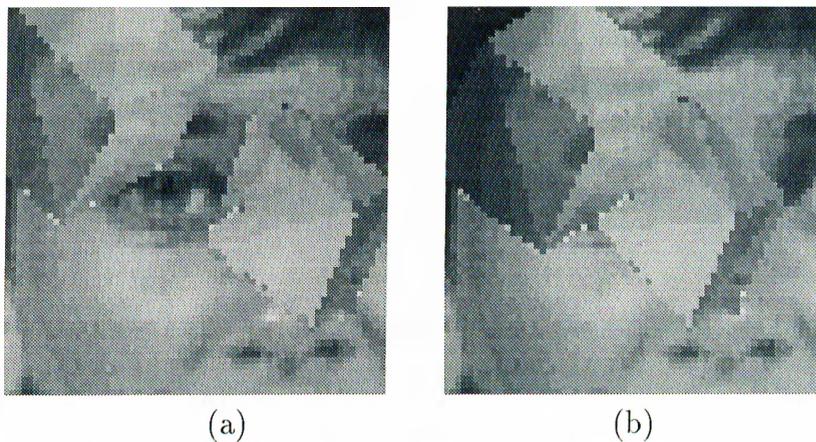


Figure 3.4: Original (a) first and (b) second frames of *Cubes* sequence

As a final step, noise analysis is performed on the proposed non-rigid 3-D motion estimation method. Two frames of *Cube* sequence are corrupted with Gaussian noise, resulting with frames, having SNR_{peak} values as 28 *dB* and 43 *dB*. The minimization of Equation 3.14 is achieved using these noise corrupted input data. In Figure 3.6,

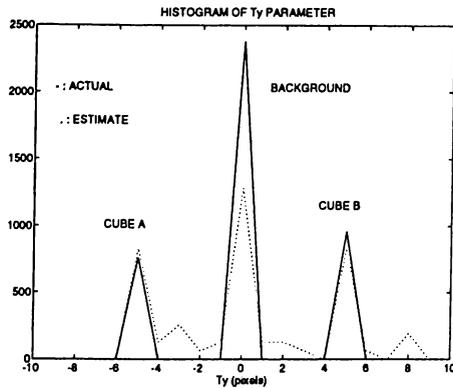


Figure 3.5: Histogram of t_y parameter for “Cubes”. True values are shown using solid, whereas the estimates with dotted lines.

the w_x and t_y components of 3-D motion parameters are presented using histogram representation. It is observed that while the injected noise on input increases, the performance of the estimation degrades, as expected. For the input with SNR_p equals to 40 dB, the results are acceptable.

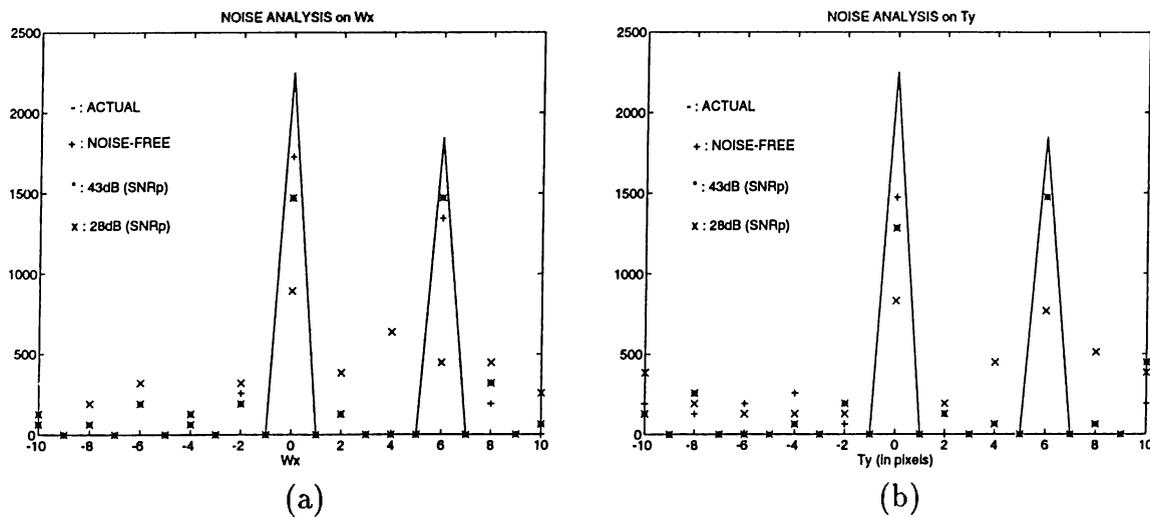


Figure 3.6: The estimation results of motion parameters (a) w_x and (b) t_y for input frames with different SNR_{peak} values.

3.4 Discussion on the Motion Models

In this chapter, two different 3-D motion models are analyzed : rigid and non-rigid. While non-rigid motion analysis is more general, it is difficult to obtain both motion and deformable structure by only observing consecutive video frames. Moreover, the computational complexity due to six motion parameters at each point of a non-rigid object is also ineffective for real-time communication applications. Hence, it is better to complete the simulations on this algorithm at this point, since the method of hierarchical rigidity is almost impossible to be utilized in a low bit-rate video coder. Nevertheless, rather than video coding, in the field of non-rigid motion analysis, the proposed non-rigid motion model is a novel approach to the challenging deformable motion analysis problem. The concept of hierarchical rigidity is promising, since it handles both global rigid and local non-rigid motions in one formulation. However, in the proposed method, depth estimation is still an open problem to be overcome before the application of the algorithm to areas that require non-rigid motion analysis.

In contrast with non-rigid motion models, it can be stated that rigid motion analysis is more preferable in video compression algorithms, since rigid motion description is ultimately efficient with low complexity. Consequently, when the aim is video coding utilizing 3-D motion information, rigid body analysis should be preferred to non-rigid counterpart. However, the most dominant drawback of the utilized E-matrix method is its susceptibility to input noise and errors. For small images (e.g., QCIF size), the unavoidable quantization of 2-D motion vectors at pel accuracy, also degrades results considerably. For example, in artificial sequences with known 2-D motion, if the true 2-D motion vector accuracy utilized in 3-D motion estimation step is high, then E-matrix method will work without any problems. Even the truncation of the floating true motion components into integer numbers affects the accuracy of E-matrix method (Table 3.1). Hence, in case of using small sized (e.g. QCIF size) images, due to unavoidable accuracy problems, 3-D motion estimates always contain some error.

On the other hand, in the proposed rigid 3-D motion estimation method, by rejecting

the outliers of the dense 2-D motion field, the overall performance of the 3-D motion estimation algorithm, which uses 2-D motion matches, improves considerably compared to the case that uses all the available 2-D motion vectors without any selection. Hence, rejection of outliers is a necessary step in the algorithm. In the computer simulations, as the obtained test parameters indicate, the estimated 3-D motion parameters are acceptable. The selection of motion vectors are achieved by the help of a performance indicator. The superior performance of the proposed rigid 3-D motion estimation algorithm is a result of utilizing RANSAC and the proposed indicator. This indicator compare the input variables and the model both implicitly and explicitly. The two test parameters based on eigenvalues implicitly require the selected input motion vectors to be a projection of 3-D motion. On the other hand, the other three parameters compare explicitly the 2-D projections of the estimated 3-D motion and structure (model) with the input 2-D motion vectors. Hence, the joint utilization of these test parameters results in better estimates compared to the least-squares solution of the E-matrix.

Although the proposed rigid motion model has obvious advantages that are explained in the previous paragraph, it might also be more advantageous to select between 2-D and rigid 3-D motion models according to the motion of the current object rather than using only the proposed rigid 3-D motion model. The rigid 3-D motion model is unsuccessful for non-rigid motions or incorrectly segmented articulated motions. 2-D motion models (especially MRF-based models), which are more flexible for any kind of motion, will still survive and obtain better performance with respect to its 3-D rigid counterpart in such situations. Hence adaptive motion model selection may improve the performance of any system with some increase in the complexity of the algorithm. In such a system, if the motion of an object is found out to be non-rigid, instead of encoding 3-D motion parameters and depth field, dense 2-D motion field obtained by minimizing Equation 2.1 can be utilized for that object.

In this chapter, a novel rigid 3-D object-based motion estimation method is proposed. The simulation results show that this method can be easily inserted into an object-based video compression algorithm. The compression performance of such a coding

algorithm depends on estimation and efficient encoding of the depth field, since rigid motion description is very efficient. Depth analysis is examined in detail in the next chapter.

Chapter 4

Depth Analysis in 3-D Motion Models

Depth analysis is necessary for any video coding scheme which uses 3-D motion models. Without having depth information of a point on the image, it is not possible to find the next coordinate of the same point after the motion. Hence, for motion compensated prediction of the intensities, after finding the 3-D motion parameters, a depth should be estimated and afterwards encoded for each point of an object.

If the main application area is determined as video coding, there are two main difficulties for depth analysis. First of all, the estimated depth field should be as robust as possible to input noise and errors, since it is very likely to feed noisy measurements and observations as inputs to the depth estimation algorithm. Moreover, the obtained depth field should also be encoded very efficiently in order to use 3-D motion models in video coding algorithms for low bit-rates.

Some of the rigid 3-D motion estimation algorithms in the literature are capable of finding a depth field. While direct methods in 3-D motion analysis only find a depth field for some specific motions such as pure rotation or pure translation [77], non-rigid motion estimation methods always require an initial estimate of the depth

field that is usually obtained from an extra sensor data. On the other hand, feature-based methods [65], such as E-matrix, can only find the depth values for locations which have correspondences between frames. Hence, the obtained sparse depth field has to be interpolated by somehow in order to be used in motion compensation for compression. Apart from all of these, all the depth estimation algorithms in the literature try to find a “true” depth field which is absolutely necessary for the applications in computer and robot vision. However, finding the correct depth field with many unnecessary details may not be preferable from lossy video coding point of view. For this case, depth analysis should be re-analyzed carefully taking into account both rate and distortion.

In the next sections, depth fields are examined in order to solve two main problems which are noise immunity and efficient encoding. After defining the sources of error, a *MAP* formulation is proposed to find a robust depth estimator in the next section. Simulations are conducted to test the performance of the algorithm. Afterwards, encoding of the depth field is examined in the rate-distortion sense. An efficient novel algorithm, which encodes the implicitly available true depth field by taking both distortion and bit-rate into account, is proposed. The performance of the algorithm is demonstrated by giving some experimental results. At the final section, a discussion on the similarities of the proposed two algorithms is given.

4.1 Noise Immune Depth Estimation

Depth estimation is usually achieved after the 3-D motion parameters are obtained. Thus, 3-D motion estimation errors might severely affect the performance of depth analysis. Since many of the 3-D motion estimation algorithms, including the proposed method in the previous chapter, are susceptible to noise, there should be an extra effort to maintain error robustness during depth estimation.

Dense depth field can be estimated using a *MAP* formulation, similar to Equations 2.1 and 3.14. Using the “true” (error-free) intensity, $\mathcal{I}_{t,t-1}$, and 3-D motion, \mathcal{M} , fields, it is

possible to obtain the depth field, \mathcal{Z} , exactly and this relation is shown in Figure 4.1. However, there is no longer an exact relation when these parameters are observed with some noise. Hence, \mathcal{Z} field should be estimated by taking noise into account. Using the observed noise contaminated consecutive intensity fields, $\tilde{\mathcal{I}}_{t,t-1}$, and the observed 3-D motion field, $\tilde{\mathcal{M}}$, which may also contain some error due to 2-D and 3-D motion estimation steps, *MAP* estimate of the depth field can be found by maximizing a conditional probability distribution. Moreover, this distribution can also be written as Gibbsian. A similar Gibbs energy function can be found in [127].

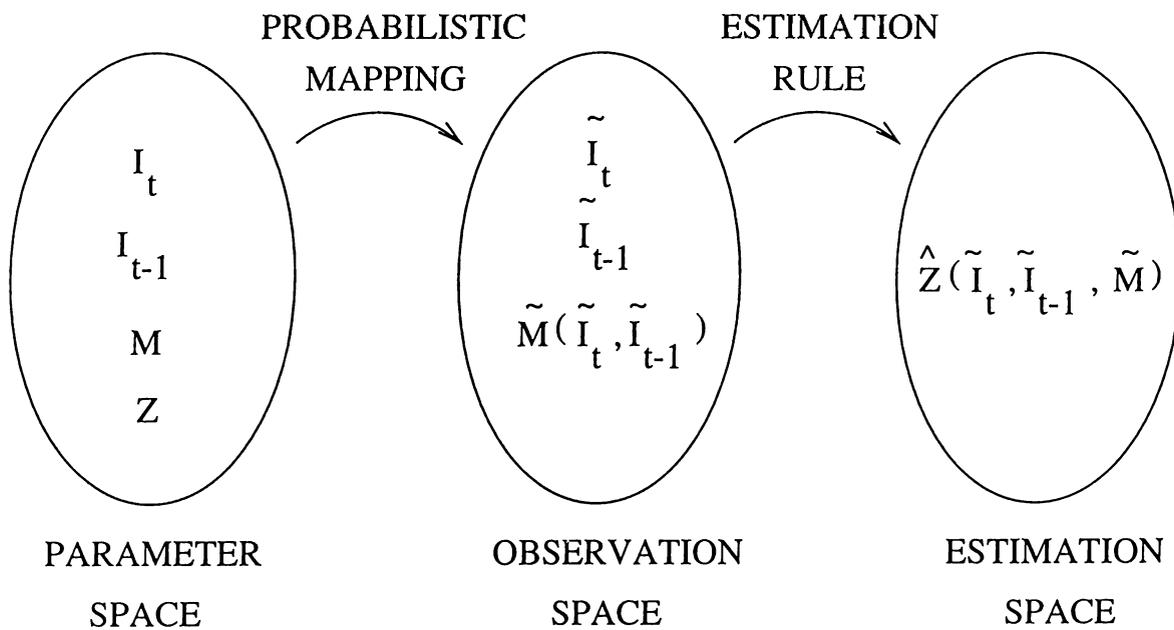


Figure 4.1: Depth estimation using MAP formulation.

4.1.1 Formulation

MAP estimate of the depth field can be found by maximizing the probability density below :

$$\max_{\mathcal{Z}} \{ P(\mathcal{Z} | \tilde{\mathcal{M}}, \tilde{\mathcal{R}}, \tilde{\mathcal{I}}_t, \tilde{\mathcal{I}}_{t-1}) \} . \quad (4.1)$$

If this density can be written as a Gibbsian, the corresponding depth estimate will be

equal to

$$\hat{\mathcal{Z}} = \arg\{\min_{\mathcal{Z}} \mathcal{U}(\mathcal{Z} | \tilde{\mathcal{M}}, \tilde{\mathcal{R}}, \tilde{\mathcal{I}}_t, \tilde{\mathcal{I}}_{t-1})\} . \quad (4.2)$$

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

$$\mathcal{U}(\mathcal{Z} | \tilde{\mathcal{M}}, \tilde{\mathcal{R}}, \tilde{\mathcal{I}}_t, \tilde{\mathcal{I}}_{t-1}) = \mathcal{U}_N + \lambda_Z \mathcal{U}_Z , \quad (4.3)$$

where

$$\mathcal{U}_N = \sum_{\mathbf{x}_p \in \Lambda} \left(\tilde{\mathcal{I}}_t(\mathbf{x}_p(t)) - \tilde{\mathcal{I}}_{t-1}(\mathbf{x}_p(t-1)) \right)^2 , \quad (4.4)$$

$$\mathcal{U}_Z = \sum_{\mathbf{x}_p \in \Lambda} \sum_{\mathbf{x}_{p,c} \in \eta_{\mathbf{x}_p}} (Z_p(\mathbf{x}_p, t) - Z_p(\mathbf{x}_{p,c}, t))^2 \cdot \delta \left(\tilde{\mathcal{R}}(\mathbf{x}_p) - \tilde{\mathcal{R}}(\mathbf{x}_{p,c}) \right) . \quad (4.5)$$

In the equation above, $\mathbf{x}_p(t-1)$ is the previous coordinate of the object point, \mathbf{p} , corresponding to the current coordinate $\mathbf{x}_p(t) = (x_p(t), y_p(t))$, and these coordinates are related by Equation 3.8. $\mathbf{x}_{p,c}(t)$ is the neighbor coordinate of $\mathbf{x}_p(t)$ defined in $\eta_{\mathbf{x}_p}$.

In Equation 4.5, \mathcal{U}_Z term 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. \mathcal{U}_Z term can also be chosen differently while still supporting the smooth variation of the depth field. The difference between the spatial differentials of the neighboring depth values can be utilized rather than simply taking the difference between depth values. In this case, the smoothness of the depth fields will be more emphasized. Obviously the smooth variation of depth field is not valid along object boundaries which are segmented previously by the \mathcal{R} field. On the other hand, similar to Equations 2.1 and 3.15, \mathcal{U}_N term models the difference between the current intensity and its motion compensated prediction as a Gaussian noise.

The minimization of Equation 4.3 can be achieved by similar global optimization methods which are examined in Section 2.3.2. Since there is only one unknown (depth value) at each image point, the computation time is considerably smaller with respect to 2-D motion estimation and segmentation.

Using the previously estimated dense 2-D motion vectors, a $Z_p(\mathbf{x}_p, t)$ value can also be found by linearly solving Equation 3.8 independently at each location. However, such

an attempt might result in degraded results since the performance of this estimation is susceptible to both 2-D and 3-D motion parameter errors and there might be some “untrustable” estimates among the dense 2-D motion vectors. The E-matrix method finds the depth values in a similar noise-prone way [93].

The *MAP* estimate, \hat{Z} , is a dense depth field, consisting of $\hat{Z}(x_p, t)$ defined at each point on the image. Hence, the intensity of all points can be motion compensated (i.e., predicted by the 3-D motion parameters and the depth value at that point using Equation 3.8) from the previous reconstructed frame at the receiver, if the 3-D motion parameters and dense depth field are transmitted for each object. An object-based 3-D motion and depth estimation algorithm is proposed using the methods explained up to this point. The corresponding flowchart is shown in Figure 4.2. However, in order to apply this algorithm in video coding, the encoding of this dense depth field should be achieved; a solution to this challenging problem is presented in Section 4.2.

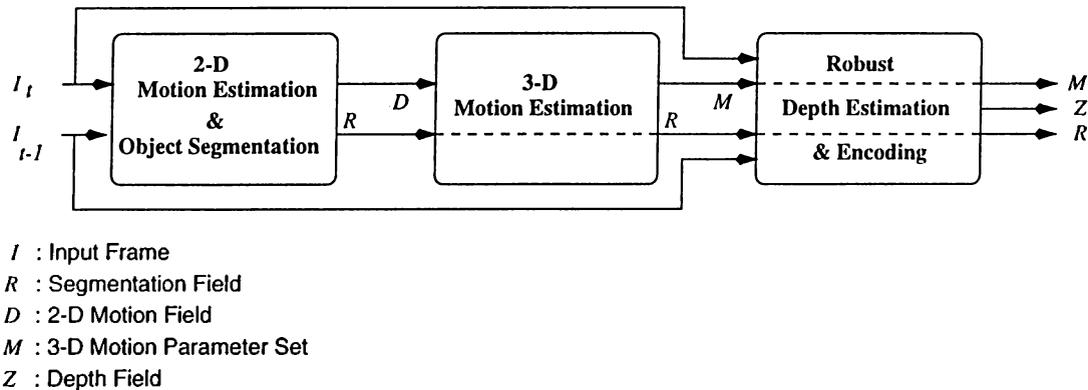


Figure 4.2: The proposed rigid 3-D object-based motion and depth estimation scheme which can be used in object-based video coding.

4.1.2 Simulations

In this section, some experimental results are presented to evaluate the performance of the proposed *MAP*-based depth estimation algorithm. Since the main advantage of the algorithm is its noise immunity, the algorithm should be compared with some

conventional depth estimators in noisy sequences. The experiments are conducted in two phases, in which artificial and standard sequences are used, respectively.

ROBUST DEPTH ESTIMATION USING ARTIFICIAL DATA

In the first phase, the intensities of an artificial frame pair, whose true depth and 3-D motion parameters are known beforehand, are injected with Gaussian noise. Using these frames, depth is found by both the E-matrix and the proposed *MAP*-based method. Since the superiority of utilizing RANSAC in 3-D motion estimation is observed during the preceding simulations, this error-robust version is used to find the 3-D motion parameters in the E-matrix method. In order to quantitatively compare two methods for different noise levels, five *quality parameters*, which are very similar to the test parameters in Section 3.2.1 are defined as follows :

- $Q_1 \doteq \frac{\sum_{\mathbf{x} \in R_i} |D_{2D}^x(\mathbf{x}) - D_{True}^x(\mathbf{x})|}{\sum_{\mathbf{x} \in R_i} |D_{True}^x(\mathbf{x})|}$, where D_{2D}^x is the horizontal component for the 2-D projection of the estimated 3-D motion and D_{True}^x is the horizontal component for the *true* projected motion for the Object i .
- $Q_2 \doteq \frac{\sum_{\mathbf{x} \in R_i} |D_{2D}^y(\mathbf{x}) - D_{True}^y(\mathbf{x})|}{\sum_{\mathbf{x} \in R_i} |D_{True}^y(\mathbf{x})|}$, where D_{2D}^y is the vertical component for the 2-D projection of the estimated 3-D motion and D_{True}^y is the vertical component for the *true* projected motion for the Object i .
- $Q_3 \doteq \frac{n_t}{N}$, where n_t is the number of negative depth values for N tested points at time t .
- $Q_4 \doteq \frac{\sum_{\mathbf{x} \in R_i} |Z_{True}(\mathbf{x}) - \hat{Z}(\mathbf{x})|}{\sum_{\mathbf{x} \in R_i} |Z_{True}(\mathbf{x})|}$, where \hat{Z} is the estimated and Z_{True} is the *true* depth value at the corresponding location for the Object i .
- $Q_5 \doteq \frac{1}{N} \sum_{\mathbf{x}_p \in R_i} (\tilde{I}_i(\mathbf{x}_p(t)) - \tilde{I}_{t-1}(\mathbf{x}_p(t-1)))^2$, where the relation between $\mathbf{x}_p(t)$ and $\mathbf{x}_p(t-1)$ is given using the Equation 3.8.

After these definitions, the simulations can be evaluated. Two consecutive frames from the noise-free artificial frame sequence *Salecube* and the ideal segmentation of the moving cube in the second frame are shown in Figure 4.3. In these frames, the background (everything except the cube) is stationary and depth field of this part is beyond the scope of this dissertation (indeed, since the background is stationary, it can be taken as a poster with no depth variation).

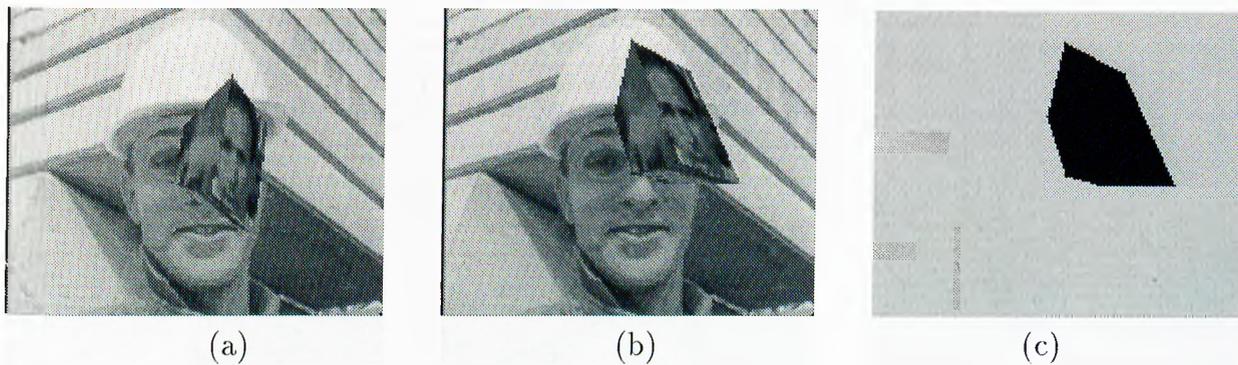


Figure 4.3: Original (a) first and (b) second frames of *Salecube* sequence. (c) The ideal segmentation result.

In order to test noise immunity, Gaussian noise is injected into both frames of this sequence. For different noise levels, the resulting noisy second frames are presented in Figure 4.4.

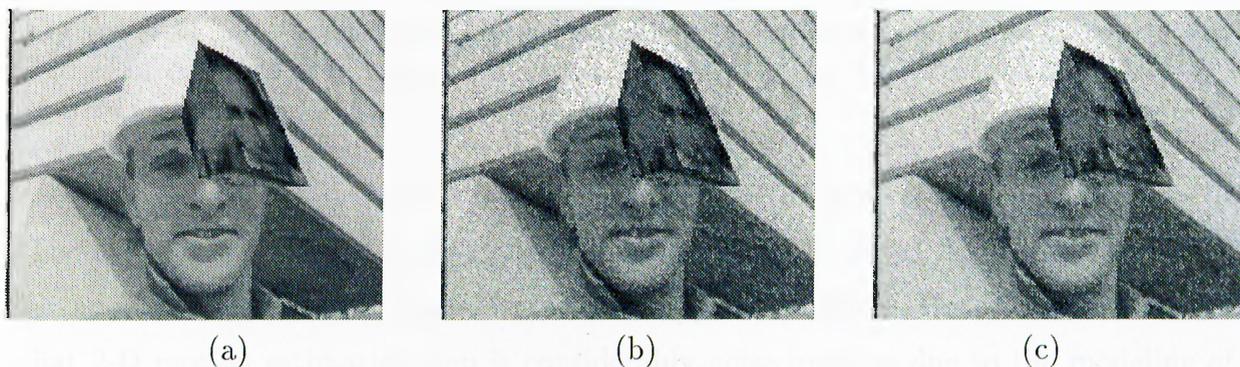


Figure 4.4: Second frame from *Salecube* sequence. The results after noise injection : (a) 35 dB, (b) 25 dB, (c) 15 dB.

In the first step of this phase, the noise immunity of the 2-D motion estimation algorithm is tested, since correct 2-D projections of the real motion is known. The

results are given in Table 4.1. Some typical results are also shown in Figure 4.5.

Table 4.1: Noise analysis of 2-D motion estimation step for *Salecube* sequence. Two quality parameters, $Q_{1,2}$ and the SNR_{peak} of the reconstructed second frame are tabulated for different noise levels.

λ	Q_1	Q_2	$SNR_{peak}(dB)$
Noiseless	0.1460	0.1945	31.68
45 dB	0.1456	0.1956	31.66
35 dB	0.1491	0.2002	29.36
25 dB	0.1512	0.2564	25.13
15 dB	0.2293	0.2611	22.46

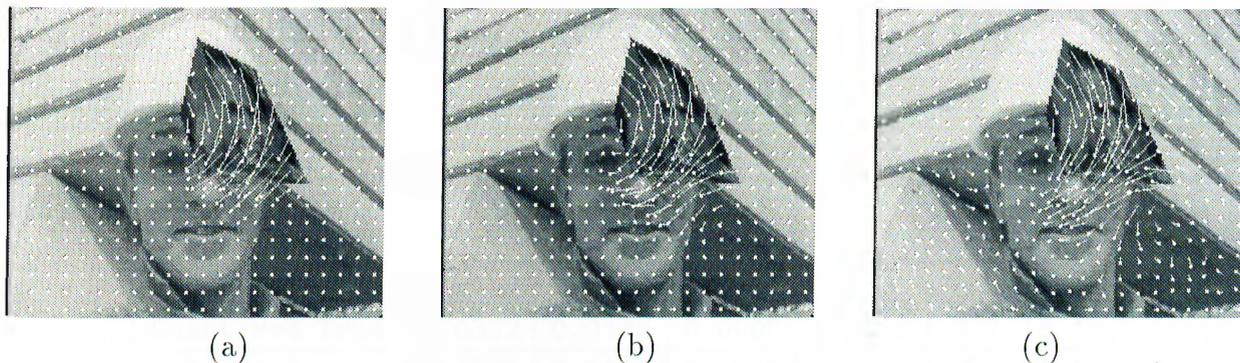


Figure 4.5: The needlegram representation of the motion between first and second frames of *Salecube* sequence. The results are obtained for (a) true, (b) noise-free, (c) 15 dB cases.

As it is expected, in Table 4.1 the performance of the motion estimates decreases as the level of noise inherent in the image frames increases. However, Figure 4.5 shows that the amount of error in 2-D motion estimation step is not critical. Hence, it can be stated that 2-D motion estimation step is considerably noise immune due to the modeling of noise in Gibbs formulation (Equation 2.1). Sensitivity analysis can also be conducted to understand the effects of noise better.

Using the results of 2-D motion estimation step, 3-D motion and structure analysis is achieved. Depth estimates are obtained using both the conventional E-matrix (with

RANSAC) [93] and MAP-based formulation of Equation 4.3. The quality parameters are used to evaluate the similarity of the estimated depth with its true counterpart. Apart from different noise levels, in order to see the effect of 2-D motion estimation, the true (quantized) projected motion, which can be the output of an ideal 2-D motion estimation algorithm, is also utilized in this step. The results are tabulated in Tables 4.2 and 4.3.

Table 4.2: The results of the noise analysis of depth fields for *Salecube* sequence using E-matrix method.

Input	Q_1	Q_2	Q_3	Q_4	Q_5
Quantized True	0.0354	0.0206	0.0127	0.3799	388.33
Noise-free	0.1181	0.2404	0.0000	0.3147	570.17
45 dB	0.1504	0.2066	0.0048	0.4900	632.51
35 dB	0.1227	0.2283	0.0000	0.4966	767.71
25 dB	0.3277	0.2490	0.0409	0.5113	1665.72
15 dB	0.4670	0.2823	0.1065	0.7897	4331.14

Table 4.3: The results of the noise analysis of depth fields for *Salecube* sequence using MAP-based formulation

Input	Q_1	Q_2	Q_3	Q_4	Q_5
Quantized True	0.0332	0.0175	0.0000	0.3356	327.16
Noise-free	0.1076	0.2410	0.0000	0.3147	274.64
45 dB	0.1220	0.2033	0.0000	0.4743	297.99
35 dB	0.1092	0.2251	0.0000	0.4951	301.73
25 dB	0.2020	0.1989	0.0000	0.4767	451.60
15 dB	0.3090	0.1856	0.0000	0.5468	1691.49

The most important observation in Tables 4.2 and 4.3 is the difference between the rate of change of quality parameters corresponding for the E-matrix and proposed method, while the input error increases. With small amount of noise, both algorithms have almost similar performances while estimating the depth. However, the quality of the depth estimates for the E-matrix method degrades considerably with respect to MAP-based formulation while the input noise increases. Some typical resulting images are shown in Figures 4.6, 4.7 and 4.8.

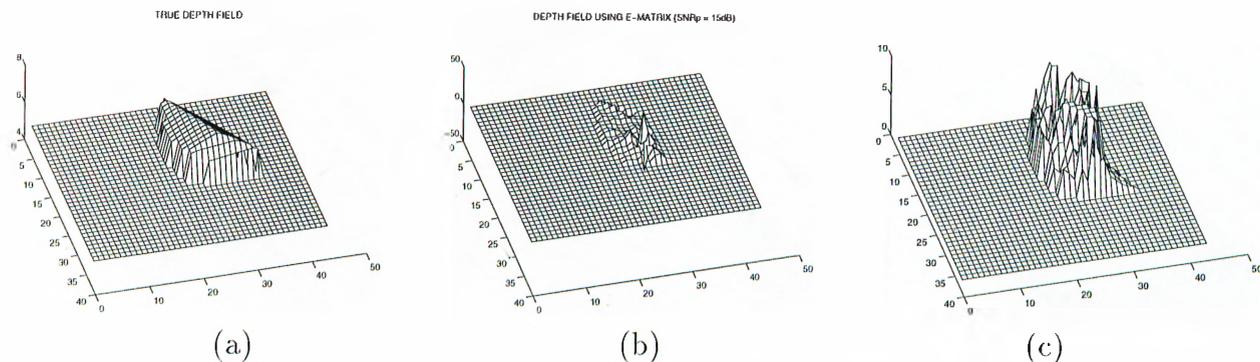


Figure 4.6: The mesh representations of the depth fields for the second frame of *Salecube* sequence. The results are obtained for (a) true, (b) 15 dB E-matrix, (c) 15 dB proposed algorithm, cases.

In Figure 4.6(b), the estimate depth field has gross errors due to noise susceptibility of the E-matrix method. As it can be observed from the corresponding depth, which is obtained from *MAP* estimation, the depth field improves both visually and quantitatively (Table 4.3). The needlegrams in Figure 4.7 also show that noise immunity of the proposed method is better.

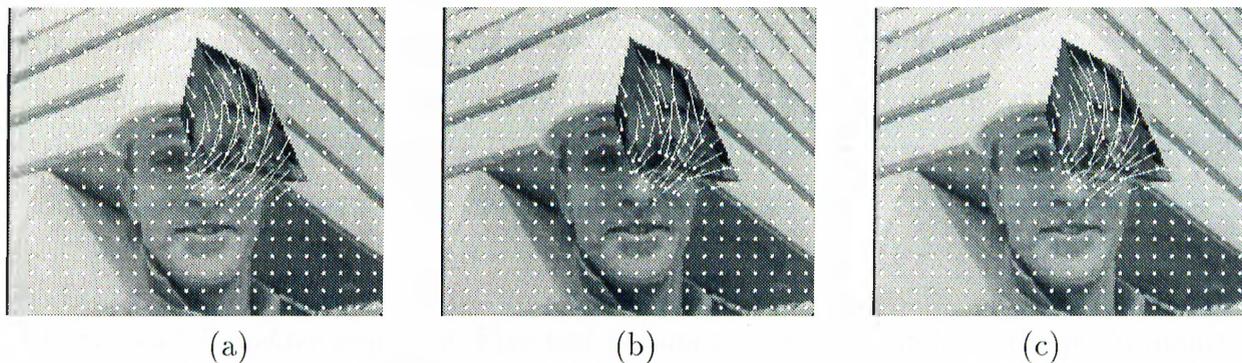


Figure 4.7: The needlegram representations of the 2-D motion field which is obtained after projecting the estimated 3-D motion and depth field of the second frame of *Salecube* sequence. The results are obtained for (a) true, (b) 15 dB E-matrix, (c) 15 dB proposed algorithm, cases.

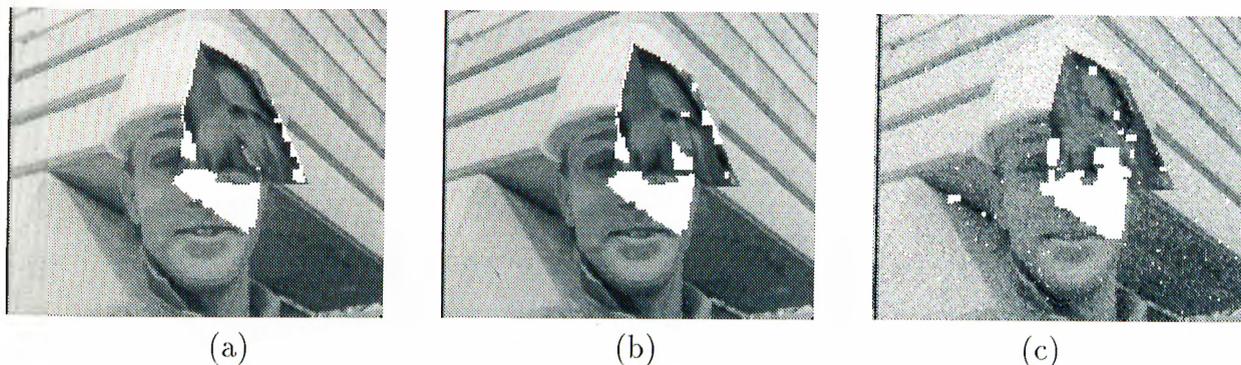


Figure 4.8: The reconstructed frame of the second frame of *Salecube* sequence which is obtained using the the projected 2-D motion field of the estimated 3-D motion and depth field. The results are obtained using MAP-based method for (a) noise-free, (b) 45 dB, (c) 25 dB, cases.

ROBUST DEPTH ESTIMATION USING REAL DATA

In the second phase of the experiments, 38th and 41th frames of *Mother and Daughter* sequence, and the previously obtained segmentation field (Figure 2.7) are used to evaluate the performance of the proposed technique. Similar to the previous phase, the original frames are contaminated with Gaussian noise. Using the proposed robust version of the E-matrix method, 3-D motion estimates are obtained for both noise-free and noisy data. The 3-D motion estimates are compared using the test parameters ($T_{1,2,3,4,5}$) and performance indicator (P) of Section 3.2.1 in Table 4.4

Table 4.4: Noise analysis of 3-D motion estimation step for 38th and 41th frames of *Mother and Daughter* sequence. Five test parameters, $T_{1,2,3,4,5}$ and the the performance indicator, P , are tabulated for noise-free and noisy cases.

λ	T_1	T_2	T_3	T_4	T_5	P
Noise-free	0.467	0.928	0.178	0.000	0.000	0.388
15 dB	1.274	0.651	0.114	0.000	0.655	0.271

The effect of noise on 3-D motion parameter estimation can be observed in Table 4.4. Using the estimated parameters, depth field is found for both noisy and noise-free cases

using E-matrix and MAP-based methods. The results are shown in Figure 4.9.

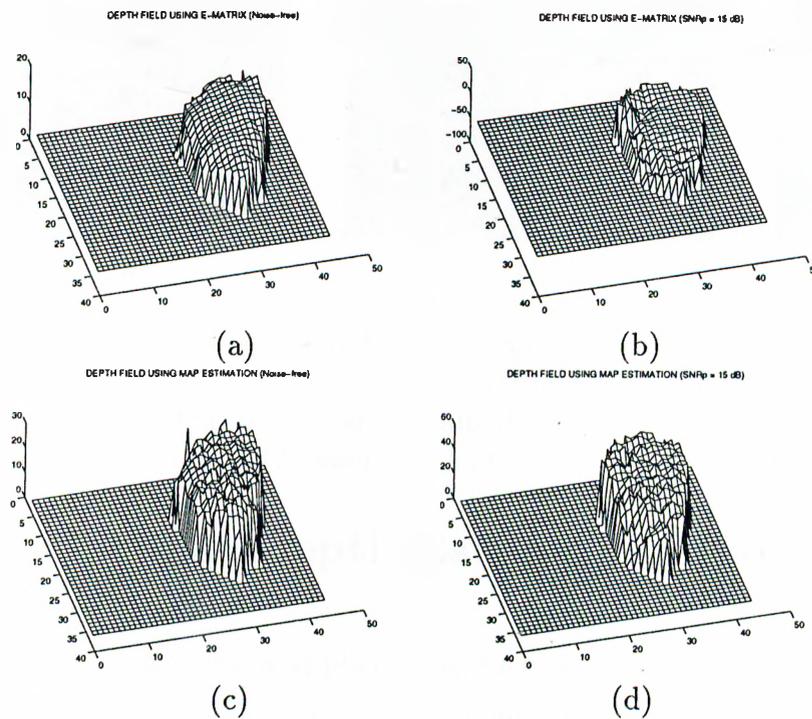


Figure 4.9: The depth maps of the 41th frame of the *Mother and Daughter* sequence. The results are obtained using the E-matrix method for (a) noise-free, (b) 15 dB cases and also MAP-based method for (c) noise-free, (d) 15 dB, cases

Using the estimated depth values and 3-D motion parameters, the reconstructed 41th frame of *Mother and Daughter* sequence is shown in Figure 4.10 for noise-free cases using E-matrix and the proposed MAP-based formulation.

In Figure 4.10(b), it can be seen that the reconstructed face is deformed. Since this linear method does not take into account the intensity matches, on contrary to MAP estimation, the reconstructed video frames might be severely distorted; this is a severe drawback from coding point of view.

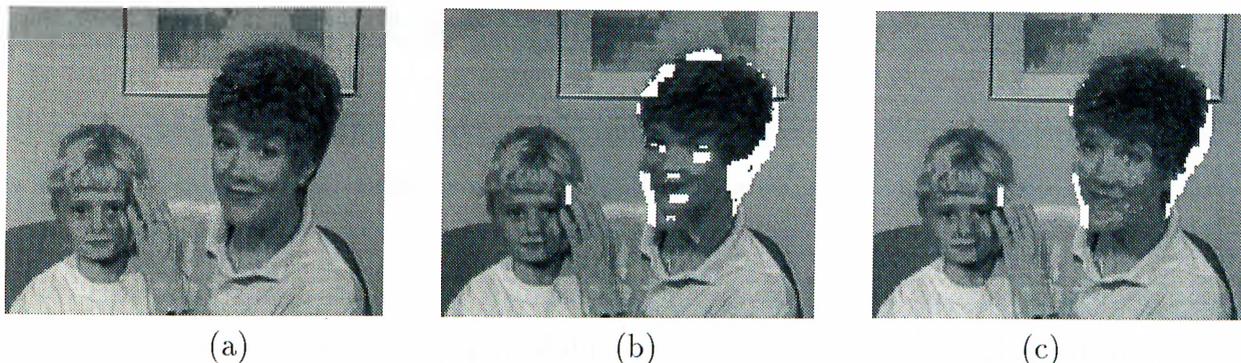


Figure 4.10: The reconstructed noise-free 41th frame of *Mother and Daughter* sequence which is obtained using the the projected 2-D motion field of the estimated 3-D motion and depth field. The results are obtained for noise-free cases. (a) Original (b) using E-matrix method (c) *MAP*-based method.

4.2 Optimal Depth Estimation and Encoding

In many computer vision applications, such as robotics, estimation of *physically true* depth values is necessary. However in video coding, depth estimation problem can be solved in a different manner since true depth has secondary importance with respect to motion compensation between consecutive frames and encoding of this dense depth field. Furthermore, due to projection from the 3-D world to 2-D scenes, the physically true depth field may not be solved. In the rest of this section, the term “true” depth field means the set of depth values which lead to the exact intensity matches between consecutive frames. Hence, an approach to encode the depth field between two frames can be a joint optimization of the *distortion* of the reconstructed frame and the *bit-rate* of the encoded depth field. Moreover, such an approach can be formulated to find a dense depth field which can not be obtained by most of the other methods [1]. It should be also noted that none of the current video coding methods with 3-D motion models propose a method for the efficient encoding of the depth field [20, 95, 96, 83, 94], except for some depth encoding algorithms in stereo video coding applications [128].

4.2.1 Theoretical Limits of Depth Encoding

Since any 3-D scene can be assumed as an output of a random source, the depth field of a scene will be a random field. The depth fields and also intensity frames obtained after perspective projection of the scene are only some mappings from this random source to a set of numbers and hence they are also random fields, having associated probability distributions. The assignment of probability to a depth field is meaningful if it matches the frequency of occurrence of that field in the real world; it is assumed that such an assignment is made. Using this probability measure, number of bits to encode this depth field can be determined according to basic principles of information theory [129]. However, such an encoding approach is lossless and it may not be preferable in most of the very low bit-rate applications.

Rate-distortion theory [129] seeks for the minimum achievable rate of a source to be encoded under a distortion constraint. It is assumed that an unknown “process” exists between the true value at the source and reconstructed value at the receiver and rate-distortion theory guarantees reaching the minimum bit-rate using this process for an arbitrary distortion value [129]. According to this theory, rate and distortion have a mutual relation as it is shown in Figure 4.11. Entropy constrained vector quantization is one of the applications of rate-distortion theory to image coding [130].

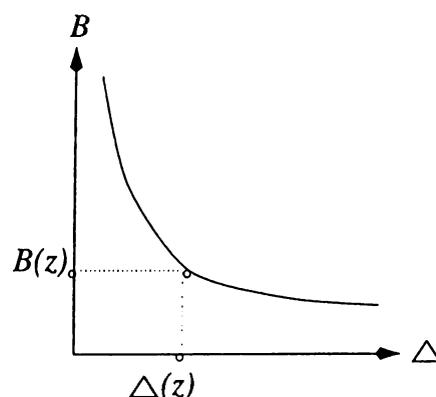


Figure 4.11: Rate (B) versus Distortion (Δ)

Inspired from rate-distortion theory, an algorithm can be proposed to find the dense

depth values to be encoded. However, it should be noted that the number of bits to encode the unknown true depth field, which have zero distortion, is assumed to be higher than the “target” bit-rate or at least minimizing this bit-rate is still preferable. Otherwise lossless encoding of the true depth field with sufficient number of bits, yielding zero distortion, is the global optimum solution. Hence the aim is to decrease the bit-rate by sacrificing from quality in an effective way. For a defined distortion measure and a probability distribution of depth field source, a theoretical rate-distortion function exists. This function gives the relation between the distortion and the minimum amount of bits to encode the depth field that creates the corresponding distortion. For any given distortion value, there exists a minimum, but the procedure of finding such a minimum is unknown. A possible solution for finding the minimum is to minimize a function \mathcal{J} which takes into account both bit-rate and distortion with respect to the depth field to be encoded. Hence the minimization process maps the true depth field to the encoded counterpart while taking into account distortion and bit-rate.

4.2.2 Selection of Encoding Criteria

In order to find a depth field to encode, a function, $\mathcal{J}(\Delta, \mathcal{B})$, which represents both distortion Δ and bit-rate \mathcal{B} , can be minimized. There are many different ways to approach this *vector optimization* problem and the method of *objective weighting* [131] is a possible choice, which can be written as

$$\mathcal{J}(\Delta, \mathcal{B}) = \Delta + \lambda_0 \cdot \mathcal{B} \quad (4.6)$$

where λ_0 is a constant which reflects the weighting between two different quantities Δ and \mathcal{B} . Before achieving joint optimization of bit-rate and depth, a distortion criteria and a measure of bit-rate should be defined.

DISTORTION CRITERIA

It is possible to define the distortion between the true and reconstructed depth values using input frame intensities. The distortion criteria, Δ can be defined as the average error over objects between the original and reconstructed frames as

$$\Delta = \frac{1}{N} \sum_{\mathbf{x} \in R_i} (I_t(\mathbf{x}) - \hat{I}_t(\mathbf{x}))^2 \quad (4.7)$$

where N is the total number of object points in region R_i . I_t is the original frame which can also be written as below with the assumption that it is in a non-occlusion region, the illumination does not change, and there is no noise:

$$I_t(\mathbf{x}) = I_{t-1}(\mathbf{x} - \mathbf{D}_{2D}(Z(\mathbf{x}, t))) \quad (4.8)$$

The reconstructed frame, \hat{I}_t is also equal to

$$\hat{I}_t(\mathbf{x}) = I_{t-1}(\mathbf{x} - \mathbf{D}_{2D}(\hat{Z}(\mathbf{x}, t))) \quad (4.9)$$

In the above equation $Z(\mathbf{x}, t)$ is the true and $\hat{Z}(\mathbf{x}, t)$ is the encoded depth values at location \mathbf{x} , In the rest of this section t is dropped for notational simplicity and those depth functions are denoted as $Z(\mathbf{x})$ and $\hat{Z}(\mathbf{x})$. Each location \mathbf{x} on the image plane at time t has a corresponding 3-D object point \mathbf{X} under perspective projection. Each \mathbf{X} has an assigned 3-D motion vector $\mathbf{M}_{3D}(\mathbf{X})$ which indicates its 3-D displacement from time $t - 1$ to t (Figure 2.1). The perspective projection of $\mathbf{M}_{3D}(\mathbf{X})$ is denoted as $\mathbf{D}_{2D}(\mathbf{x})$. Therefore, $\mathbf{D}_{2D}(\mathbf{x})$ depends implicitly on the depth component of \mathbf{X} , which is $Z(\mathbf{x})$. To stress this dependence the notation $\mathbf{D}_{2D}(Z(\mathbf{x}))$ is used in Equation 4.8, 4.9 and 4.15.

Since the true depth is unknown, it is better to use the lhs of Equation 4.8 instead of the rhs. Using this measure, the unknown true depth is implicitly available at the lhs of Equation 4.8 in the intensity information at frame I_t . In other words, the difference between true and encoded depth fields are defined in a nonlinear way as in Equation 4.7. Such a nonlinear distortion measure is more preferable than taking directly the difference between true and estimated depth values, because the dense true depth values can not be obtained at each point. Besides compensation of intensities is more appropriate for video coding applications.

BIT-RATE OF ENCODED DEPTH

In order to find an estimate for the bit-rate of the encoded depth field, a probability measure for the depth field must be defined. According to the source coding theorem, it is possible to find the number of bits required to encode any depth field by using this probability distribution. Although it is impossible to find an exact distribution of the dense depth fields existing in 3-D world, some assumptions can be made. In many indoor scenes, it is more likely to observe objects with smooth depth variations except at object boundaries. Hence a Gibbs distribution taking into account these observations can be written.

For each segmented object in the scene, the joint conditional probability distribution function of the encoded depth field, \mathcal{Z} , can be written as

$$\mathbf{P}(\mathcal{Z}) = \frac{e^{-\mathcal{U}_{\mathcal{Z}}(\mathcal{Z}) \cdot k}}{\sum_{\mathcal{Z}} e^{-\mathcal{U}_{\mathcal{Z}}(\mathcal{Z}) \cdot k}} \quad (4.10)$$

where k is the energy constant; the denominator is the normalization factor and $\mathcal{U}_{\mathcal{Z}}$ is the Gibbs energy function. Since the only *a priori* information is having a smooth depth field over each segmented object, the energy function of the Gibbs distribution can be written as

$$\mathcal{U}_{\mathcal{Z}}(\mathcal{Z}) = \sum_{\mathbf{x} \in R_i} \sum_{\mathbf{x}_c \in \eta_{\mathbf{x}}} \left(\hat{Z}(\mathbf{x}) - \hat{Z}(\mathbf{x}_c) \right)^2 \quad (4.11)$$

where the sum is over all points \mathbf{x} of the i th object, segmented by the region R_i and $\eta_{\mathbf{x}}$ is the neighborhood of \mathbf{x} . Such a distribution function gives higher probability to smooth surfaces, which is not contradictory to indoor studio scenes.

The required number of bits, \mathcal{B} , of the depth field to be transmitted to receiver side is simply equal to

$$\mathcal{B} = -\log_2(\mathbf{P}(\mathcal{Z})) \quad (4.12)$$

which can be written as

$$\mathcal{B} = k \cdot \log_2 e \cdot \sum_{\mathbf{x} \in R_i} \sum_{\mathbf{x}_c \in \eta_{\mathbf{x}}} \left(\hat{Z}(\mathbf{x}) - \hat{Z}(\mathbf{x}_c) \right)^2 + c(k) \quad (4.13)$$

where constant $c(k)$ does not depend on \mathcal{Z} and is equal to

$$c(k) = \log_2 \left(\sum_{\mathcal{Z}} e^{-U_{\mathcal{Z}} \cdot k} \right) \quad (4.14)$$

The value of $c(k)$ parameter is simply equal to $\frac{N}{2} \log_2 \left(\frac{\pi}{4k} \right)$ since the Gibbs distribution, which is defined in Equation 4.11, is Gaussian and the normalization factor of a normal distribution is explicitly known. Hence Equation 4.13 gives the required number of bits to encode the transmitted depth field, \mathcal{Z} . According to this bit-rate measure, smooth surfaces will need less number of bits to be encoded and this result again supports intuitive reasoning.

MINIMIZATION OF THE ENCODING CRITERIA

Distortion and bit-rate is jointly optimized by minimizing Equation 4.6 with respect to \mathcal{Z} and this can be written as

$$\min_{\mathcal{Z}} \left\{ \left(\frac{1}{N} \sum_{\mathbf{x} \in R_i} \left(I_t(\mathbf{x}) - I_{t-1}(\mathbf{x} - \mathbf{D}_{2D}(\hat{Z}(\mathbf{x}))) \right)^2 \right) + \lambda \left(\sum_{\mathbf{x} \in R_i} \sum_{\mathbf{x}_c \in \eta_{\mathbf{x}}} \left(\hat{Z}(\mathbf{x}) - \hat{Z}(\mathbf{x}_c) \right)^2 \right) \right\} \quad (4.15)$$

Since $c(k)$ parameter does not depend on \mathcal{Z} , it is removed from Equation 4.15. k and $\log_2(e)$ constants can be multiplied with λ_0 constant and hence this product is defined to be λ .

Minimization of Equation 4.15 is a global optimization problem and Simulated Annealing (SA) [48] is a possible, computationally demanding but optimal solution. A suboptimal version of SA is ICM algorithm [51] which gives comparable results with SA with much less computation time, when it has good initial estimates or applied in a multiresolution manner [64].

For different choices of λ , different values for rate and distortion are obtained. For a given bit-rate (or distortion), the corresponding distortion (or bit-rate) is optimal. λ may be specified externally. Equivalently, some external constraints may be used to imply a λ . For example, there might be some upper limits or some “target” values for bit-rate and/or

distortion in low bit-rate video coding applications, these limits can be added to the minimization problem as extra constraints so that a λ value can be selected. Although, $c(k)$ parameter is removed from minimization, in order to calculate a theoretical bit-rate value, this parameter should be considered after assigning an appropriate value to parameter k . Since it is almost impossible to find the “true” value of k , it is better to fix a target distortion value, instead of bit-rate, to select among λ values. Hence the λ value is chosen so that the corresponding distortion is nearest to the target distortion value among all possible λ values. Since it is computationally costly to make minimization for all values of λ , a suboptimal solution can be selecting the best λ among a predetermined small discrete set.

Given previous reconstructed frame and 3-D motion parameters, the unknown depth field at present frame is determined by taking into account both reconstructed frame quality and the amount of bits required to encode this depth field. It is possible to obtain better results with respect to any encoding algorithm which does not take into account the rate-distortion dilemma. Hence, this approach is well-suited for video coding applications with a 3-D motion model.

THE DEPTH ENCODER

The intensity of any point belonging to an object in the current frame will be predicted from a corresponding intensity in the previous reconstructed frame using only the encoded 3-D motion parameters and depth field. Assuming three rotation angles and three translation values can be encoded by very few bits, the compression performance of this method mainly depends on depth encoding. Using the methods explained in the previous sections, an algorithm is proposed in order to encode the depth field between two consecutive frames.

Although finding an efficient depth field to be encoded is explained, approaching to the theoretical bit-rate limit is still unknown. Since it is impossible to give a codeword to all existing depth fields according to their probabilities, another coding strategy must

be followed. Predictive coding can be applied to remove the redundancy existing in the depth field. By linearly predicting each depth value using its causal neighbors and sending only the prediction error, some compression is possible. Before transmission, the redundancy in this error field can further be removed by using one of the lossless compression algorithms. The lossless encoding is achieved as follows: every depth value is predicted from its casual horizontal and vertical neighbors (\mathbf{x}_{hor} and \mathbf{x}_{ver} respectively) simply as

$$\hat{Z}_e(\mathbf{x}) = 0.5 \left(\hat{Z}(\mathbf{x}_{\text{ver}}) + \hat{Z}(\mathbf{x}_{\text{hor}}) \right) \quad (4.16)$$

and the prediction error, $\hat{Z} - \hat{Z}_e$, is encoded in a lossless manner using Lempel-Ziv coding[129] . Although sophisticated schemes may be employed, this simple prediction gives satisfactory results.

Hence, the overall algorithm can be summarized as below :

1. Find 2-D correspondences between two 2-D consecutive frames (projections of the 3-D scene) and segmentation of the current image by minimizing Equation 2.1.
2. Find 3-D motion parameters (rotation matrix and translation vector) using E-matrix method (with RANSAC) by the help 2-D correspondences for each object, separately.
3. Using segmented 3-D motion parameters and two input frames, minimize Equation 4.15 for a given λ to find the best depth field.

If λ is not given externally, it is possible to continue as follows :

- 3.b Repeat step 3 for various values of λ .
- 3.c Choose the best depth estimate among different λ values according to a “target” distortion.
4. Encode the selected best depth field using a lossless compression method.

Using this algorithm, the intensities of each segmented rigid object are reconstructed at the receiver by using the previous available frame, 3-D motion parameters (rotation matrix and translation vector) and encoded depth field.

4.2.3 Simulations

Experiments on depth encoding are conducted again in two phases. In the first phase, an artificial sequence, whose 3-D motion parameters and depth values are known, is used. Hence in this phase, motion estimation is not considered and it is assumed that all the necessary motion parameters were successfully estimated. The aim of this phase is to observe the performance of the method without taking into account errors due to motion estimation. In the second phase of experiments, frames from standard sequences are used to estimate 3-D motion and depth field of objects. The results of this phase give the required number of bits to encode the moving objects for a given distortion value.

DEPTH ENCODING USING ARTIFICIAL DATA

Two consecutive frames (64×64) of the artificial *Cube* sequence with known 3-D motion and depth field, are shown in Figure 3.1. In these frames, a cube is rotating along x , y and z axes with 6 degrees each and translating with 6 pixels at the same time. The focal length of the system is equal to 50 pixels and the average depth of the cube from the origin is 60 pixels. Hence the cube is moving near the image plane and perspective deformations are visible, although the cube remains rigid. The background is a 2-D poster of a girls head. The segmentation of the frames is also achieved by hand and shown in Figure 4.14. The optical axis, which is shown as the z -axis in Figure 2.1, passes through the upper left corners of these images.

By the help of the 3-D motion parameters and the segmentation field, the minimization of Equation 4.15 is achieved using a multiscale version of the deterministic ICM algorithm [51, 64], which is explained in the previous chapters as MCR. Using the method of objective weighting, for different values of λ , different rate-distortion pairs are found by minimizing Equation 4.15 using the MCR algorithm. The obtained depth field is inserted into Equations 4.7 and 4.13 to obtain the distortion, Δ , and computed bit-rate, \mathcal{B} , values, respectively. These values are tabulated in the second and third columns of Table 4.5, respectively.

Table 4.5: The experimental results for *Cube* sequence. For different values of λ , Equation 4.15 is minimized to obtain Δ and \mathcal{B} (with $k = 0.5$) values. Bit-rate of the depth field is obtained after lossless encoding of the prediction error field.

λ	Δ	\mathcal{B}	Bit-rate(bits/object)
True Depth	0	2115	13328
1	0.001	1388	6128
10	0.105	1268	5880
100	0.740	1157	5224
1000	13.427	941	5152
10000	35.683	851	4872
100000	41.382	801	4704

Note that, \mathcal{B} values are theoretical limits for the required number of bits to encode the corresponding depth fields for a given value of $k = 0.5$. Obviously, \mathcal{B} values can only be accepted as theoretical limits if the proposed probability distribution of the depth fields can exactly match the frequency of occurrence of the corresponding fields in the real world and such an exact match for the probabilities is not guaranteed. Even if the distribution is chosen correctly, the arbitrary selection of k parameter still causes problems in determination of the theoretical bit-rate. Nevertheless, when the algorithm presented in Section 4.2.2 is used with the arbitrary selection $k = 0.5$, the values tabulated on the fourth column of Table 4.5 are obtained. The similar behavior between \mathcal{B} and the experimental results on the third column of the same table supports the validity of the overall formulation (Figure 4.11). Different selection of k might increase the similarity of the two curves even more. Therefore, as Table 4.5 and Figure 4.11 indicate, the theoretical and practical results are in harmony. Finally, it can be concluded that \mathcal{B} can be accepted as the theoretical bit-rate limit with the assumption that the proposed probability distribution and k parameter are correct.

During experiments distortion is increasing while bit-rate is decreasing, as expected (Figure 4.11). In Figure 4.12, the computed (Equation 4.13 with arbitrary selection of

$k = 0.5$) and the experimental bit-rate and distortion curves are plotted for λ values shown in Table 4.5. The selection of best λ among them can be done according to a target distortion chosen by the user. In Figure 4.13, the true and encoded depth fields ($\lambda = 1000$) are shown. The encoded depth field is much smoother especially at the intersection of cube faces with respect to true field and hence better compression is achieved in this way. However the distortion increases as a result of this smoothing.

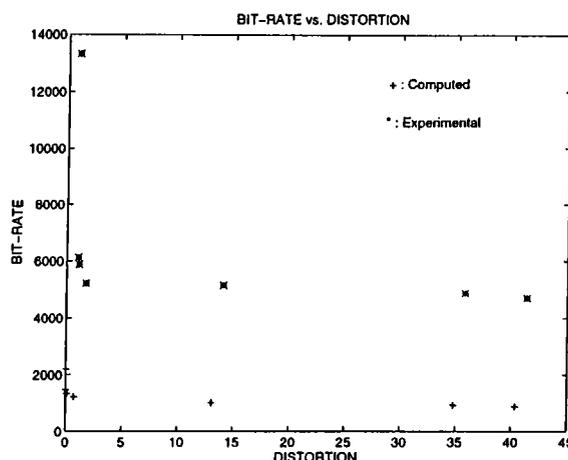


Figure 4.12: Bit-rate vs. distortion curve for computed and experimental bit-rate values for “Cube” sequence for different values of λ (tabulated in Table 4.5).

In Figure 4.14, current frame is reconstructed using 3-D motion parameters, previous frame and encoded depth field ($\lambda = 1000$). The SNR_{peak} is equal to 41 dB inside the moving cube. The needlegram of projected 3-D motion using the encoded depth field supports visual motion. The TU regions are also found according to motion compensation error between original and reconstructed frames and locates the occluded region at the top of the cube. TU regions are segmented by using Equation 2.1 where \mathbf{D} is replaced with \mathbf{D}_{2D} (see Chapter 5).

DEPTH ENCODING USING REAL DATA

Two different frame pairs are used in order to check the performance of the proposed depth encoding scheme. These pairs are shown in Figure 2.3 and 2.6, respectively. For

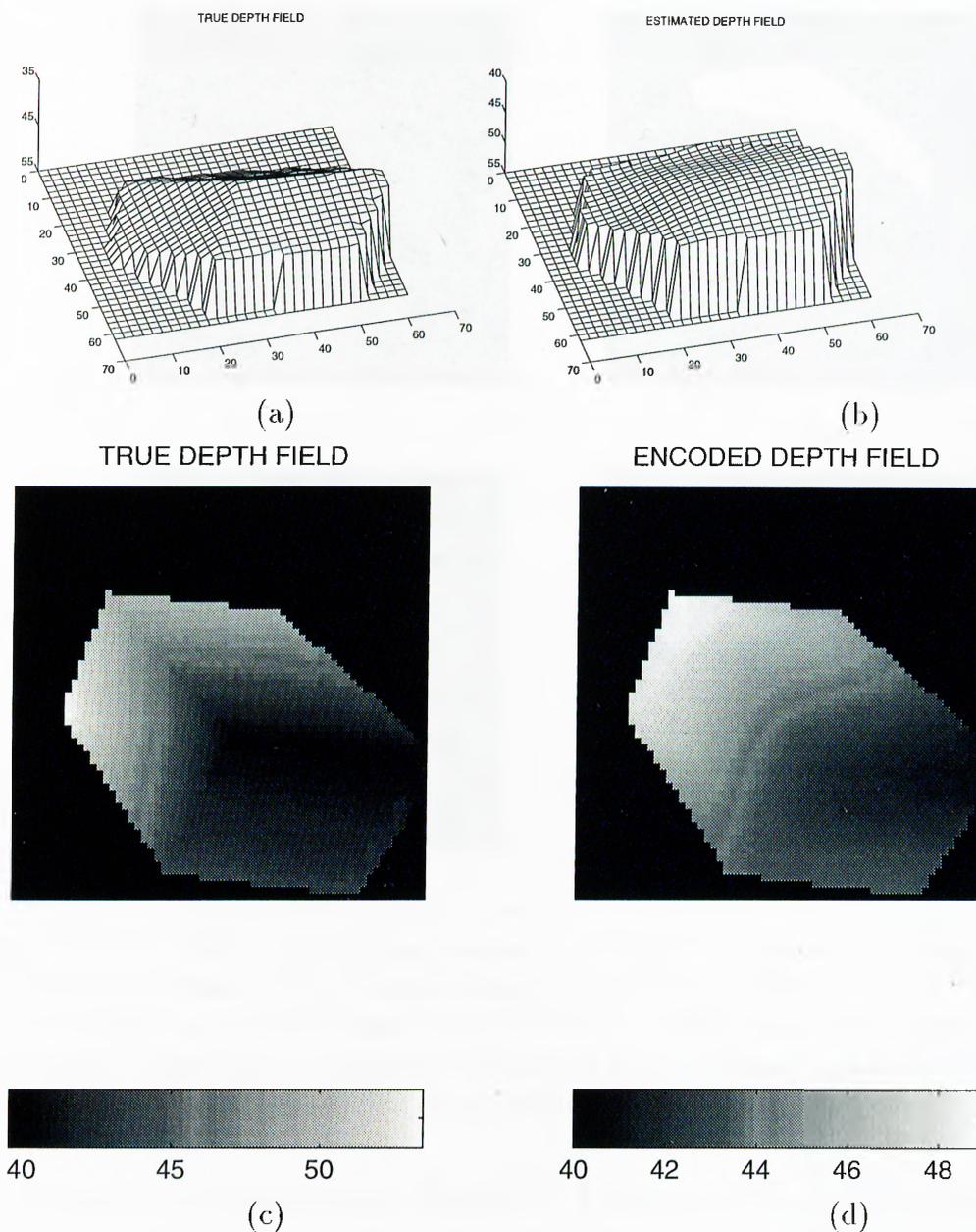


Figure 4.13: The mesh representations of the (a) true and (b) encoded depth fields of the current frame of the “Cube” sequence. (c) Depth field with intensity description (color-bar shows the depth levels with respect to intensities). Note that the assigned depth values for the background is dummy since it can not be determined by any means.

both frame pairs, the 3-D motion parameters of the segmented objects are found using the proposed algorithm in Section 3.2.1.

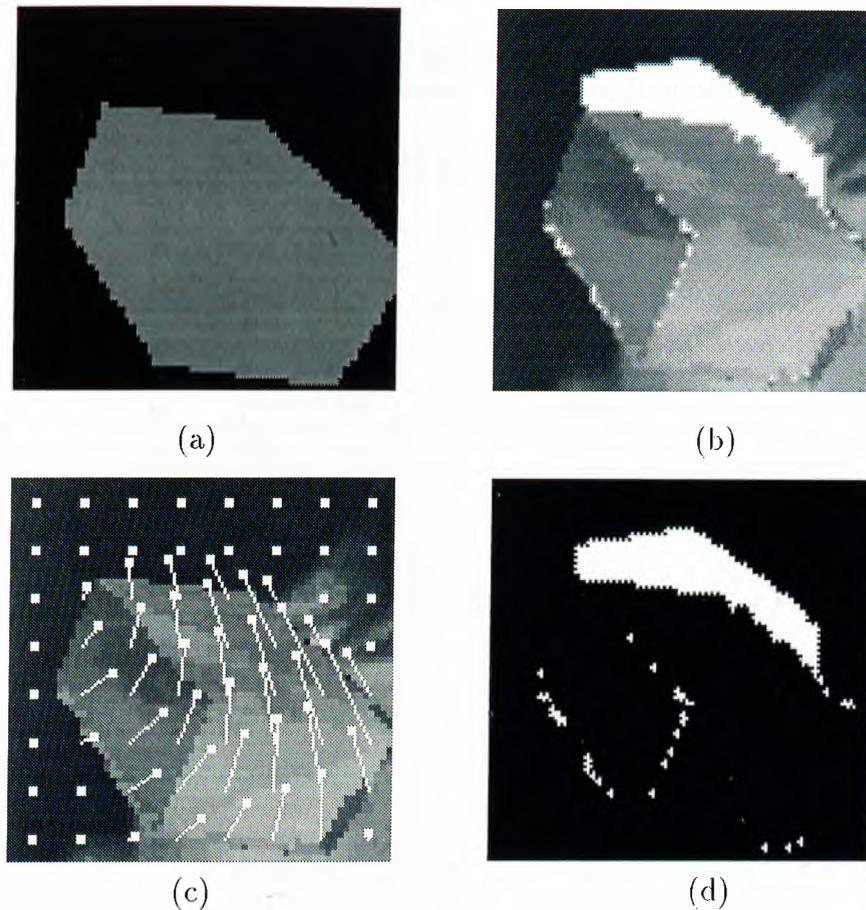


Figure 4.14: The experimental results for “Cube” sequence; (a) Segmentation, (b) Reconstructed frame using encoded depth with TU areas detected, (c) The projection of 3-D motion as a “needlegram” ($\mathbf{D}_{2D}(\hat{Z}(\mathbf{x}, \mathbf{t}))$ of Equation 4.9 is represented by the vector whose direction is from the thicker end to the thinner end of the pin where the thinner end shows $\mathbf{x}(t)$), (d) TU areas (white).

Considering the frame pair in Figure 2.6, Equation 4.15 is minimized again by using the MCR method for various values of λ and the results are shown in Figure 4.15 for an arbitrary value of $k = 0.5$ and tabulated in Table 4.6.

In the table above, the experimental bit-rate values are also shown after lossless encoding of the prediction error of the depth field. As it is expected, the distortion is decreasing for increasing number of bits to encode the depth field. If the target distortion is chosen to be 60, which corresponds to approximately 30 dB SNR_{peak} then the best value of λ can be chosen as 5. In Figure 4.16, the reconstructed current frame, which is

Table 4.6: For different values of λ , Equation 4.6 is minimized to obtain Δ and \mathcal{B} (with arbitrary $k = 0.5$) values. Bit-rate is obtained after encoding of the prediction error.

λ	Δ	\mathcal{B}	Bit-rate(bits/object)
1	33	9200	14928
5	60	4586	10312
10	65	4147	9752
15	69	3904	8928
25	75	2848	7512
50	93	2455	6288
100	118	2288	5656
1000	243	2118	3560

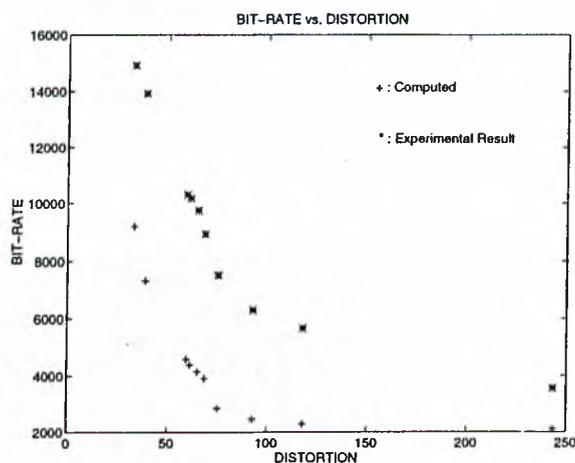


Figure 4.15: For different values of λ , corresponding rate-distortion pairs;

obtained using the estimated 3-D motion parameters, previous frame and the encoded depth field, is shown for $\lambda = 5$. The TU areas are segmented by using Equation 2.1 (\mathbf{D} is replaced with \mathbf{D}_{2D}) and the SNR_{peak} of the overall image is around 33 dB except these TU regions.

Further simulations are achieved for a multi-object scene, which is shown in Figure 2.3. After the necessary 3-D motion estimation step, Equation 4.15 is minimized for various values of λ . The distortion values for each object are tabulated in Table 4.7. On the same table, the experimental bit-rate values are also shown after lossless encoding

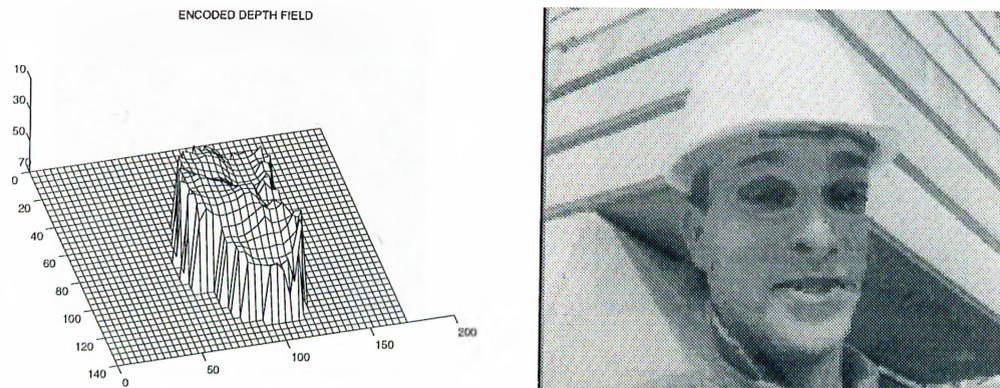


Figure 4.16: For the segmented head, (a) Encoded depth field and (b) reconstructed frame using the encoded depth field and motion parameters, for $\lambda = 5$

of the prediction error of the estimated depth fields.

Table 4.7: The experimental results for “Salesman” sequence. For each object and different values of λ , Equation 4.15 is minimized to obtain the corresponding Δ and bit-rate values.

λ	Object 1		Object 2		Object 3		Object 4		Object 5	
	Δ	Bits	Δ	Bits	Δ	Bits	Δ	Bits	Δ	Bits
1	70.6	5432	118.8	3904	221.8	6592	23.3	2536	2316.2	1240
10	155.4	2656	191.1	3601	227.6	5696	26.8	2320	2317.5	1200
50	176.4	1472	199.2	3248	258.3	4704	34.4	2201	2317.3	1144
100	177.2	1402	203.4	3224	281.6	4344	49.2	2152	2318.5	1136
1000	184.3	1304	836.2	2512	442.9	2608	216.5	1848	2318.2	1168
10000	201.2	1264	1524.9	1272	590.6	1288	981.8	1296	2319.0	1160

In Figure 4.17, the reconstructed current frame is shown for $\lambda = 100$. The TU areas are segmented by using Equation 2.1 and the visual quality of the reconstructed frame is acceptable. A significant part of object 5 is successfully segmented as TU. As expected, the projections of the 3-D motions are meaningful for the rigid objects 1, 3 and 4. The obtained depth values for the objects are also represented in the same figure for the same value of λ .

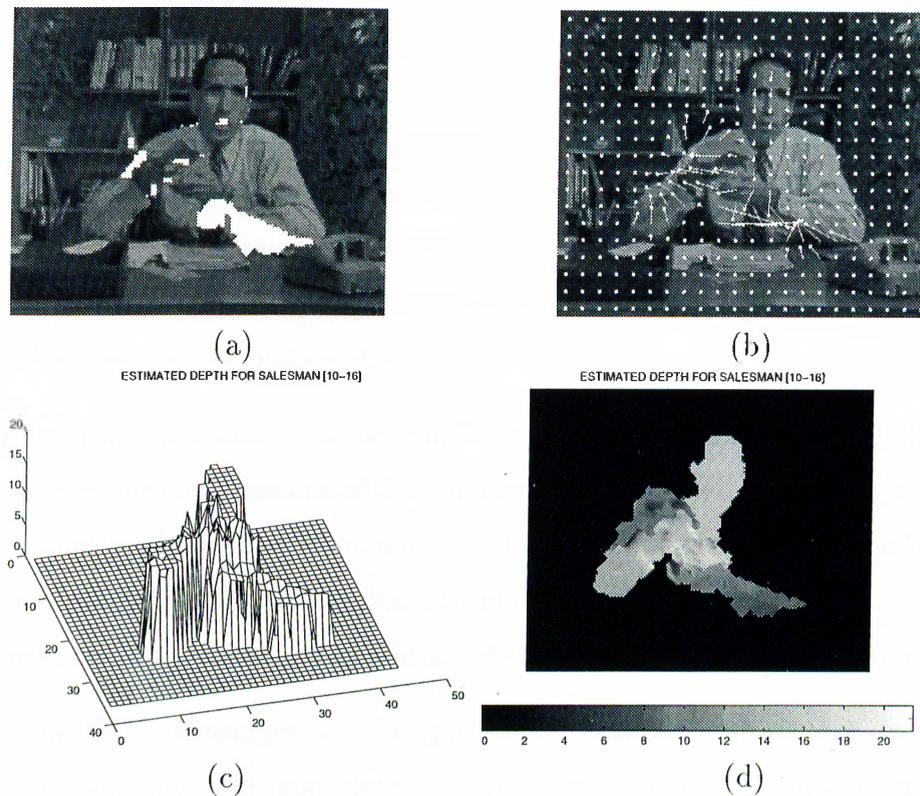


Figure 4.17: The results of 3-D motion and depth estimation for *Salesman* sequence; (a) Motion compensated current frame using 3-D motion parameters and encoded depth field (TU areas are segmented) (b) Needlegram of 2-D projection of 3-D motion; Encoded depth field with (c) mesh and (d) intensity representations.

4.3 Discussion on Depth Estimation and Encoding

In this chapter, two different methods to obtain a depth field are examined. One of the methods gives the depth field simply as the *MAP* estimate. The second method gives a depth field which is suitable for encoding. The similarity between Equations 4.3 and 4.15 is noteworthy. Since the same a priori probability density for the depth field is used in both formulations, this is an expected result. Obviously, if the distortion function in Equation 4.7 or the Gaussian noise between intensities in Equation 4.5 is replaced with different counterparts, the similarity will be diminished. Finally, it can be stated that these approaches yield the “best” dense fields with respect to noise immunity and optimal encoding with a similar formulation.

The proposed two algorithms not only find depth values in an optimum way, but they also obtain a dense depth field which is necessary for motion compensating intensities at each point. It should be noted that the number of locations whose depth values can be determined by the E-matrix method is limited by the number of trustable 2-D correspondences between frames. Hence the depth values usually can not be determined at each location by using the conventional methods. Finding a dense depth field is an important advantage of the proposed methods.

The proposed depth encoding algorithm finds and encodes a dense depth field for any two consecutive frames and associated 3-D motion parameter set, but during experiments it is observed that better compression and quality are obtained whenever the 3-D motion parameter set represents an acceptable motion between the two frames. Hence 3-D motion estimation is a critical factor which determines the overall performance.

Apart from the advantages of the proposed methods, which are explained in the previous paragraphs, the complexity of the algorithms is an important point to examine. For both robust depth estimation and efficient depth encoding procedures, the computational complexity of the overall procedures are significant due to nonlinear minimization. However, compared to the well-known MRF-based 2-D motion estimation algorithms [15], the complexity is lower by a ratio of $N \times N$ to N , where N is the number of quantized levels of the search space for each unknown. Therefore, the computational complexity is not prohibitive.

Before concluding the discussion on depth analysis, it should be noted that the required number of bits to encode a depth field is still high for very low bit-rate applications according to the simulation results. However, the encoded depth field belongs to a rigid object and the temporal redundancy in this field is ultimately high. Therefore, the real benefits will be achieved when longer sequences with more than two frames are encoded. It can be concluded that the efficient encoding of the depth fields puts 3-D motion models as alternatives to the current motion models in object-based video coding algorithms.

Chapter 5

Utilization of 3-D Motion for Occlusions and Temporal Interpolation

During the last decade, 3-D motion models have found applications in video coding to predict intensities between frames. By the help of 3-D motion, the temporal redundancy between frames are usually reduced either using some pre-defined wireframes [113] or direct usage of 3-D motion and structure information [83, 132]. However, in addition to motion compensated prediction, 3-D motion models can be further utilized in different areas associated with video coding, like motion compensated temporal interpolation and detection of occlusion areas. Both of these methods will be examined in the next sections.

5.1 Detection of Occlusion Areas using 3-D Motion Models

Occlusions occur as a result of openings and closings of some regions due to the movements of objects between consecutive frames. While such an occluding region can

be observed on one frame, it is not visible on the other one. Accordingly, such regions are defined as either *covered* or *uncovered*. If a region of an object is covered or uncovered due to its own motion (e.g. especially during rotation of the object around an axis passing through itself), such a case is called *self-occlusion*. Occlusions can also be defined as the temporal non-stationarities in video sequences. Hence, they can be called as *temporally unpredictable* regions, since temporal prediction is not possible in these areas.

Occlusion detection is an important issue in video compression, especially in object-based approaches. Since such covered/uncovered areas are temporally unpredictable from previous frames, they should be detected and encoded without temporal information. In DCT-based video coders, motion compensated prediction of such regions gives high prediction errors which cause the encoded DCT coefficients to require more bits. Moreover, in order to achieve successful temporal interpolation in every such decoder, occlusions should be detected and interpolation must be achieved appropriately at these blocks taking into account covered and uncovered regions. In addition, in object-based video coders, occlusion detection is an important issue for making correct segmentations. Since dense motion vector fields, which are usually required in object-based schemes, contain some outliers due to occlusions, such motion vector groups might lead to wrong classifications. In such cases, the correct moving object boundary, which might be on the border of the occluding region and the moving object, might degrade considerably. Hence, in any video coding application, occlusion detection is necessary.

The only way to detect occlusion regions is to examine the intensity prediction error between frames. In the literature, 2-D motion models are usually utilized to predict intensities and consequently to detect occlusions [43, 44, 81]. Among all, MRF-based methods give better results, since they model not only the intensity mismatches, but also the clustering property of occlusions, i.e. it is more probable to observe occluding points forming clusters, rather than single points. However, even in a true occluding region, the prediction error can be small if a 2-D motion vector, which achieves an incorrect intensity match, is found by only using local (motion and intensity) information. Since in most of the 2-D motion estimation algorithms, local neighborhood is used during the

estimation process, such a situation is possible, especially when the occluding regions have similar textural properties with their neighboring regions. Hence, a possible solution for this problem is to utilize global 3-D motion information, which might not permit such incorrect matches in a local neighborhood.

5.1.1 Improved Detection of Occlusion using 3-D Motion

In 3-D motion models, given 3-D motion parameters, the projection of the object motion onto 2-D image plane is controlled by a constraint, called the *epipolar constraint* [94]. This equation linearly relates the 3-D motion parameters with the positions of the object point on the 2-D image plane [86]. Using Equation 3.9, this linear relation between the current, $\mathbf{x}_p(t)$, and previous, $\mathbf{x}_p(t-1)$, 2-D coordinates of an object point \mathbf{p} can be written as

$$y_p(t-1) = a x_p(t-1) + b x_p(t) + c y_p(t) + d \quad (5.1)$$

where the parameters (a, b, c, d) are functions of the elements of matrix E in Equation 3.9 and they are known, if 3-D motion parameters are given. The linear equation above is called the *epipolar line* [94]. The relations above are illustrated in Figure 5.1.

Hence, given 3-D motion parameters and current position of the point, previous position must be on a straight line whose orientation is determined by the epipolar constraint. In order to find the previous exact position of the point on the line, the depth information of the corresponding point must be used. Hence, if the 3-D motion parameters of an object are found beforehand, the previous coordinate of the object point is tightly constrained and has only one degree of freedom, which is the depth value at that point. On the other hand, in 2-D motion models, there are two degrees of freedom, which are simply the horizontal and vertical components of the motion vector. Hence in the 2-D case, it is more probable to make erroneous intensity matches which not only violate the true motion, but also lead to incorrect detection of occlusions.

If 3-D motion parameters are given, the detection of the occluding regions should

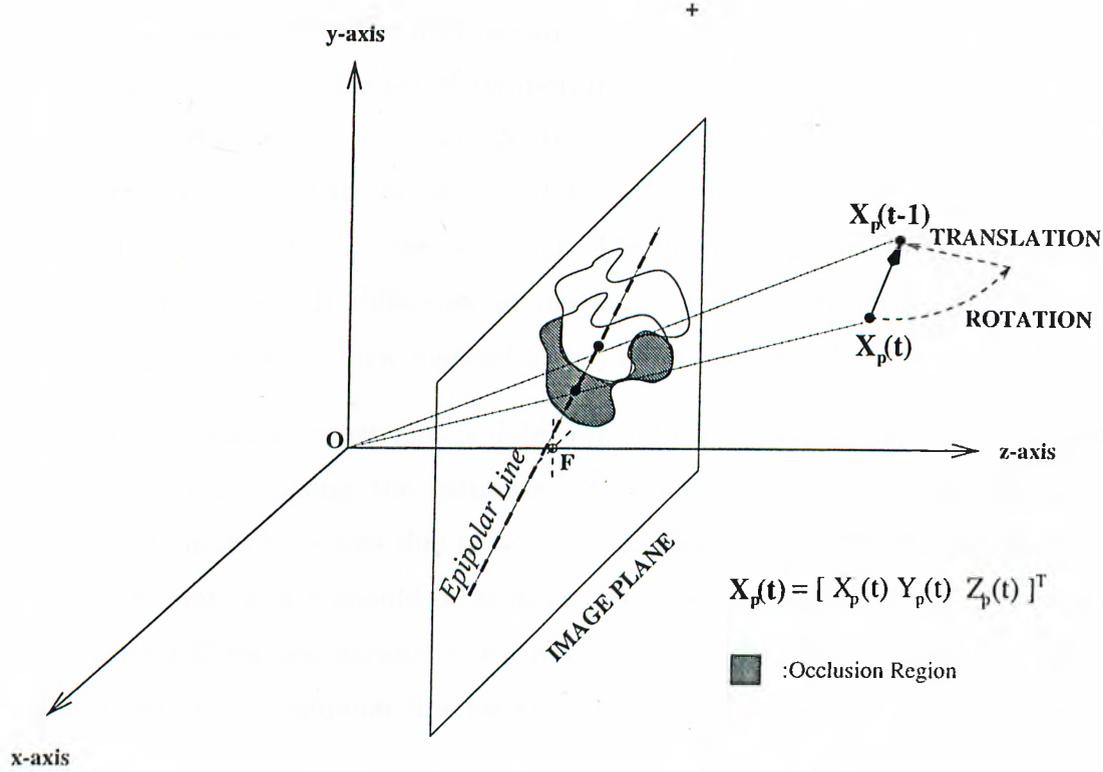


Figure 5.1: The epipolar constraint.

be achieved jointly with depth estimation. For such joint estimation problems, Gibbs formulation gives good results [132]. Given two frames, $\mathcal{I}_{t,t-1}$, the segmentation field, \mathcal{R} and 3-D motion parameters, \mathcal{M} , joint estimation of the depth, \mathcal{Z} , and occlusion, \mathcal{S} fields can be formulated by using the Gibbs energy function, \mathcal{U} , as follows :

$$\mathcal{U}(\mathcal{Z}, \mathcal{S} | \mathcal{I}_t, \mathcal{I}_{t-1}, \mathcal{R}, \mathcal{M}) = \mathcal{U}_n + \lambda_Z \mathcal{U}_Z + \lambda_s \mathcal{U}_s \quad (5.2)$$

$$\begin{aligned} \mathcal{U}_n &= \sum_{\mathbf{x} \in R_i} (I_t(\mathbf{x}) - I_{t-1}(\mathbf{x} - \mathbf{D}_{2D}(\mathbf{x}, Z)))^2 (1 - S(\mathbf{x})) + S(\mathbf{x}) T_s \\ \mathcal{U}_Z &= \sum_{\mathbf{x} \in R_i} \sum_{\mathbf{x}_c \in \eta_{\mathbf{x}}} (Z(\mathbf{x}) - Z(\mathbf{x}_c))^2 \\ \mathcal{U}_s &= \sum_{\mathbf{x} \in R_i} \sum_{\mathbf{x}_c \in \eta_{\mathbf{x}}} [1 - \delta(S(\mathbf{x}) - S(\mathbf{x}_c))] \end{aligned}$$

In the equation above, \mathbf{D}_{2D} is the projection of 3-D motion onto the image plane and it depends on the depth value, Z , at that point. The equation is valid for object i , which is previously segmented into the region R_i and \mathbf{x}_c denotes a neighbor of \mathbf{x} . It is probable

that region R_i contains both the moving object and the occlusion region near this object. Hence, the aim is finding this set of temporally unpredictable points in object i . After minimizing \mathcal{U} with respect to \mathcal{Z} and \mathcal{S} , the obtained occluding points not only have prediction errors higher than the threshold T_s (supported by the term \mathcal{U}_n), but also try to form regions (supported by the term \mathcal{U}_s). The detection of occlusions are similar in Equation 2.1 with the only difference at motion models. In Equation 2.1, the motion models is 2-D, whereas this new method uses 3-D motion model.

The global (epipolar) constraint is determined by 3-D motion parameters and these parameters are found using the estimated 2-D motion vectors which contain some untrustable 2-D motion vectors due to occlusions. Hence, it should be emphasized that 3-D motion estimation step should be as robust as possible, so that it can eliminate these outliers during 3-D motion parameter estimation. Otherwise, these outliers will disturb the orientations of each epipolar line for every object point.

5.1.2 Simulations

The experiments on occlusion detection is tested on different frame pairs, such as *Salecube* (Figure 4.3), *Mother and Daughter* (Figure 2.7) and *Salesman* (Figure 2.3). The 3-D motion parameter estimation is achieved using the proposed method in the previous Chapter 3. After minimizing Equations 2.1 and 5.2, the results in Figures 5.2, 5.3 and 5.4 are obtained in order to compare 2-D and 3-D motion based occlusion detections, respectively. From three figures, it can be clearly observed that 3-D motion based occlusion detection achieved better results for finding the correct regions for uncovered areas in the current frame.

5.1.3 Discussion

The simulation results show that the proposed method has better performance compared to conventional 2-D motion model based occlusion detectors. There are some reasons for

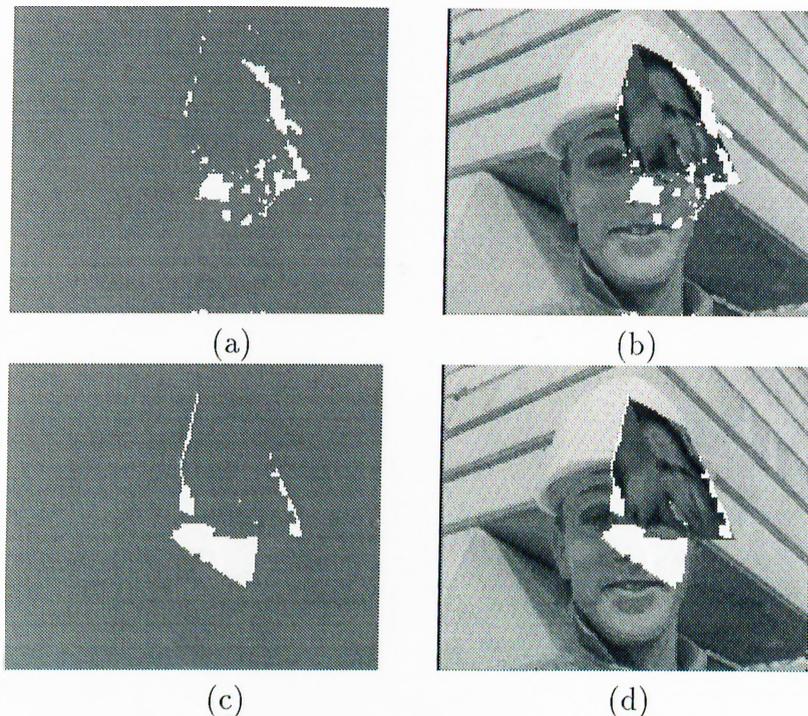


Figure 5.2: The occlusion regions for the second frame of the *Salecube* sequence. The results are obtained for 2D and 3-D motion models; (a) Temporally Unpredictable regions using 2-D motion, (b) Reconstructed frame by the help of 2-D motion, (c) Temporally Unpredictable regions using 3-D motion, (d) Reconstructed frame by the help of 3-D motion and structure.

this superiority. The proposed algorithm uses not only global constraints (the epipolar constraint), but also neighboring (local) relations (\mathcal{U}_s term in Equation 5.2), which force occluding points to form regions. The joint estimation of depth and occlusions by Gibbs formulation also improves the overall performance compared to sequential estimation of both. The acceptable performance of the proposed occlusion detection scheme is also a direct result of the robust 3-D motion estimation algorithm, based on RANSAC. If the E-matrix method is applied directly without RANSAC, the 2-D motion vectors of the occluding regions should lead to erroneous 3-D motion parameter estimates and consequently wrong epipolar constraints. In such a case, some parts of a moving rigid body, might be obtained as occluded.

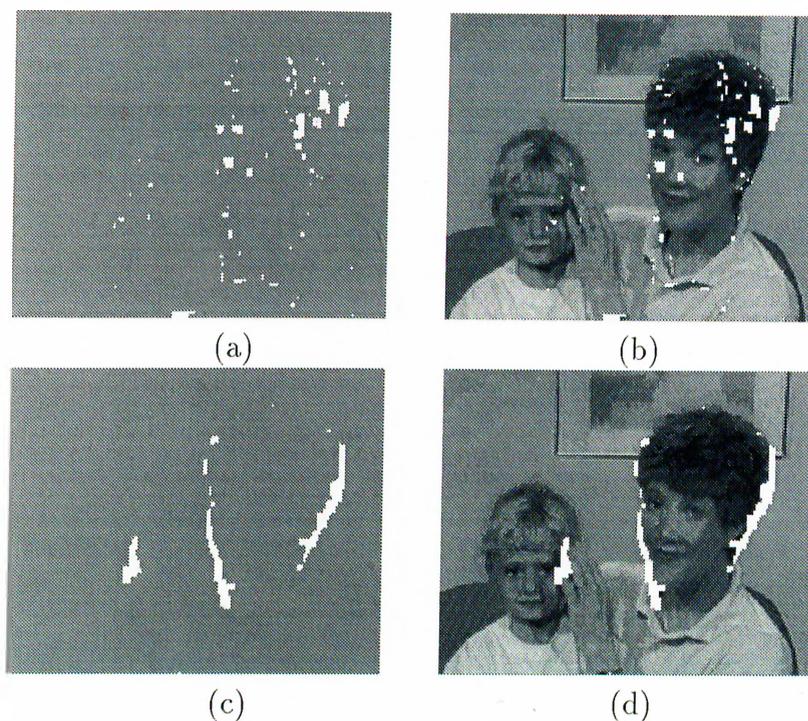


Figure 5.3: The occlusion regions for the 41th frame of the *Mother and Daughter* sequence. The results are obtained for 2D and 3-D motion models; (a) Temporally Unpredictable regions using 2-D motion, (b) Reconstructed frame by the help of 2-D motion, (c) Temporally Unpredictable regions using 3-D motion, (d) Reconstructed frame by the help of 3-D motion and structure.

5.2 Motion Compensated Temporal Interpolation

In video signal analysis, Motion Compensated (MC) processing is necessary for three major areas : predictive coding, noise reduction and sampling structure conversion [45]. While MC predictive coding is the cornerstone of all current video coding standards, noise in video signals can also be removed better using low-pass filtering along motion trajectories. On the other hand, sampling structure conversion is related with spatio-temporal interpolation. Some applications of this area are frame-rate increase, interlace to progressive conversion and general standards conversion (e.g. PAL \leftrightarrow SECAM, NTSC \leftrightarrow HDTV).

In some video coding applications, it is necessary to decrease the frame rate of the

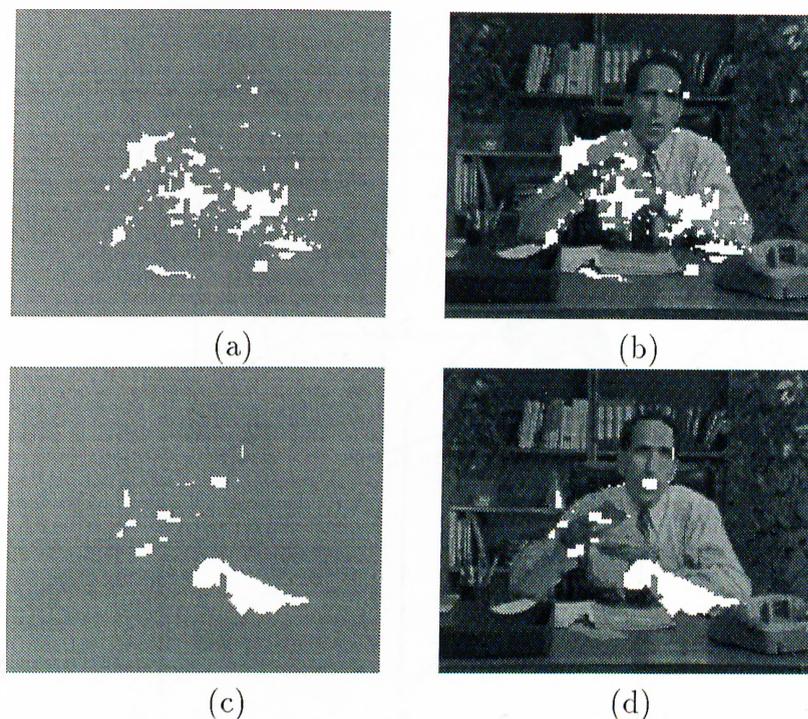


Figure 5.4: The occlusion regions for the 16th frame of the *Salesman* sequence. The results are obtained for 2D and 3-D motion models; (a) Temporally Unpredictable regions using 2-D motion, (b) Reconstructed frame by the help of 2-D motion, (c) Temporally Unpredictable regions using 3-D motion, (d) Reconstructed frame by the help of 3-D motion and structure.

original sequence in order to save bits before transmission. However, the original number of frames must be obtained in the decoder from the reconstructed frame sequence which has a lower frame-rate. The straightforward method is either the repetition of the available frames or the interpolation of the frames linearly in the temporal domain to fill the unavailable frames. However, for high motion areas, while the former solution creates the effect of “jerkiness”, the latter leads to “blurring” around motion edges. A possible remedy for frame increasing problem is to achieve interpolation along motion trajectory rather than the temporal axis and this approach is called MC temporal interpolation.

As stated in the previous chapter, occlusion detection is an important issue in temporal interpolation. The intensities in the covered or uncovered regions of an object, should be simply transferred from the previous or current frames appropriately rather than making some interpolation between intensities. This situation is illustrated in

Figure 5.5.

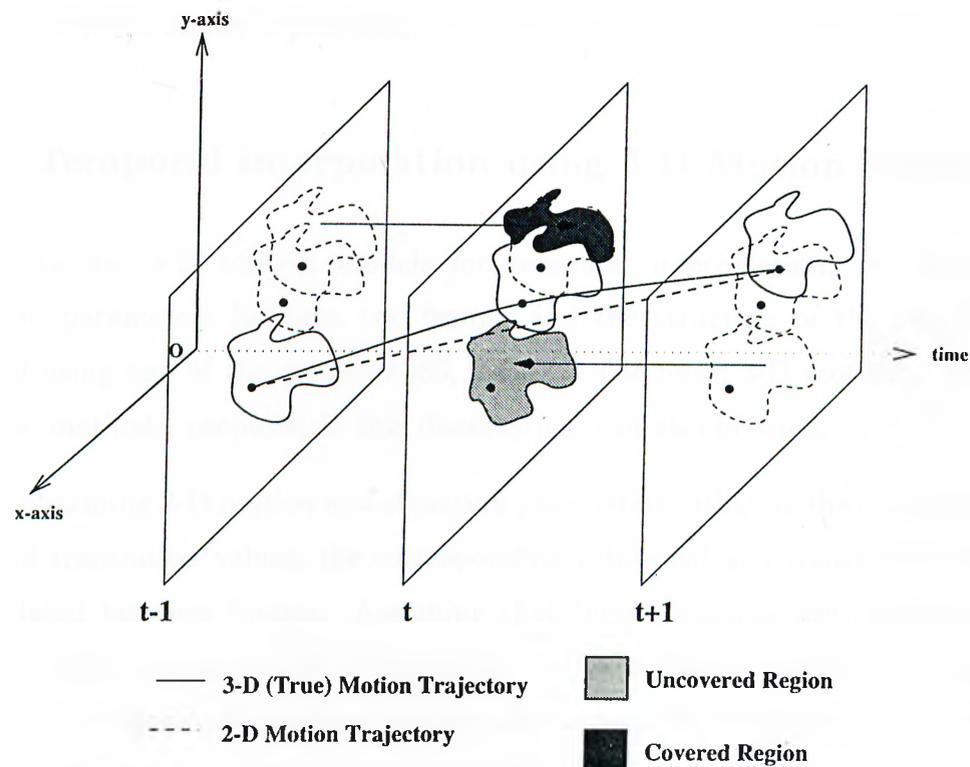


Figure 5.5: Motion compensated temporal interpolation with the corresponding motion trajectories of 2-D and 3-D models and occlusion areas.

Since MC processing is a well-known subject, there are many approaches to solve this problem. The popular Bayesian [133, 43, 45] or block-based [134, 2] motion compensated temporal interpolation methods both use 2-D motion model which lack the non-linear motion trajectory modeling between frames. Such methods can only obtain linear motion trajectories. In Figure 5.5, the modeling of motion trajectories using 2-D and 3-D models is illustrated.

Similar to occlusion detection, 3-D motion models have possible advantages over the conventional 2-D motion models for temporal interpolation between frames. Except for the assumption of rigidity, 3-D motion models do not have any other assumption which will contradict with the true motion trajectory in the scene. In [134], it is shown that during MC temporal interpolation, the estimated global camera motion has better

performance compared to conventional block-based methods. In the next section, a MC temporal interpolation, which uses 3-D motion models for not only global camera but also objects in the scene, is proposed.

5.2.1 Temporal Interpolation using 3-D Motion Models

In order to use 3-D motion models for temporal interpolation, the rotation and translation parameters between two frames and the structure of the rigid body are estimated using one of the methods [86, 2, 132]. The novel 3-D motion and structure estimation methods, proposed in this dissertation, can also be used.

After obtaining 3-D motion and structure parameters, utilizing the obtained rotation angles and translation values, the corresponding rotational and translational velocities are calculated between frames. Assuming that these velocities are constant between frames, for each frame to be interpolated, an associated rotation and translation parameter set is obtained from the velocity information. For example, assuming that the rotation angle around x - axis between available current and previous frames at time $t + \Delta$ and $t - \Delta$, respectively, is Θ_x . If the number of frames to interpolate in between is equal to N , the corresponding rotation angle will be equal to

$$\Theta_x^n = n \cdot \frac{\Theta_x}{N + 1} \quad (5.3)$$

where n takes values between 1 to N . After achieving similar formulations for the rotation angles around y and z - axes, the rotation matrix, which will be used to map the intensities of the moving object from the current frame to the n th interpolated frame, can be written using Equation 3.3 as

$$\mathbf{R}^n = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos(\Theta_x^n) & \sin(\Theta_x^n) \\ 0 & -\sin(\Theta_x^n) & \cos(\Theta_x^n) \end{bmatrix} \cdot \begin{bmatrix} \cos(\Theta_y^n) & 0 & -\sin(\Theta_y^n) \\ 0 & 1 & 0 \\ \sin(\Theta_y^n) & 0 & \cos(\Theta_y^n) \end{bmatrix} \cdot \begin{bmatrix} \cos(\Theta_z^n) & \sin(\Theta_z^n) & 0 \\ -\sin(\Theta_z^n) & \cos(\Theta_z^n) & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (5.4)$$

Similar to above formulation, corresponding translational values are equal to

$$\mathbf{T}^n = \begin{bmatrix} n \cdot \frac{T_x}{N+1} \\ n \cdot \frac{T_y}{N+1} \\ n \cdot \frac{T_z}{N+1} \end{bmatrix} \quad (5.5)$$

where T_x , T_y and T_z are the translational values between the available current and previous frames for (x, y, z) axes, respectively. After finding the corresponding 3-D motion parameters, \mathbf{R}^n and \mathbf{T}^n , for each frame to be interpolated, by the help of the estimated depth information, the frames in between are reconstructed by compensating intensities through the trajectories determined by 3-D motion and structure parameters. At this point, it should be noted that, if a point of an object occludes another point of the same object after motion, the observability of these points are decided according to their depth values; i.e., the intensity of the point, which has a depth value closer to the image plane, is used while reconstructing the frame.

The method explained above is a one-way process and the intensities of only one frame is used (carried) to find the corresponding pixels at the interpolated frame. However, temporal interpolation using motion data can be also achieved in a *bi-directional* fashion [43], which means that for 3-D models, motion, depth and occlusion regions are found between two available frames in both directions and used to interpolate the missing frame together, taking into account covered and uncovered regions appropriately (Figure 5.5). In such bi-directional algorithms, for better performance, the estimated motion and occlusion fields are usually defined on the frame to be interpolated. However, for 3-D motion models, it is difficult to devise a new algorithm which will find the 3-D motion and structure field on the interpolated frame using the other two available frames.

Since the compensated pixels, which are moved along the motion trajectory, pass through the interpolated frame at non-grid points, some kind of spatial (e.g. bilinear) interpolation should be applied in order to find the intensities at grid locations. Moreover, the problems due to occlusions should also be handled appropriately.

5.2.2 Simulations

The simulations are conducted on an artificial sequence. It is assumed that 3-D motion, depth and segmentation are known a priori. Consequently, the utilized motion information (both 2-D and 3-D) in this simulation, are ideal. The first 3 frames are shown in Figure 5.6,

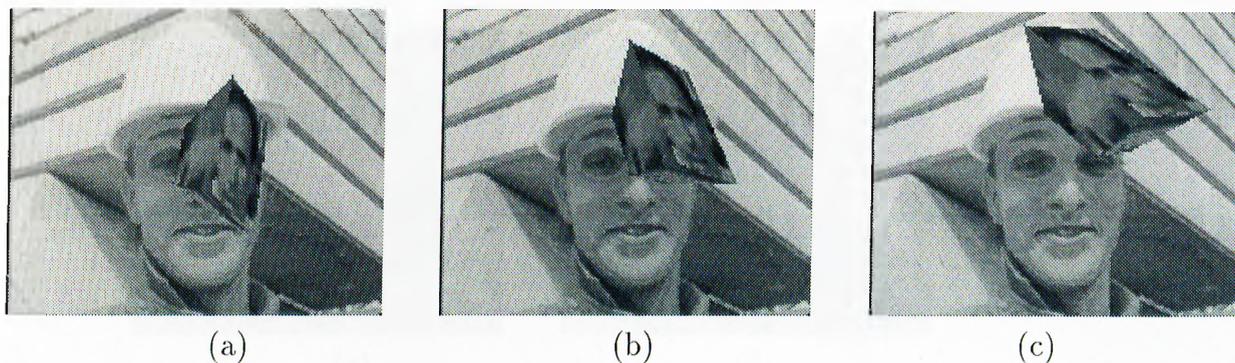


Figure 5.6: The original first 3 frames of *Salecube* sequence. (a) First, (b) second and (c) third frame.

In the simulation, the second frame is reconstructed using first and third frames by the help of motion information. Figure 5.7 compares the 2-D and 3-D motion-based temporal interpolations of the second frame.

The difference frames in Figure 5.7(a) and (c) clearly shows the performance between 2-D and 3-D motion based temporal interpolations. 2-D motion does not have the capability of modeling either rotations or translations along camera axis. The reconstructed cube for 2-D case, is smaller compared to its original size and this is due to insufficient modeling of non-linear motion trajectory of the cube. However, 3-D motion does not have these drawbacks.

5.2.3 Discussion

Although, the proposed method works for any rigid motion, the advantages of this scheme can be realized when the object motion (rotation and/or translation) is around/along

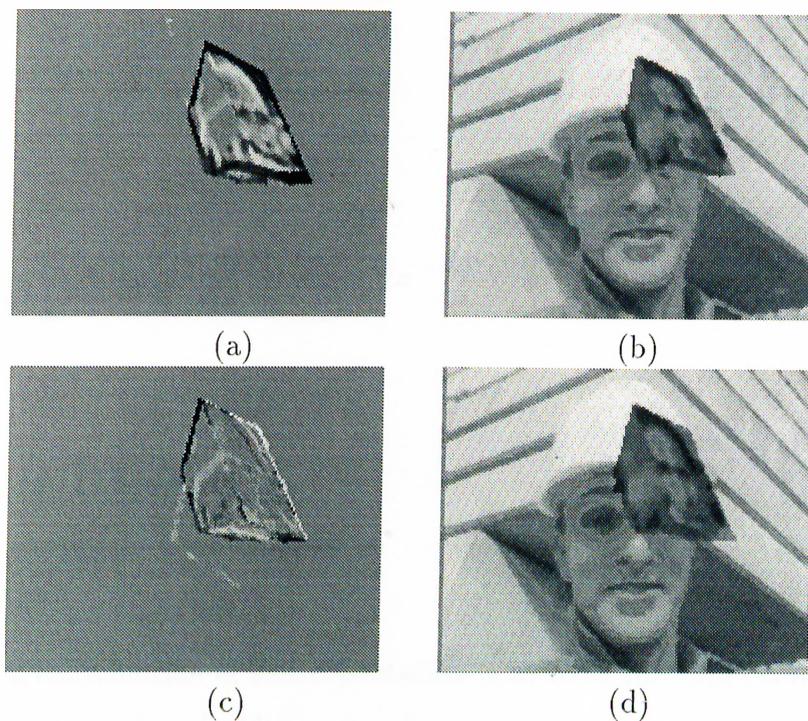


Figure 5.7: The error between the original second and the interpolated frame using (a) 2-D motion and (c) 3-D motion models. The reconstructed second frame using temporal interpolation by the help of (b) 2-D motion and (d) 3-D motion model.

Z-direction. In such cases, the motion trajectories is very non-linear and any 2-D motion model does not model these non-linearities without using any 3-D motion information.

For low resolution images, which contain small motion, the approximation of the non-linear true motion path with a 2-D motion vector might not create much degradation. Hence, it is more suitable to use the proposed MC temporal interpolation method, when the MC prediction for video compression is also achieved using 3-D motion models.

If the proposed one-way interpolated scheme can be improved to work in bi-directional, the performance of the overall algorithm will improve. However, as stated previously, such bi-directional formulations are not as simple as it is in 2-D cases, compared to 3-D motion models.

Chapter 6

Conclusions

The main contribution of this dissertation is to show the applicability of the 3-D motion models in object-based video coding by proposing some new tools for full video codecs. While object-based methods tries to find new horizons within the scope of MPEG-4, the current 2-D motion models do not promise a bright future for most of the very low bit-rate or some other novel applications, such as 3-D TV. This dissertation proposes some tools, not a full-codec, that can be utilized in the complete algorithms for the compression of video signals.

In order to construct a full-codec using these tools, some important points, as well as some future research topics, are stated in the last section of this chapter. A summary of contributions is given in the next section.

6.1 Contributions

In object-based video coding schemes, segmentation of the scene into semantically meaningful objects is compulsory. Moreover, if 3-D motions of the objects have to be estimated, the performance of this segmentation method becomes much more critical. A novel 2-D motion estimation and segmentation algorithm is developed and presented

in Section 2.4. As the simulation results in Section 2.4.2 clearly indicate, not only the segmentation results, but also the estimation of 2-D motion, which is necessary in the 3-D motion estimation step, is successful using the proposed hybrid algorithm (Section 2.4) which is a combination of deterministic and stochastic powerful segmentation algorithms.

The popular rigid 3-D motion estimation method E-matrix has given good results for high-resolution images with some sparse feature correspondences. When the application area is object-based compression, which requires motion segmentation using a dense motion field for very low bit-rates which usually requires low-resolution video input, the conventional method definitely needs some adjustments and improvements. The novel method based on RANSAC as proposed in Section 3.2.1 improves the performance of 3-D rigid motion estimation using E-matrix considerably and the simulation results (Section 3.2.2), based on quantitative criteria, show that it is possible to use noise susceptible E-matrix algorithm safely after the insertion of RANSAC-based extension.

Although, the proposed novel non-rigid motion estimation method in Section 3.3.1 deserves more interest, after it is understood that non-rigid motion description is not suitable for video compression, further research is useless beyond the scope of this dissertation. However, for applications in which modeling is more important than compression, this novel method might be a good starting point to improve deformable motion analysis methods.

The results of depth analysis might be the most interesting and important contributions of this dissertation. Noise immune estimation and optimal encoding problems for depth fields are tried to be solved by proposing two different methods in Sections 4.1.1 and 4.2.2, respectively. Eventually, these different algorithms are converged to similar formulations. The utilization of the same probabilistic models and some kind of quality measures in both algorithms should be the reason for having equivalent results. The idea behind robust depth estimation (simply *MAP* estimation) is well-known, although the formulation is novel. On the other hand, lossy depth estimation concept is first in the literature. For a long time, the researchers on computer vision community have been trying to find “true” depth fields, which are necessary in their own

applications. Hence, it is an important contribution to propose how to obtain a lossy depth field which is also suitable for coding purposes in the rate-distortion sense.

Apart from motion based intensity prediction, in video coding applications further utilization of 3-D motion models is also possible. The simulation results in Section 5.1.2 show that occlusion detection using 3-D motion models and Gibbs formulation give much better results with respect to their counterparts based on 2-D motion models. The global constraint due to the 3-D motion of an object forces occluded points not to make incorrect intensity matches that mislead occlusion analysis.

Similar to occlusion detection, 3-D motion models have advantages over the conventional 2-D counterparts for frame-rate increase using motion compensation. The advantage of utilizing 3-D motion models is a direct result of better representation of the true motion trajectory of an object in the spatio-temporal domain. The consequence of better trajectory modeling is the improved quality during the interpolation of the missing frames. If the proposed method in Section 5.2.1 can be improved to work bi-directional, the obtained simulation results might be surpassed.

Apart from the contributions stated above, in this dissertation there are numerous open questions which requires further research.

6.2 Possible Future Topics

Although, the main theme of this dissertation is on 3-D motion models, the segmentation is achieved using 2-D motion information. This is due to the fact that 2-D motion analysis is necessary to find a correspondence set for the 3-D motion estimation step. However, a final region merging step, which will be based on 3-D motion parameter sets, can be used to refine the initial segmentation in which only 2-D motion data is utilized. For an object rotating along the camera axis, while 2-D motion based segmentation might divide this object into multiple regions due to the high differences between projected 2-D motion vectors, 3-D motion parameters corresponding to these regions will almost

be equal to each other. Hence, 3-D motion based segmentation is promising concept for future research.

Since the utilized 3-D motion estimation algorithm is feature-based, it needs robust correspondences as input. The Gibbs formulated 2-D motion estimation algorithm can be further improved, so that instead of only intensity information, some motion invariant features, such as edges and corners [81], are also utilized in the motion estimation step. While such an approach increases the complexity by a small amount, the robustness of the 2-D motion vector set against noise or illumination changes is expected to increase considerably. The utilization of color for robust correspondence determination is also another open issue to examine.

As it can be easily observed from the depth encoding simulation results, the number of bits that is used to encode a depth field of a moving object in one frame is still high, even when an efficient lossy depth encoding scheme is utilized. Hence, in order to use 3-D motion models in very low bit-rate algorithms, the temporal redundancy in not only motion but also depth should be decreased. The temporal redundancy reduction in depth can be achieved in two different ways. The depth estimate of the initial frame pair can be converted into a wireframe which needs less information to transmit in consecutive frames. Afterwards, motion analysis can be achieved on the wireframe. In this way, the tight (head and shoulder) constraints on the current wireframes will be relaxed and it will be possible to apply knowledge-based video coding to any scene and any moving object. In the second approach, a “depth accumulator” should be designed, so that the structure of the rigid body, which definitely does not change, is not transmitted after every frame pair, but rather some error differences between the current depth estimate and the corresponding depth at the accumulator will be transmitted. Both of these approaches might decrease the temporal redundancy and they require more research.

As it is stated before, the proposed MC temporal interpolation should also be upgraded to bi-directional case, by the help of a novel 3-D motion and structure estimation scheme in which 3-D motion parameters and structure are defined on the frame to be interpolated.

The extension of all the proposed algorithms to multi-view systems, is also left as a future work. Obviously, depth analysis will be much more easy and robust in multi-view systems. By using a number of views of a scene, the depth field of the overall environment can be obtained very easily. In such a case, 3-D motion estimation will be achieved after the depth field is estimated. 3-D motion estimation is usually much trivial in cases where the depth field is obtained beforehand. Hence, it can be stated that the utilization of the multiple views of the same scene will be advantageous while 3-D motion models are being used.

Obviously, the most important future work after this dissertation should be the design of a full object-based video coder which uses 3-D motion models. The integration of the proposed tools with some standardized object-based coders of the near-future can also be achieved. All the methods, which are explained in the chapters of this dissertation, are possible alternatives to existing methods in such object-based full codecs.

Although, the proposed tools in this dissertation are suitable for any video coding application, the estimation of the depth information for the moving objects can also be utilized in a futuristic digital TV application. In this mono-view TV, during broadcasting, every user (viewer) selects the *view of observation* of the scene interactively and independently. By the help of 3-D motion and (accumulated) structure information, the scene will be observed from an arbitrary (user-defined) angle, rather than the recording camera angle. The same approach can also be applied on the previously recorded moving pictures or TV programs. Without extra hardware, the only way to fulfill such an achievement is to utilize 3-D motion models.

Bibliography

- [1] J.K. Aggarwal and N. Nandhakumar “On the Computation of Motion from Image Sequences-A Review,” *IEEE Proceedings*, vol. 76, pp. 917–935, August 1988.
- [2] A.M. Tekalp. *Digital Video Processing*. Prentice Hall, 1995.
- [3] A. Ikonomopoulos M. Kunt and M. Kocher “Second-generation Image Coding Techniques,” *IEEE Proceedings*, vol. 73, pp. 549–574, April 1985.
- [4] M.J. Biggar, O.J. Morris and A.G. Constantinides “Segmented-Image Coding: Performance Comparison with the Discrete Cosine Transform,” *IEE Proceedings*, vol. 135, pp. 121–132, April 1988.
- [5] H.G. Mussmann, M. Hotter and J. Ostermann “Object-oriented analysis-synthesis coding of moving images,” *Signal Processing : Image Communication*, vol. 1, pp. 117–138, October 1989.
- [6] J.O. Limb and J.A. Murphy “Measuring the Speed of Moving Objects from Television Signals,” *IEEE Trans. on Communications*, pp. 474–478, April 1975.
- [7] C. Cafforio and F. Rocca “Methods of Measuring Small Displacement of Television Images,” *IEEE Trans. on Information Theory*, vol. 22, pp. 573–579, September 1976.
- [8] B.K.P. Horn and B.G. Shunck “Determining Optical Flow,” *Artificial Intelligence*, vol. 17, pp. 185–203, 1981.

- [9] J. L. Barron, D. J. Fleet and S. S. Beauchemin “Performance of Optical Flow Techniques,” *International Journal of Computer Vision*, vol. 12, pp. 43–77, January 1994.
- [10] A.N. Netravali and J.D. Robbins “Motion-Compensated Television Coding, Part-1,” *AT & T Technical Journal*, vol. 58, pp. 629–668, March 1979.
- [11] J.R. Jain and A.K. Jain “Displacement Measurement and Its Application in Interframe Image Coding,” *IEEE Trans. on Communications*, vol. 29, pp. 1799–1808, December 1981.
- [12] A. Verri and T. Poggio “Motion Field and Optical Flow: Qualitative Properties,” *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 11, pp. 490–498, May 1989.
- [13] J. Hutchison, C. Koch, J. Luo and C. Mead “Computing Motion Using Analog and Binary Resistive Networks,” *IEEE Trans. on Computer*, vol. , pp. 52–63, March 1988.
- [14] F. Heitz and P. Bouthemy “Multimodal Motion Estimation and Segmentation using Markov Random Fields,” in *Proceedings of Int. Conf. on Pat. Recog. 90*, pp. 378–383, 1990.
- [15] J. Konrad and E. Dubois “Bayesian Estimation of Motion Vector Fields,” *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 14, pp. 910–927, September 1992.
- [16] C. Stiller and B. Hurtgen “Combined Displacement Estimation and Segmentation in Image Sequences,” in *Proceedings of Fibre Optic Networks and Video Compression*, Berlin, Germany, April 1993.
- [17] N. Diehl “Object-Oriented Motion Estimation and Segmentation in Image Sequences,” *Signal Processing : Image Communication*, vol. 3, pp. 23–56, 1991.

- [18] H. Sanson “Joint Estimation and Segmentation of Motion for Video Coding at Low Bitrates,” in *COST211ter European Workshop*, Hannover, Germany, December 1993.
- [19] J.M. Odobez and P. Bouthemy “Robust multiresolution estimation of parametric motion models in complex image sequences,” in *Proceedings of 7th EUSIPCO, Edinburgh, Scotland, September*, pp. 411–414, 1994.
- [20] M. Hotter and R. Thoma “Image Segmentation based on Object Oriented Mapping Parameter Estimation,” *Signal Processing*, vol. 15, pp. 315–334, 1988.
- [21] G. Adiv “Determining Three-Dimensional Motion and Structure from Optical Flow Generated by Several Moving Objects,” *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 7, pp. 384–402, July 1985.
- [22] D.W. Murray and B.F. Buxton “Scene Segmentation from Visual Motion Using Global Optimization,” *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 9, pp. 220–228, March 1987.
- [23] M. Chang, M.I. Sezan and A.M. Tekalp “A Bayesian Framework for Combined Motion Estimation and Scene Segmentation in Image Sequences,” in *Proceedings of IEEE ICASSP 94*, pp. 221–224, 1994.
- [24] O. J. Morris, M. J. Lee, A. G. Constantinides “Graph Theory for Image Analysis : an approach based on the Shortest Spanning Tree,” *IEE Proceedings*, vol. 133, pp. 146–152, April 1986.
- [25] M. Bierling “Displacement estimation by hierarchical blockmatching,” in *Proceedings of SPIE Visual Communications and Image Processing 88*, pp. 942–951, 1988.
- [26] R. Chellappa and A. Jain, ed. *Markov Random Fields , Theory and Application*. Academic Press, 1993.

- [27] S. Geman and D. Geman “Stochastic Relaxation, Gibbs Distributions, and the Bayesian Restoration of Images,” *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 6, pp. 721–741, November 1984.
- [28] S.T. Barnard “A Stochastic Approach to Stereo Vision,” in *Proceedings of Fifth Nat. Con. on Artif. Int.* 86, pp. 676–680, 1986.
- [29] C. Chang and S. Cahtterjee “Multiresolution Stereo - A Bayesian Approach,” in *Proceedings of Int. Conf. on Pat. Recog.* 90, pp. 908–912, 1990.
- [30] H. Derin and H. Elliot “Modelling an Segmentation of Noisy and Textured Images Using Gibbs Distribution,” *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 9, pp. 39–55, January 1987.
- [31] C.O. Acuna “Texture Modeling Using Gibbs Distributions,” *CVGIP-Image Understanding*, vol. 53, pp. 212–220, March 1991.
- [32] S. Geman, D.E. McClure and D. Geman “A Nonlinear Filter for Film Restoration and Other Problems in Image Processing,” *CVGIP-Graphical Models and Image Processing*, vol. 54, pp. 281–289, July 1992.
- [33] R.D. Morris and W.J. Fitzgerald “Replacement Noise in Image Sequences - Detection and Interpolation by Motion Field Segmentation,” in *Proceedings of IEEE ICASSP 94*, pp. 245–248, 1994.
- [34] M. Cooper. *Markov Random Fields , Theory and Application*, pp. 335–367. Academic Press, 1993.
- [35] R. Cohen. *Markov Random Fields , Theory and Application*, pp. 307–334. Academic Press, 1993.
- [36] S. Krishnamachari and R. Chellappa “An Energy Minimization Approach to Building Detection in Aerial Images,” in *Proceedings of IEEE ICASSP 94*, pp. 13–16, 1994.

- [37] D. Geman and G. Reynolds “Constrained Restoration and Recovery of Discontinuities,” *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 14, pp. 367–383, March 1992.
- [38] C. Bouman and K. Sauer “A Generalized Gaussian Image Model for Edge-Preserving MAP Estimation,” *IEEE Trans. on Image Processing*, vol. 2, pp. 296–310, July 1993.
- [39] F. Heitz and P. Bouthmey “Multimodal Estimation of Discontinuous Optical Flow Using Markov Random Fields,” *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 15, pp. 1217–1232, December 1993.
- [40] J. Konrad and E. Dubois “Estimation of Image Motion Fields: Bayesian Formulation and Stochastic Formulation,” in *Proceedings of IEEE ICASSP 88*, pp. 1072–1075, 1988.
- [41] J. Konrad and E. Dubois “Multigrid Bayesian Estimation of Image Motion Fields using Stochastic Relaxation,” in *Proceedings of IEEE Int. Conf. on Comp. Vision 1988*, pp. 354–360, 1988.
- [42] M. Chang, A.M. Tekalp and M.I. Sezan “Motion Field Segmentation using Adaptive MAP Criterion,” in *Proceedings of IEEE ICASSP 93*, pp. 33–36, 1993.
- [43] R. Depommier and E. Dubois “Motion Estimation with Detection of Occlusion Areas,” in *Proceedings of IEEE ICASSP 92*, pp. 269–272, 1992.
- [44] S. Iu “Robust Estimation of Motion Vector Fields with Discontinuity and Occlusion using Local Outliers Rejection,” in *SPIE Visual Communications and Image Processing 93*, pp. 588–599, 1993.
- [45] E. Dubois and J. Konrad “Motion Estimation and Motion-Compensated Filtering of Video Signals,” in *Proceedings of IEEE ICASSP 93*, pp. 95–98, 1993.
- [46] J. Zhang and J. Hanauer “The Mean Field Theory for Image Motion Estimation,” in *Proceedings of IEEE ICASSP 93*, pp. 197–200, 1993.

- [47] A. A. Alatan and L. Onural “Object-based 3-D motion and structure estimation,” in *Proceedings of IEEE Int. Conf. on Image Processing '95, Washington D.C., October*, pp. I 390–393, 1995.
- [48] S. Kirkpatrick, C.D. Gelatt and M.P. Vecchi “Optimization by Simulated Annealing,” *Science*, vol. 220, pp. 661–680, 1983.
- [49] P. Carnevalli, L. Coletti and S. Patarnello “Image Processing by Simulated Annealing,” *IBM Journal of Research and Development*, vol. 29, pp. 569–579, 1985.
- [50] G.L. Bilbro and W.E. Snyder “Optimization of Functions with Many Minima,” *IEEE Trans. on Systems Man and Cybernetics*, vol. 21, pp. 840–849, July/August 1991.
- [51] J. Besag “On the Statistical Analysis of Dirty Pictures,” *J.R. Statist. Soc.*, vol. 48, pp. 259–302, 1986.
- [52] P.B. Chou and C.M. Brown “The Theory and Practice of Bayesian Image Labelling,” *International Journal of Computer Vision*, vol. 4, pp. 185–220, 1990.
- [53] I.M. Abdelqader, S.A. Rajala, W.E. Snyder and G.L. Bilbro “Energy Minimization Approach to Motion Estimation,” *Signal Processing*, vol. 28, pp. 291–309, 1992.
- [54] J. Zhang “The Mean Field Theory in EM Procedures for Markov Random Fields,” *IEEE Trans. on Signal Processing*, vol. 40, pp. 2570–2583, October 1992.
- [55] A. Blake and A. Zisserman. *Visual Reconstruction*. MIT Press, 1987.
- [56] K. Rose, E. Gurewitz and G.C. Fox “Constrained Clustering as an Optimization Method,” *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 15, pp. 785–794, August 1993.
- [57] J. Konrad and E. Dubois “Comparison of Stochastic and Deterministic Solution Methods in Bayesian Estimation of 2D Motion,” *International Journal of Image and Vision Computing*, vol. 9, pp. 215–228, August 1991.

- [58] D. Terzopoulos “Image Analysis using Multigrid Relaxation Methods,” *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 8, pp. 129–139, March 1986.
- [59] W. Enkelmann “Investigations of Multigrid Algorithms for the Estimation of Optical Flow Fields in Image Sequences,” *Computer Vision Graphics and Image Processing*, vol. 43, pp. 150–177, 1988.
- [60] B. Gidas “A Renormalization Group Approach to Image Processing Problems,” *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 11, pp. 164–180, February 1989.
- [61] J.M. Laferte, P. Perez and F. Heitz “Global Non-linear Multigrid Optimization for Image Analysis Tasks,” in *Proceedings of IEEE ICASSP 94*, pp. 533–536, 1994.
- [62] Z. Kato, J. Zerubia and M. Berthod “Satellite Image Classification using Modified Metropolis Dynamics,” in *Proceedings of IEEE ICASSP 92*, pp. –, 1992.
- [63] C. Bouman and K. Sauer “Nonlinear Multigrid Methods of Optimization in Bayesian Tomographic Image Reconstruction,” in *Proceedings of SPIE Conf. on Neural and Stoc. Methods in Image and Sig. Proc.*, pp. –, 1992.
- [64] 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, January 1994.
- [65] T. S. Huang and A. N. Netravali “Motion and Structure from Feature Correspondences: A Review,” *IEEE Proceedings*, vol. 82, pp. 252–268, February 1994.
- [66] R.Y. Tsai and T.S. Huang “Uniqueness and Estimation of Three-Dimensional Motion Parameters of Rigid Objects with Curved Surfaces,” *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 6, pp. 13–27, January 1984.

- [67] X. Hu and N. Ahuja “Sufficient Conditions for Double and Unique Solution of Motion and Structure,” *CVGIP-Image Understanding*, vol. 58, pp. 161–176, September 1993.
- [68] J.W. Roach and J.K. Aggarwal “Determining the Movement of Objects from a Sequence of Images,” *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 2, pp. 554–562, November 1980.
- [69] A. Sommerfeld. *Mechanics of Deformable Bodies*. Academic Press, 1950.
- [70] R.Y. Tsai and T.S. Huang “Estimating Three-Dimensional Motion Parameters of a Rigid Planar Patch,” *IEEE Trans. on Signal Processing*, vol. 29, pp. 1147–1152, December 1981.
- [71] T. Wun, R. Chellappa and Q. Zheng “Experiments on Estimating Egomotion and Structure Parameters Using Long Monocular Sequences,” *International Journal of Computer Vision*, vol. 15, pp. 77–103, February 1995.
- [72] W.N. Martin and J.K. Aggarwal. *Motion Understanding, Robot and Human Vision*, pp. 329–352. Kluwer Academic Publishers, Boston, 1988.
- [73] K. Shoemake “Animating Rotation with Quaternions,” in *Proceedings of SIGGRAPH’85*, pp. 245–254, San Francisco, July 1985.
- [74] J. D. Foley, A. Dam, S. K. Feiner and J. F. Hughes. *Computer Graphics : Principles and Practice*. Addison Wesley, 1995.
- [75] A. Blake and A. Zisserman. *Introduction to Robotics : Mechanics and Control*. Addison Wesley, 1986.
- [76] A.N. Netravali and J. Salz “Algorithms for Estimation of Three-Dimensional Motion,” *AT & T Technical Journal*, vol. 64, pp. 335–346, 1985.
- [77] B.K.P. Horn and E.J. Weldon Jr. “Direct Methods for Recovering Motion,” *International Journal of Computer Vision*, vol. 2, pp. 51–76, 1988.

- [78] S. Peleg and H. Rom “Motion Based Segmentation,” in *Proceedings of Int. Conf. on Pat. Recog. 90*, pp. 109–111, 1990.
- [79] R. J. Holt and A. N. Netravali “Motion from Optic Flow : Multiplicity of Solutions,” *Journal of Visual Communication and Image Representation*, vol. 4, pp. 14–24, March 1993.
- [80] B. K. P. Horn. *Robot Vision*, pp. 401–417. MIT Press, Cambridge, 1986.
- [81] J. Weng, N. Ahuja and T. S. Huang “Matching Two Perspective Views,” *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 14, pp. 806–825, August 1992.
- [82] J. Weng and T.S. Huang “Estimating Motion and Structure from Line Matches: Performance Obtained and Beyond,” in *Proceedings of Int. Conf on Pat. Recog. 90*, pp. 168–172, 1990.
- [83] A. Zakhor and F. Lari “Edge-Based 3-D Camera Motion Estimation with Applications to Video Coding,” *IEEE Trans. on Image Processing*, vol. 2, pp. 481–498, October 1993.
- [84] S. Negahdaripour “Multiple Interpretations of the Shape and Motion of Objects from Two Perspective Images,” *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 12, pp. 1025–1039, November 1990.
- [85] C.P. Jerian and R. Jain “Structure from Motion - A Critical Analysis of Methods,” *IEEE Trans. on Systems Man and Cybernetics*, vol. 21, pp. 572–, May/June 1991.
- [86] J. Weng, N. Ahuja and T.S. Huang “Optimal Motion and Structure Estimation,” *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 15, pp. 864–884, September 1993.
- [87] A. Kara, D. M. Wilkes and K. Kawamura “3-D Structure Reconstruction from Point Correspondences between Two Perspective Projections,” *CVGIP-Image Understanding*, vol. 60, pp. 392–397, November 1994.

- [88] E. Steinbach and B. Girod “Estimation of Rigid Body Motion and Scene Structure from Image Sequences using a Novel Epipolar Transform,” in *Proceedings of IEEE ICASSP 96, Atlanta, May 6-10*, pp. IV 1991–1994, 1996.
- [89] T.S. Huang. *Determining Three-Dimensional Motion and Structure from Two Perspective Views*, pp. 333–354. Academic Press, 1986.
- [90] S. D. Blostein and T. S. Huang. *Motion Understanding, Robot and Human Vision*, pp. 329–352. Kluwer Academic Publishers, Boston, 1988.
- [91] D.W. Murray and B.F. Buxton. *Experiments in the Machine Interpretation of Visual Motion*. MIT Press, 1990.
- [92] S. Maybank. *Theory of Reconstruction from Image Motion*. Springer-Verlag, 1993.
- [93] J. Weng, T.S. Huang and N. Ahuja. *Motion and Structure from Image Sequences*. Springer-Verlag, 1993.
- [94] B. Girod “Image Sequence Coding Using 3-D Scene Models,” in *Proceedings of Visual Communications and Image Processing*, volume 3, pp. 1576–1591, Chicago, IL, October 1994.
- [95] H. Morikawa and H. Harashima “3D Structure Extraction Coding of Image Sequences,” *Journal of Visual Communication and Image Representation*, vol. 2, pp. 332–344, December 1991.
- [96] H. Li, P. Roivainen and R. Forchheimer “3-D Motion Estimation in Model-Based Facial Image Coding,” *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 15, pp. 545–555, June 1993.
- [97] 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. on Signal Processing*, vol. 30, pp. 525–534, August 1982.
- [98] T. S. Huang “Modelling, Analysis and Visualization of Nonrigid Object Motion,” in *Proceedings of Int. Conf on Pattern Recognition 90*, pp. 361–364, 1990.

- [99] S. Ullman “Maximizing Rigidity : The incremental recovery of 3D structure from rigid and nonrigid motion,” *Perception*, vol. 13, pp. 255–274, 1984.
- [100] A. Pentland and B. Horowitz “Recovery of Nonrigid Motion and Structure,” *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 13, pp. 730–742, July 1991.
- [101] A. Pentland and S. Sclaroff “Closed-form Solutions for Physically Based Shape Modelling and Recognition,” *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 13, pp. 715–729, July 1991.
- [102] D. Terzopoulos, J. Platt, A. Barr and K. Fleischer “Elastically Deformable Models,” *Computer Graphics*, vol. 21, pp. 205–214, July 1987.
- [103] M. Kass, A. Witkin and D. Terzopoulos “Snakes : Active Contour Models,” *International Journal of Computer Vision*, vol. , pp. 321–331, 1988.
- [104] D. Metaxas. *Physics-based Modelling of Nonrigid Objects for Vision and Graphics*. PhD thesis, University of Toronto, 1992.
- [105] D. Terzopoulos and D. Metaxas “Dynamic 3D models with Local and Global Deformations : Deformable Superquadrics,” *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 13, pp. 703–714, July 1991.
- [106] D. Terzopoulos and K. Waters “Analysis and Synthesis of Facial Image Sequences Using Physical and Anatomical Models,” *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 15, pp. 569–579, June 1993.
- [107] S. Han, D.B. Goldgof and K.W. Bowyer “Using Hyperquadrics for Shape Recovery from Range Data,” in *Proceedings of the 4th Int. Conf. on Computer Vision, Berlin, May 11-14*, pp. 492–496, 1993.
- [108] T. McInerney and D. Terzopoulos “A Finite Element Method for 3-D Shape Reconstruction and Nonrigid Motion Tracking,” in *Proceedings of 4th. Int. Conf. on Computer Vision, Berlin , 11-14 May*, pp. 518–523, 1993.

- [109] R. J. Holt and A. N. Netravali “Motion of Nonrigid Objects from Multiframe Correspondences,” *Journal of Visual Communication and Image Representation*, vol. 3, pp. 255–271, September 1992.
- [110] C. W. Chen , T. S. Huang and M. Arrott “Modelling, Analysis, and Visualization of Left Ventricle Shape and Motion by Hierarchical Decomposition,” *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 16, pp. 342–356, April 1994.
- [111] T. Chen, W. Lin and C. Chen “Artificial Neural Networks for 3-D Motion Analysis-Part II : Nonrigid Motion,” *IEEE Trans. on Neural Networks*, vol. 6, pp. 1394–1401, November 1995.
- [112] D.P. Huttenlocher, J.J. Noh and W.J. Rucklidge “Tracking Non-rigid Objects in Complex Scenes,” in *Proceedings of 4th. Int. Conf. on Computer Vision, Berlin , 11-14 May*, pp. 93–101, 1993.
- [113] K. Aizawa and T.S. Huang “Model-based Image Coding : Advanced Video Coding Techniques for Very Low Bit-Rate Applications,” *IEEE Proceedings*, vol. 83, pp. 259–271, February 1995.
- [114] J. Ostermann “Object-based analysis-synthesis coding based on the source model of moving rigid 3D objects,” *Signal Processing : Image Communication*, vol. 6, pp. 143–161, May 1994.
- [115] G. Bozdağı. *Three Dimensional Facial Motion and Structure Estimation in Video Coding*. PhD thesis, Bilkent University, January 1994.
- [116] G. Martinez “Automatic analysis of flexibly connected rigid 3D objects for an OBASC,” in *Proceedings of Int. Picture Coding Symposium 94*, pp. –, 1994.
- [117] H. Busch “Subdividing Non Rigid 3D Objects into Quasi Rigid Parts,” in *Proceedings of the 3rd. IEE Int. Conf. on Image Processing and it’s applications*, pp. –, 1989.

- [118] J. Philip “Estimation of Three-Dimensional Motion of Rigid Objects from Noisy Observations,” *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 13, pp. 61–66, January 1991.
- [119] M. A. Fischler and R. C. Bolles “Random Sample Consensus : A Paradigm for Model Fitting,” *Communications of ACM*, vol. 24, pp. 381–395, June 1981.
- [120] A. Azarbayejani and A. Pentland “Recursive Estimation of Motion, Structure and Focal Length,” *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 17, pp. 562–575, June 1995.
- [121] 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.
- [122] X. Hu and N. Ahuja “Motion Estimation under Orthographic Projection,” *IEEE Trans. on Robotics and Automation*, vol. 7, pp. 848–853, December 1991.
- [123] K. Kanatani “Structure and Motion from Optical Flow under Orthographic Projection,” *Computer Vision Graphics and Image Processing*, vol. 35, pp. 181–199, 1986.
- [124] M. Yamamoto “A General Aperture Problem for Direction Estimation of 3-D Motion Parameters,” *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 11, pp. 528–536, May 1989.
- [125] M. Yamamoto “A Segmentation Method Based on Motion from Image Sequence and Depth,” in *Proceedings of Int. Conf. on Pat. Recog. 90*, pp. 230–232, 1990.
- [126] A. A. Alatan and L. Onural “Gibbs Random Field Model Based 3-D Motion Estimation by Weakened Rigidity,” in *Proceedings of IEEE Int. Conf. on Image Processing '94, Austin, November*, pp. II 790–794, 1994.

- [127] R. Laganiere and A. Mitiche “Direct Bayesian Interpretation of Visual Motion,” in *IMACS Int. Symposium on Singal Processing, Robotics and Neural Networks*, pp. 140–144, 1994.
- [128] D. Tzovoras, N. Grammailidis and M. G. Strintzis “Depth Map Coding for Stereo and Multiview Image Sequence Transmission,” in *Proc. of the Inter. Workshop on Stereo and 3-D Imaging, Santorini, Greece*, pp. 75–80, 1995.
- [129] T. Cover. *Elements of Information Theory*. Wiley, 1991.
- [130] P. Chou, T. Lookabaugh and R.M. Gray “Entropy-constrained Vector Quantization,” *IEEE Trans. on Acoustics, Speech and Signal Processing*, vol. 37, pp. 31–42, January 1989.
- [131] W. Stadler. *Multicriteria Optimization in Engineering and in the Sciences*. Plenum Press, 1988.
- [132] A. A. Alatan and L. Onural “Joint Estimation and Optimum Encoding of Depth Field for 3-D Object-based Video Coding,” in *Proceedings of IEEE Int. Conf. on Image Processing ‘96, Lausanne, Switzerland, September*, pp. II 871–874, 1996.
- [133] C. Bergeron and E. Dubois “Gradient-Based Algorithms for Block-Oriented MAP Estimation of Motion and Application to Motion-Compensated Temporal Interpolation,” *IEEE Trans. on Circuits and Systems for Video Technology*, vol. 1, pp. 72–85, March 1991.
- [134] S. Tubaro and F. Rocca. *Motion Analysis and Image Sequence Processing*, chapter Motion Field Estimators and their Application to Image Interpretaton, pp. 153–187. Kluwer Academic Publishers, Boston, 1993.

Vita

A. Aydın Alatan was born in Ankara, Turkey, in 1968. He received his 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. His research interests are Image/Video Compression, Object-based Coding, Motion Analysis, 3-D Motion Models, Non-rigid Motion Analysis, Gibbs Models, Rate-Distortion Theory, Active Meshes.