

WATERMARKING
FOR 3D REPRESENTATIONS

A THESIS SUBMITTED TO
THE GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCES
OF
MIDDLE EAST TECHNICAL UNIVERSITY

BY

ALPER KOZ

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR
THE DEGREE OF DOCTOR OF PHILOSOPHY
IN
ELECTRICAL AND ELECTRONICS ENGINEERING

AUGUST 2007

Approval of the Thesis

“WATERMARKING FOR 3D REPRESENTATIONS”

Submitted by **Alper Koz** in partial fulfillment of the requirements for the degree of **Doctor of Philosophy in Electrical and Electronics Engineering** by,

Prof. Dr. Canan Özgen

Dean, Graduate School of **Natural and Applied Sciences** _____

Prof. Dr. İsmet Erkmén

Head of Department, **Electrical and Electronics Engineering** _____

Assoc. Prof. Dr. A. Aydın Alatan

Supervisor, **Electrical and Electronics Engineering, METU** _____

Examining Committee Members

Assoc. Prof. Dr. Gözde Bozdağı Akar*

Electrical and Electronics Engineering, METU _____

Assoc. Prof. Dr. A. Aydın Alatan**

Electrical and Electronics Engineering, METU _____

Prof. Dr. Uğur Halıcı

Electrical and Electronics Engineering, METU _____

Prof. Dr. Levent Onural

Electrical and Electronics Engineering, Bilkent University _____

Assist. Prof. Dr. Çağatay Candan

Electrical and Electronics Engineering, METU _____

* Head of the examining committee

** Supervisor

I hereby declare that all information in this document has been obtained and presented in accordance with academic rules and ethical conduct. I also declare that, as required by these rules and conduct, I have fully cited and referenced all material and results that are not original to this work.

Name, Last Name: **Alper Koz**

Signature:

ABSTRACT

WATERMARKING FOR 3D REPRESENTATIONS

Koz, Alper

Ph.D., Department of Electrical and Electronics Engineering

Supervisor: Assoc. Prof. Dr. A. Aydın Alatan

August 2007, 156 Pages

In this thesis, a number of novel watermarking techniques for different 3D representations are presented. A novel watermarking method is proposed for the mono-view video, which might be interpreted as the basic implicit representation of 3D scenes. The proposed method solves the common flickering problem in the existing video watermarking schemes by means of adjusting the watermark strength with respect to temporal contrast thresholds of human visual system (HVS), which define the maximum invisible distortions in the temporal direction. The experimental results indicate that the proposed method gives better results in both objective and subjective measures, compared to some recognized methods in the literature.

The watermarking techniques for the geometry and image based representations of 3D scenes, denoted as *3D watermarking*, are examined and classified into three groups, as 3D-3D, 3D-2D and 2D-2D watermarking, in which the pair of symbols identifies whether the watermark is embedded-detected in a 3D model or a 2D projection of it. A detailed literature survey on 3D-3D watermarking is presented that mainly focuses on protection of the intellectual property rights of the 3D geometrical representations. This analysis points out the specific problems in 3D-3D geometry watermarking, such as the lack of a unique 3D scene representation, standardization for the coding schemes and benchmarking tools on 3D geometry watermarking.

For 2D-2D watermarking category, the copyright problem for the emerging free-view televisions (FTV) is introduced. The proposed watermarking method for this original problem embeds watermarks into each view of the multi-view video by utilizing the spatial sensitivity of HVS. The hidden signal in a selected virtual view is detected by computing the normalized correlation between the selected view and a generated pattern, namely rendered watermark, which is obtained by applying the same rendering operations which has occurred on the selected view to the original watermark. An algorithm for the estimation of the virtual camera position and rotation is also developed based on the projective planar relations between image planes. The simulation results show the applicability of the method to the FTV systems.

Finally, the thesis also presents a novel 3D-2D watermarking method, in which a watermark is embedded into 3-D representation of the object and detected from a 2-D projection (image) of the same model. A novel solution based on projective invariants is proposed which modifies the cross ratio of the five coplanar points on the 3D model according to the watermark bit and extracts the embedded bit from the 2D projections of the model by computing the cross-ratio. After presenting the applicability of the algorithm via simulations, the future directions for this novel problem for 3D watermarking are addressed.

Keywords: Video Watermarking, Temporal Contrast Thresholds, 3D watermarking, Geometry Watermarking, Free View Watermarking, Human Visual System, Projective Invariants

ÖZ

3 BOYUT GÖSTERİMLER İÇİN GÖRÜNMEZ DAMGALAMA YÖNTEMLERİ

Koz, Alper

Doktora, Elektrik-Elektronik Mühendisliği Bölümü

Tez Yöneticisi: Doç. Dr. A. Aydın Alatan

Ağustos 2007, 156 sayfa

Bu tezde farklı 3 boyut (3B) gösterimler için yeni damgalama yöntemleri önerilmiştir. İlk olarak, 3B sahnelerin en basit için gösterimi olarak da kabul edilebilecek tek bakışlı video için yeni bir yöntem önerilmiştir. Yöntem, videolarda damgalama sonrası çokça karşılaşılan kırpışma türü bozulmaları engellemek amacıyla görünmez damgayı, damga büyüklüğünü insan görme sisteminin (İGS) zamansal kontrast eşik değerlerine göre sınırlayarak eklemektedir. Yöntemin, literatürdeki bilinen çeşitli yöntemlere göre görünmezlik, dayanıklılık ve kapasite açısından daha iyi sonuçlar verdiği gösterilmiştir.

Tezin ikinci bölümünde, geometri ve imge tabanlı 3 boyut gösterimlerin görünmez damgalanması anlamında kullanılan, 3B damgalama yöntemleri incelenmiş ve bu yöntemler damganın 3B modele veya 3B modelin 2B izdüşüm imgelerine konulmasına ve 3B modelden veya 2B imgeden çıkarılmasına göre, 3B-3B, 3B-2B ve 2B-2B damgalama şeklinde üç sınıfa ayrılmıştır. Tezde, 3B geometrik yapıların telif hakkının korunması sorunu üzerine odaklanan 3B-3B damgalama yöntemleri hakkında ayrıntılı bir literatür taraması sunulmuştur. Yapılan araştırma 3B-3B geometri damgalamaya özgün, 3B sahne gösterimlerinin tek olmaması, kıyaslama araçlarının eksikliği ve standart sıkıştırma algoritmalarının bulunmaması gibi sorunları ortaya koymuştur.

2B-2B damgalama alanında, çok bakışlı televizyonlarda üretilen sanal görüntünün korunmasına yönelik yeni bir damgalama problemi sunulmuştur. Bu özgün problem için önerilen yöntem görünmez damgayı çok bakışlı videonun herbir görüntüsüne İGS'nin konumsal duyarlılığına göre eklemektedir. Herhangi bir konum ve açı için oluşturulmuş sanal görüntüdeki damgayı belirlemek için verilen görüntü ile aynı konum ve açı için oluşturulmuş damga arasındaki ilinti hesaplanmaktadır. Ayrıca, verilen bir görüntü için sanal kameranın konumu ve açısını belirleyen bir yöntem geliştirilmiştir. Sonuçlar önerilen yöntemin çok bakışlı televizyonlara uygulanabilirliğini ortaya koymuştur.

Son olarak, 3B-2B damgalama alanında, damgayı üç boyutlu modele koyduktan sonra damgayı modelin iki boyuttaki izdüşümlerinden çıkaran yeni bir yöntem önerilmiştir. İzdüşümsel değişmezler tabanlı önerilen yöntem 3B model üzerindeki beş düzlemsel noktanın çapraz oranını verilen damga bitine göre değiştirmekte ve saklanan bilgiyi 3B modelin herhangi bir 2B izdüşümünden çapraz oranı hesaplayarak çıkarmaktadır. Damga sezici için değişik senoryolar düşünülmüş, deneyler ile uygulanabilirliği ortaya konmuş ve son olarak 3B-2B alanındaki gelecek çalışmalar ayrıntılı olarak tartışılmıştır.

Anahtar Kelimeler: Video Damgalama, İnsan Görme Sistemi, Zamansal Duyarlılık, 3B Damgalama, Geometri Damgalama, Çok Bakışlı Video Damgalama, İzdüşümsel Değişmezler

To my family.

ACKNOWLEDGEMENTS

First and foremost, I would like to express my sincere gratitude to my advisor, Assoc. Prof. A. Aydın Alatan, for his valuable supervision and support in every step taken in the preparation of this thesis. I am very pleased to study under his supervision for six years of my M.S. and Ph.D. studies. I can honestly say that his convincing, encouraging and friendly attitude has converted this long and difficult period into a more enjoyable and rewarding experience for me.

I would like to acknowledge the members of my thesis committee, Prof. Levent Onural, Prof. Uğur Halıcı, Assoc. Prof. Gözde Bozdağı Akar and Assist. Prof. Çağatay Candan for their support and suggestions which improved the quality of the thesis. I would also like to thank Prof. Levent Onural and Assist. Prof. Çağatay Candan once more for their valuable feedbacks in my thesis progress presentations during the last three years.

I would also like to gratefully acknowledge the researchers who directly contributed to this thesis. I would like to thank Dr. George Triantafyllidis of Informatics and Telematics Institute, Greece, for hosting me in his institute in Spring 2006 and for his collaboration and discussions on 3D-3D geometry watermarking. This visit constructed the background of the third chapter of this thesis. I wish to thank my friend Cevahir Çıgla for his collaboration on free view watermarking which constituted the fourth chapter of this thesis. I would like to sincerely thank Prof. Ingemar Cox of University College London (UCL), England for hosting me in his laboratory in Fall 2006 and introducing me the idea of 3D-2D projective invariant watermarking for 3D objects. I would also like to acknowledge Dr. Gwenael Doerr of UCL for his discussions and contributions on the same topic. This visit to UCL formed the background of the fifth chapter of the thesis.

I am also grateful to my instructors at METU. Much of the background for the work in this thesis was taught me by the professors in METU. I thank all of them for dedication to teaching.

I would like to thank all of my friends in METU during my graduate years. I would like to specially thank Umut Orguner for sharing his notes and books every time I need during my graduate years, Yucel Ozbek for sharing the long nights during the preparation period of qualification examination and Ayhan Yilmaz and Eren Gürses for sharing our laboratory for three years. I would like to thank all the present and previous members of METU multimedia laboratory, Evren, Ahmet, Elif, Oytun, Serdar, Engin, Yoldaş, Neslihan, Anıl, Oğuz, Çağdaş, Murat, Emrah, Berkan, Özlem, Bertan, Erman, Özgü and others, for their friendship and collaboration. I also wish to thank Aydın Biçer, Burhan Gültekin, Turan Demirci and Eray Özçelik for their valuable friendship during my graduate and undergraduate years in METU.

This thesis is the most important fruit of my life of education. Therefore, I would also like to thank all of my teachers during my education, especially my primary school teacher, Hacer Serdar, and secondary school teacher Rıdvan Bal for showing me the correct directions in my education. They are always irreplaceable for me.

Finally and most importantly, I would like to thank my family for all of their help, support and love during all my life. This work is dedicated to them.

This thesis is partially supported by European Commission within FP6 under Grant 511568 with the acronym 3DTV and The Scientific and Technological Research Council of Turkey under International Research Fellowship Programme.

TABLE OF CONTENTS

ABSTRACT	iv
ÖZ	vi
ACKNOWLEDGEMENTS.....	ix
TABLE OF CONTENTS	xi
CHAPTER	
1. INTRODUCTION.....	1
1.1 Watermarking Applications	2
1.2 Requirements for Watermarking.....	3
1.3 Motivation.....	5
1.4 Main Contributions of the Thesis.....	9
1.5 Outline of Thesis.....	10
2. OBLIVIOUS SPATIO-TEMPORAL WATERMARKING OF MONO-VIEW VIDEO BY EXPLOITING HUMAN VISUAL SYSTEM.....	12
2.1 Overview of Video Watermarking Methods.....	13
2.1.1 Video Watermarking Methods Discarding HVS	14
2.1.2 Video Watermarking by Implicit Utilization of HVS.....	15
2.1.3 Video Watermarking Based on HVS Properties.....	16
2.2 Proposed Watermarking Method.....	17
2.2.1 Temporal Contrast Thresholds.....	18
2.2.2 Watermark Embedding	19
2.2.3 Watermark Detection.....	23
2.3 Experimental Results	24
2.3.1 Case 1: Equal Watermark Energy Insertion under Various Attacks..	28
2.3.2 Case 2: Invisibility under Various Attacks	32
2.3.3 Case 3: Equal BER during Extraction for Compression Attack	34

2.3.4 Case 4: Capacity under Various Attacks.....	35
2.4 Conclusions.....	39
3. 3D-3D GEOMETRY WATERMARKING	41
3.1 3D Watermarking: Taxonomy	41
3.2 3D-3D Geometry Watermarking.....	44
3.2.1 Requirements of 3D-3D Geometry Watermarking.....	45
3.2.2 Fundamental Problems in 3D-3D Geometry Watermarking	46
3.3 3D Geometry-Based Watermarking Methods.....	48
3.4 Spatial Domain Methods	52
3.4.1 Modifying Vertex Coordinates	52
3.4.2 Methods Based on other Geometric Primitives	58
3.4.3 Methods based on 2D (Image) Watermarking.....	61
3.5 Transform Domain Methods.....	63
3.5.1 3D Geometry Watermarking based on Spectral Analysis	64
3.5.2 3D Geometry Watermarking based on Wavelet Transformation	68
3.6 Comparisons and Discussions on Geometry Watermarking	72
3.7 Summary and Conclusions.....	73
4. 2D-2D WATERMARKING FOR FREE-VIEW TELEVISION	77
4.1 Introduction	78
4.2 Light Field Rendering.....	80
4.3 Proposed Watermarking Method.....	83
4.3.1 Watermark Embedding	83
4.3.2 Watermark Detection.....	84
4.4 Experimental Results	85
4.4.1 Imperceptibility Tests	86
4.4.2 Robustness Tests.....	87
4.5 Analysis of Transformations on Watermark Pattern during LFR.....	90
4.5.1 Proposed Solution for Case I	91
4.5.2 Proposed Solution for Case II	93

4.5.3 Proposed Solution for Case III.....	95
4.5.4 Decomposing the Rendered Image into Regions.....	101
4.6 <i>Proposed Solution for Bilinear Interpolation in LFR</i>	108
4.6.1 Determining Planar Projective Transformation between Focal Plane and Image Plane.....	108
4.6.2 Robustness Results	111
4.7 <i>Conclusions</i>	112
5. 3D-2D PROJECTIVE INVARIANT WATERMARKING FOR 3D OBJECTS....	114
5.1 <i>Previous Work on 3D-2D Watermarking</i>	114
5.2 <i>Projective Invariants</i>	116
5.3 <i>Proposed 3D-2D Watermarking Method</i>	118
5.3.1 Identification of Coplanar Five Points at the Embedder.....	120
5.3.2 Modification of the Feature Point Locations by the Embedder	125
5.3.3 Identification at the Detector	127
5.4 <i>Simulation Results</i>	131
5.5 <i>Conclusions and Future Directions</i>	133
6. SUMMARY and CONCLUSIONS.....	135
APPENDIX A.....	139
APPENDIX B.....	142
REFERENCES	144
VITA.....	154

CHAPTER 1

INTRODUCTION

As a result of the vast growth in digital multimedia technologies, the unbearable simplicity in the duplication of a digital content rapidly, perfectly and without limitations in numbers has resulted with a new problem for the content owners, that is copyright protection. *Watermarking* is proposed as a remedy to this problem [1]. A *watermark*, a secret imperceptible message, is embedded into the original data in such a way that it is detectable as long as the perceptible quality of the content stays at an acceptable level. The owner of the original data proves his/her ownership by extracting the watermark from the watermarked content, in a case of multiple ownership claims.

A general scheme for digital watermarking is given in Figure 1.1. A message is imperceptibly embedded into the original visual content at the embedder. Then, the embedded content is delivered through a channel. This channel might represent degradations, transformations and possible signal processing operations on the visual data. Next, the watermark message is tried to be detected from the available content at the detector. The detector may also utilize the original data during detection, depending on the application.

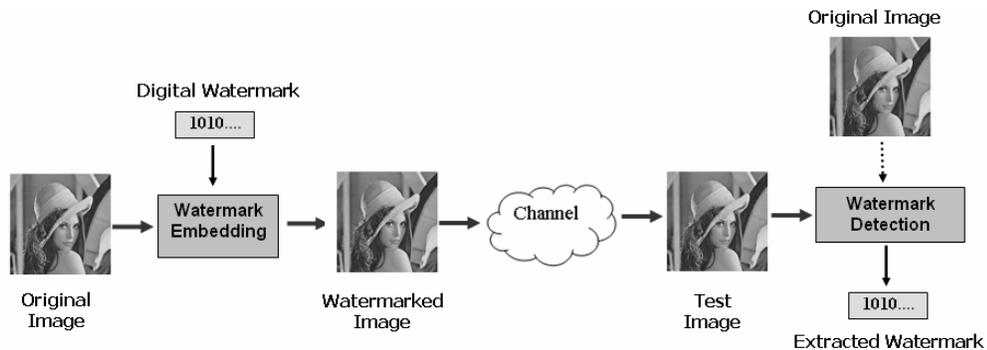


Figure 1.1 General scheme for watermarking

1.1 Watermarking Applications

Although the main motivation behind digital watermarking is copyright protection, its application area is much wider, including *broadcast monitoring*, *fingerprinting*, *authentication* and *covert communications* [2]-[5]. In broadcast monitoring, by embedding watermarks into commercial advertisements, one can monitor whether the advertisements are broadcasted at the correct instants by means of an automated system [2]. In such an application, the system receives the broadcast data and searches for watermarks identifying where and when the advertisement is broadcasted. The same process could also be used for video and audio clips. Musicians and actors would like to ensure that they receive accurate royalties for broadcasts of their performances.

Fingerprinting is another novel approach to trace the source of illegal copies [2]-[3]. The owner of the digital data may embed different watermarks into the copies of digital content that are customized for each recipient. In this manner, the owner is able to identify the customer by extracting the watermark, if the data is supplied to third parties.

Watermarking could also be utilized for *authentication* of any change that might have been occurred on the digital content [5]. If any tampering has occurred on the content, the same change will also take place on a fragile watermark embedded to the digital content, providing information about which part of the content has been altered.

Covert communication is another possible application of digital watermarking [2]-[3]. A secret message can be embedded imperceptibly as a watermark to the digital image or video to provide information from the sender to the intended receiver, while maintaining low probability of intercept by other unintended receivers.

There are also non-security related applications of digital watermarking, such as indexing of videos, movies and news items, where markers and comments can be inserted by search engines [3]. Another application is the detection and concealment of image/video transmission errors [6]. In this application, for block-based coded images, a summarizing data of every block is extracted and hidden to another block by data hiding. At the decoder side, this data is used to detect block errors.

1.2 Requirements for Watermarking

A digital watermarking process is usually evaluated based on the following criteria: perceptual transparency of the watermark, robustness against encountered attacks, computational complexity of the algorithm, bit-rate of data embedding process, false positive rate of watermark detection at the detector, recovery of data with or without access to the original signal, the speed of embedding and retrieval process, computational complexity of the embedding and detection algorithm, the ability of the embedding and retrieval module to integrate into standard encoding and decoding process [1], [3], [7].

The constraints that are used during the evaluation process might differ with respect to the application. For example, in a video indexing application, the robustness of the watermarking method against attacks is not important, since there is usually a slight possibility that the video might pass through any signal processing operation. In the covert communication application, it is better to use a blind watermarking scheme which does not require the original data during the detection process, if TV broadcasting is used as a communication channel. Moreover, if the application is copyright protection, the insertion/detection of the watermark is a time-consuming procedure. On the other hand, in a broadcast monitoring application, the speed of the watermark detection algorithm should be fast enough to follow real-time broadcasting.

When the main motivation behind digital watermarking and most of the other applications are considered, the major and common requirements to provide useful and effective watermarks might be given as *imperceptibility*, *robustness* against intended or non-intended signal operations and *capacity*.

Imperceptibility refers to the perceptual similarity between the original and watermarked data. The owner of the original data mostly does not tolerate any kind of degradations in the original data. Therefore, the original and watermarked data should be perceptually equivalent. The imperceptibility of the watermark is usually tested by means of some subjective experiments [8].

Robustness can be defined as the ability to detect the watermark, after the watermarked data has passed through a particular signal processing operation. The attacks, against which a watermarking method should be robust, depend on the application. For instance, while only the robustness against the transmission of the data in a channel is sufficient for the broadcast monitoring application, this is not the case for copyright protection application of digital watermarking. For such a case, totally unknown signal processing operations are usually applied to the watermarked data by the intruders. Hence, in copyright protection, the watermarking scheme is required to be robust to any possible signal processing operation, as long as the quality of the watermarked data preserved.

Capacity refers to the ability to verify and distinguish between different watermarks with an arbitrarily low probability of error, as the number of differently watermarked versions of an image increases [9]. In other words, it can be defined as the amount of information that can be hidden to the data without producing any perceptible distortions on the content.

There is a strong *trade-off* between these three requirements. If the capacity is increased, this might yield visible deformations in the content. If robustness is needed by means of increasing watermark strength, this might also lead to degradation of imperceptibility. On the other hand, the *capacity* requirement is inversely related with *robustness*. It is obvious that as the number of bits in the hidden information is increased, it will be more difficult to extract the hidden watermark without any bit error, after the attacks on the content.

The optimum compromise between these requirements is also dependent on the application. While the number of hidden bits, in some applications, such as broadcast monitoring, should be sufficient to differentiate all broadcasts from each other, some others, such as copyright protection, might require only one-bit of hidden information, which indicates the owner of the content. A watermarking scheme should take all of this trade-off into account to achieve the optimum solution.

1.3 Motivation

The research on digital watermarking has first initiated on still images [1]. Then, the trend has shifted towards the watermarking of video due to the variety of applications specific to moving pictures, such as copy and copyright protection of digital versatile disks (DVD) [10], broadcast monitoring [11]-[12], video authentication [13] and fingerprinting [3].

The wider area of applications for video watermarking has also created some additional difficulties in the two fundamental requirements of watermarking, namely *robustness* and *imperceptibility*. Regarding the robustness, the variety of attacks is quite diverse for video, including not only the spatial attacks that might occur on the video frames, but also the usual distortions in the temporal direction, such as temporal synchronization loss, frame rate conversions, analog-digital and digital-analog conversions during the video transmission [3]-[4]. Moreover, a video watermarking scheme should be resistant to a number of hostile attacks, such as *collusion* attack [4], [14], as well.

Imperceptibility requirement in video watermarking is another challenging problem, compared to its image counterpart, due to the additional dimension existing in video. The watermark embedding system should not only yield spatially invisible watermarks for each frame of the video, but it should also take the temporal dimension into account, in order to avoid any flickering type of distortion in time. While some of the methods [15]-[29] in the literature solve this problem by only allowing arbitrarily small modifications within frames in some transform domains, some others [30]-[36] simply use implicit spatial properties of human visual system (HVS), such as luminance masking, spatial masking and contrast masking. In addition, some approaches [37]-[41] exploit explicitly the spatial thresholds of HVS to determine the location and strength of the watermark. However, none of the former approaches has focused on guaranteeing temporal invisibility and achieving maximum watermark strength, which should yield maximum robustness along the temporal direction.

After the spread of 3D-content based applications, such as computer animation, computer gaming, virtual and augmented realities and VRML-type of internet applications, the

watermarking research has also extended to the protection of the ownership rights of the original 3D material, as well as prevent unauthorized duplication or tampering [42]-[46]. The applications requiring 3D information have first aroused in the field of computer graphics in order to create a realistic visual content from the available 3D models. Then, video games, animation films and stereoscopic displays have emerged. In the foreseeable future, it is expected to witness more sophisticated technologies, such as free-view televisions (FTV), where the user will have an opportunity to select his/her arbitrary view, while watching a real scene (Figure 1.2) and, finally, 3D holographic televisions, which are expected to generate the exact replica of a real scene via the specially designed displays in our homes (Figure 1.3). These new technologies have also increased the necessity to protect the digital 3D information for the content owners; hence, watermarking is proposed as a candidate solution.

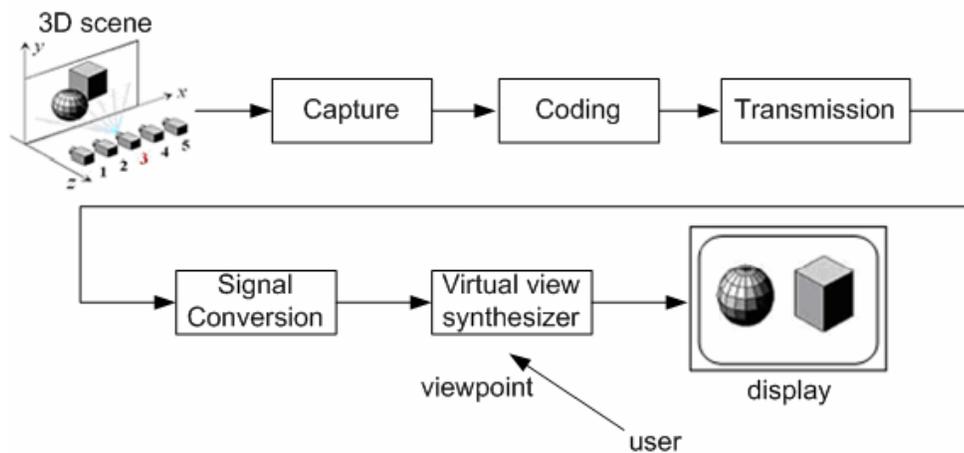


Figure 1.2 A general scheme for free view television

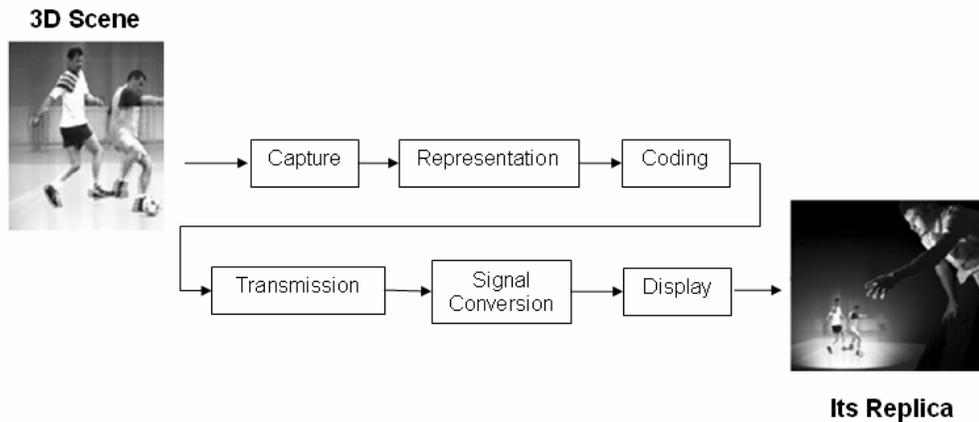


Figure 1.3 A 3DTV system [48]. Reprinted with kind permission of SPIE.

This thesis proposes a number of novel watermarking methods for different representations of 3D scenes. It should be emphasized that the video sequences are also interpreted as the basic implicit representations of the observed 3D scenery. While a novel watermarking method for this simplest representation of 3D scenes is proposed for imperceptibility problem in video watermarking in the first stage of the thesis, the fundamental representations of 3D scenes, i.e. geometrical representations and image based representations, are also examined in the second stage.

In general, the term, *3D watermarking*, in this dissertation, implies the watermarking of these fundamental representations of a 3D scene. The watermarking technology aims to protect the 3D representation by embedding hidden data into the main components of the representation. Considering the dimensions of the scene representations and the resulting components after an application, the watermarking methods in the literature can be classified into three groups, which could be denoted as *3D-3D*, *3D-2D* and *2D-2D*. The first pair of symbols identifies, whether the watermark is embedded into a 3D model or a 2D projection of it, and similarly the second pair of symbols identifies whether the watermark is detected in the 3D model or the 2D projection. The definitions and requirements of the corresponding watermarking problems are different in each of the aforementioned groups.

The first group, *3D-3D watermarking*, mostly focuses on protection of the intellectual property rights of the 3D geometrical structure, which is the most significant part of any 3D scene [42]-[44]. The watermark is embedded into the 3D geometric structure of an object existing in the scene and tried to be extracted from the 3D geometry after typical attacks on the geometry. These attacks might include rotation, translation, uniform scaling, polygon simplification, randomization of points, mesh compression, remeshing, mesh smoothing operations, cut operations, local transformations, global transformations and other operations on the geometry that changes the structure while preserving the visual quality in a desired level [42].

Although much of the literature on 3D watermarking focuses on 3D-3D, major applications require watermark to be detected in the 2D projection, either a 2D image or video sequence. This requirement is certainly applicable for computer animation, computer gaming, and virtual and augmented realities. The applications based on virtual view synthesis, e.g. free-view TV and 3D-TV, also require their rendered scenes to be protected. 3D-2D and 2D-2D watermark classes represent these scenarios.

3D-2D watermarking, aims to extract the watermark, which is originally hidden into the 3-D object, from the resulting images or videos (obtained after projection of 3-D object onto 2-D image planes); thus, protecting any visual representation of the object. The watermark can be both embedded to the geometry or the texture of the object [45]-[46]. The attacks on the texture, not only include the operations on the texture of the object, such as subsampling, JPEG compression or malicious attacks, but also involve modifications in texture mapping or distortions in the geometrical descriptions of the object [45]. Therefore, the problem is much more complicated compared to *3D-3D watermarking*.

In this taxonomy, the final category, *2D-2D watermarking*, aims to protect the image-based representation of a 3D scene. While the first two groups try to protect the intellectual property rights for the two important components of a traditional representation of a 3D scene, geometry and texture, the third group approaches to this problem, by watermarking sequences of images, which represent the 2D projections of

the same 3D scene, and extracting the watermark from any 2D rendered image, generated for an arbitrary angle of the scene via these sequences.

A practical application of this third category is being expected in the copyright protection of multi-view video content for the emerging new technologies, such as *free-view Televisions* (FTV), in which TV-viewers freely select the viewing position and angle for the transmitted multi-view video. Noting that a TV-viewer might also record a personal video from the arbitrarily selected views and misuse this content, it is apparent that copyright problems also exist and should be solved for free-view video. However, it should be noted that the problem is more complicated compared to conventional single-view video watermarking.

1.4 Main Contributions of the Thesis

The main objective of this dissertation is to explore new watermarking techniques for different 3D representations. For each of the aforementioned area, our goal is to present the specific requirements, examine the state-of-the-art methods, perform a categorization among these methods and develop novel and superior techniques, considering the specific requirements of each area.

The major contributions of this thesis can be summarized as follows:

- The imperceptibility problem in video watermarking is studied in detail. As a result of this analysis, a new method, which eliminates the common flickering type of distortion in the existing video watermarking schemes by utilizing the temporal sensitivity of human visual system (HVS), is developed. The proposed method is shown to be superior to two well-known methods in the literature [20], [33] after performing extensive tests on the robustness, imperceptibility and capacity of these methods.
- A detailed literature survey on 3D-3D watermarking is presented. Although most of the 3D watermarking methods in the literature belong to this category, the literature still lacks such a complete survey for 3D-3D watermarking. The presented survey reviews the requirements, as well as the problems in geometry

watermarking, compared to image and video watermarking, classifies the methods in the literature and summarizes pros and cons of each category. The solutions in the literature are examined separately based on their robustness against the main attacks in geometry watermarking, in order to give a more concise evaluation on 3D geometry watermarking.

- A novel problem for watermarking research is introduced, as a solution to the copyright protection of multi-view video content for the emerging *Free-View Television* (FTV). The proposed watermarking method aims to protect the virtual views in FTVs that are generated by the use of image based rendering (IBR) technologies on the transmitted multi-view video. Two solutions are proposed for different interpolation algorithms (nearest neighborhood and bilinear interpolation) in *light field rendering*, which is expected to be one of the fundamental technologies in FTV systems [47].
- A novel watermarking method based on projective invariants for 3D-2D watermarking is proposed. The method alters 3-D properties of a model such that the invariant measure, which is extracted from the 2D projection, is the same for any 2-D projection. In particular, a well-known projective invariant, 5-point cross ratio, is exploited during watermark embedding. The advantages of the proposed method compared to previous studies are given, as well as, a discussion on different scenarios for the detector.

1.5 Outline of Thesis

This thesis is organized as follows:

Chapter 2 gives an overview of the video watermarking methods with a taxonomy based on different solution approaches to the imperceptibility problem. The proposed video watermarking method based on temporal sensitivity of HVS is presented in detail and the comparative results with two well-known methods [20], [33] are given.

Chapter 3 presents an extensive literature survey on 3D-3D watermarking methods.

Chapter 4 introduces the requirements and problems in 2D-2D watermarking category. In particular, the main application of this category in free-view televisions is examined. This chapter gives the details of the proposed watermarking method for free-view video¹.

Chapter 5 presents 3D-2D watermarking scenario. This class of 3D watermarking algorithms is particularly challenging, since 3-D information is not directly available at the watermark detector. This chapter presents the details of the proposed method based on projective invariants. The future directions are presented.

Chapter 6 summarizes the dissertation and its main contributions.

¹ Free view (multi-view) video watermarking might also be interpreted as an intermediate step between video watermarking and 3D-3D (geometry) watermarking. However, we prefer to give multi-view watermarking in 3D watermarking categorization considering the image-based representation of 3D scenes.

CHAPTER 2

OBLIVIOUS SPATIO-TEMPORAL WATERMARKING OF MONO-VIEW VIDEO BY EXPLOITING HUMAN VISUAL SYSTEM

Mono-view video might be interpreted as the oldest and simplest representation of 3D scenes. One of the common problems in watermarking of this basic representation of 3D scenes is temporal distortions occurring between frames, while viewing the watermarked video. A video watermarking scheme, which inserts different watermarks into video frames independently without taking the temporal dimension into account, usually yields *flicker effect* in video. In particular, this distortion is due to the differences, resulting from watermarking operation, between the intensities of pixels at the same position in two successive video frames.

The maximum invisible difference between the intensities of the successive pixels is determined according to temporal limits of human visual system (HVS), which are called *temporal contrast thresholds* in psycho-visual research [49]-[50]. In order to avoid any visible distortions in the watermarked video, the modifications in the temporal direction should not exceed the temporal contrast thresholds. Yet, these temporal limits of HVS are not exploited in video watermarking research, to the best knowledge of the author. In order to achieve maximum robustness without any visible distortion in the video, the subjective limits of HVS in the temporal direction, should be fully utilized within a watermarking scheme.

In this chapter, we propose an oblivious video watermarking method, which does not need the original video during the watermark detection process, in order to solve the imperceptibility problem in video watermarking in an efficient way. The proposed method embeds maximum possible watermark energy, satisfying invisibility in the temporal direction by exploiting temporal contrast thresholds of HVS. *Temporal contrast*

threshold, $T(u,v,z)$, is defined as the contrast of a modulating temporal domain sinusoidal function, when the temporal fluctuations in the visual target become visible [50]. According to this definition, the modifications, which are smaller than the temporal contrast thresholds in the temporal direction of the visual target, should be imperceptible.

The outline of this chapter is as follows. In the next section, a detailed overview of the video watermarking methods is presented with a taxonomy based on the solution approaches to the imperceptibility problem in video. Section 2.2 gives the details of the proposed method and the computation of the temporal contrast thresholds utilized in the method. The proposed method is tested against common video processing attacks, such as additive Gaussian noise, lossy video compression, frame rate conversions and temporal shifts in Section 2.3. A detailed comparison of the proposed technique with two well-known methods from the literature is also presented in this section. Finally, the chapter is concluded in Section 2.4.

2.1 Overview of Video Watermarking Methods

In video watermarking literature, the imperceptibility problem is usually treated in three different ways. The first group of methods allows only arbitrarily small modifications for watermarking on video data in any transform domain, in order not to cause any visible distortion. The methods in the second group approach the same problem by utilizing some implicit properties of HVS to satisfy the trade-off between imperceptibility and robustness. Spread spectrum-based methods [7] form the majority of this group. The techniques exploiting basic features of HVS, such as *luminance* and *spatial sensitivity* [51], [52] can also be included into this group due to their simplicity. The final group is mainly based on higher-level features of HVS, such as *frequency masking* [53], *entropy masking* [54], *motion entropy masking* [55], *foveation phenomenon* [56] and some temporal properties of HVS, all of which clearly increase the complexity of the algorithm. The major application for most of the methods in this group is copyright protection, where complexity or real-time performance is not a serious requirement.

2.1.1 Video Watermarking Methods Discarding HVS

One of the most popular approaches in this group working in the spatial domain embeds the watermark bit into the least significant bit of the image pixels [15]. The drawback of the method is its susceptibility against signal processing operations. In a DCT domain approach [16], some of the predetermined DCT coefficients in each DCT block are forced to a specified order based on the watermark bit. A similar approach [17] embeds the watermark bit into the Intra- (I) frames of MPEG coded video by changing the relationship of mid-frequency coefficients of neighboring blocks. In a DWT domain approach [18], Y-frames of the video is decomposed into sub-levels and the low frequency components are watermarked by using a controlled quantization process [19]. In a different approach [20], the watermark is embedded in 3-D discrete Fourier domain. For each element of the watermark sequence, two magnitude coefficients are selected in the transform domain and the element of the watermark sequence is added onto one of them, based on its sign.

Some other methods in this group utilize the compressed video as input, where the watermark is directly embedded into the coded coefficients or motion vectors after video compression [21]-[24]. In one of such methods [21], which is proposed for MPEG compressed video, the variable length codes (VLCs) are changed with some other codewords, which have the same run length (only a quantized difference of one) and the same codeword length, according to the watermark bit. In this manner, the bit-rate of the video is tried to be kept constant and the distortion in the compressed data becomes limited [21]. The method in [22] also exploits VLC codewords to achieve real-time detection, as well as keeping the bit-rate unchanged. The improvement of the method [22] compared to [21] is its robustness against specific attacks for video watermarking, such as collusion. Biswas et. al. [23] proposes another watermarking strategy for MPEG compressed video by means of adaptively embedding the watermark into DCT coefficients of the compressed bit-stream. Similar methods are also proposed for H.264 compressed video sequences as well [24]. Another approach [25] changes the magnitude of the energy difference of high frequency DCT coefficients of two groups of 8x8 blocks according to a watermark bit. The method, which uses only intra frames for watermarking, is extended by embedding the watermark also into P frames in [26]. In a totally different approach [27]-[29], the direction of the motion vectors in the compressed

video is slightly modified for watermarking. Unfortunately, none of these methods take HVS into account in their algorithms for watermark insertion to fully exploit human perception.

2.1.2 Video Watermarking by Implicit Utilization of HVS

In this class of methods, the imperceptibility problem is treated by using HVS properties implicitly. Spread spectrum-based (SS) methods form the majority of this group. SS approaches for watermarking is first introduced for still images by Cox *et.al.* [7]. In their seminal method, after a global DCT transform on the image, the watermark is weighted by a fraction of a DCT coefficient and added to the same DCT coefficient. In fact, weighting the watermark energy according to the value of the frequency coefficient is an approximation for the contrast masking principle of HVS [53], which refers to the decrease in the visibility of a signal in the presence of another signal. In SS methods, the decrease in visibility is assumed to be linearly proportional with the DCT coefficient magnitude, although it is nonlinearly dependent for the actual case [51], [53]. As a drawback of this assumption, it is possible to encounter visible artifacts in the images, especially in large flat regions [9].

SS approaches are also adapted to video watermarking by means of 3-D multi-resolution decomposition. In [30], after video is decomposed into sub-signals at different resolutions by using a 3-D wavelet transform, the watermark is added to the high-pass wavelet coefficients at each resolution, excluding the lowest level by using the same procedure in [7]. In a similar approach [31], 3-D shape adaptive wavelet transform is applied to a group of objects (GOO) of video. In the same way, the watermark is inserted only to the high-pass wavelet coefficients. In another approach [32], the watermark is added in the 3D DWT domain by weighting the mark through a defined mask according to the variance and the luminance of the 3D subbands. Due to the same underlying reasoning as in the SS image watermarking case, the disadvantage for these methods is a possible fluctuation type of distortion in the temporal direction of the video.

The techniques, which exploit the basic features of HVS, such as luminance and spatial sensitivity, might also be included into this group due to their simplicity. The spatial sensitivity refers to the phenomena for which HVS differentiates the modifications in the

low contrast regions more easily, compared to the changes in high contrast regions. One of the most influential work for watermarking of the compressed video utilizes the spatial sensitivity by using arbitrarily smaller watermark weight for low-pass regions, whereas a higher watermark weight for the high contrast regions of a video frame [33]. Later, a similar approach is proposed for copy protection of DVD video [10]. In [34], the same principle is used for the watermarking of video objects. In this approach, wavelet transform is applied to the video objects and a watermark is inserted to the coefficients of high frequency bands by means of weighting with a visual mask, based on spatial characteristics of the subband image. In [35], the spatial sensitivity of HVS is exploited in the DCT domain. For each 8x8 DCT Y-block of I-frames of MPEG coded video, the embedding strength and region of the watermark is determined, based on the total energy of the AC coefficients in that block, while being compliant with the spatial sensitivity of HVS. In addition to spatial sensitivity, luminance sensitivity of HVS is also utilized for video object watermarking [36]. In this case, the watermark is weighted by a visual mask based on the fact that human eye is less sensitive to contrast in dark or bright areas compared to the middle luminance areas [51].

In all the aforementioned methods, there is often a global scaling factor, which satisfies the trade off between the imperceptibility and robustness. Obviously, the drawback of such an approach is the requirement for a control process to satisfy the imperceptibility, every time the watermark is embedded into a different content.

2.1.3 Video Watermarking Based on HVS Properties

The last group consists of the methods that use more sophisticated models of HVS. Most of the methods in this group use explicit perceptual measures, called *contrast thresholds* to determine the location and strength of the watermark signal. Contrast threshold, in general, refers to the minimum level of the contrast that a sinusoidal grating is observable [53]. In vision research, the increase in contrast thresholds for many different phenomena of HVS, such as luminance masking [51], contrast masking [51], entropy masking [54], motion entropy masking [55], temporal masking [50] and foveation [56] is analyzed by means of perceptual tests. The thresholds under all these phenomena are simply called as *Just Noticeable Differences* (JNDs) [51]. In [9], JNDs under luminance and contrast masking are exploited to determine the strength and embedding location of the

watermark. The watermark is inserted into the 8x8 block DCT coefficients of an image which are larger than JNDs to satisfy imperceptibility and robustness to compression attack. In another approach [37], entropy masking, which refers to the effect of the interaction between neighboring blocks to the JND values, is considered for an improvement. In [38], a method based on the foveation phenomenon of HVS that embeds watermark energy into the peripheral of the foveation point is proposed. The method is shown to be superior to other HVS-based methods [9], [57] since the watermarking energy is comparatively increased due to higher JNDs under foveation phenomenon. The aforementioned JND-based methods [9], [38] are also extended to MPEG-coded video, by means of watermarking *I-frames* and applying a simple linear interpolation of the watermarks in time between two consecutive I-frames, in order to avoid temporal flickering occurring in every intra frame of the video. As a drawback, in these methods, there is inefficiency in utilization of the “available space” for data hiding, since inter-frames are excluded from watermarking.

The imperceptibility problem, especially temporal flickering, is also approached in some methods [39]-[41] by considering the temporal dimension, as well as spatial dimensions, during watermark embedding. In [39], a temporal wavelet transform is applied to each video shot and a different watermark, weighted according to spatial and contrast masking characteristics of HVS, is embedded to each wavelet coefficient frame. Another watermarking scheme [40], dealing with flickering problem, is proposed, especially, for digital cinema and it embeds the watermark to the mean luminance value of Y-component of the video signal. In order to avoid flickering along temporal dimension, they embed the same watermark bit into a number of successive frames. Later, the method is extended [41] by means of applying temporal low pass filtering to watermark before embedding, in order not to cause a major difference between consecutive pixels.

2.2 Proposed Watermarking Method

As examined in detail in the previous section, the temporal sensitivity limits of HVS are not fully utilized in the existing video watermarking methods. Accordingly, it is not possible to solve the flickering problem in each case and still satisfy the maximum watermark strength in the temporal direction, which yields the maximum robustness. In

order to achieve maximum robustness without any visible distortion in the video, the limits of HVS in the temporal direction, i.e. *temporal contrast thresholds*, should be fully utilized by a watermarking scheme.

The proposed method exploits the temporal contrast thresholds of HVS to determine the maximum strength of the watermark for oblivious detection scenario. Oblivious detection property makes the application area of the proposed scheme wider, spreading from copyright protection and alteration detection of original data to covert communications and broadcast monitoring. In the following subsection, the computations of the temporal contrast thresholds are explained. Next, the details of the proposed watermark embedding and detection are presented.

2.2.1 Temporal Contrast Thresholds

Temporal contrast threshold, $T(u,v,z)$, is defined as the contrast of a modulating temporal sinusoidal function, when the temporal fluctuations in the modulated visual target become visible [50]. The visual experiments for temporal contrast thresholds are conducted in [49] and [50]. In the experiments [49], a visual stimulus is produced by first computing an image composed of a square array of 8x8 pixel blocks within each of which is placed a DCT function of the same spatial frequency, (u,v) . Then, each basis function is modulated by a Gabor function in time, which is the product of a Gaussian and a sinusoid of temporal frequency, z , and phase [49]. The generated visual pattern is presented to a subject standing at a specific distance. The subject is checked whether the temporal variation in the modulated target is distinguishable, beginning from a target with zero contrast. When the response is negative, the amplitude of the sinusoid is increased until the target becomes visible to the subject. The ratio of the amplitude of the sinusoid to the mean luminance of the visual pattern at which the target become visible is denoted as *temporal contrast threshold* and its reciprocal is called as *temporal contrast sensitivity* [50].

In [49], temporal contrast thresholds for a number of spatial DCT frequencies and temporal frequencies are determined by subjective visual tests and then, the thresholds

are modeled as a product of a global threshold parameter T_o , a temporal function, $T_z(z)$, a spatial function, $T_f(u, v)$, and an orientation function, $T_a(u, v)$:

$$T(u, v, z) = T_o \cdot T_z(z) \cdot T_f(u, v) \cdot T_a(u, v) \quad (2.1)$$

where,

$$T_z(z) = \left| \frac{-1 + \exp\left(\frac{1 + i2\pi\tau_0 z}{\tau_0 w_s}\right)}{-1 + \exp\left(\frac{1}{\tau_0 w_s}\right)} \right| \quad (2.2)$$

$$T_f(u, v) = \exp\left(\pi \frac{(u^2 + v^2)}{f_0^2} \left(\frac{p}{16}\right)^2\right) \quad (2.3)$$

$$T_a(u, v) = 2^{\frac{\beta-1}{\beta}} \left/ \left(1 - \pi \frac{4ru^2v^2}{(u^2 + v^2)^2}\right)\right. \quad (2.4)$$

In the proposed method, this model is utilized for the computation of temporal contrast thresholds. The following parameters are selected to compute $T_f(u, v)$ and $T_a(u, v)$ [49]: $p=32$ pixels/degree; $r= 0.1$; $\beta = 2$; $f_0= 17.2$. T_o is taken as 0.0126 [49]. In order to compute $T_z(z)$, the available graph in [49] is utilized, rather than using the first-order approximation in the given equation. The validity of these models is also tested and verified for the computer generated visual patterns of 8-bit resolution on a computer screen by means of subjective experiments, whose details are given in Appendix A.

2.2.2 Watermark Embedding

The overall structure of the proposed watermarking method is given in Figure 2.1. The proposed method mainly consists of two parts. In the first part, the video sequence is separated into shots and transformed into (u, v, z) domain by means of applying spatial 2D DCT transform followed by a DFT transform in the temporal direction in order to utilize the temporal contrast thresholds. In the second step, watermark signal, which is generated by using m-sequences, is embedded by means of modifying one of the

randomly selected coefficient pairs in the transform domain based on the sign of the watermark element.

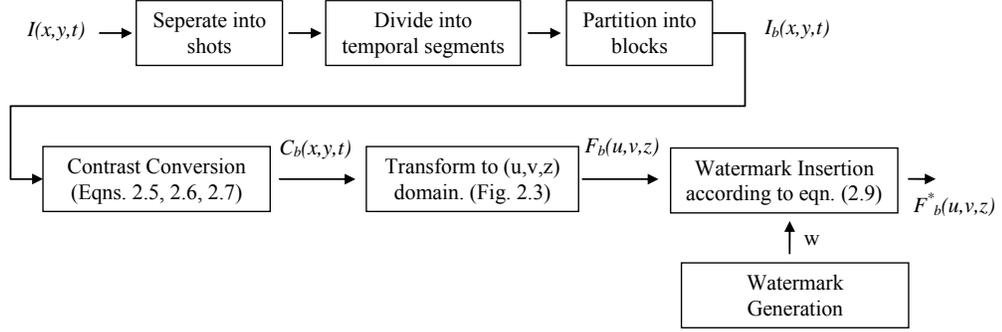


Figure 2.1 Overall structure of the watermark insertion process

A segmentation of the video into consecutive shots is obtained after applying any shot boundary detection algorithm [58], in order to obtain temporally stationary signals. This decomposition yields a more suitable framework for exploiting the temporal contrast thresholds, which are measured for a visual pattern of constant mean value [49]-[50].

Then, each shot is separated into the temporal segments of K frames. The same message is embedded into each temporal segment. Next, each segment is divided into the blocks of $8 \times 8 \times K$ pixels as illustrated in Figure 2.2.

The video signal for each block is denoted as $I_b(x, y, n)$, where x and y are the spatial coordinates in a block, n is the frame number and b denotes the block number. For each block, the intensities are converted into contrast values, where contrast is defined as the ratio of the AC component of the video signal to the DC component [50]. The DC component of the block video signal, $I_{b,DC}$, is simply defined as the time-space average of the block [50]:

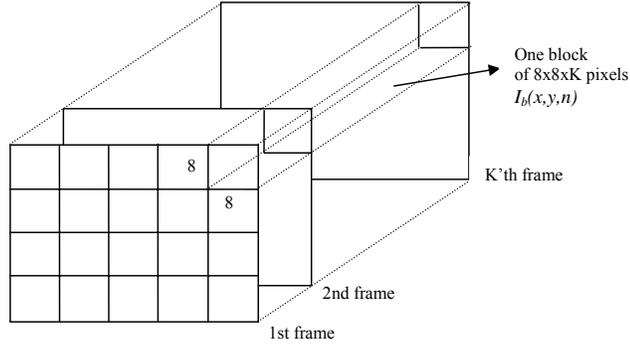


Figure 2.2 Partition of each temporal segment of K frames into blocks of $8 \times 8 \times K$ pixels.

$$I_{b,DC} = \frac{1}{8 \times 8 \times K} \sum_n \sum_x \sum_y I_b(x, y, n) \quad (2.5)$$

while the AC component of the signal is defined as:

$$I_{b,AC}(x, y, n) = I_b(x, y, n) - I_{b,DC} \cdot \quad (2.6)$$

Hence, the spatio-temporal contrast sequence for each block is obtained by:

$$C_b(x, y, n) = \frac{I_{b,AC}(x, y, n)}{I_{b,DC}} \cdot \quad (2.7)$$

Next, the contrast signal for each block, $C_b(x, y, n)$, should be transformed into (u, v, z) domain in order to exploit the temporal contrast thresholds. For this purpose, 2-D DCT (8×8) transform is applied to the contrast signal for each frame (i.e. fixed n) and the resulting signal is denoted as $f_b(u, v, n)$, as shown in Figure 2.3. Then, a temporal 1-D DFT is applied to the resulting signal in the temporal direction for each u and v value. The final transformed signal is denoted, as $F_b(u, v, z)$.

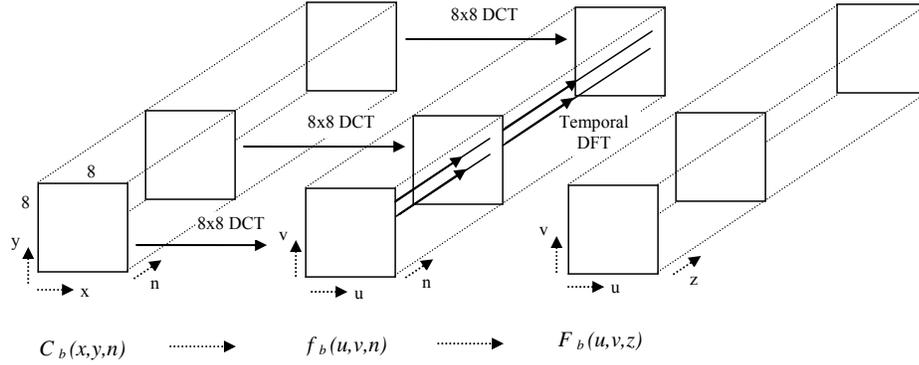


Figure 2.3 The transformation of $C_b(x, y, n)$ into (u, v, z) domain.

Since the proposed method should spread the watermark among all blocks of $8 \times 8 \times K$ pixels, the watermark sequence is first partitioned into equal portions according to the number of the blocks in the temporal segment. Then, elements of the watermark sequence corresponding to each block are embedded to the magnitude coefficients of $F_b(u, v, z)$. In order to satisfy imperceptibility, the watermark strength is limited such that its maximum value does not exceed the temporal thresholds, $T(u, v, z)$. The method uses the given model in section 2.2.1 to compute the thresholds during watermark insertion.

In the proposed method, every watermark is generated by using m-sequences. The stego-signal, i.e. the message which will be inserted to the video, is converted into a binary form, resulting of M bits. For each bit (+1,-1), an m-sequence of length N is generated and these m-sequences are summed up, after multiplying each sequence with its representing bit [20]. Then, the resulting sequence is normalized by the expected value of the maximum absolute element of the resulted sum sequence:

$$w = \frac{\sum_{i=1}^M b_i v_i}{E(\max(\text{abs}(\sum_{i=1}^M b_i v_i)))} \quad (2.8)$$

where w denotes the resulting watermark, v_i is the m-sequence for the current bit, b_i , and E corresponds to the expectation operation.

In order to embed each element of the watermark sequence to any block of $8 \times 8 \times K$ pixels, two random coefficients are selected from $F_b(u, v, z)$ for each watermark element. DC-coefficient is excluded from this selection process due to its high sensitivity to any modifications. During insertion, if the watermark symbol is greater than (or equal to) zero, the watermark sequence element is added to the first coefficient after multiplying with the contrast threshold for that coefficient. If it is less than zero, it is added to the second coefficient by using the same procedure. Assuming $F_b(u_1, v_1, z_1)$ and $F_b(u_2, v_2, z_2)$ denote the selected 1st and 2nd coefficients, the watermark insertion process can be summarized as

$$\begin{aligned} \left| F_b^*(u_1, v_1, z_1) \right| &= \left| F_b(u_1, v_1, z_1) \right| + w_i T(u_1, v_1, z_1) & w_i \geq 0 \\ \left| F_b^*(u_2, v_2, z_2) \right| &= \left| F_b(u_2, v_2, z_2) \right| + |w_i| T(u_2, v_2, z_2) & w_i < 0 \end{aligned} \quad (2.9)$$

where w_i refers to i^{th} element of the watermark sequence, w , and $F_b^*(u, v, z)$ is the watermarked signal. Similarly, the next element of the watermark sequence (of size N) is also inserted into the same $8 \times 8 \times K$ block by choosing some other random coefficient pairs. In summary, all the elements of the watermark sequence are equally spread among all $8 \times 8 \times K$ blocks.

2.2.3 Watermark Detection

In order to detect the watermark, after converting intensities of $8 \times 8 \times K$ block pixels into contrast values, the contrast video is transformed into (u, v, z) domain. A new vector, w' , whose i^{th} element is the difference between the coefficients which are selected for the i^{th} element of w , is constructed [20]. Hence, w' can be expressed as follows:

$$w' = w \otimes T + n = \sum_{i=1}^M b_i v_i \otimes T + n \quad (2.10)$$

where \otimes represents the element-by-element multiplication of the two vectors. T is the vector consisting of the temporal thresholds corresponding to the watermarked coefficient

of the transformed video. The term, n , corresponds to the noise added to each watermarked coefficient due to the possible signal processing operations or attacks.

The message bits are extracted by means of an inner product between w' and v_j :

$$B'_j = \langle w', v_j \rangle \cong E(T) \left(\sum_{i=1}^M b_i \langle v_i, v_j \rangle \right) + \langle n, v_j \rangle \quad (2.11)$$

Using the orthonormality properties of m-sequences [59], (2.11) can be written as:

$$B'_j \cong E(T) (b_j N - (M - 1)) + \langle n, v_j \rangle \quad (2.12)$$

$$b'_j = \text{sign}(B'_j) = \text{sign}(b_j) = b_j \quad (2.13)$$

This result is valid, since $N \gg M$ and $\langle n, v_j \rangle$ is negligible compared to N due to the fact that the distribution of n can be approximated by a zero mean Normal distribution. Comparing to the method in [20], the main improvement of the proposed method is obtained by inclusion of the temporal contrast thresholds into this formulation. This inclusion emerges as a multiplicative term, $E(T)$, (average of the temporal thresholds), in (2.13) and increases the difference between the terms $\langle n, v_j \rangle$ and $(b_j N - (M - 1))$. Consequently, this results a decrease in bit error rate and still guarantees the temporal imperceptibility of the watermark, as it will be justified in the next section.

2.3 Experimental Results

The experiments on the proposed approach are conducted in 4 stages with comparisons against two oblivious watermarking methods [20], [33] by utilizing *Foreman and Mother* sequences (352x288, 192 frames). The tested cases can be summarized as follows:

- Case 1: Equal watermarking energy insertion is enforced for all methods, various attacks in different levels are considered and bit error rate (BER) is measured during extraction (Tables 2.1, 2.2, 2.3, 2.4).

- Case 2: Invisibility criteria is enforced for all methods, various attacks in different levels are considered and BER is measured during extraction (Tables 2.5, 2.6, 2.7).
- Case 3: Equal BER during extraction is enforced for all methods after only the compression attack. The visual quality and (PSNR) values are compared.
- Case 4: The same number of bits within a set of values is embedded for all methods, various attacks are considered and BER is measured during extraction (Tables 2.8, 2.9).

For comparison purposes, the following oblivious methods from the literature are implemented: The first method is a transform domain method, in which 3-D DFT is applied to each 8x8 video block of 16 frames [20]. As in the proposed approach, an element of the watermark sequence, formed by m-sequences, is added to one of the randomly selected middle frequency coefficients, based on the sign of the watermark element. The watermark is only added to mid-frequency coefficients in order to satisfy the trade-off between robustness and imperceptibility, since HVS is more sensitive to changes in the low frequency coefficients and a lossy compression can distort the high frequency coefficients [20]. The other method [33] is a popular approach from the literature, which adds the watermark into the pixel values of a line-scanned video, after multiplying with a local adaptive scale factor. The local scaling factor is varied according to spatial masking properties of HVS by means of filtering each video frame by a 3x3 high pass filter [33]. This operation decreases the watermark strength in the flat regions of a frame, where HVS is more sensitive and increases the embedded watermark energy in the detailed regions, where HVS is less sensitive.

As the initial experiment, a message of 88 bits is embedded into the Y-component of each video sequence for all tested methods. In the proposed approach, the sequences are divided into temporal segments of 16 frames and 88 bits are embedded into each segment. An m-sequence of length 8191 ($2^{13}-1$) is generated for each bit of this message, during the construction of the watermark sequence from the message. The watermark sequence is equally spread among all blocks of 8x8x16 pixels. Such an approach approximately corresponds to 6 watermark coefficients embedded into each spatio-temporal block. For the proposed method, typical samples from the original and

watermarked frames for *Foreman* and *Mother* sequences are shown in Figure 2.4.(a), (b), (c) and (d), respectively. The visual equivalence of the watermarked and original videos are confirmed by presenting the sequences to a number of subjects. Based on these perceptual tests (see [39] for the complete experimental procedure), it is concluded that the embedded watermark produces no visible distortion in the original video. For viewing purposes, the original and watermarked video sequences for the proposed method are available at URL: http://snap-mmrg.eee.metu.edu.tr/paper_ieee/watermarked_sequences .

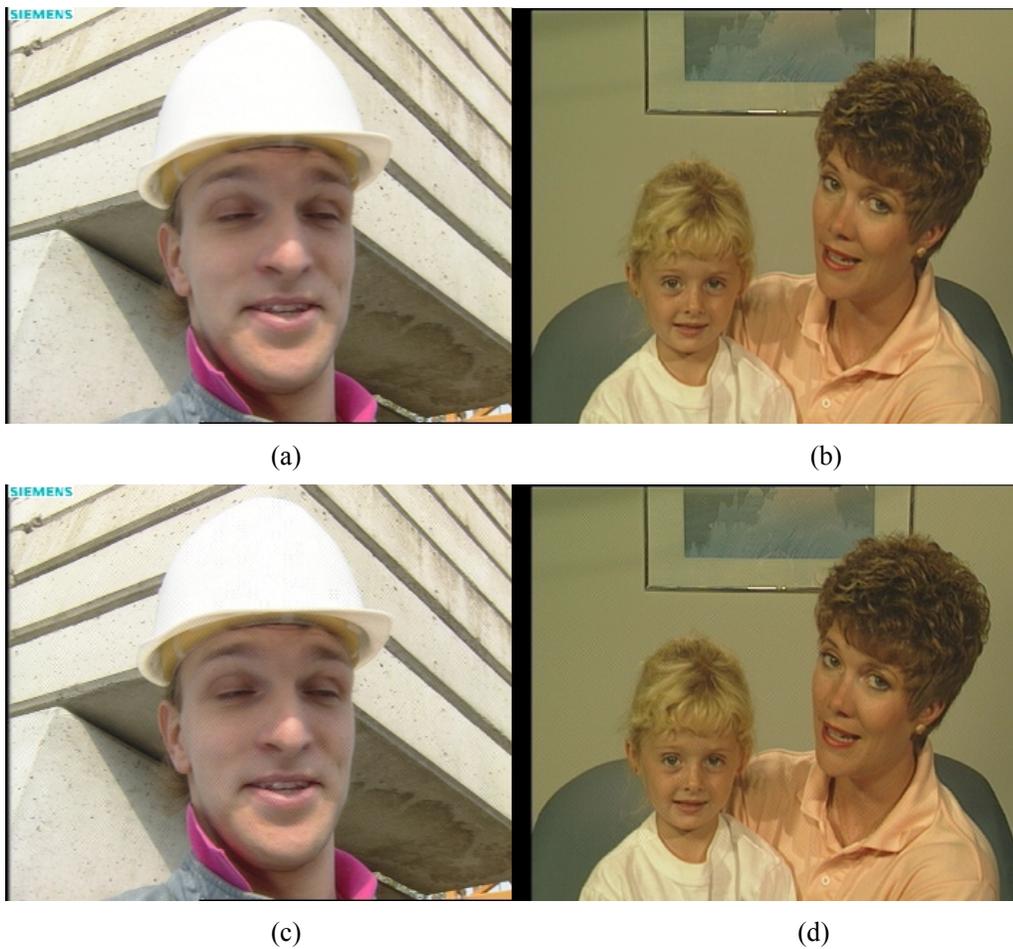


Figure 2.4 (a), (b) Original frames from *Foreman* and *Mother* sequences. **(c), (d)** Watermarked frames by the proposed method



(e)

(f)



(g)

(h)

Figure 2.4 (Continued) (e), (f) Watermarked frames by the compared method in [20] (g), (h) Watermarked frames by the compared method in [33] (Average PSNR between original and watermarked sequences are around 39 and 43 dBs for *Foreman* and *Mother*, respectively).

In order to equalize the added watermark energy for all methods, the watermark strengths for the compared methods [20], [33] are increased until the resulting watermark energies in their corresponding domains are approximately the same with that of the proposed method. In this manner, based on the randomly selected frequency coefficient pairs, average *PSNR* value for *Foreman* sequence results as 39 dB in the proposed method, whereas *Mother* sequence yields 43 dB. Typical frames from the two watermarked sequences for the compared methods [20], [33] are shown in Figure 2.4 (e), (f), (g) and

(h). The fluctuation type of distortion can be observed clearly in all test sequences for the method in [20] for not considering a perceptual limit for the embedded watermark strength. On the other hand, the watermarked sequences of [33] look visually equivalent with the original sequences at the first sight. However, in the motion-free regions in the watermarked sequence (e.g. the “SIEMENS” overlay text in *Foreman* and the spatial edges at the background of *Mother*), a flickering type of distortion is noticeable due to embedding different strengths to the same spatial motion-free locations. For visualization purposes, the original and watermarked sequences for the compared methods are available at URL: http://snap-mmrg.eee.metu.edu.tr/paper_ieee/comparison_case1

2.3.1 Case 1: Equal Watermark Energy Insertion under Various Attacks

In the first phase of the experiments, the method is tested against the common signal processing operations, such as additive white Gaussian noise, video coding at different bit rates, frame rate conversions and temporal shifts that might occur during the detection stage, while the embedding energy for the watermarks is kept equal for all methods. However, it should be emphasized that *spatial* synchronization issues of the video frames, such as synchronization after spatial shifts, scaling, rotation and cropping are beyond the scope of this thesis. Some approaches for synchronization in the presence of spatial shifts and scaling are described in [10], [60] and [61]. Similar ideas, such as periodically embedding the watermark into each frame [60] or utilization of a template to detect the geometric changes between video frames [61] could also be adapted for the proposed method.

The results are tabulated in terms of the average percentage of the erroneously extracted bits from each segment (i.e. bit error rate, *BER*) for synchronization problems at different temporal shifts, additive white Gaussian noise (AWGN) at different PSNR values, ITU H.263+ video coding at different quantization levels and frame rate decrease by half (after dropping one of each consecutive two frames) at Tables 2.1, 2.2, 2.3 and 2.4, respectively. Note that BER results are computed as the ratio of the total number of erroneously extracted bits from 12 temporal segments of 16 frames to the total number of embedded bits, which can be written as

$$\frac{\sum_{i=1}^{12} e_i}{12.88}, \quad (2.14)$$

where e_i refers to the number of erroneously extracted bits from each segment. It should be noted that 88 bits are embedded to each temporal segment.

In Table 2.2, it should be noted that AWGN-I indicates addition of noise in the watermark embedding domain (e.g. quantization noise in a compression operation), whereas AWGN-II stands for noise insertion in the spatial (intensity) domain (e.g. sensor noise or A/D & D/A conversions). Hence, for the same noise levels, while AWGN-II is resulting with similar PSNR values as expected, AWGN-I might yield different PSNRs, since noise is added in a different (transform) domain. Moreover, in Table 2.2, AWGN-II is not used for the method by Hartung *et.al.* [33], since this method is purely in the spatial domain. On the other hand, at Table 2.4, in order to convert the frame rate to the original rate before watermark detection, the dropped frame is replaced by the average of two neighboring frames.

As it can be observed in Table 2.1, while the proposed method can successfully extract the watermark without any bit error for all temporal shifts, the method in [20] can detect the message without any bit error up to a temporal shift of 9 frames, since this method also embeds the watermark in a shift invariant domain (3D DFT magnitude coefficients). Unfortunately, the spatial domain method in [33] is not invariant to any temporal shifts, since this method embeds the watermark directly into the pixel values of the line scanned video in the spatial domain.

According to the results in Table 2.2, all three methods are able to extract the information without any error, up to 14-15 dBs. Since visual quality of such noisy sequences is quite low, a comparison with respect to average BER is not significant for those very low PSNR levels. Note that BERs in each row in the table correspond to different noise PSNRs for each method in the table.

In order to visualize the performance of the methods against compression more clearly, the average BER values in Table 2.3 are also illustrated in Figure 2.5 for *Foreman* sequence. It can be observed that the proposed temporal sensitivity based method is superior to the other methods.

Finally, the robustness results for frame rate conversion are tabulated in Table 2.4. While the proposed method and the transform domain method [20] are able to extract the hidden message without any bit errors, the performance of the spatial domain method [33] is quite poor, since this attack mainly distorts the video in the spatial domain, where the watermark is embedded in [33].

Table 2.1 BER results (Eq. 2.14) for temporal shifts during the detection stage (*Foreman* sequence). The temporal segmentation begins from k^{th} frame.

Synchronization Attack (BER %)	Temporal shift, k				
	$k=2$	$k=5$	$k=9$	$k=12$	$k=15$
Proposed Method	0	0	0	0	0
Deguillaume <i>et al.</i> [20]	0	0	3.20	0	0
Hartung <i>et al.</i> [33]	47.25	48.67	46.59	45.26	51.70

Table 2.2 BER results (Eq. 2.14) for additive White Gaussian noise (AWGN) at various PSNR levels (*Foreman* sequence). (I: AWGN in transform domain. II: AWGN in spatial domain.)

Proposed Method AWGN (I)		Deguillaume <i>et al</i> [20] AWGN(I)		Hartung <i>et al</i> [33] AWGN(I)		Proposed Method AWGN(II)		Deguillaume <i>et al</i> [20] AWGN(II)	
PSNR (dB)	BER (%)	PSNR (dB)	BER (%)	PSNR (dB)	BER (%)	PSNR (dB)	BER (%)	PSNR (dB)	BER (%)
37.30	0	43.20	0	43.30	0	43.24	0	43.14	0
29.43	0	37.55	0	37.45	0	37.64	0	37.69	0
23.66	0	31.63	0	31.53	0	31.21	0	31.21	0
18.22	0	25.66	0	24.65	0	24.66	0	24.66	0
15.16	0.28	19.85	0	18.90	0.09	18.90	0	18.90	0
13.11	4.92	14.43	1.42	13.95	0.28	13.97	4.36	13.95	9.65

Table 2.3 BER (%) results (Eq. 2.14) for ITU H.263+ coding at various quantization levels and bit-rates (*Foreman* sequence).

<i>ITU H.263+ CODING</i>	Proposed Method		Deguillaume <i>et al</i> [20]		Hartung <i>et al</i> [33]	
	Quantization Level	Bit-rate (Kbits/s)	Bit Error Rate (%)	Bit-rate (Kbits/s)	Bit Error Rate (%)	Bit-rate (Kbits/s)
2	4443	0	4236	0	4114	0
4	2197	0	1900	0	1854	0
6	1390	0	1131	0	1136	0.19
8	982	0	790	0.09	805	1.61
10	745	0	613	0.09	625	6.44
12	596	0	509	0.47	516	9.19
14	496	0	440	2.08	444	12.31
16	427	0	391	3.79	393	15.53
18	378	0.19	356	7.20	357	19.32
20	343	0.76	330	7.77	330	22.25
22	316	2.18	308	10.23	309	23.39
24	295	5.59	291	12.22	292	25.66

Table 2.4 BER results (Eq. 2.14) for frame rate decrease by half (*Foreman* sequence).

Frame Rate Conversion Attack	Proposed Method	Deguillaume <i>et al</i> [20]	Hartung <i>et al</i> [33]
BER (%)	0	0	22.16

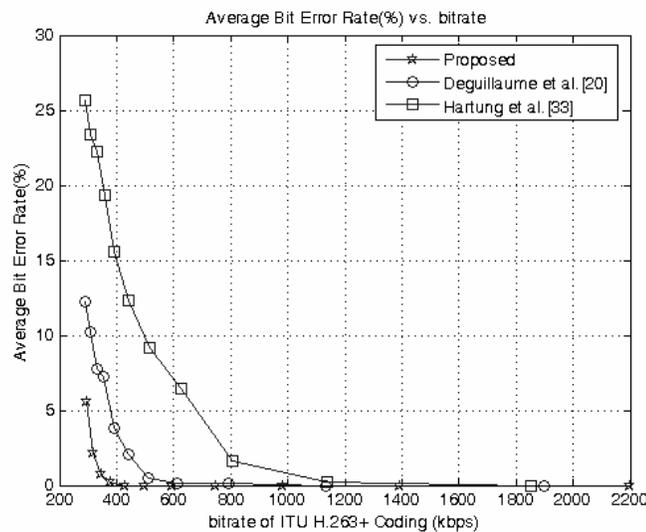


Figure 2.5 BER (%) results (Eq. 2.14) after ITU H.263+ coding for different bit-rates.

2.3.2 Case 2: Invisibility under Various Attacks

In the second stage of the simulations, the embedded watermark energies for the three methods are set to the levels, where the distortions in the watermarked sequence for all approaches are invisible ($PSNR$ is around 45 dB for *Foreman* at which the distortions are invisible both for [20] and [33]). Note that the proposed method does not allow modifying the watermark strength for any coefficient, since its strength is adjusted based on temporal contrast thresholds, which are fixed for a given frequency. Instead, different coefficients with lower temporal contrast thresholds can be selected for watermarking in order to adjust the $PSNR$ to the desired level for the proposed method. Typical frames from the watermarked sequences with no visible distortions are shown in Figure 2.6. The complete video sequences are also available at:

http://snap-mmrg.eee.metu.edu.tr/paper_ieee/comparison_case2

The robustness tests for AWGN, ITU H.263+ coding and frame-rate conversion are repeated for this case and tabulated in Table 2.5, 2.6 and 2.7, respectively. As in Case-1, all three methods are able to extract the message for AWGN attack with no bit errors, up to $PSNR$ levels of 18-19 dBs, where the video quality becomes quite poor. In ITU H.263+ coding tests, the proposed method is superior to the other methods. Similar to the previous case, the performance of [33] for the frame-rate conversion attack is far from acceptable.

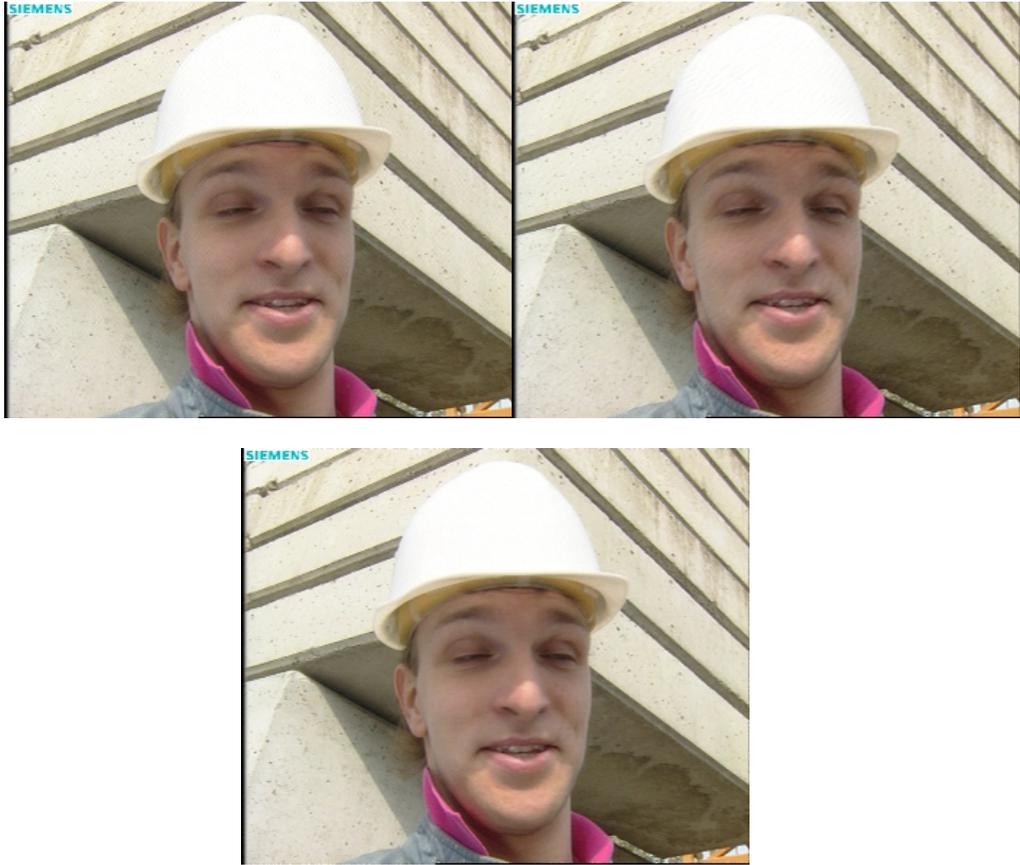


Figure 2.6 Typical frames from watermarked *Foreman* sequences for the proposed method (top left), compared methods in [20] (top right) and [33] (bottom).
(Average PSNR~45 dB.)

Table 2.5 BER (%) results (Eq. 2.14) for *Foreman* sequence against AWGN at different noise PSNR levels. (PSNR~ 45 dB)

Proposed Method AWGN (I)		Deguillaume <i>et al</i> [20] AWGN(I)		Hartung <i>et al</i> [33] AWGN(I)		Proposed Method AWGN(II)		Deguillaume <i>et al</i> [20] AWGN(II)	
PSNR (dB)	BER (%)	PSNR (dB)	BER (%)	PSNR (dB)	BER (%)	PSNR (dB)	BER (%)	PSNR (dB)	BER (%)
23.65	0	25.66	0	24.65	0	24.66	0	24.65	0
18.24	0	19.84	1.51	18.89	0.95	18.89	3.69	18.89	0.28
13.14	3.88	14.45	23.20	13.95	6.34	13.96	25.94	13.96	20.45

Table 2.6 BER (%) results (Eq. 2.14) for *Foreman* sequence against ITU H.263+ Coding at different quantization levels and bit-rates.

ITU H.263+ CODING	METHOD					
	Proposed Method		Deguillaume <i>et al</i> [20]		Hartung <i>et al</i> [33]	
Quantization Level	Bitrate (kbps)	Bit Error Rate (%)	Bitrate (kbps)	Bit Error Rate (%)	Bitrate (kbps)	Bit Error Rate (%)
2	3781	0	3568	0	3553	0
4	1736	0	1602	0.09	1608	1.13
6	1067	0	1004	0.57	1006	6.91
8	758	1.33	731	2.94	730	12.93
10	591	5.77	581	8.05	581	19.41

Table 2.7 BER results for *Foreman* sequence against frame rate conversion

BER (%) after Frame Rate Conversion	<i>Foreman</i> sequence
Proposed Method	0
Deguillaume <i>et al</i> [20]	0
Hartung <i>et al</i> [33]	21.59

2.3.3 Case 3: Equal BER during Extraction for Compression Attack

In the third stage of comparisons, BER value that is observed during the message extraction stage is set equal for all three methods, and then, the perceptibility and embedded watermark energy values are compared. For ITU H.263+ compression attack, first, the quantization level, for which the bit error rate is immediately larger than 5 %, is determined for the proposed method. Then, the watermark strengths for the compared two methods [20], [33] are increased, such that the bit error rate after ITU H.263+ coding at the same quantization level is approximately equal to 5 %.

Typical frames from *Foreman* for the compared methods [20], [33] are given in Figure 2.7. Compared to the original watermarked sequence in Figure 2.4 (c), a distortion on the face of *Foreman* is quite visible for [20]. These differences are more visible, while viewing its video. For the other method [33], a flickering type of distortion is distinguishable, also during video viewing. The average PSNR values for these sequences

are around 36 and 31 dBs, for the methods in [20] and [33], respectively, whereas the PSNR for the proposed method is 39 dB, as stated before. From the point of the required watermark energy to achieve the same bit error rate, the proposed method is clearly the best among the three methods. The resulting sequences are available at the following URL: http://snap-mmrg.eee.metu.edu.tr/paper_ieee/comparison_case3



Figure 2.7 Typical watermarked frames from the *Foreman* sequence for the compared methods in [20] (left frame) and [33] (right frame). Strength of watermarks are adjusted to give a bit error-rate of 5 % for ITU H.263+ coding at the bit rate of approximately 300 kbps.

2.3.4 Case 4: Capacity under Various Attacks

In this section, experimental results for determining the capacity of the proposed watermarking method are presented and embedded bit number is changed. During watermark embedding, due to the normalization operation in (8), PSNR values always remain at 39 dB and 43 dB for *Foreman* and *Mother*, respectively, in each case for different number of bits are inserted. Hence, due to normalization, it is possible to embed quite large number of bits without any distortion, as well as minimum robustness against attacks. Therefore, the effect of embedding different number of bits to the proposed method should be tested, considering some typical attacks. These attacks are selected such that the maximum number of bits that can be embedded into 192-frame

video, yielding a negligible BER. The results are tabulated in Table 2.8 for *Foreman*, after a number of “mild” attacks, such as AWGN-I (noise PSNR \cong 36dB), AWGN-II (noise PSNR \cong 36 dB), ITU H.263 Coding (bit-rate \cong 800 kbps, coding PSNR \cong 36 dB), frame rate conversion and temporal desynchronization, for different number of embedded bits. The results show that the proposed method can survive up to around 500 bits for the temporal shift and ITU H.263+ Coding, whereas approximately 1500 bits could remain for AWGN I, II and Frame Rate Conversion for the mentioned levels.

Table 2.8 BER (Eq. 2.14) against typical attacks for different number of embedded bits for *Foreman* sequence. (AWGN-I PSNR \cong 36 dB; AWGN-II PSNR \cong 36 dB; *ITU H.263* + Coding Bit-rate \cong 800 kbps, Q:9, Coding PSNR \cong 36 dB; Temporal Shift, $k=9$)

<i>BER (%)</i> <i>Foreman</i>	Number of Embedded Bits				
	<i>500 bits</i>	<i>750 bits</i>	<i>1000 bits</i>	<i>1250 bits</i>	<i>1500 bits</i>
Attack 1 : AWGN I	0	0	0	0	0.01
Attack 2 : AWGN II	0	0	0	0	0.02
Attack 3 : ITU H.263 +	0.81	2.13	3.36	5.07	6.80
Attack 4 : Temporal shift	0	0.05	0.09	0.34	0.64
Attack 5 : Frame Rate Conv.	0	0	0	0	0.01

In the second part of the capacity tests, the effect due to the length of m-sequences is also examined. For this purpose, the number of bits embedded into the video is increased and bit-error rates for the typical attacks are determined for the m-sequences of length 8191 ($2^{13}-1$) and 16383 ($2^{14}-1$). Since more coefficients are modified when the length of the m-sequences is doubled, the watermark energy embedded into the video increases for this case. BERs for different number of embedded bits are tabulated in Table 2.9. Only the effective attacks, where BER reaches around 5 % in the previous robustness tests, are tabulated to show clearly the decrease in BER. BER values are also plotted with respect to the number of embedded bits for both cases for the AWGN I of noise PSNR 13-14 dB in Figure 2.8. A decrease in BER is quite clear for doubling the length of m-sequence. A typical frame is presented in Figure 9 (a) with a PSNR of 35 dB.

Table 2.9 BER results (Eq. 2.14) for different number of embedded bits for *Foreman* sequence for m-sequences of lengths, $(2^{13}-1)$ and $(2^{14}-1)$.

(AWGN I - PSNR \approx 13-14 dB; AWGN II- PSNR \approx 13-14 dB; *ITU H.263* + Coding – Bitrate \approx 330 kbps, Q: 24)

<i>BER (%)</i> <i>Foreman</i>	Number of Embedded Bits									
	88		132		176		220		264	
<i>m-seq. length</i>	$2^{13}-1$	$2^{14}-1$	$2^{13}-1$	$2^{14}-1$	$2^{13}-1$	$2^{14}-1$	$2^{13}-1$	$2^{14}-1$	$2^{13}-1$	$2^{14}-1$
AWGN I	4.92	0.19	7.82	0.94	11.64	2.36	13.21	3.33	16.25	6.53
AWGNII	4.36	0.28	8.33	1.83	11.45	4.54	15.18	4.62	16.69	8.27
ITU H.263+	5.59	0.18	8.77	1.32	12.68	3.92	13.14	4.92	16.28	5.90

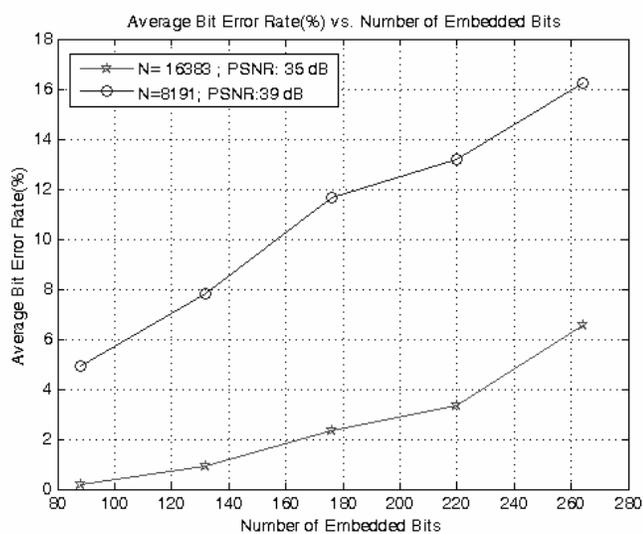


Figure 2.8 BER (%) vs number of embedded bits for different lengths of m-sequences after the AWGN I of noise PSNR around 13-14 dB.

In order to examine the maximum length of m-sequences that yields a visual distortion in the video, the length of the m-sequences is also increased to $2^{15}-1$, where 24 coefficients on the average are modified for watermarking in each block of $8 \times 8 \times 16$ pixels, where the

resulting PSNR is about 29 dB. A typical frame from watermarked *Foreman* is shown in Figure 2.9 (b). Some distortions are quite apparent on the white cap of the *Foreman*, as well as some flickering type of distortions in the video. The flickering is due to the fact that the modifications at the coefficients yield a visual distortion after 29 dB's, due to the visual coupling between different frequency components that can effect the invisibility of each other, although none of the modifications in each frequency coefficient is larger than the temporal sensitivity thresholds [50]. For viewing purposes, the original and watermarked video sequences for different m-sequence lengths are available at http://snap-mmrg.eee.metu.edu.tr/paper_ieee/capacity_tests.



(a)

(b)

Figure 2.9 Typical frames from the watermarked *Foreman* (a) $N=(2^{14}-1)$ PSNR \cong 35 dB and (b) $N=(2^{15}-1)$ PSNR \cong 29 dB

In the final step of the comparisons, the embedded watermark energy is again set equal for all the compared methods and the relation between the maximum number of embedded bits and BER is examined for a typical average attack around 36 dB. The resulting BERs for ITU H.263+ coding of 800 kbps, yielding PSNR of 36 dB, is plotted as a function of embedded bits in Figure 2.10 and the proposed method has the lowest BER.

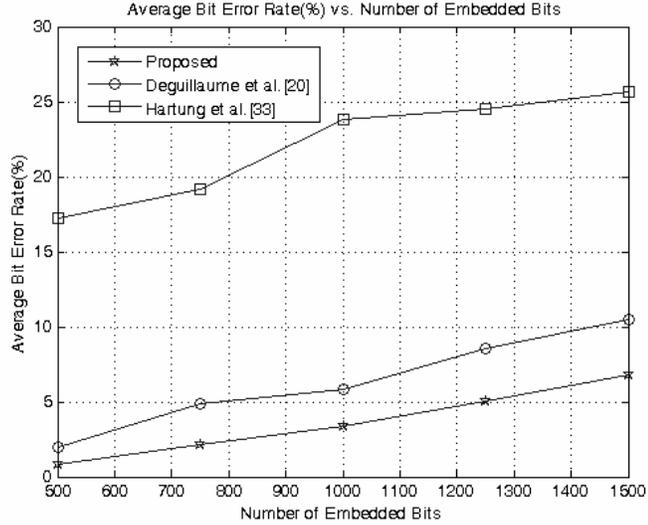


Figure 2.10 BER vs. number of embedded bits after ITU H.263 + coding of about 800 kbps for the compared methods. The watermark PSNR is 39 dB for all of the methods.

2.4 Conclusions

A novel oblivious video watermarking method is proposed based on the temporal sensitivity of HVS. The method embeds the watermark by exploiting the temporal contrast thresholds. The robustness results show that the watermarking scheme can survive typical video attacks, such as additive Gaussian noise, video compression and temporal synchronization loss. The results also clearly indicate that the proposed method is superior to the compared two other well-known methods in the literature, one of which is based only on the spatial sensitivity of HVS [33] and the other which implicitly uses HVS characteristics [20]. One may conclude that integrating temporal characteristics of HVS into a watermarking scheme increases the robustness against common temporal signal processing operations, in terms of bit error rate and mainly solves the imperceptibility problem, which usually shows itself as a flickering in video watermarking. The proposed method is clearly shown to be competitive or better against some respected methods from the literature.

The method is applied between shot boundaries in order to work with the temporally stationary signals. An extension to this method can also be proposed by using not only the temporal contrast but also the temporal masking characteristics of HVS, which corresponds to a significant decrease in the sensitivity of HVS at the shot boundaries. Moreover, the method can be further improved by embedding the watermark into chromatic components as well as the Y-component of the video.

CHAPTER 3

3D-3D GEOMETRY WATERMARKING

In this chapter, a detailed literature survey on 3D-3D geometry watermarking techniques is presented. Although most of the 3D watermarking methods in the literature belong to this category, the literature still lacks such a complete survey for 3D-3D watermarking. Therefore, rather than investigating new solutions and methods in this relatively mature category, the state of the art for 3D-3D geometry watermarking methods is analyzed in this thesis.

The chapter begins with a brief summary on fundamental 3D representations and a classification on 3D watermarking techniques. The general framework, specific requirements and problems of geometry watermarking is given in Section 3.2. The geometry watermarking methods are classified into two groups, as *spatial domain methods* and *transform domain methods*, according to the embedding domain of watermark in Section 3.3. The methods in each category are overviewed in Section 3.4 and 3.5, respectively. The comparisons and discussions are given in Section 3.6. Finally, Section 3.7 concludes the chapter.

3.1 3D Watermarking: Taxonomy

During the last decade, watermarking has been one of the most active research topics, attracting the interest of researchers with different backgrounds, such as signal processing, communication and information theory, cryptography, and computational vision. However, a great deal of this research effort has focused on digital watermarking of still images, video [1]-[4] or audio streams [62]. Accordingly, the watermarking technology for these media is on its way for reaching to a maturity. On the contrary, 3D watermarking can still be accepted to be immature, although there are many applications in which 3D content is utilized and consumed.

In general, the term, *3D watermarking*, in this chapter, implies the watermarking of any representation of a 3D scene. The watermarking technology aims to protect a 3D representation by embedding hidden data into the main components of a representation. Based on the main components constructing a scene representation, can be achieved as illustrated in Figure 3.1. While the representations in the left side of this plot are mainly dependent on the geometrical structure of the scene, the representation in the rightmost part is purely based on the images that are captured by the cameras appropriately located in the scene.

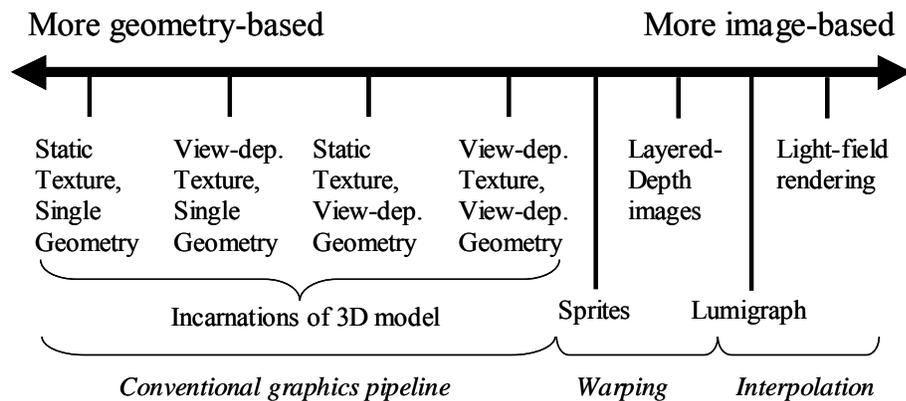


Figure 3.1 Categorization of scene representations according to [63]. Reprinted with kind permission of Springer Science and Business Media.

In the geometry-based representation, there are three fundamental components of a scene: *geometry*, *texture* and a *map* that defines the relation between the texture and the geometry. While forming an arbitrary view of a scene during the rendering applications, these three components are used accordingly in relation with the given lighting conditions of a scene. On the other hand, in the image based representation, a scene is represented by only the 2D projections of the scenes which are the images simply captured by cameras. The arbitrary view generation of a scene is achieved by using some special interpolation techniques on the original camera images.

Considering the dimensions of these basic components of each representation, the watermarking methods in the literature can be classified into three groups:

- 3D-3D (Geometry) Watermarking
- 3D-2D Watermarking
- 2D-2D Watermarking

The first pair of symbols identifies whether the watermark is embedded in the 3D model or a 2D projection of it, and similarly the second pair of symbols identifies whether the watermark is detected in the 3D model or a 2D projection.

In 3D-3D watermarking category, the watermark is embedded into the 3D geometric structure of an object used in a scene and tried to be extracted from the 3D geometry after any attacks on the geometry. The second category aims to embed the watermark into the 3D object and then, to extract the watermark from the resulting images or videos, obtained after projection of 3-D object into 2-D image planes. The last group embeds the watermark into the 2D projections of a 3D object and tries to extract the watermark from 2D renderings of the object. Table 1 summarizes the aforementioned categories in 3D watermarking.

Table 1 3D Watermarking categories, their protected components and typical attacks

	Protected Component	Typical Attacks
3D-3D Watermarking	Geometry Representation	Geometrical Transformations, Mesh operations (Compression, simplification, smoothing...)
3D-2D Watermarking	Geometry or Texture of 3D Object	Any processing on the texture, Geometrical distortions
2D-2D Watermarking	Images of 3D Object	Multi view Compression, Image-based Rendering Operations

3D-3D geometry watermarking is well studied in the literature on watermarking. However, a detailed overview is still missing in the literature. In this chapter, an extensive literature survey on geometry watermarking is presented beginning from a brief review on specific problems and requirements of geometry watermarking with a comparison to image and video watermarking, in the next section.

3.2 3D-3D Geometry Watermarking

Since geometry is the fundamental information source of a 3D scene in many related applications, the 3D watermarking research mostly focused on the geometrical information of the scene. Therefore, most of the methods in the literature belong to this group. A general scheme for the geometry watermarking is given in Figure 3.2. The watermark is imperceptibly embedded to the object and the watermarked object is delivered to the channel. The channel might include any distortion on the geometry that could happen due to any malicious or non-malicious utilization of the object. For instance, the model after the channel in Figure 3.2 is the translated, rotated, scaled and smoothed version of the originally watermarked model. Then, the watermark is attempted to be detected from the tested object at the extraction part. Depending on the extraction algorithm, the original model might not be required in some applications.

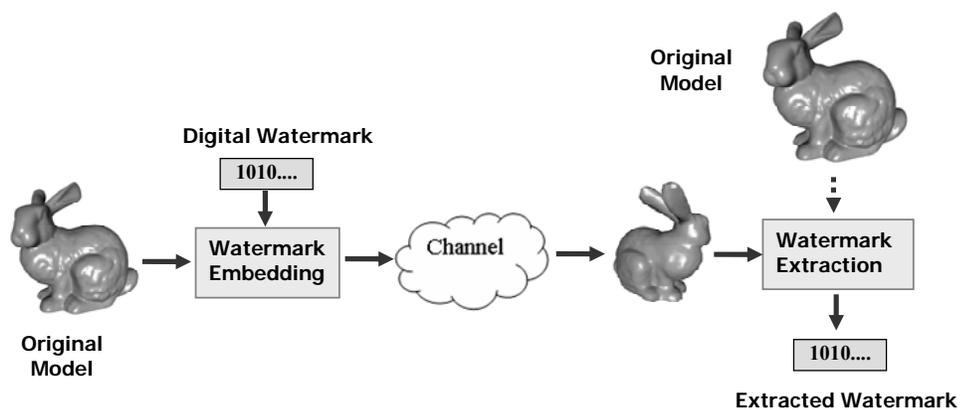


Figure 3.2 A general scheme for 3D-3D geometry watermarking

3.2.1 Requirements of 3D-3D Geometry Watermarking

Similar to the general requirements for the image and video watermarking, a watermarking system designed for the geometrical structure of a 3D scene or a 3D object should satisfy the requirements on *capacity*, *robustness* and *imperceptibility*.

Regarding the *capacity* requirement, the system should allow embedding of nontrivial amounts of data. In addition, a watermarking scheme should be able to distinguish between different watermarks with a low probability of error as the number of differently watermarked version of a digital content increases [9]. This requirement is critical, especially for public watermarking schemes, where a different watermark is embedded into each content for every licensor or buyer. Actually, the capacity requirement has no significant differences from those of image and video watermarking.

As a second requirement, a watermarking system for the scene geometry should be robust to all geometric or topological operations, as long as the visual quality of the geometric model is not severely degraded. The geometrical and topological operations on the model may include the following [42]:

- Rotation, translation, and uniform scaling
- Polygon simplification (often needed to achieve adequate rendering speed)
- Randomization of points
- Re-meshing (re-triangulation); generating equal shaped patches with equal angles and surface
- Mesh smoothing operations
- Cut (sectioning) operations—removing parts of the model as in backface culling
- Local deformations

In addition to these requirements, the distortions in the geometry due to the watermarking should be imperceptible. However, the imperceptibility requirement for a geometrical model is not a trivial problem; in fact, this requirement is more complicated, compared to that of image and video watermarking. The following section presents the problems of 3D geometry model watermarking, while fulfilling the mentioned requirements.

3.2.2 Fundamental Problems in 3D-3D Geometry Watermarking

The 3D geometry model watermarking has some specific problems due to its different type of representation, compared to image and video signals. These problems are tabulated in Table 3.2.

Table 3.2 Problems in 3D geometry watermarking compared to the image and video watermarking.

	Image and Video	3D meshes
<i>Representation</i>	Two and three dimensional functions on a manifold grid	Lack of unique representation, different mesh can represent the same surface
<i>Synchronization</i>	Scan line ordering	Requires a fixed orientation and a position of data in space
<i>Handling and Editing</i>	Well-defined standards for compression (JPEG, JPEG 2000, MPEG, H.26X,), Well-analyzed transformations used in transmission, compression, filtering. (DCT, FFT, wavelet etc.)	No well-accepted standard for compression, Multi-resolution representation techniques are comparatively new.
<i>Robustness</i>	Common signal processing during compression and transmission, synchronization attacks, cropping, etc.	A high number of the diverse attacks, including geometry transformations (translation, scaling, rotation, affine etc.), local modifications, topological changes.
<i>Imperceptibility</i>	Experimentally well studied perceptual analysis inherited from vision research to watermarking research.	No experimental study on the perceptual limits of the watermark for the geometry

According to this table, the first problem emerges in the synchronization of the watermark sequence to the 3D geometry data, since there is no implicit ordering of the data in 3D models. While the data in images and video frames are scan-line ordered, 3D model data, such as vertices, edges and faces, could only be ordered, after a fixed orientation and a position of data in space [42]. Therefore, most of the 3D watermarking methods in the literature first, convert the 3D model into a translation, rotation and scale

invariant domain, then, apply the watermarking scheme to the model. For example, in order to achieve translation invariance, center of the mass of the object is mostly chosen as the origin of the x -, y - and z -axes [64], [65]. The rotation invariance is satisfied by placing the 3D object into a location such that the principal component of the object coincides with the z -axis of the xyz -space. The scale invariance is usually achieved by normalizing r (radius) components of the vertices of the model such that the distance of the furthest vertices to the origin is equal to one [65]. It is obvious that an attack could also distort the main parameters of the object that are being used to achieve translation, rotation and scale invariance, which converts the watermark extraction into a more difficult problem.

Another problem in 3D model watermarking is the lack of uniqueness in the representation of the model data. Since it is possible to represent the same surface with different vertices, edges and faces, a scheme that embeds the watermark into the values of the geometric primitives of a model should also be able to extract the watermark from any other representations, as long as the visual quality of the model is not degraded. This approach makes the watermark embedding and extraction in 3D models a more complicated problem.

As mentioned before, another challenge in 3D geometry watermarking is to achieve robustness against a high number of diverse attacks. In addition to translation, rotation and scaling of the object, the general transformations, such as affine transformations, are also common [64], [65]. Furthermore, it is possible to make some local deformations to reshape a part of the model. Topological modifications, such as resection of a desired part of a model, mesh simplification or re-meshing, might also be applied to the model. A watermarking scheme for 3D geometry model should survive all such attacks, as long as the visual quality of the model is preserved at acceptable levels.

Handling and editing of the 3D geometry model are also more serious problems compared to image and video counterparts [42]. This case might involve complex geometrical and topological operations, as previously mentioned. It is also difficult to extend the common operations that are used in 2D signal processing for 3D content.

The final problem in 3D watermarking is related to the imperceptibility requirement. In image or video watermarking [4],[9],[38], invisibility is satisfied by using some explicit perceptual measures [66], termed as *just noticeable differences (JNDs)*, to determine the watermark embedding location and strength. *JNDs* are determined in vision research by setting some perceptual experiments on the cosine patterns at different spatial and temporal frequencies. If the method does not exploit this kind of perceptual metrics, at least, *PSNR* is used in order to make an assessment on the watermark energy and imperceptibility. However, in 3D case, setting of the perceptual experiments and the determination of *JNDs* is not a trivial problem. Furthermore, there is not a metric for general evaluation of invisibility and watermark energy in 3D processing, such as *PSNR* for the image case.

3.3 3D Geometry-Based Watermarking Methods

As depicted before, the main trend in 3D watermarking focuses on the watermarking of 3D geometry-based data, since the most valuable component of a scene in the majority of the 3D applications is its geometry. In fact, *geometry watermarking* in this chapter implies the watermarking of the geometry component of an object(s), residing in a scene.

An object could be represented by points, meshes or voxels. Among these representations, mesh-based representation is more commonly used in 3D applications. Hence, in watermarking research, 3D mesh watermarking is examined in more detail, compared to the other representations of 3D geometry. Therefore, the classification in this chapter is presented, considering mainly the watermarking methods that are applicable to 3D meshes. However, a few existing methods on 3D point and voxel representations are also presented in the given classification, accordingly.

In 3D mesh representation, an object is formed of geometric primitives, such as points, lines, polygons, polyhedrons and connected polyhedrons [43],[67] (see Figure 3.3). A watermark scheme could exploit the geometrical values of these primitives, such as coordinates of a point, length of a line, area of polygon, volume of a polyhedron, ratios of the areas of two polygons, two quantities that define a set of similar triangles, ratios of volumes of two polyhedrons, etc. Some examples of geometric primitives for a mesh

representation are shown in Figure 3.4. In addition to these geometrical primitives, it is also possible to embed the watermark by changing the topology of the 3D model. For instance, a watermarking scheme can use one of the two alternative ways of triangulating a quadrilateral or two different mesh sizes in order to embed a watermark bit of 1 or 0, as shown in Figure 3.5 [67]. While the methods exploiting the geometrical attributes form the majority of the geometry based methods, the approaches based on the topological properties of 3D geometrical model receives less attention in the literature.

The methods on 3D geometry watermarking can be divided into two main groups, based on the embedding domain of the watermark: spatial domain and transform domain methods. The general classification of 3D geometry watermarking methods is given in Figure 3.6. Spatial domain methods embed the watermark directly into the geometrical values of the geometrical primitives. In general, most of the methods in this group first extract perceptually significant geometric primitives from the model and then embed the watermark into those primitives by using a specially proposed method (Figure 3.7). The other class is the transform domain methods, where a 3D object is decomposed into subsignals by applying a 3D geometry based transformation, such as wavelet transformation [69] or mesh spectral analysis [70]. In this group of methods, after applying the transformation to the mesh model, watermark is embedded to the resulting transform coefficients (Figure 3.8). The members of each group are explained in detail in the following sections. Next, the advantages and disadvantages of each group are presented.

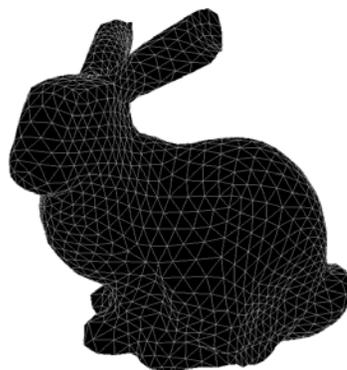


Figure 3.3 An example for a mesh representation: *Bunny* mesh [68]

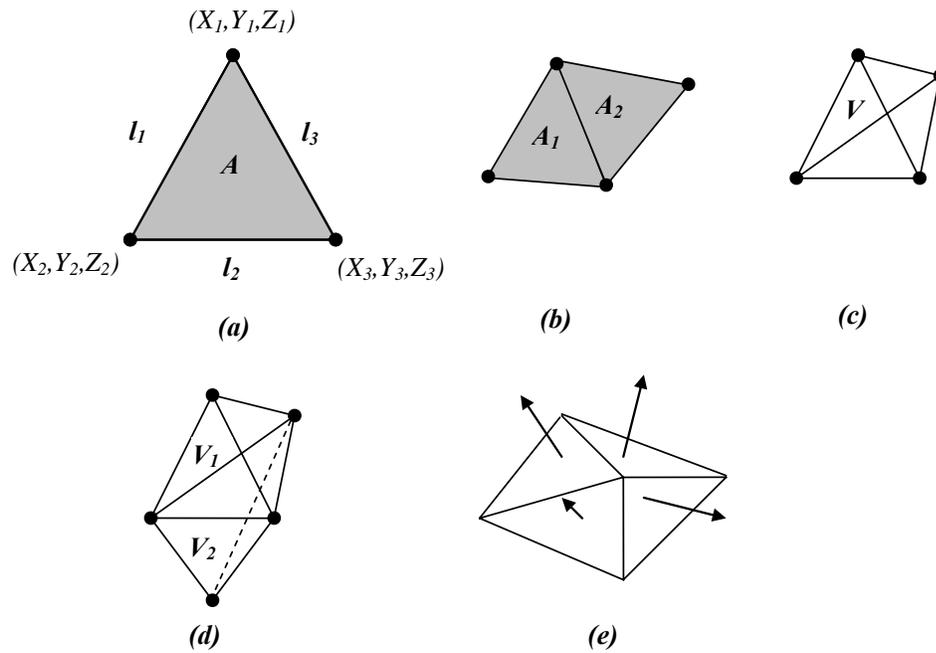


Figure 3.4 Some of the geometric primitives of a mesh model that could be exploited for watermarking. (a) Coordinates of the vertices (X, Y, Z) , length of the lines (l) , area of a triangle (A) , (b) Ratio of the areas of two triangles (A_1/ A_2) , (c) Volume of a tetrahedral (V) , (d) Ratio of the volumes of two tetrahedral (V_1/ V_2) , (e) Normal vectors of the triangles forming the mesh.

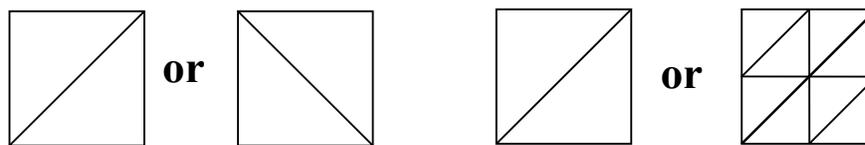


Figure 3.5 Some examples of alternative topological structures for watermarking [67].

Reprinted with permission from IEEE. Copyright 1998 IEEE.

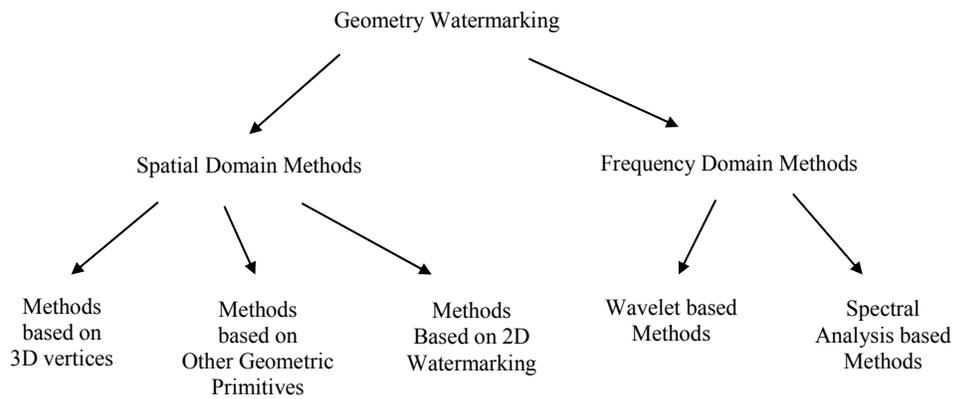


Figure 3.6 Categorization of 3D geometry-based watermarking

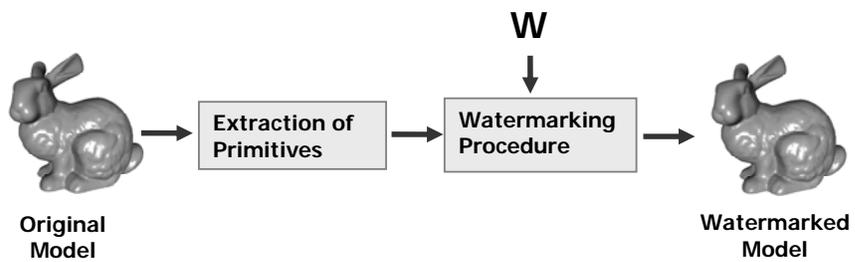


Figure 3.7 A general scheme for the spatial domain methods

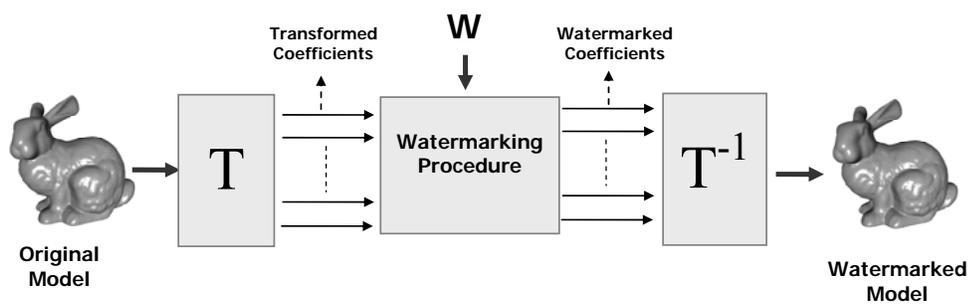


Figure 3.8 A general scheme for the transform domain methods

3.4 Spatial Domain Methods

Spatial domain methods could be categorized into 3 groups (Figure 3.6). While the first group basically embeds the watermark into the vertex coordinates of a 3D mesh model, the second group exploits the other geometric primitives of a 3D mesh before adding the watermark. The third category follows a different strategy by means of utilizing the 2D image watermarking methods for watermarking of the meshes, after transforming the 3D geometry into a two dimensional form. Although there is quite limited number of representatives for this group in the literature, it is still categorized in a different category due to its originality.

3.4.1 Modifying Vertex Coordinates

The methods of this group apply modifications on the vertex coordinates of a 3D mesh model to embed the watermark. While some of the approaches directly add the watermark into the vertex coordinates (Figure 3.9 (a)), some others change the position of a vertex, with respect to the watermark bit, into one of the two predefined regions around the location of the vertex (Figure 3.9 (b)). These different approaches are examined in the following sections.

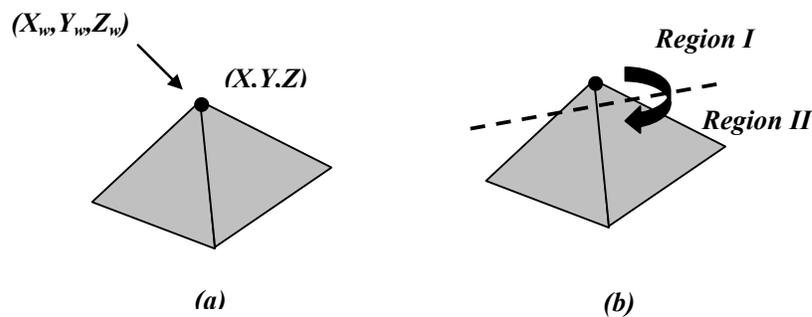


Figure 3.9 Two different approaches to modify the vertex coordinates for watermarking: (a) adding a watermark pattern (X_w, Y_w, Z_w) into the coordinates of the vertex directly, (b) moving the vertex into one of the predefined regions around the vertex with respect to watermark bit.

3.4.1.1 Embedding Watermark directly onto Vertex Coordinates

The first method in this group [71] modifies the vertex coordinates of a 3D model by modulating the watermark signal with a global scaling factor and a masking weight and then, adding to the coordinate values. The masking weight for each vertex is determined by taking the average of the differences between the positions of the connected vertices to that vertex. Since the locations, where sharp variations occur, contains the perceptually significant part of a 3D model, such a masking weight can give perceptually acceptable results. In fact, more deliberate masking weights should be exploited by using well defined edge detectors which are tailored for 3D geometrical models. The method is shown to be robust to the additive random noise, compression by MPEG-4 SNHC and mesh simplification [71]. The advantage of the method is in its simple implementation, compared to other spatial domain methods.

In a similar method [72], the watermark is again embedded in an additive manner to the coordinates of the vertices. The scheme also varies the strength of the watermark signal adaptively with respect to the local geometry and embeds the watermark information into the model by modifying the distance of the vertices to the centre of the model. The method distributes the information corresponds to a bit of the watermark over the entire model via vertices scrambling. Such a scrambling of the watermark among the entire model yields a more resistive scheme against cropping attack, as well as mesh simplification and additive noise, where most of the transform domain methods fail.

In [65], the watermark is embedded into the radius component of the vertices in spherical coordinate system. In this technique, the 3D object is first translated into the new coordinate system, so that the center of mass coincides with the origin of axes. Then, the vertices of the object are converted to the spherical coordinates, as (r, θ, φ) . All components of r_j ($j = 1, 2, \dots, N$) are sampled into K bins with the uniform distance by

$$Q(r_j) = INT((r_j - r_{min}) / (r_{max} - r_{min})) \times K + 0.5, \quad (3.1)$$

where K is selected according to the watermark bit number. r_{max} and r_{min} , correspond to the maximum and minimum value of the r -component. Sample means, $E_k(r)$ of $r = Q^{-1}(k)$, in each of the sampled bins, are calculated and arranged in the descending order,

according to the density of vertex per bin. The binary watermark, w_k , is embedded into the sample means in the descending order as:

$$E_k'(r) = (1 + \alpha R_k) \times E_k(r). \quad (3.2)$$

If $w_k = 0$, then R_k is chosen as -1. Otherwise, R_k is set 1. The parameter, α , is the embedding strength. In order to extract the watermark without the original mesh, the indices of bins into which the watermark is embedded, and sample mean $E_k(r)$ before watermark embedding, are sufficient. The critical point in the embedding process is the parameter, α . In order to satisfy the trade off between imperceptibility and robustness, the mesh model is projected onto two constraint convex sets, (one for robustness and the other for imperceptibility), and the range for parameter, α , is determined according to these convex sets. In the detection scheme, the watermarked object passed from the same operation up to the determination of sample means, E_k^w . Then, E_k^w 's are compared to E_k at each bin to extract the watermark bit. The method shows robustness to scaling and rotation without the use of the original object in the detection stage. However, in the case of affine transformation, the method requires the original object to determine the necessary transformations before the extraction process.

3.4.1.2 Modifications on Position of the Vertices w.r.t Watermark Bit

The methods in this approach make changes in the position of the vertices to satisfy some requirements specially defined for each watermark bit. For instance, one of the methods in this group [73]-[74] enforces some limitations on the probability distribution of the vertices, surrounding a vertex to embed the watermark to that vertex. The first step in the method is to translate the 3D object such that the center of the mass corresponds to the origin of the axes. Then, in order to achieve rotation invariance, the object is rotated so that its principal component, u , coincides with the z-axis. Next, the model is converted into spherical coordinates and the watermark is embedded to the r -component (i.e. the distance of a vertex to the origin of the axes) for the purpose of scale invariance. In order to achieve robustness against mesh simplification, every watermark sample is embedded to a set of vertices instead of one vertex.

In the method, r - Θ plane (spherical coordinates) is decomposed into subranges, Θ_j , and a random variable, $d_r(u_i^s)$, is formed by using the r component of the vertex, u_i^s , where u_i^s denotes the spherical coordinates of the vertice, u_i as,

$$d_r(u_i^s) = r(u_i^s) - H(u_i^s). \quad (3.3)$$

H is a local neighborhood operation of the vertices around u_i^s . H satisfies an approximation function of $r(u_i^s)$ that depends on the neighborhood of u_i^s . The operator H is chosen so that $d_r(u_i^s)$ follows a Gaussian distribution with variance σ^2 , and zero mean. In the method, the distribution of $d_r(u_i^s)$ is modified in each subrange, Θ_j , according to the watermark bit. If the watermark bit is equal to 1, r -component of the some of the vertices, v_i^s , that have $d_r(v_i^s) > b\sigma$, is altered in order to fall inside $(0, b\sigma)$. If the watermark bit is -1, r -component of the some of the vertices, v_i^s , such that, $d_r(v_i^s) < -b\sigma$, is altered in order to fall inside $(-b\sigma, 0)$. This operation changes the probabilities, $prob(d_r(v_i^s) > b\sigma)$ and $prob(d_r(v_i^s) < -b\sigma)$, in each subrange Θ_j of the watermarked 3D object. In the original object, these probabilities are equal to $G(-b)$, where G shows the error function, *erfc*, for a Gaussian distribution of variance σ^2 , and zero mean. In the watermarked object, these probabilities will be smaller than $G(-b)$, which is the main logic during watermark detection. The probability, $prob(d_r(v_i^s) > b\sigma)$, is computed by means of calculating the ratio of the number of vertices, that satisfy $d_r(v_i^s) > b\sigma$, to the number of the vertices in the model. The probability, $prob(d_r(v_i^s) < -b\sigma)$, is calculated in a similar way. The method is claimed to be the first blind 3D object watermarking algorithm (i.e. does not require the original mesh during the detection process) robust to translation, rotation, scaling and mesh simplification. However, one of the typical attacks, cropping, is not considered in the robustness tests.

Another method changing the position of the vertices for watermark embedding is proposed in [75]. In this method, the vertices are located in one of the two predefined regions around the selected vertex according to the watermark bit. These regions are

determined with respect to the local moments of the vertices in the neighborhood of the selected vertices for watermarking. The algorithm consists of two steps [75]. In the first step, a chain of vertices and their neighborhoods are selected and ordered. The most appropriate vertices for watermarking are those from mesh areas consisting of small polygons in the 3D object. Such regions are equivalent to the image regions in image with texture, details or noise that has been considered appropriate for image watermarking. In order to ensure the imperceptibility of the watermarks, modifications can be produced only in the regions consisting of small polygons. A threshold, $T(V_i)$, depending on the distance $D(V_i)$, which is the distance of a vertex, V_i , to its neighborhood vertices, is used in order to select those small areas for watermarking.

In the second step of the method [75], locations of selected vertices are changed according to their local neighborhood moments and the embedded information bit. The watermark code is embedded into a set of B vertices and their neighborhoods. Two separate geometric areas are considered in the space defined by the set $\{V_i, N(V_i)\}$, one for embedding a bit of 0, and the other for embedding a bit of 1. The watermarked vertex is moved into one of the two regions, according to corresponding watermark bit. The method defines two parallel planes by using the geometry of its neighborhood, $N(V_i)$, in order to determine the separate geometric areas. In a recent work [76], the method is improved by means of defining bounding ellipsoids, instead of parallel planes, for each watermark bit. In the detection stage, the same procedure for vertex selection and ordering is applied to the test object and the watermark is retrieved by determining the region where the vertex is placed. The method in [76] makes a comparison between this new approach and the method in [75], and indicates the superiority of the approach defining bounding ellipsoids to the parallel plane approach in terms of imperceptibility and robustness. The method shows robustness to the common attacks, such as rotation, scaling and other affine transformations changing the vertex order. Especially, robustness against cropping attack is one of the exceptional advantages of the method.

In [77], [78] a new 3D watermarking technique, based on a *generalized Radon transform* (GRT), is described. It should be noted that the (GRT) is not a transformation from a signal processing perspective, as commonly understood. It is a variation of the radial integration transform (RIT) [78], namely the cylindrical integration transform (CIT),

which simply integrates the information of a 3D model on cylinders, beginning from its centre of mass.

Figure 3.10 illustrates the computation of the CIT for model's vertices: the dots indicate the model's vertices, the line segments L_i indicate the lines, which end on the surface of the bounding unit sphere and the cylinders CYL_i indicate the cylindrical integration area. The proposed watermarking technique embeds a specific model identifier, which represents the 3D model's descriptors, into the vertices of the 3D model via modifications on their locations. The method is robust to geometric distortions such as translation, rotation and uniform scaling. Additionally, the scheme is robust against any vertex reordering. However, the method is vulnerable to attacks, such as mesh smoothing operations, cropping and local deformations, since such operations change the shape of 3D model, which then cannot be used as a query model.

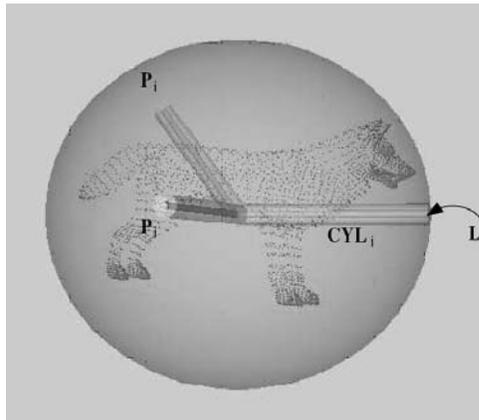


Figure 3.10 Cylindrical integration transform (CIT) [78]. Reprinted with permission from IEEE. Copyright 2004 IEEE.

Other than these methods proposed for copyright protection, there are also some fragile watermarking methods which are focusing on the authentication of integrity of 3D meshes [79]. The first method [79] in this category embeds a fragile watermark by iteratively perturbing vertex coordinates until a predefined hash function applied to each

vertex matches the other predefined hash function applied to that vertex. Since their algorithm relies heavily on an ordered traversal of vertices, it is capable of detecting object cropping [80]. However, authentication after local modifications, vertex reordering and some particular attacks, such as floating point truncation or quantization are not taken into account in the method. In [80], the aforementioned problems are handled. The proposed method could not only achieve localization of malicious modifications, but also robust to certain incidental data processing, such as quantization of vertex coordinates and vertex reordering.

3.4.2 Methods Based on other Geometric Primitives

The methods in this class perform the changes in the geometric quantities of a 3D model other than the coordinates of the vertices. The method in [42], which is one of the prior and well-known methods on 3D model watermarking, embeds the watermark information into the surface geometry. The scheme first maps surface normals onto a unit sphere, and then subtly alter groups of similar normals in order to embed the watermark bits.

It should be noted that possible operations on a 3D model, such as re-meshing, mesh simplification, cropping, randomization of points and other similar attacks, might cause substantial changes in model vertex and face set configuration, adjacencies, and topology without changing the perceived quality of the model. In other words, there exists nearly an infinite amount of meshes representing or approximating one particular surface. Therefore, the watermark should be embedded into some other geometric primitives of the model that should not be affected from the aforementioned attacks as long as the visual quality of the model is not degraded. The idea in [42] is to use collections of surfaces as an embedding primitive. If the representation of a model changes due to these operations, the new vertex face set configuration should maintain global surface characteristics, regarding size, orientation, and curvature, at least of perceivable features. Otherwise, a significant loss of visual quality should occur. Embedding and retrieval process consist of several stages, as illustrated in Figure 3.11.

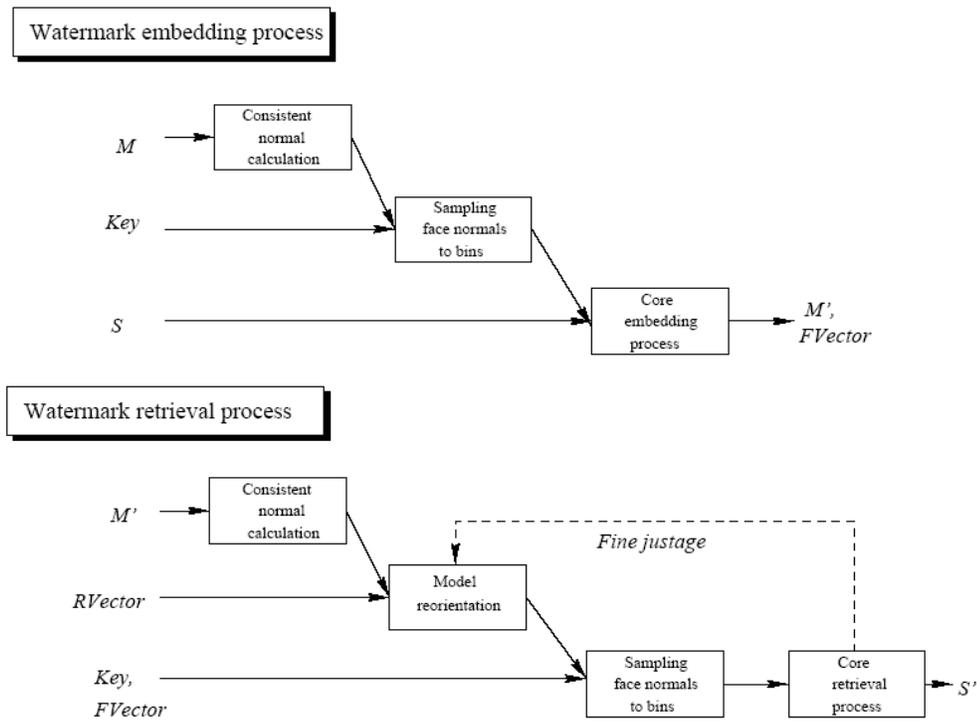


Figure 3.11 The dataflow between stages in embedding and retrieval process in [42].

Reprinted with kind permission of SPIE.

In this method [42], the embedding process takes an original model M , a key and a bit string S of length N as input. Next, the system calculates *consistent* face normals from the actual face normals of M . The original model M with consistent normals or parts of it, denoted as $RVector$ in the illustration, is needed to be stored. From the key, N non overlapping bins, bin centers and radii, are derived. In the following step, the consistent face normals are sampled by using these bins. The core embedding process now modifies model vertices in order to change bin contents (normals) with respect to certain measures, called *feature types*. The output of embedding process are the model M' , which is denoted as *watermarked copy* of M , in which the bit-string S had been embedded, the N original feature values, called $FVector$, which are needed as reference values in the retrieval process and the mentioned $RVector$.

The retrieval process in [42] takes the watermarked model M' , the key, the feature vector $FVector$ and certain additional original features, denoted as $RVector$, as input. First consistent normals are calculated from the actual face normals of M' . Next, M' is reoriented with respect to the original model, M . For this process the $RVector$ information is required. Then, the bins are constructed from the key (or alternatively the center and radii information is contained in the $FVector$). In the core retrieval process, the actual feature values are calculated and compared against those in $FVector$ which yields the bit string S' as a result. If S' differs from S , and a constant part of S indicate that actual reorientation accuracy is insufficient, further orientations are tested until S' matches the constant fraction of the bit string S , sufficiently.

With this system, Benedens [42] primarily aims robustness against:

- Randomization of points
- Mesh altering (re-meshing) operations or attacks
- Polygon simplification

However, the method demonstrates robustness against only simplification attacks [42]. Moreover, one drawback of the algorithm is the large amount of a priori data needed, before watermark retrieval. A more relevant drawback is the amount of preprocessing needed before the watermarking core algorithm can be applied.

Another scheme exploiting the normal vector distributions of surface patches of a 3D model is proposed in [81]. The method embeds the watermark into the consistent normal vectors. In order to satisfy robustness against the partial geometric deformation, equal number of watermark bits is embedded to each patch of subdivided mesh. The method uses EGI distribution of Benedens' algorithm [42] to achieve robustness against simplification and remeshing attacks. The main improvements against [42] are the robustness of the method against cropping attack by means of spreading the watermarks into different patches and exploiting the normal vector distribution of each patch instead of the normal distribution of the entire model, which is the case in [42].

Another well-known method in this group is proposed in [43], [67]. The paper presents mainly three watermarking methods, namely, *triangle similarity quadruple algorithm*

(*TSQ*), *tetrahedral volume ratio embedding (TVR)* and *mesh density pattern embedding algorithm (MDP)*. In the first algorithm, a pair of dimensionless quantity, such as a pair of angles in a triangle or the ratio of the lengths of the adjacent sides of a triangle and the ratio of the height of a triangle to its base, is used as the geometrical embedding primitive to watermark triangle meshes. Since these dimensionless ratios are invariant to rotation, translation and uniform scaling, the watermark is robust against those operations. In addition, the watermark also survives resection and local deformation, since subscript arrangement to determine the watermark embedding locations and repeated embedding are utilized during watermark insertion. In the second method, the ratio of the volumes of a pair of tetrahedrons is selected as the embedding primitive. Since the ratio of the volumes of two polyhedrons is invariant to affine transformation, the watermark also survives from affine transformation, as well as the aforementioned attacks in the first method. The tetrahedral volume ratio (TVR) algorithm, show near optimal properties with respect to capacity, execution speed, and monitoring capabilities. The significant drawbacks include vulnerability to re-meshing operations, polygon simplification, and point randomization. Nevertheless, their algorithm is well-suited for embedding public watermarks. The third method, *Mesh Density Pattern Embedding (MDP)*, first tessellates the given curved surfaces, and then, embeds a visible pattern by modulating the sizes of triangles in the output mesh. This simple method survives from any type of geometrical transformations. However, it is vulnerable to a re-meshing attack that generates patterns with mostly identical shapes (angles and size).

3.4.3 Methods based on 2D (Image) Watermarking

In the last category, a different method based on 2D image watermarking is presented. Although, there is only one representative of this group in the literature, it is categorized as a different group, due to its novelty. The proposed method [64], first extracts a 2D image from a 3D model and then, exploits a DCT-based image watermarking algorithm instead of the usual 3D watermarking methods, which mainly embeds the watermark by performing slight modifications on the vertex coordinates of the 3D models. Extracting a 2D image from a 3D model also allows using any other image watermarking algorithm during 3D watermarking.

In the extraction process of the 2D image, a cylinder, denoted as *scanning cylinder*, is placed around the object. The radius and height of the virtual cylinder are set such that the 3D model slightly resides inside the cylinder. As a result, the cylinder undergoes the same amount of uniform scaling as the 3D model does, which achieves robustness to scaling attack. Then, $N_v \times N_u$ grid on the side face of the cylinder, as shown in Figure 3.12, is constructed. The grid points, where the ranges are calculated, are denoted by $(u_r, v_r) \in \{(0,0), \dots, (N_v - 1, N_u - 1)\}$. The vertical line ($v_r = 0$) lies in the direction of s_3 and v_r increases along the counterclockwise direction.

Figure 3.13 shows the general situation of virtual ranging in [64]. From (u_r, v_r) on the grid, which corresponds to \mathbf{q}' in (x,y,z) coordinate, a line is drawn towards the point \mathbf{q} on the opposite side of the cylinder. The triangle \mathbf{abc} represents one of the triangular faces which intersects the line segment $\mathbf{q}'\mathbf{q}$. Note that the triangle \mathbf{abc} is facing the point \mathbf{q}' , i.e., the order of vertices is $\mathbf{a} \rightarrow \mathbf{b} \rightarrow \mathbf{c}$. The l is the range value that is to be found for every (u_r, v_r) on the grid. The corresponding 2D image of l values is linearly mapped to $[0,255]$ and a DCT based algorithm is used for watermarking. The method is robust against mesh simplification and noise.

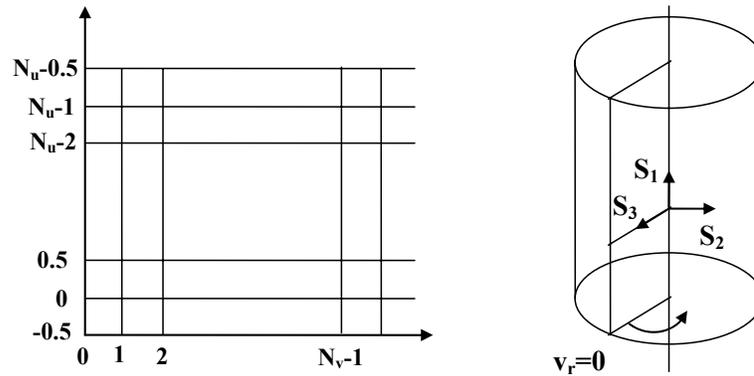


Figure 3.12 2D grid on the side face of the scanning cylinder [64]. Reprinted with permission from IEEE. Copyright 2002 IEEE.

3.5.1 3D Geometry Watermarking based on Spectral Analysis

Spectral methods embed the watermark into the resulting signals after a transformation is applied to the model in order to decompose the mesh model into low frequency and high frequency contents. The transformations are constructed by using a set of orthogonal basis defined over the entire 3D mesh model. Most of the methods in this group embed the watermark in an additive manner after modulating the watermark with a local scaling factor in the direction of the orthogonal basis. The differences between the methods are in the selection of the orthogonal basis for the construction of the transformation.

In the first method of this group [82], a technique derived from *progressive meshes* [89], is utilized to construct the orthogonal basis over the meshes. *Progressive meshes* method is one of the multi-resolution surface representations that share similar properties with the frequency-based transformations on the images. The representation automatically determines the perceptually significant parts of the surface. In the proposed method, the original mesh is first converted into a coarse base mesh and a sequence of refinement operations by using the techniques in *progressive meshes*. Then, the method defines the orthogonal basis for each of these refinements over their corresponding neighborhood in the original mesh. The watermark is added to the 3D coordinates of the mesh vertices, after it is multiplied by the orthogonal basis and a global scaling factor, adjusting the watermark strength. In this way, the spread-spectrum principles that are used in image watermarking, are adapted to embed information into the basis functions corresponding to perceptually significant features of the model. The method in [82] also proposes a solution to another challenge in 3D watermarking, namely, extracting the watermark after the attacks that modifying not only the vertex positions, but also the structure of vertex sampling. In order to address this challenge, an optimization technique to resample the attacked mesh using the original mesh connectivity is developed. With the proposed technique, the robustness against mesh simplification and other operations that preserve shape is obtained. In addition, the method also survives from classical attacks on the 3D meshes, such as smoothing, additive random noise, similarity transformation and other attacks, due to the spread spectrum principles used during watermark embedding.

In another method based on the spectral analysis [70], the orthogonal basis for the transformation is selected as the eigenvectors of a *Laplacian* matrix derived from

connectivity of polygonal meshes. The method first forms the *Laplacian* matrix, whose elements are determined by using only the number of connections of each vertex to the other vertices in the model. Then, *eigenvalue* decomposition is applied to the *Laplacian* matrix to compute the mesh spectra. The decomposition produces a sequence of eigenvalues and a corresponding sequence of eigenvectors of the matrix. While spectral coefficients of the smaller eigenvalues represent global shape features, spectral coefficients of the larger eigenvalues represent local or detail shape features. Projecting the coordinate of a vertex onto a normalized eigenvector produces a mesh spectral coefficient of the vertex. The watermark is embedded into the mesh spectral coefficients by exploiting a spread spectrum approach similar to the previous approach [82]. However, rather than using the randomly generated numbers from a Gaussian distribution as a watermark, as in [82], an information sequence of 0 and 1's are embedded to the model. Due to the spread spectrum techniques utilized during watermark embedding, this method is also robust against similarity transformations (translation, rotation and scaling), random noise added to vertex coordinates, mesh smoothing and partial resection of meshes. However, one lack of the method compared to [82] is the weakness against the connectivity change that can occur, during remeshing or mesh simplification, since mesh spectral analysis depends on the connectivity of the mesh. Another disadvantage of the method is the high computational cost of the numerical method used for the spectral analysis. Such a high computational cost precluded the analysis and watermarking of mesh models having more than a few thousand vertices.

In a recent version of this approach by the same researchers [83], the method in [70] is improved to handle the mentioned problems. In order to provide robustness against mesh connectivity, *mesh alignment* and *remeshing* steps are included to the system before the watermark extraction is applied to the tested mesh. *Mesh alignment* is achieved by minimizing the distance between the surfaces of the watermarked mesh and the tested mesh. In the *remeshing* step, geometry of the tested mesh is resampled by using the connectivity of originally watermarked mesh. The second improvement of the method is in the robustness of the method against attacks that combine cropping with the geometric transformations, mesh simplification, smoothing, and other interferences. This is achieved by means of *per patch alignment*, instead of *per model alignment*. In the watermarking process, the model is first decomposed into suitable patches into which the

watermark is embedded repeatedly. Such an alignment with respect to each patch enabled extraction of watermarks after cropping followed by geometric transformation. The last improvement of the method is achieved in the computational cost. By using an efficient method, called *Arnoldi* method, in the eigenvalue decomposition of the *Laplacian* matrix, the performance of the method is increased more than ten times. Such an achievement made it possible to analyze a much larger mesh region for shape features to be modified for watermarking.

A similar technique to [70] also uses the eigenvectors of the *Laplacian* matrix to define the transformation applied to the 3D model [90]. The difference of the method is in the watermarking embedding scheme. Instead of using a spread spectrum based additive watermarking technique, the method uses a substitutive scheme which spreads the watermark over the three spectral axes after the transformation. Given that the (P_i, S_i, R_i) is one set of spectral coefficients obtained after the transformation is applied to the model, the method changes the middle coefficient of the set such that it becomes closer to the minimum or maximum coefficient of the set according to the watermark bit. The scheme is robust against spectral compression, random noise added on the vertices coordinates and other common geometrical transforms, such as translation, rotation and scaling. However, the watermark fails after the attacks changing the connectivity of the model.

The last method in this group is based on the spherical harmonic transformation [91]. First, the method maps the coordinate information of 3D meshes to a unit sphere and then applies spherical harmonic transformation, which can be interpreted, as a combination of Fourier transform and latitude transform on the vertical and horizontal angles of the spherical coordinates, respectively. The watermark is embedded to the resulting transformation coefficients in an additive manner. The method is proved to be robust against noise addition, filtering, enhancement, rotation, translation and resampling with no need of mesh alignment and remeshing. One advantage of the method compared to the previous methods can be denoted as not requiring a remeshing operation to provide robustness to resampling. The method is also robust to cropping, vertex reordering and simplification with preprocessing. However, the method requires additional processing in

order to decrease the serious distortion resulting from the spherical harmonic transformation, which increases the computational complexity of the method.

Other than these methods watermarking the mesh representation of a geometric model, there are also some methods for the watermarking of point representations. In [92], a watermarking system is proposed for 3D models represented by unstructured clouds of point samples. The algorithm is similar to the previous work proposed by Ohbuchi [70], [83] by means of adapting the specific parts to point clouds. In a preprocessing step, the model is decomposed into a set of disjoint patches as in [83]. Each patch is transformed into the frequency domain, where the watermark is encoded into the spectral coefficients. The final watermarked model is then obtained by applying an inverse transform to each patch. The proposed method is of non-blind type, which requires the original model during the watermark extraction performed on a given point cloud. During the extraction, the original object and the potentially marked test object are aligned by using a registration process. Then, the surface of the test object is resampled with the resolution of the original object. After a frequency transformation, the watermark is extracted using both the resampled test geometry and the original object. Then the correlation is computed between the extracted and original watermark.

Finally, an algorithm for watermarking of 3D volume data, based on spread spectrum technique, is proposed in [93]. It is important to note that a 3D volume data is a 4D signal with three coordinates corresponding to 3D position and a value of the signal at that position. In fact, the watermark component in this method is the intensity information of the volume. The method is robust and invisible in the sense that 2D rendered image of this watermarked volume is perceptually indistinguishable from that of the original volume. The method is different from the previous methods due to the transform domain where the watermark is embedded. In the method, 3D DCT of the volume data is computed and the watermark is added similar to the spread spectrum watermarking method [7]. The method is robust against many attacks, such as geometrical distortions, addition to constant offset to voxel values, addition of Gaussian and non-Gaussian noise, low pass, high pass and median filtering, local exchange of voxel slices etc.

3.5.2 3D Geometry Watermarking based on Wavelet Transformation

The second group in transform domain methods is based on *wavelet* transformation [94]. Similar to the spectral analysis based methods, this group of methods applies a wavelet transformation which decomposes the 3D mesh into the subsignals in different resolutions and then, embeds the watermark into each resolution coefficients by using spread spectrum techniques. The difference of this group from the previous group is the use of wavelet transformation, which represents a mesh model in a more compact and natural form consisting of multiple resolutions.

The first method [69] in this group is specially proposed for robustness against affine transformation. The method first decomposes a 3D polygonal mesh by using *lazy wavelets* induced on 3D polygonal meshes. Then, the watermark is added to some geometric measures on wavelet coefficients which are invariant to affine transformation. During the embedding process, a mechanism to limit the maximum geometric distortion on the vertices is also utilized to control the imperceptibility of the watermark. The watermark embedding process is illustrated in Figure 3.14. During detection, both the watermarked and original meshes are passed through the wavelet transformation and the difference between these wavelet coefficients are calculated for detection (Figure 3.15). The scheme is shown to be resistant against affine transformation, partial resection, and random noise added to vertex coordinates, and other attacks. As a limitation, the method requires the mesh to have 1-to-4 subdivision connectivity. In addition, robustness against remeshing attack is not handled in the method.

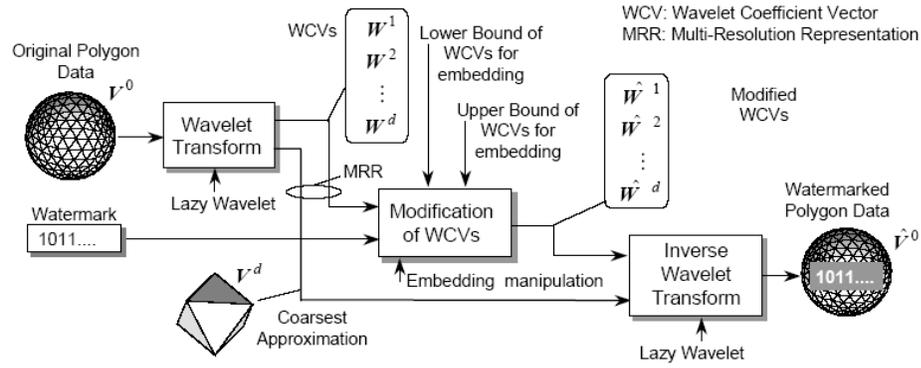


Figure 3.14 Watermark Embedding Process [69]. Reprinted with kind permission of Springer Science and Business Media.

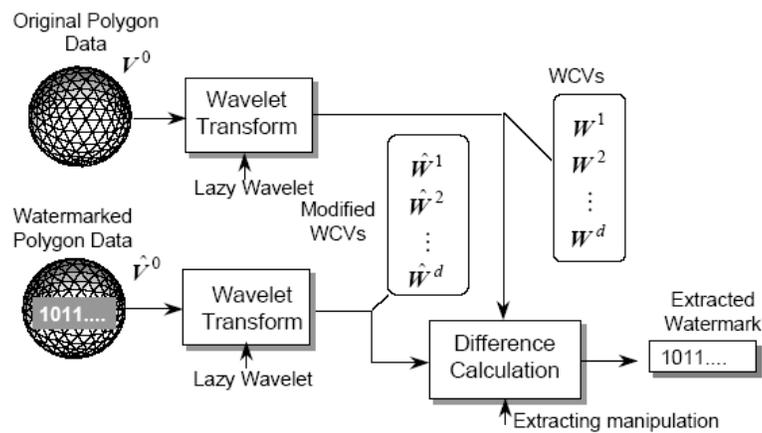


Figure 3.15 Watermark Extraction Process [69]. Reprinted with kind permission of Springer Science and Business Media.

The advantages of [69] can be summarized as follows: First of all, the area where the watermark is embedded can be easily determined by selecting the largest wavelet coefficient vectors, so that the embedded data can be perceptually invisible. Secondly, the embedded watermark can be spread into various resolution levels. Hence, the localization

of the watermark at high resolutions provides the ability to identify the distinct region of the watermarked polygon data, on which the local modification is achieved, while the global spreading of the watermark at low resolutions makes the embedded watermark invariant for the local modification on the geometry. This strategy makes the watermark more robust than the one in the spatial domain method. Thirdly, from the property of the wavelet decomposition, the watermark can be embedded not only into the original polygon data, but also into the approximated polygon data at several higher resolution levels only by executing the one embedding process. Finally, the method based on the wavelet transform by using the lazy wavelet enables to define a clear geometrical relation between the overall geometric tolerance and the upper bound of modification on each wavelet coefficient vectors. Therefore, the geometric error can be easily controlled between the watermarked polygon data and the original one.

Another method based on multi-resolution processing is proposed in [86]. The method is based on a multi-resolution decomposition of polygonal mesh shapes developed by Guskov *et.al.* [95] which separates a mesh into detail and coarse feature sequences by repeatedly applying local smoothing combined with shape difference. After the decomposition of the model, the watermark is added into the resulting coefficients after a multiplication with a local and a global factor as in the spread spectrum based methods. The method has shown good robustness against vertex reordering, noise addition, simplification, filtering and enhancement, cropping etc. In addition, the watermarking algorithm integrates nicely with the other signal processing tools developed by Guskov *et.al.* [95] which also constitutes a generic applicable multi-resolution framework in which other proposed techniques fit in as well. However, as most of the other transform domain methods, the scheme requires a registration and re-sampling stages before watermark extraction to bring the attacked mesh model back into its original location, orientation, scale, topology and resolution level. In addition, the method requires the original mesh to detect the watermark

In [87], a blind watermarking technique which exploits a class of 3D wavelets based on subdivision surface is presented. The method assumes that the mesh to be decomposed by wavelet analysis is a semi-regular mesh, which is obtained by regularly subdividing an irregular mesh. The method decomposes the 3D mesh into sublevels as illustrated in

Figure 3.16. The watermark is added to each wavelet coefficients at a suitable resolution level after a multiplication with a local factor. The overall scheme of the watermarking process is given in Figure 3.17. Watermark detection is accomplished by computing the correlation between the watermark signal and the tested mesh. Robustness against geometric transformations, such as rotation, translation and uniform scaling, is achieved by embedding the watermark in a normalized version of the host mesh, obtained by means of Principal Component Analysis. The method is shown to be robust against additive noise, low pass filtering, translation, rotation, scaling and cropping. However, the robustness against remeshing operation is not handled in the method. The method is extended in [88] by means of using a roughness measure to improve the imperceptibility.



Figure 3.16 Representations of a 3D model in different wavelet sublevels.

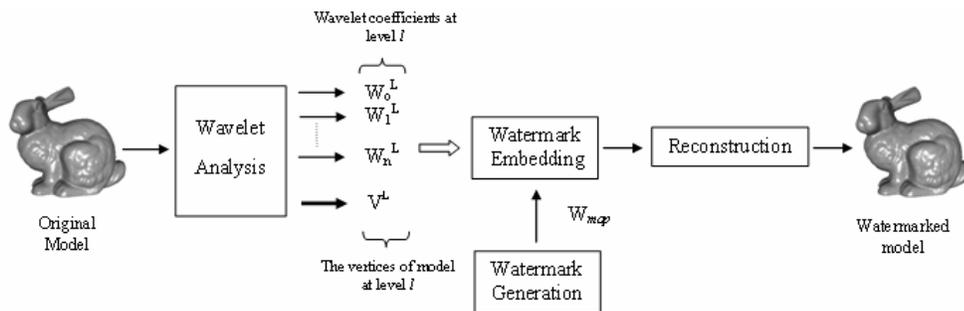


Figure 3.17 The general scheme of the algorithm in [87].

3.6 Comparisons and Discussions on Geometry Watermarking

A brief comparison between the two main categories of geometry watermarking methods is presented in Table 3.3. Based on this table, compared to the spatial domain methods, transform domain methods generally satisfy the trade off between the imperceptibility and robustness in a more reliable manner, since the applied transformation mostly separates the perceptually significant and insignificant part of the 3D model. Therefore, it is easier to find the perceptually significant part of the model to embed the watermark in this category. The second advantage is due to the ability for spreading the watermark into each resolution after the transformation. Such a spreading makes the watermark more robust, especially against compression, noise addition and local deformations. Since the watermark is deeply embedded into each resolution of the model, watermark components reside inside the 3D model as long as the visual quality is preserved. However, the transform domain methods are computationally more complicated, compared to that of the spatial domain methods, which preclude their use in practical applications. On the other hand, spatial domain methods are easier in implementation and more robust especially against cropping attack, where most of the transform domain methods fail.

Table 3.3 Comparisons between different categories in 3D geometry watermarking

	Pros	Cons
Spatial Domain Methods	Lower complexity, Robustness against Cropping	Difficulty in finding perceptually significant regions, Weakness against local deformations
Transform Domain Methods	Robustness against compression and noise addition, Well-integration with visual perception	Higher Computational Cost, Weakness against Cropping

In Table 3.4, the proposed solutions against the main attacks in geometry watermarking is presented. Since one of the important problems in geometry watermarking is the diversity of the attacks, a separate table for the proposed solutions could be useful to achieve a

complete evaluation. In summary, the first four attacks in Table 3.4, *translation*, *rotation*, *scaling* and *affine transformation* are handled in the literature by means of embedding the watermark into the invariant primitives of the mesh model under mentioned attacks. *Cropping* is solved by spreading the watermark into the entire model. For the robustness against *additive noise* and *compression*, transform techniques which embed the watermark into different resolutions are utilized. Finally, *mesh simplification* and *remeshing* is solved by resampling the attacked mesh using the original mesh connectivity.

Finally, the methods are tabulated in Table 3.5 with respect to their robustness against the main attacks. The methods are also categorized as *non-blind* or *blind* in this table. While the *non blind* methods require the original mesh in the detection process, the *blind* methods do not have such a constraint. It is difficult to speak about the superiority of one method to another method due to the lack of a unique 3D scene representation, coding scheme and a benchmarking on 3D geometry watermarking. However, if only the robustness against possible common attacks on 3D geometry is considered in the evaluation, the transform domain methods, [86], [83], [91] seem more promising.

3.7 Summary and Conclusions

In this chapter, a detailed literature survey on 3D geometry watermarking techniques is presented. Realizing that 3D-3D watermarking methods are relatively mature compared to the other two new categories of 3D watermarking, an extensive state-of-the-art survey for 3D-3D geometry watermarking methods is given in this thesis, instead of investigating new approaches and solutions in this category.

The 3D-3D geometry watermarking methods in the chapter are classified, as *spatial domain methods*, where the watermark is embedded into the geometric values of the geometric primitives, such as the coordinates of the points, length of a line, area of a polygon, volume of a polyhedron, and *transform domain methods*, where the watermark is embedded into the resulted coefficients, after a 3D geometry-based transformation is applied to the 3D geometry data. Pros and cons of each category are summarized in Table 3.3. Briefly, transform domain methods should be more appropriate for determining the

significant portions of the 3D object, hence more robust to the compression and noise attacks. On the other hand, spatial domain approaches are easier to implement and robust against the geometrical attacks, such as cropping.

The proposed solutions in the literature to the main attacks in geometry watermarking are examined separately, in order to give a more concise evaluation on 3D geometry watermarking. The robustness of the methods against main attacks is also compared. Although it is difficult to state the superiority of a method against another, due to the lack of a unique 3D scene representation, coding scheme and a benchmarking on 3D geometry watermarking, some methods are found out to be more promising, based on their robustness against possible common attacks on 3D geometry.

One important consequence of the chapter is to realize that the existing 3D watermarking methods in the literature mostly work on the 3D geometry-based representation of 3D scenes. As pointed before, in such a representation, the scene geometry is modeled by using meshes, point clouds or voxels and then, texture and reflectance maps are used as the additional descriptions on top of the model. Currently, such methods are mostly proposed to protect the geometry description of a 3D object that is used in computer graphics applications. However, considering that the geometry-based representations might also be a suitable framework for the future technologies, such as 3DTV, some suitable extensions of these methods might be good candidates for the solution of the copyright problem and other related problems, such as authentication, content labeling, and content protection, in the coming applications.

Table 3.4 Attacks and proposed solutions in 3D watermarking

Attacks	Proposed Solutions
1. Translation	<ul style="list-style-type: none"> Positioning the object, as the center of mass corresponding to the origin of the coordinate system [73] Using invariant metrics under <i>translation</i>, as an embedding primitive (e.g. a pair of angles in a triangle of a mesh [43][67])
2. Rotation	<ul style="list-style-type: none"> Rotating the object such that its principal component coincides with the z-axis of the coordinates system [73]. Using metrics invariant under <i>rotation</i> as an embedding primitive (e.g. a pair of angles in a triangle of a mesh [43], [67])
3. Scaling	<ul style="list-style-type: none"> Normalizing the radial components of the vertices for the model such that the distance of the furthest vertices to the origin is equal to one [65]. Using metrics invariant under <i>scaling</i> as an embedding primitive (e.g. a pair of angles in a triangle of a mesh [43], [67])
4. Affine Transformation	<ul style="list-style-type: none"> Using the original mesh to estimate and recover the distortions due to affine transformation [77]. Embedding the watermark into invariant metrics under affine transformation (e.g. ratio of the volumes of a pair of tetrahedrons in a mesh model [43], [67])
5. Cropping	<ul style="list-style-type: none"> Distributing the information corresponds to a bit of the watermark over the entire model via vertex scrambling [77]. Spreading the watermarks into different surface patches of the mesh model and using the normal vector distribution of each patch for watermark embedding [81].
6. Additive Noise	<ul style="list-style-type: none"> Embedding the watermark into the coefficients of each resolution after a frequency based transformation is applied to the mesh model. (Transform Domain Methods, [69], [86])
7. Compression	<ul style="list-style-type: none"> Embedding the watermark into the coefficients of each resolution after a frequency based transformation is applied to the mesh model. (Transform Domain Methods [69], [86])
8. Mesh Simplification	<ul style="list-style-type: none"> Embedding each watermark sample into a set of vertices, instead of one vertex [73]. Resampling the attacked mesh using the original mesh connectivity [82]. Using invariant metrics under mesh simplification as an embedding primitive, such as normals of collections of surfaces [42].
9. Remeshing	<ul style="list-style-type: none"> Using invariant metrics under mesh simplification, such as normals of collections of surfaces, as an embedding primitive [42] . Resampling the attacked mesh using the original mesh connectivity [82].

Table 3.5 Methods and their robustness against major attacks in geometry watermarking. The corresponding attack to the given number is given in Table 3.4. (B: blind, NB: non-blind, NH: Not Handled in robustness tests, ✓: robust, X: not robust)

	Methods	Ref. No.	Category Type (NB or B)	Robustness Against Attacks (# in Table 3.4)									
				1	2	3	4	5	6	7	8	9	
SPATIAL DOMAIN METHODS	Based on Vertice Coordinates	[71]	NB	NH	NH	NH	NH	NH	✓	✓	✓	NH	
		[72]	NB	✓	✓	✓	✓	✓	✓	✓	✓	✓	
		[65]	B	✓	✓	✓	X	NH	✓	✓	✓	NH	
		[73][74]	B	✓	✓	✓	NH	NH	✓	NH	✓	NH	
		[75][76]	B	✓	✓	✓	✓	✓	✓	NH	✓	NH	
		[77][78]	B	✓	✓	✓	NH	X	X	X	X	X	
	Based on Other Primitives	[42]	B	✓	✓	✓	NH	X	NH	NH	✓	✓	
		[81]	B	✓	✓	✓	NH	✓	NH	NH	✓	✓	
		[43] – TSQ	B	✓	✓	✓	X	✓	✓	NH	X	X	
		[43]- TVR	B	✓	✓	✓	✓	✓	✓	NH	X	X	
		[43]- MDP	B	✓	✓	✓	✓	✓	✓	NH	X	X	
	Based on 2D Methods	[64]	B	✓	✓	✓	NH	X	✓	X	✓	X	
	TRANSFORM DOMAIN METHODS	Spectral Analysis Based Methods	[82]	NB	✓	✓	✓	NH	X	✓	✓	✓	✓
			[70]	NB	✓	✓	✓	NH	X	✓	✓	✓	X
[83]			NB	✓	✓	✓	NH	✓	✓	✓	✓	✓	
[90]			B	✓	✓	✓	NH	NH	✓	✓	✓	X	
[91]			NB	✓	✓	✓	NH	✓	✓	✓	✓	✓	
Wavelet Based Methods		[69]	NB	✓	✓	✓	✓	NH	✓	✓	✓	NH	
		[86]	NB	✓	✓	✓	NH	✓	✓	✓	✓	✓	
		[88]	NB	✓	✓	✓	NH	✓	✓	✓	✓	NH	

CHAPTER 4

2D-2D WATERMARKING FOR FREE-VIEW TELEVISION

Image-based rendering (IBR) techniques render novel views directly from input images, mostly with no utilization of any explicit information related to the 3D scene structure [96]. With the advances in IBR technology during the recent years, generation of a realistic arbitrary view of a scene from a number of original views became easier and faster. As a major application of this progress, *free-view TV*, in which TV-viewers select freely the viewing position by the application of IBR algorithms on the transmitted multi-view video, has emerged. Noting that the TV-viewer might also record a *personal* video for this arbitrarily selected view and misuse this content, it is apparent that copyright and copy protection problems should also exist for free-view TV (FTV).

In this chapter, a watermark embedding and detection method focusing on this problem is proposed for free-view video. The watermark is embedded into every image in all the sequences, which represent a 3D scene implicitly, and tried to be extracted from any 2D generated image, rendered for an arbitrary angle of the camera from these sequences. In particular, the watermarking scheme utilizes spatial masking properties of the human visual system (HVS) to achieve the *trade-off* between imperceptibility and robustness.

This chapter is organized as follows: The next section introduces the specific requirements in free-view video watermarking, compared to the conventional single-view video watermarking. Section 4.2 gives a brief summary of *light field rendering* (LFR) techniques, which are utilized to produce a projection of a scene, corresponding to an arbitrary view-point, by only using the available views of the same scene. Due to the low production costs and simple hardware implementation, it is expected that this representation might be the dominating technology in *free-view TV* systems [47]. The main interpolation methods in LFR, namely, *nearest neighborhood interpolation* and

bilinear interpolation, are also summarized in this section. Next, the details of the proposed watermarking method are given in Section 4.3. The robustness results for watermark detection from an arbitrarily generated view are presented in Section 4.4. The detection scheme assumes that the position and orientation of the virtual camera is given a priori in this approach. In order to extend the method for the case of an unknown virtual camera position and orientation, the transformations on the watermark pattern due to *nearest neighborhood interpolation* and *bilinear interpolation* in LFR are examined in Sections 4.5 and 4.6, respectively. Based on this analysis, the camera localization and homography estimation methods are proposed and the robustness results for the unknown virtual camera position and orientation are presented. Finally, the chapter is concluded in Section 4.7.

4.1 Introduction

Image-based rendering (IBR) has been developed in the last decade as an alternative to the traditional 3D geometry-based rendering techniques. IBR aims to produce a projection of a scene, corresponding to an arbitrary view-point, by using only the available views of the same scene. IBR has shown to yield more natural views, compared to traditional 3D geometry-based methods [96]. Due to its advantages, IBR has attracted much attention from the research community, as well as the industry, and yet, real-time *free-view TV* systems have emerged, where the viewing position and angle could be selected, as a result of the application of IBR algorithms to the transmitted multi-view video [47], [97] (Figure 1.2).

Noting that a TV-viewer might also record a personal video from the arbitrarily selected views and misuse this content, it is apparent that copyright problems also exist for multi-view video. In a possible scenario, the owner of the multi-view content should prove his/her ownership, not only on the original views of the multi-view video, but also on any *virtual* view, which is generated by the viewer using IBR from the original views.

Among many alternative digital rights management methods, the copyright problem for visual data has already been approached by embedding hidden imperceptible information, denoted as *watermark*, into image and video content [1]-[4]. Hence, *watermarking* can be

utilized for embedding information into the multi-view content about the owner and extracting this ownership information even from the rendered virtual views. However, the problem is more complicated compared to the conventional single-view video watermarking due to the fact that rendering becomes an important noise source on the watermark signal.

First of all, beside the robustness to common video processing and multi-view video processing operations, the main characteristics problem for multi-view video watermarking is to extract the embedded signal from a virtual video sequence, generated for an arbitrary view (Figure 4.1). In order to extract the watermark from such a rendered view, the watermark detection scheme should involve an estimation procedure for the virtual camera position and orientation, where the rendered view is generated. In addition, the watermark should also survive from image-based rendering operations, such as frame interpolation between neighbor cameras and pixel interpolation inside each camera frame.

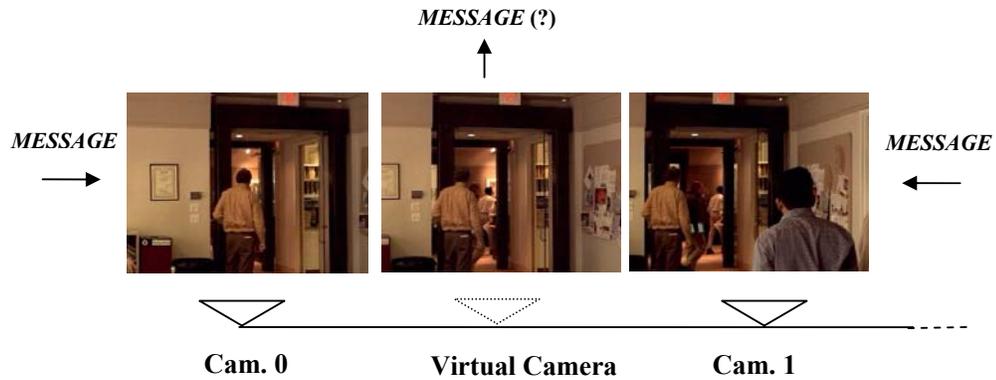


Figure 4.1 The watermarking problem for free-view television

IBR also extends the imperceptibility requirement for multi-view video watermarking. A video watermark should be spatially invisible in each frame and temporally invisible in the temporal dimension of the video. Moreover, in multi-view video watermarking, the

watermark should be embedded to the adjacent original camera frames such that it does not yield any temporal distortions in the resulting virtual video during rendering.

In the literature, the most popular IBR representation is the *light field* approach [98], due to its simplicity and natural-looking outputs. Light field-based methods only require the original images to construct the imagery views and it is possible that this representation might be the dominating technology in *free-view TV* systems, due to the low production costs and simple hardware implementation [47]. Therefore, the proposed watermarking method is specially tailored for the free view TV systems, whose arbitrary views are generated by using light field rendering (LFR).

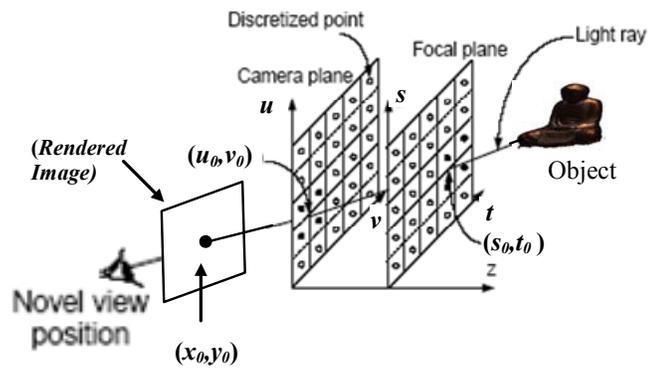
4.2 Light Field Rendering

Light field rendering (LFR) is a simple and robust method for generating novel views from arbitrary camera views by combining and resampling the available images without utilizing any depth information about the scene. The basic idea behind this technique is a representation of the *light field*, as the radiance at a point in a given direction in regions of space free of occluders (free space). This representation characterizes the flow of light through free space in a static scene with fixed illumination [98].

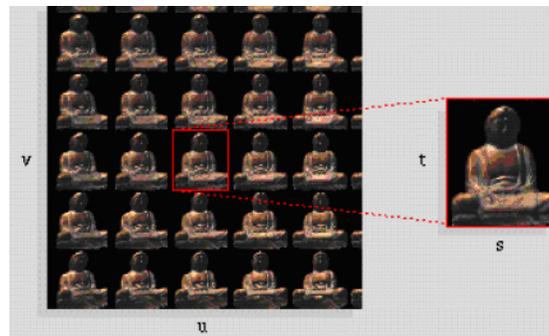
A practical way to create (capture) light fields is to assemble a collection of images by a number of cameras where the intensity of the pixels of the camera views corresponds to the radiance of the light rays passing through the pixel locations and camera centers. A general configuration for LFR is given in Figure 4.2 (a) with two parallel planes namely, *camera* (uv) plane and *focal* (st) plane [98].

Camera plane can be interpreted as the plane where the cameras are located. *Focal* plane is parallel to the camera plane which is utilized for parameterization of the light field. A light ray is parameterized with four variables, (u_o, v_o, s_o, t_o) , where (u_o, v_o) and (s_o, t_o) are the intersections of the light ray with *camera* and *focal* planes, respectively. These planes are usually discretized, so that a finite number of light rays can be recorded (Figure 4.3 (a)). If all the discretized points from the focal plane are connected to a single point on the camera plane, an image (2D array of light fields) is resulted [96]. This image is

simply the sheared version of the camera view located at that point on the camera plane. If the same process is considered for all the points on the camera plane, a 2D image array is obtained, as it is shown in Figure 4.2 (b). Hence, 4D representation (i.e. (u, v, s, t)) of the light field can also be interpreted as a 2D image array.



(a)



(b)

Figure 4.2 (a) General configuration for LFR [96] (b) A sample light field image array: *Buddha* light field [99].

While generating the virtual view of the object for an arbitrary viewing position, the light ray for each pixel of the rendered image is intersected with the *camera* and *focal* planes (Figure 4.2 (a)). Then, intensity of the pixel is calculated by linearly interpolating the

neighboring light rays in the image array. For example, the intensity of a pixel (x_o, y_o) , corresponding to the light ray (u_o, v_o, s_o, t_o) in Figure 4.2 (a), is interpolated from the intensity of the light rays connecting the solid discrete points on the two planes [96].

Based on the utilized neighbor light rays during interpolation, there are two major interpolation methods in LFR: *nearest neighborhood interpolation* and *bilinear interpolation* [98]. In nearest neighborhood interpolation-based LFR, the intensity value of each pixel for the virtual view is generated from the pixel intensities, belonging only to the nearest camera. On the other hand, in bilinear interpolation-based approach, the intensity value of each rendered pixel is obtained, as a weighted sum of pixels belonging to a number of neighboring cameras (see Figure 4.3). While the nearest interpolation is simpler to implement, it usually yields more distortion (discontinuities) in the rendered video, especially in the regions whose pixels are rendered by different (adjacent) cameras. On the other hand, bilinear interpolation gives more natural and subjectively pleasant outputs as a result of rendering operation [98].

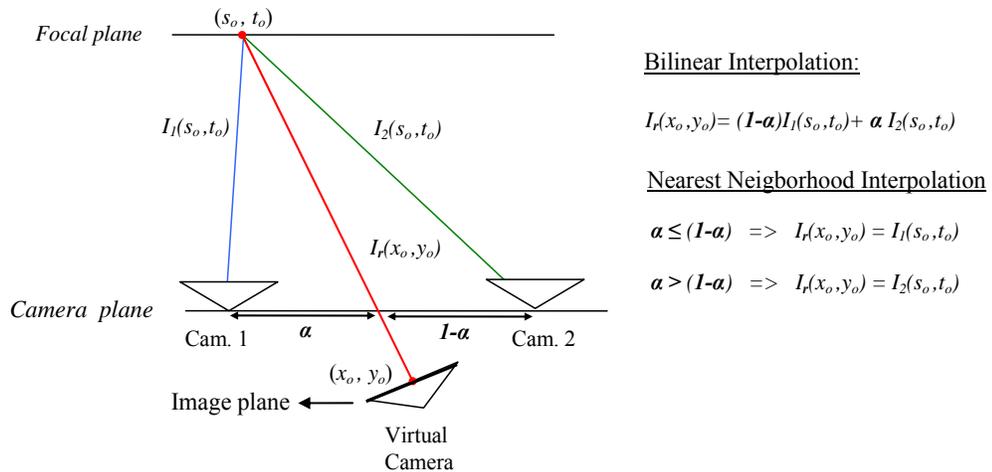


Figure 4.3 Interpolation methods in LFR: Nearest neighbor vs bilinear interpolation

4.3 Proposed Watermarking Method

In the upcoming free-view TV scenario, the broadcast stations will deliver their multi-view content to the set-top boxes of the viewers and the viewers might interact with their viewing angles by implicitly using the rendering algorithms, which process the transmitted multi-views in their set-top boxes. Hence, a watermarking algorithm should be capable of detecting a watermark, which might be transmitted within the multi-view content, from any rendered view for proving ownership or copy protection, in case of illegal usage. Unfortunately, it is not possible to detect the trace of a watermark in a rendered image by using the conventional watermarking algorithms in the literature; hence, a novel approach is required.

4.3.1 Watermark Embedding

The proposed method embeds the watermark into each image of the light field image array (Figure 4.2 (b)) by exploiting spatial sensitivity of HVS [51]. For that purpose, the watermark is modulated with the resulting output image, after filtering each light field image by a 3×3 high-pass filter, and spatially added onto the image. This operation decreases the watermark strength along the flat regions of the image, in which HVS is more sensitive, whereas increases the embedded watermark energy in the textured regions, where HVS is relatively insensitive.

There are two critical points at the embedding stage. First of all, the watermark is embedded to the light-field images, which are the sheared perspective projection of the original camera frames [98]. These frames can be easily obtained by the camera calibration information. Secondly, the watermark component, added to the intensity of each image pixel, is determined according to the intersection of the light ray corresponding to that pixel with the focal plane. The same watermark is added to the pixels of different camera frames, whose corresponding light rays are intersected at the same point in the focal plane, as illustrated in Figure 4.4. The rationale behind such a procedure is to avoid facing with the superposition of the different watermark samples from different camera frames in the interpolation step during the rendering. Otherwise, such a superposition severely degrades the performance of the correlation operation in the watermark detection stage. Moreover, such a procedure should also avoid the flickering

type of distortion during rendering, since the watermark component in any rendered view will be identical to the watermark components in the adjacent original views.

The method applies the relation

$$I_{uv}^*(s,t) = I_{uv}(s,t) + \alpha.H_{uv}(s,t).W(s,t) \quad (4.1)$$

to each light field image where I_{uv} is the light field image corresponding to the camera at the (u,v) position on the camera plane, H_{uv} is the output image after high pass filtering, α is the global scaling factor to adjust the watermark strength, W is the watermark sequence generated from a Gaussian distribution with zero mean and unit variance, and finally, I_{uv}^* is the watermarked light field image. It should be noted that (s,t) domain indicates the focal plane at which the watermark is embedded.

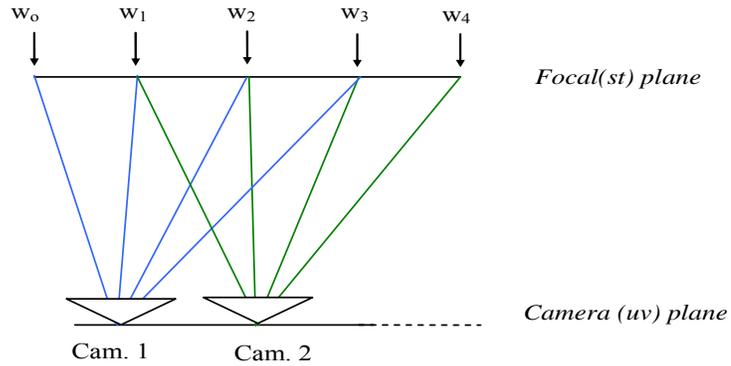


Figure 4.4 Watermark embedding methodology

4.3.2 Watermark Detection

The critical problem of free view video that is the extraction of the watermark from an arbitrary view generated via IBR, is mainly considered during the detection. The transmitted watermarked camera views which are utilized during the generation of arbitrary view in free view TV are available during detection. The detector should also have the original watermark embedded to the original camera views.

The popular correlation-based detection scheme is utilized for watermark extraction. Assuming that the position and orientation of the virtual view are known *a priori* (not available in practice and a novel algorithm is proposed to solve this problem in Section 4.5), the first step is applying the same rendering operations during the generation of an arbitrary view to the watermark pattern, W , in order to generate a “rendered watermark”, denoted as W_{ren} . After the arbitrarily selected view, (I_{ren}) , is filtered by a high-pass filter, the normalized correlation between the resulting image (\hat{I}_{ren}) and rendered watermark is determined as

$$\frac{\langle \hat{I}_{ren}, W_{ren} \rangle}{|\hat{I}_{ren}| |W_{ren}|}. \quad (4.2)$$

In the next step, the normalized correlation is compared to a threshold for the detection of the watermark. The overall structure of the watermark detection is shown in Figure 4.5.

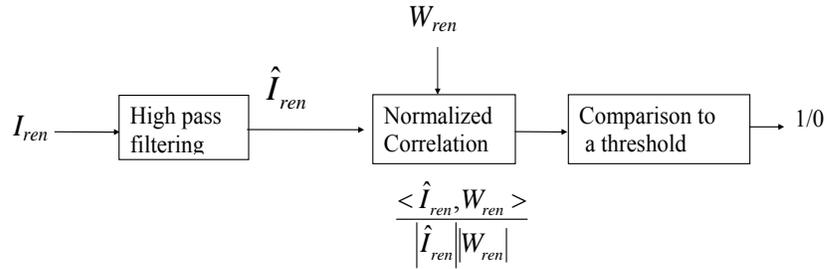


Figure 4.5 Overall structure of Watermark Detection Process

4.4 Experimental Results

Buddha and *Teapot* light fields are used during the simulations. A *Laplacian* high pass filter of size 3x3 is used during filtering operations. The global scale factor, α , is adjusted to 0.7. The parameterization of the focal and camera plane for *Buddha* light field [99] is shown in Figure 4.6. (The calibration parameters for *Teapot* light field is given in [100].)

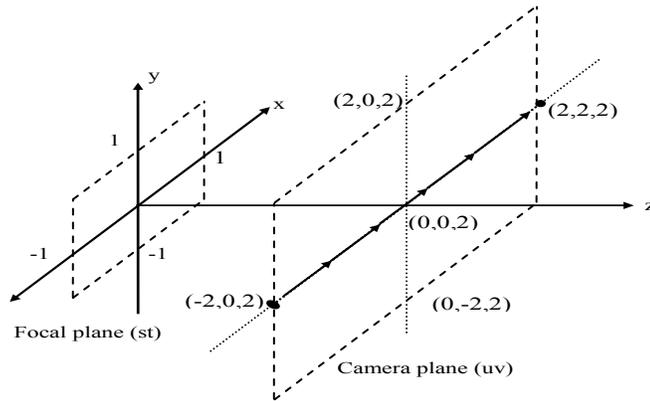


Figure 4.6 The location of camera & focal plane for *Buddha* light field

4.4.1 Imperceptibility Tests

In the imperceptibility tests, the visual equivalence of the original and watermarked rendered image is checked at the first step. Typical rendered views for the original and watermarked *Buddha* light field are presented in Figure 4.7. Virtual camera is located at $[0,0,2]$ with the normal direction of $[0,0,-1]$. Another example is given in Figure 4.8 for *Teapot* light field. Based on subjective assessment, there is no visible difference between the original and watermarked rendered views.

In the second step of the imperceptibility tests, the visual equivalence of the original and watermarked rendered video is also tested. The video is obtained by sequencing the light field images according to the trajectory (from $[-2,0,2]$ to $[2,2,2]$) on the camera plane in Figure 4.6. Since the watermark patterns embedded to adjacent frames can yield visible distortions during rendering, these tests should also be included in the experiments. The rendered video sequences from the original and watermarked *Buddha* light field for the illustrated trajectory in Figure 4.6 are given in <http://snap-mmrg.eee.metu.edu.tr/FreeView>. Based on subjective assessment, there is no visible difference between the watermarked and original rendered video sequences.

In order to show the positive effect of embedding the same watermark component into the pixels of different camera frames whose corresponding light rays are intersected at

the same point in the focal plane (see Figure 4.4), two more video sequences are generated for the same trajectory in Figure 4.6. The first video is generated from the watermarked light field by the proposed method where the *same* watermark is embedded to each light field image. However, for the second video, *different* watermark components are added into the pixels of different camera frames whose corresponding light rays are intersected at the same point in the focal plane. Although, there is no flickering type of distortion in the first video, a flicker is observable in the second video, which shows the validity of the proposed embedding strategy. It should be noted that the watermark strength is increased from 0.5 to 0.9 for both two video sequences in order to view the flickering in the second video. The video sequences are available in the aforementioned URL link.

4.4.2 Robustness Tests

During the robustness tests, the detection scheme is applied for different imagery views based on the virtual camera position and orientation. In *Buddha* light field, the camera position of $[0,0,2]$ and normal direction of $[0,0,-1]$ are taken as reference, in order to describe translation and rotation in the results. The following cases are considered during the simulations:

Case I: No translation in uv -plane and z -axis. No rotation. Camera position= $[0\ 0\ 2]$; Image plane normal= $[0\ 0\ -1]$; focal length=2 for all cases (default).

Case II: No translation in z -axis. No rotation. Only translation in uv -plane. Camera position = $[0.5\ 0\ 2]$; Image plane normal= $[0\ 0\ -1]$.

Case III: Translation in uv plane and z -axis. No rotation. Camera position= $[0.5\ 0\ 3]$; Image plane normal= $[0\ 0\ -1]$.

Case IV: Translation on uv -plane + Rotation. No translation in the z -axis. Camera position= $[2\ 0\ 2]$; Image plane normal= $[-1\ 0\ -1]$.

Case V: Translation on uv -plane and z -axis + Rotation. Camera position= $[1.5\ 0\ 2.5]$; Image plane normal= $[-1\ 0\ -1]$.

The normalized correlation values in Case I for the embedded watermark and other 100 randomly generated watermarks from a Gaussian distribution of zero mean and unit

variance are shown in Figure 4.7 (c). The embedded watermark is placed at the 50th position. Its higher value with respect to the other correlation values shows that the watermark is detected successfully. The correlation results for the other cases are shown in Table 4.1. The results for *Teapot* light field, for different virtual camera positions and orientations are given in Figure 4.8 and Table 4.2.

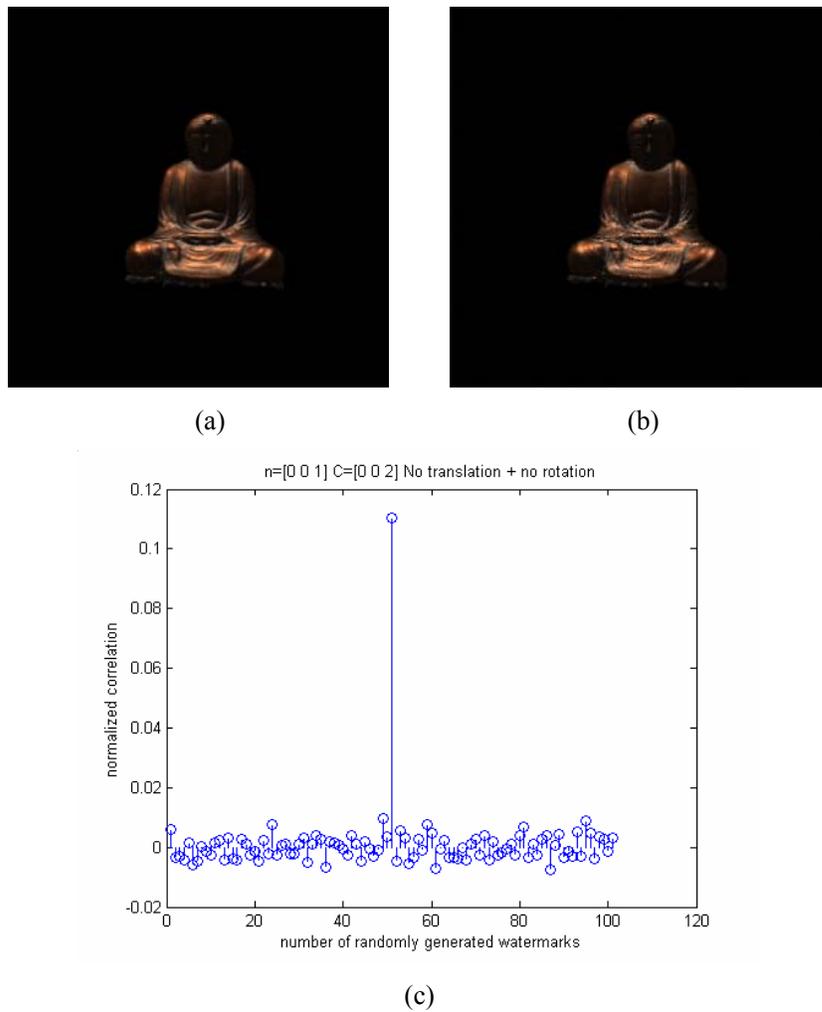


Figure 4. 7 (a) Rendered view, (b) watermarked view, (c) normalized correlation result (Eq. 4.2)

Table 4.1 Normalized correlation (4.2) for the embedded watermark and max. of the normalized correlations for 100 other watermarks in *Buddha* light field

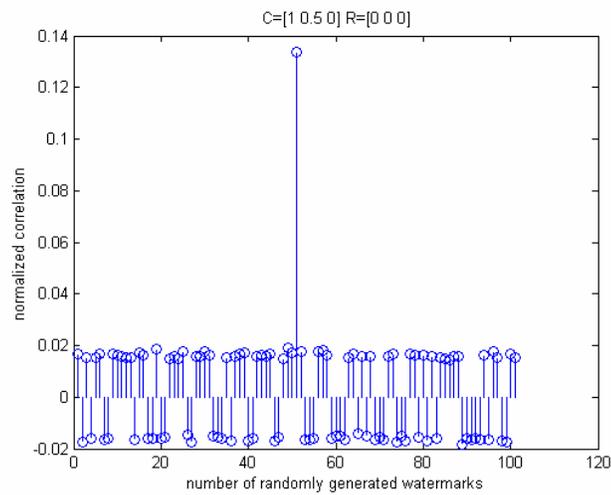
Normalized Correlation Values (Eq. 4.2)	Case II	Case III	Case IV	Case V
Embedded Watermark	0.125	0.085	0.075	0.11
Max. Other Watermark	0.010	0.009	0.010	0.009



(a)



(b)



(c)

Figure 4.8 (a) Rendered view, (b) watermarked view, (c) correlation result

Table 4.2 Normalized correlation results (Eq. 4.2) for *Teapot* light field. (The components of the vectors for the camera rotation are the angles between the image plane normal and x, y, z Cartesian coordinates, respectively)

Camera Position	[2.7 2.7 0]	[1 0.5 0]	[1.3 0 0.5]	[1 0.5 0.5]
Camera Rotation	[0 0 0]	[0 0 0]	[0 0 $\pi/8$]	[0 0 0]
Embedded Watermark	0.085	0.134	0.056	0.035
Max. Other Watermark	0.017	0.019	0.018	0.019

4.5 Analysis of Transformations on Watermark Pattern during LFR

In the previous tests, the position and orientation of the virtual camera is assumed to be known during the detection. However, this information is not available in practice. Hence, the detection algorithm should also include a procedure to determine the position and orientation of the virtual camera.

In order to handle this problem, the relation between the embedded watermark and the rendered watermark is analyzed for the two main interpolation methods in light field rendering, namely, nearest neighbor and bilinear interpolation. In this section, the analysis for nearest neighbor interpolation and the robustness results are given in consecutive sections.

Considering the virtual camera position and orientation, the problem for the nearest neighbor interpolation can be handled in three cases:

1. The virtual camera is located in the camera plane and its rotation matrix is a unit matrix. This configuration is shown in Figure 4.9. The same watermark, W , is embedded to the off-sheared images of Cameras 1 and 2. From the analysis of the operations in light field rendering, the watermark corresponding to the imagery camera, W_{ren} , will simply be a *shifted version* of W .
2. The virtual camera is again located in the camera plane and its rotation is not equal to unity (see Figure 4.9). For this case, the relation between W_{ren} and W is a planar projective transformation (homography) [101] (see Appendix-B for its explanation).

3. The virtual camera is in an arbitrary position and orientation (Figure 4.9), as the most general case. The relation between W_{ren} and W is again planar projective transformation. However, the projective transformation is different for various regions of the virtual view based on the original images used during the generation of the virtual view. Hence, this is the most difficult case for watermark detection.

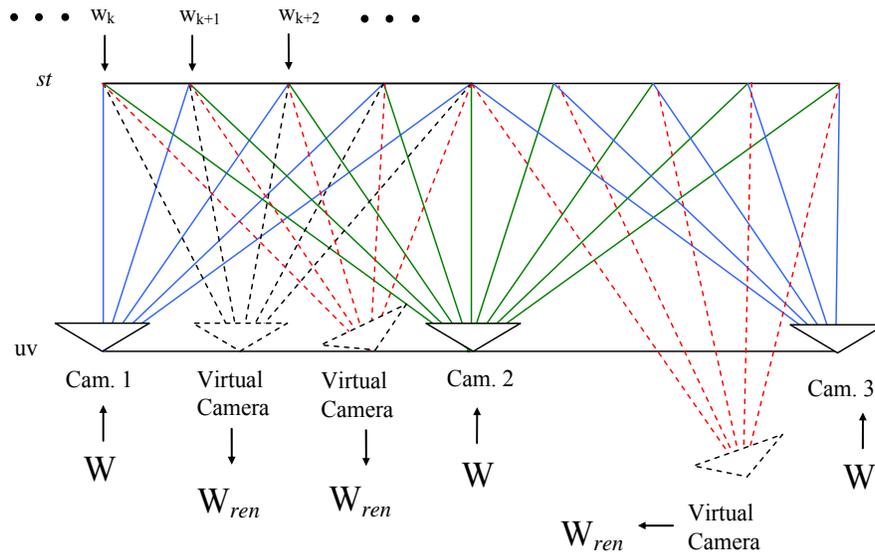
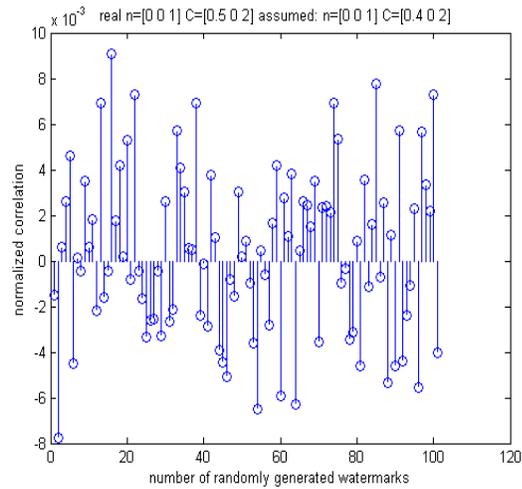


Figure 4.9 Configurations for the imagery camera position and rotation from left to right for Case 1, 2 and 3, respectively.

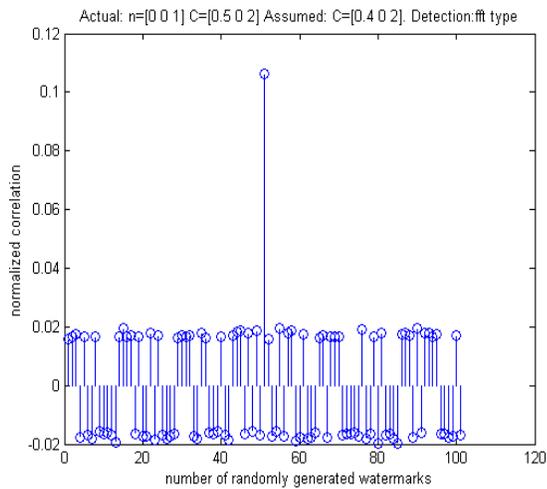
4.5.1 Proposed Solution for Case I

In order to solve the problem for the first case, the correlation is computed for the possible shifts of the watermark pattern in the detector. The computation of this operation is decreased by utilizing a fast Fourier transform (FFT) and its inverse (IDFT). Specifically, symmetrical phase only matched filtering (SPOMF) is used for correlation computations [10]. The position of the virtual camera is changed from the actual case and the normalized correlation is computed for this shifted camera position during these tests. The robustness results in Figure 4.10 show the correlations for the normal and SPOMF

type of detection when the camera position is slightly different than the actual case. While the SPOMF still detects the watermark, normal detection algorithm fails.



(a)



(b)

Figure 4.10 Normalized correlations (Eq. 4.2) graphs for (a) the conventional and (b) SPOMF detection. Actual camera position = $[0.5\ 0\ 2]$. Assumed camera position in the test = $[0.4\ 0\ 2]$

4.5.2 Proposed Solution for Case II

In Case II, the rendered image is generated from the image corresponding to the nearest neighbor camera. The relation between the rendered and the nearest neighbor original image can be approximated as a planar projective transformation [101] (see Appendix B).

In this case, there are mainly two stages in the solution. First, the nearest neighbor camera to the virtual camera should be determined among all other cameras. Second, the planar projective transformation between the rendered and the nearest neighbor original image should be estimated. Afterwards, the planar projective transformation could be applied to the original watermark to obtain the rendered watermark.

The proposed solution for finding the nearest neighbor image to the rendered image utilizes the intersections of the cross lines for any non-collinear four points. While intersections of cross lines of corresponding four points should be approximately at the same location in the rendered and nearest neighbor original image [101], they will be at different locations for the other images, as illustrated with crosses in Figure 4.11. This property can be used to determine the original image from which the rendered image is generated.

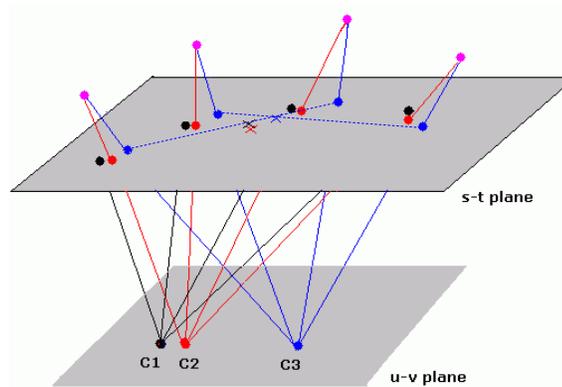


Figure 4.11 Illustration of the projection of four feature points. $C1$, $C2$ and $C3$ are camera centers for the rendered image, nearest neighbor original image and an arbitrary original image.

The procedure for obtaining the nearest neighbor light field image is as follows:

1. Find feature points in the rendered image and original images. Match the feature points in the rendered image and one of the original images [102].
2. Randomly select four matched points and find intersection point of cross lines. (red, black and blue crosses in Figure 4.11)
3. Find the mean square error (*MSE*) in 3x3 window. Slide the 3x3 window in a 7x7 search region around the intersection point and obtain the minimum mean square error.
4. Repeat step 2 and 3 for *N* times and obtain the best *percentage p* results. Record the average of *MSE*.
5. The minimum average corresponds to the nearest neighbor image to the rendered image.

(During simulations, Harris corner detector [103] is utilized for feature extraction. *N* is taken as 2000 and *p* as 60 %)

Table 4.3 gives the average of *MSE* between the rendered image and the neighbor original images. The average *MSE* is lowest for the *buddha.23.16* [100], from which the rendered image is generated, as expected (“*buddha.u.v*” corresponds to the light field image at row *u* and column *v* on the camera plane. Please see Figure 4.2).

Table 4.3 The average *MSE* between the rendered image and the original images in the *Buddha* light field.

(Virtual Camera position: [0.5 0 3]; Image plane normal:[0 0 -1])

Image No.	Average <i>MSE</i>	Image No.	Average <i>MSE</i>	Image No.	Average <i>MSE</i>
22.15	23.20	22.16	22.45	22.17	30.61
23.15	23.08	23.16	0.09	23.17	36.41
24.15	33.89	24.16	19.71	24.17	40.91

After determining the nearest neighbor original image to the rendered image, the planar projective transformation is estimated between the two images by using the RANSAC algorithm [104]. The resulting homography is applied to the embedded watermark pattern to obtain the rendered watermark, W_{ren} . Then, the SPOMF type of detection is utilized to extract the watermark from the rendered image. The normalized correlation values are given in Table 4.4 for different virtual camera positions and rotations. The normalized correlation for the embedded watermark is clearly higher than the maximum of the other correlation values which shows that the method clearly detects the watermark for the mentioned cases.

Table 4.4 Normalized correlation (Eq. 4.2) for the embedded watermark and max. of the normalized correlations for 100 other watermarks.

Camera Center	[1 0 2]	[0 -1 2]	[0 0 2]
Image plane normal	[-0.5 0 -1]	[0 0.5 -1]	[0 0.5 -2]
Embedded Watermark	0.078	0.088	0.121
Max. for the others	0.023	0.020	0.020

4.5.3 Proposed Solution for Case III

In this case, the rendered image is composed of different regions where the light fields for each region are generated from a different light field image corresponding to a different camera. An illustration is given for this scenario in Figure 4.12. The rendered image is formed of three regions and the light rays (pixels) for the first, second and third region are generated from the light rays (pixels) for the first, second and third cameras, respectively. Similar to the second case, the relation between each region of the rendered image and the corresponding region in the light field image can also be approximated as a planar projective transformation.

The rendered image, which is generated for a virtual camera located at $[0\ 0\ 2.4]$ with $[0\ 0\ 1]$ orientation, is shown in Figure 4.13 for *Buddha* light field. This rendered image is composed of four main regions each of which is generated from the light field images,

buddha.15.15, *buddha.16.15*, *buddha.15.16* and *buddha.16.16*. The relation between each region of rendered image and the corresponding regions in light field images is a different planar projective transformation (Figure 4.13). This relation can be used for the watermark detection in this scaled case. Since the same watermark is embedded to each light field image, the relation between the regions of the rendered watermark, W_{ren} , and the regions of the original watermark, W , will also be a homography as illustrated in Figure 4.14.

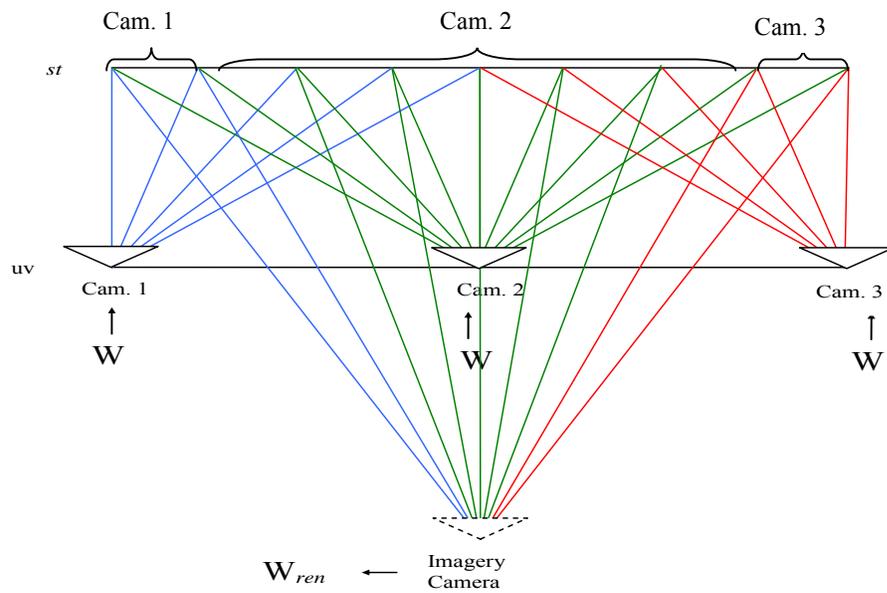


Figure 4.12 The rendered image is formed of three regions where the light rays for each region are generated from a different light field image corresponding to a different camera

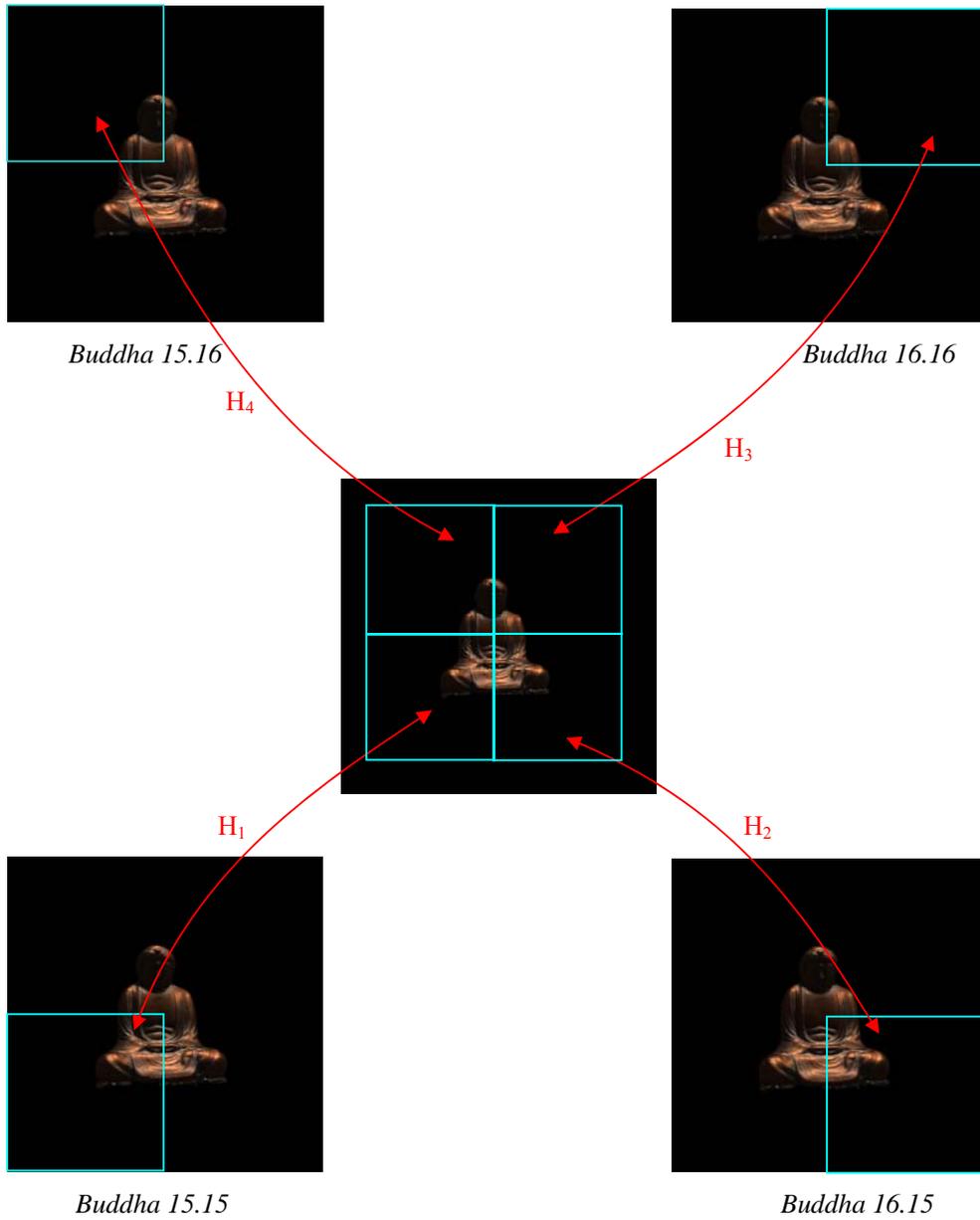


Figure 4.13 The relation between the regions of the rendered image and the original light field images. H_1 , H_2 , H_3 and H_4 correspond to the homographies between the illustrated regions.

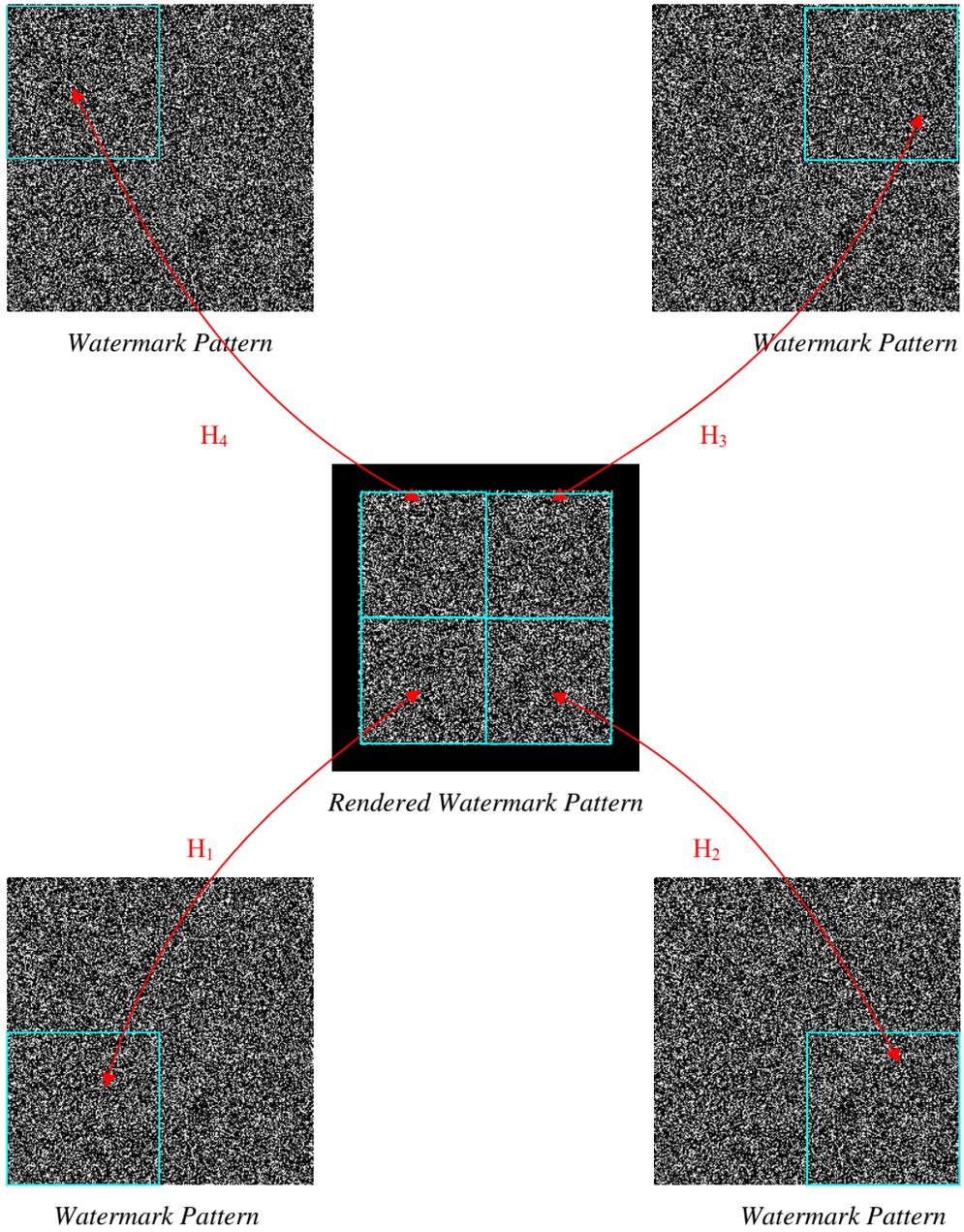


Figure 4.14 The relation between the regions of the rendered watermark and the original watermark. H_1 , H_2 , H_3 and H_4 are the homographies in Figure 4.13.

In the detection process, first of all, the rendered image should be partitioned into the regions according the original light field images from which the pixels in each region are generated. This partition process is explained in detail in the next subsection. After such a partition of the rendered image, the homographies between the regions of the rendered image and corresponding regions of the light field images are estimated by means of utilizing the scale invariant feature points [105]. Then, the estimated homography relations are applied to the corresponding regions in the original watermark, W , to generate the rendered watermark, W_{ren} (Figure 4.14). Finally, the normalized correlation between the rendered watermark and the rendered image that has passed from high-pass filtering is computed.

The normalized correlations for the original watermark and randomly generated 100 watermark pattern are shown in Figure 4.15 for the rendered image shown in Figure 4.13. The rendered image corresponds to approximately 70-80 % scaled version of the original light field images. For this case, the watermark is successfully detected. Another example is shown Figure 4.16 and 4.17. In this case, some rotation is also included in the virtual camera orientation. The watermark is still detected for this case as well.

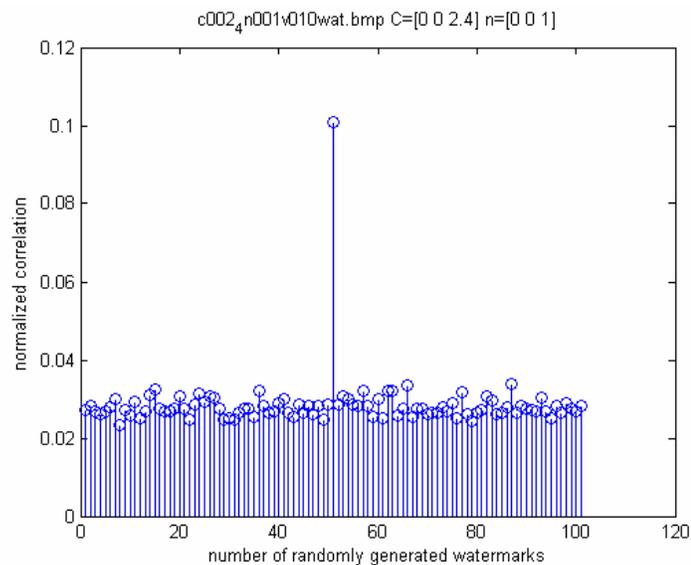


Figure 4.15 Normalized correlations results (Eq. 4.2). Camera position = $[0\ 0\ 2.4]$.
Image normal = $[0\ 0\ 1]$.

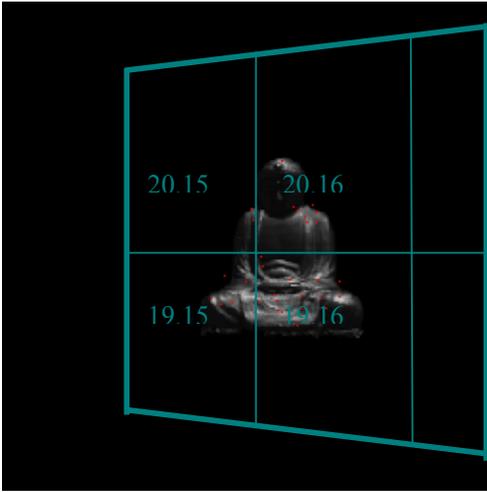


Figure 4.16 The rendered image for the virtual camera at location $[0.5 \ 0 \ 2.4]$ with orientation $[0.3 \ 0 \ 1]$. The numbers show the original image from which corresponding region is generated.

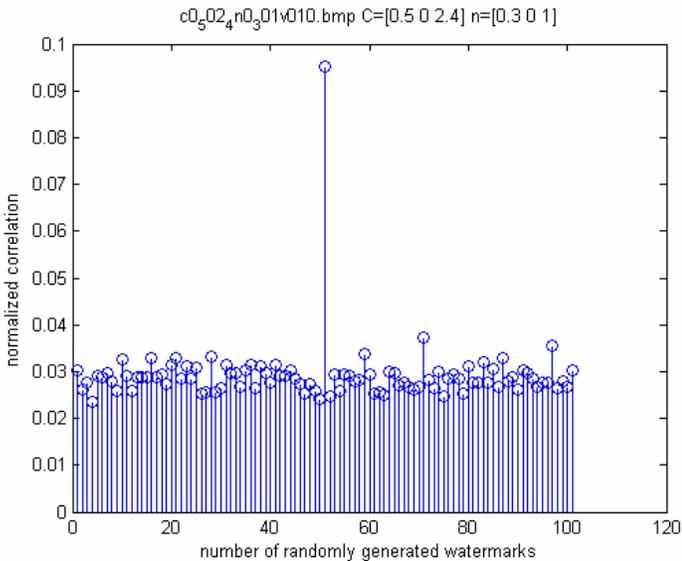


Figure 4.17 Normalized correlations results (Eq. 4.2) for the rendered image in Figure 4.16.

4.5.4 Decomposing the Rendered Image into Regions

For Case III, in the practical case, the original light field images that are used in the generation of the rendered image should also be determined automatically for a complete solution.

As the first step, the proposed solution identifies the feature points on the rendered image and decides which feature point is originating from which original light field image. In Figure 4.18, the matched points between the rendered image and the image *buddha.15.15* are shown as an example. While the matched points in the square illustrated in Figure 4.18 (b) originate from the image *buddha.15.15*, the other feature points belong to the other original light field images. Since the relation between the regions shown in square in Figure 4.18 (a) and (b) is planar projective transformation, the feature points inside the square might be selected among other feature points by utilizing the intensity difference at the intersection of the cross lines of four feature points. Similar to Case II, the intensity values at such intersections will be quite close to each other in the corresponding regions in the rendered image and its originating light field image.

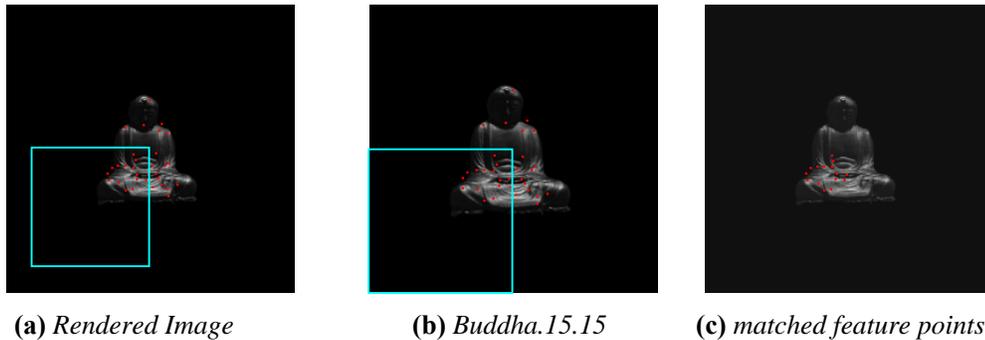


Figure 4.18 (a), (b) Matched feature points in the rendered image and the image *Buddha.15.15*. (c) The matched feature points on the rendered image originating from the image *Buddha.15.15* by the proposed algorithm.

The algorithm for finding the originating light field image of each feature point on the rendered image is as follows:

- Find the matched points between the rendered image and each of the original light field images [105].
- Determine the light field images giving the first N -highest number of feature point matches. (The number of matched feature points is highest for the original images used during rendering (see Figure 4.20))
- For each feature point in the rendered image, form a group by selecting M -nearest feature points to that feature point.
- Take four feature point pairs among the formed group and determine the intersection of cross-lines (Figure 4.21). Compute the difference of the pixel values at the intersections in the rendered image and original light field image.
- Repeat the previous step for each of the different four pairs in the group. Determine the ratio of number of four pairs which give zero difference in intensity at the cross-line intersections to the number of all four pair combinations in the group.
- If the found ratio is greater than the threshold, then assign that feature point to the matched original light field image.

(During simulations, N , M and threshold are selected as 9, 5 and 0.8, respectively. Note that the ratio of four pairs which give zero difference in intensity is utilized, instead of using MSE , as in the second case. Remember that MSE is not a robust measure between the images for the scaled cases.)

The result of the algorithm for the rendered image generated for a virtual camera located at $[0\ 0\ 2.4]$ with $[0\ 0\ 1]$ orientation and the image *buddha.15.15* is shown in Figure 4.18 (c). The selected feature points by the algorithm are mostly located on the region generated from the image *buddha.15.15*. The results for the same rendered image and the image *buddha.16.15* are shown in Figure 4.19.

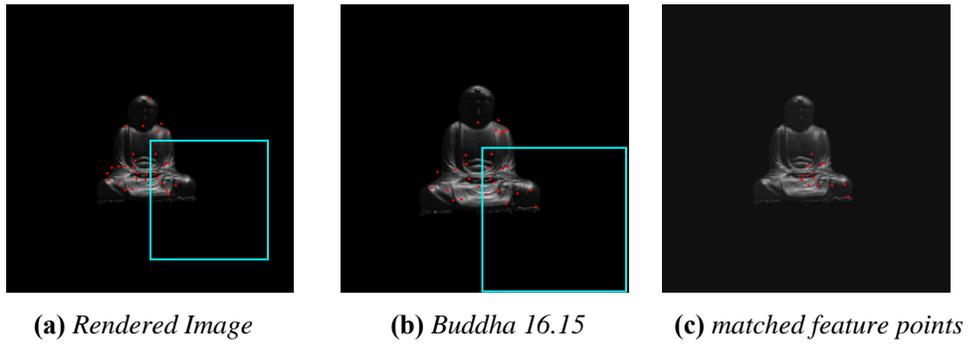


Figure 4.19 The results for the same rendered image and the image *buddha.16.15*

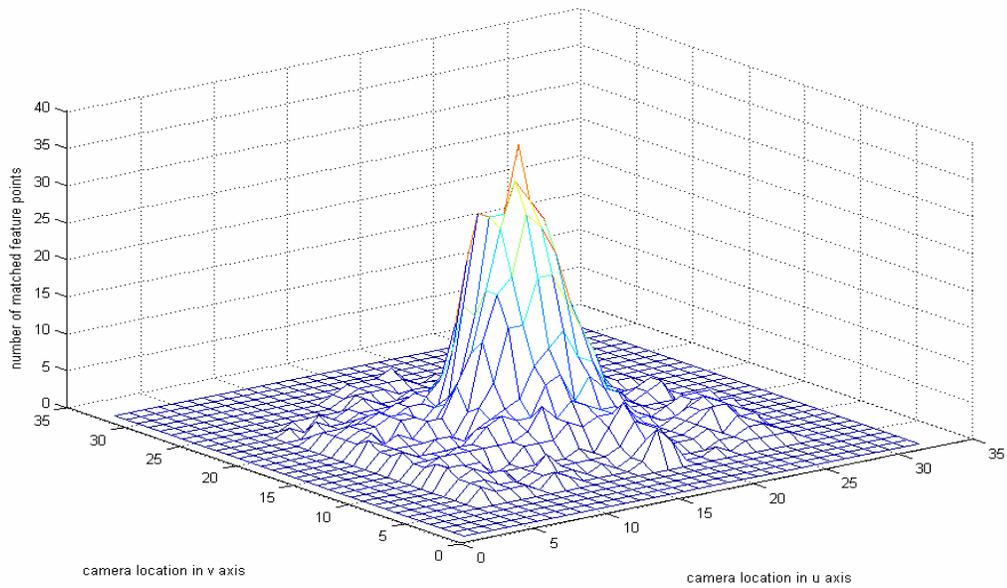


Figure 4.20 Number of matched feature points between the rendered image and original light field images with respect to the original camera position on the camera plane. Virtual camera position = $[0 \ 0 \ 2.4]$, Image normal = $[0 \ 0 \ 1]$. The original images used during rendering are *buddha.15.15*, *15.16*, *16.15* and *16.16*.

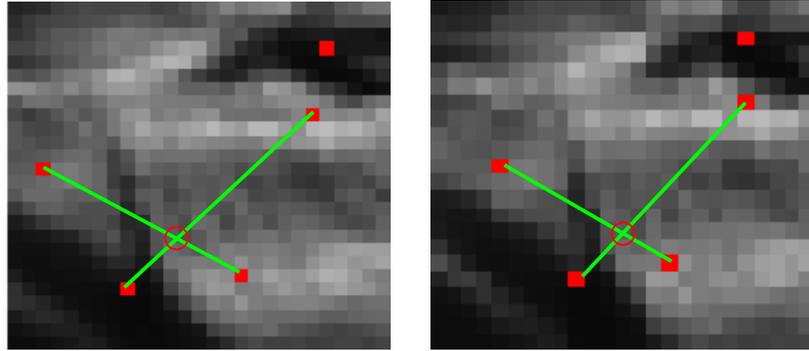


Figure 4.21 Illustration of cross-lines intersection of four matched feature points on the rendered image and one of original light field image.

After determining the original light field image of each feature point, the homographies between the rendered image and light field images are estimated by the help of the matched points. Then, the originating light field image of each pixel in the rendered image other than the feature points is determined. In order to assign each pixel to a light field image, the estimated homographies are applied to the pixel coordinates and the difference between the intensity values of the pixel and its corresponding match is computed. The pixel is assigned to the original light field image whose homography gives the minimum intensity difference. In order to increase the performance of the algorithm, the intensity difference is calculated within a 3x3 window for each pixel. An illustration of the procedure is given in Figure 4.22.

The results of the partition algorithm for the rendered image generated for an imagery camera located at $[0\ 0\ 2.4]$ with $[0\ 0\ 1]$ orientation is shown in Figure 4.23. The regions of the % 95 of all pixels are correctly determined. Another example for the rendered image generated for an imagery camera located at $[0.5\ 0\ 2.4]$ with $[0.3\ 0\ 1]$ orientation is shown in Figure 4.24. The regions of the % 90 of all pixels are correctly determined. The normalized correlation for the original watermark decreases by %5-10 due to imperfect region identification. However, this decrease is tolerable when the watermark detection performances illustrated in Figure 4.15 and 4.17 are considered.

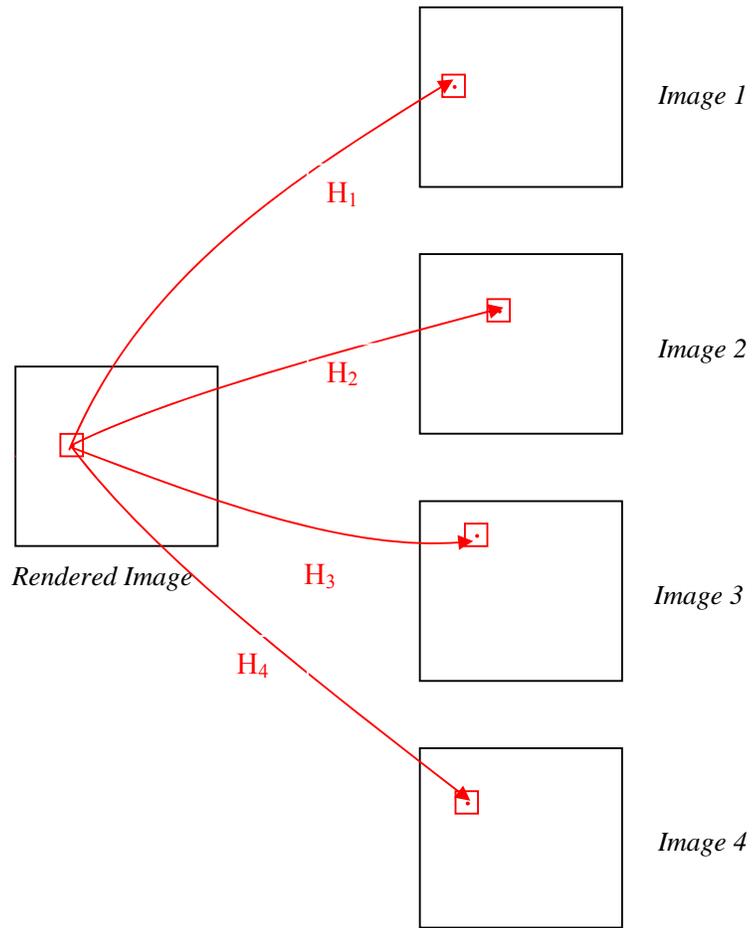
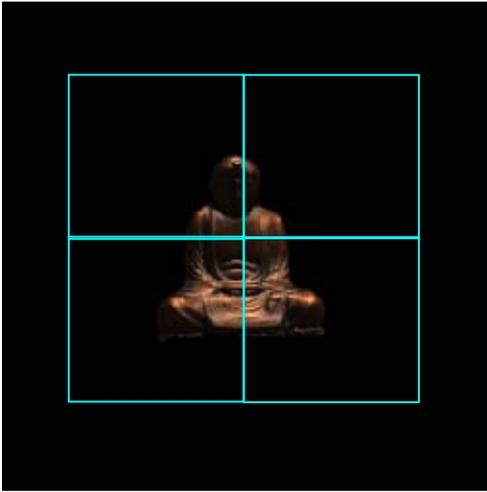
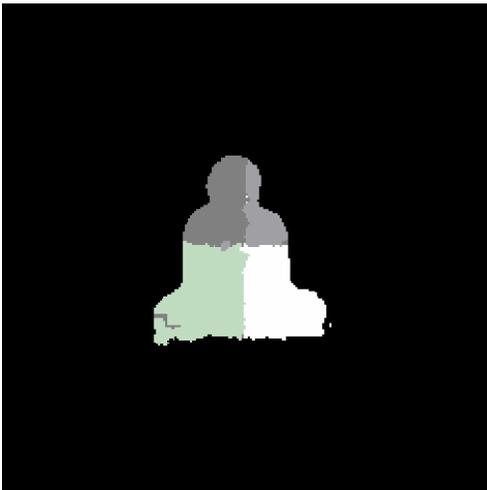


Figure 4.22 An illustration of the procedure. Each pixel is assigned to the original light field image whose homography gives the minimum intensity difference.

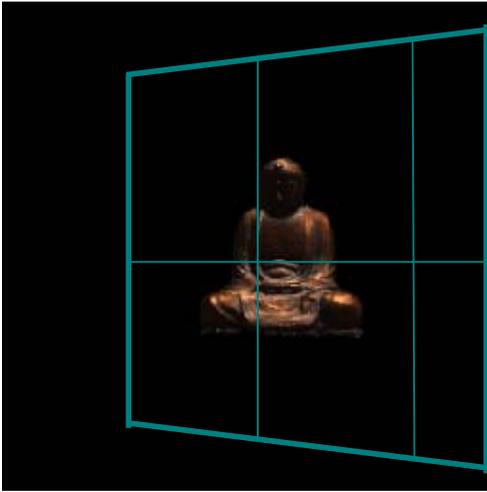


(a)



(b)

Figure 4.23 (a) Original partition for the rendered image. Camera position = $[0\ 0\ 2.4]$.
Image normal = $[0\ 0\ 1]$, (b) The results of the partitioning algorithm.



(a)



(b)

Figure 4.24 (a) Original partition for the rendered image. Camera position = $[0.5 \ 0 \ 2.4]$, Image normal = $[0.3 \ 0 \ 1]$, (b) The results of the partitioning algorithm.

4.6 Proposed Solution for Bilinear Interpolation in LFR

The other interpolation method widely used in light field rendering is bilinear interpolation. In this case, intensity of each pixel of the rendered image is calculated by the weighted sum of light rays corresponding to the four nearest neighbor original images. The weights of the cameras are determined according to the distance of the camera centers to the intersection of the corresponding light ray of the rendered image with the camera plane (Figure 4.3). Since each pixel in the rendered image is the weighted sum of the pixels coming from different cameras, it is not possible to define the relation between the rendered image and the original light field images as a projective planar transformation. Therefore, a different strategy should be followed for this case.

The proposed solution is based on the fact that the relation between the watermarks in the rendered image and the original images is same as the projective planar transformation between the focal plane and the image plane of the imagery camera. This can be justified with the following equation relation (see Fig.4.3):

$$I'_r(x_o, y_o) = (1-\alpha)(I_1(s_o, t_o) + W(s_o, t_o)) + \alpha(I_2(s_o, t_o) + W(s_o, t_o)) \quad . \quad (4.3)$$

$$\Rightarrow I'_r(x_o, y_o) = (1-\alpha)I_1(s_o, t_o) + \alpha I_2(s_o, t_o) + W(s_o, t_o) \text{ where } I'_r(x_o, y_o) = I_r(x_o, y_o) + W(x_o, y_o)$$

In (4.3), $W(s_o, t_o)$ refers to the watermark component embedded to each light ray passing through (s_o, t_o) . The other terms are illustrated on Figure 4.3. It should be noted that the corresponding weights for the watermark component during bilinear interpolation sum up to one. Hence, the relation between watermarks in (4.3) is only related with (s_o, t_o) to (x_o, y_o) transformation, which corresponds to planar projective transformation between focal plane and image plane of virtual camera (Figure 4.3).

4.6.1 Determining Planar Projective Transformation between Focal Plane and Image Plane

If the corresponding projective planar transformation between the focal plane and the image plane of the virtual camera is estimated, the watermark on the rendered image can

also be obtained by applying same transformation to the original watermark pattern. For this reason, 3-D points of the object, which are located on (intersecting with) the focal plane, should be detected and utilized (Figure 4.25). In order to determine these points, two properties resulted from light field parameterization can be used. First of all, all light rays of different light field images passing through these points should have similar intensity information. In addition, the coordinates of the pixels in each light field image corresponding to the light ray passing through these points should be close (ideally equal) to each other.

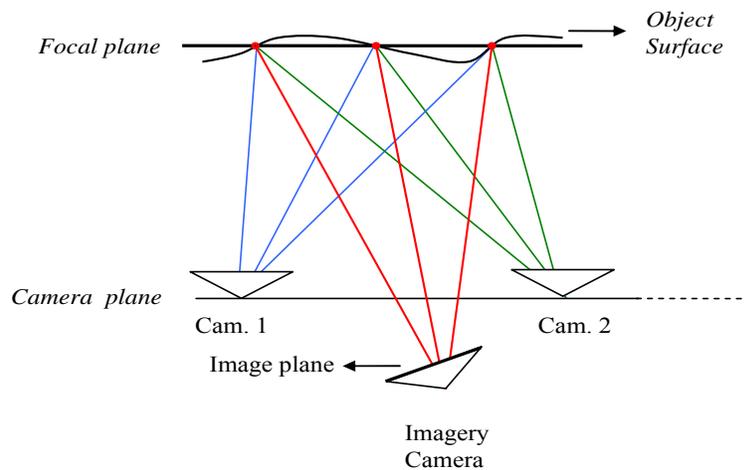


Figure 4.25 An illustration for the points of the 3-D object surface located on the focal plane

Considering the mentioned properties, the proposed algorithm to determine these crossing points is as follows:

- Find matched feature points between the rendered image and each of the light field images [105].
- Determine the light field images that give the first N highest number of feature point matches (during the experiments, N is selected as 4.)
- Form a set of matched points for each of these light field images, $\{S_1 \dots S_N\}$

- Determine the common feature points on the rendered image which are elements of each set, S_i .
- From the common feature points, choose the points whose matches in each corresponding light field image have the same pixel location and similar intensity values (Red dots for example in Figure 4.25).

In Figure 4.26, results of the algorithm for the rendered image for a virtual camera (located at $[0\ 0\ 2.4]$ with $[0\ 0\ 1]$ orientation), is illustrated for *Buddha* light field. The four original light field images which give the maximum number of matched points with the rendered image are *buddha.15.15*, *buddha.16.15*, *buddha.15.16* and *buddha.16.16*. (Note that “*buddha.u.v*” corresponds to the light field image at row u and column v on the camera plane, as in Figure 4.2 (a))

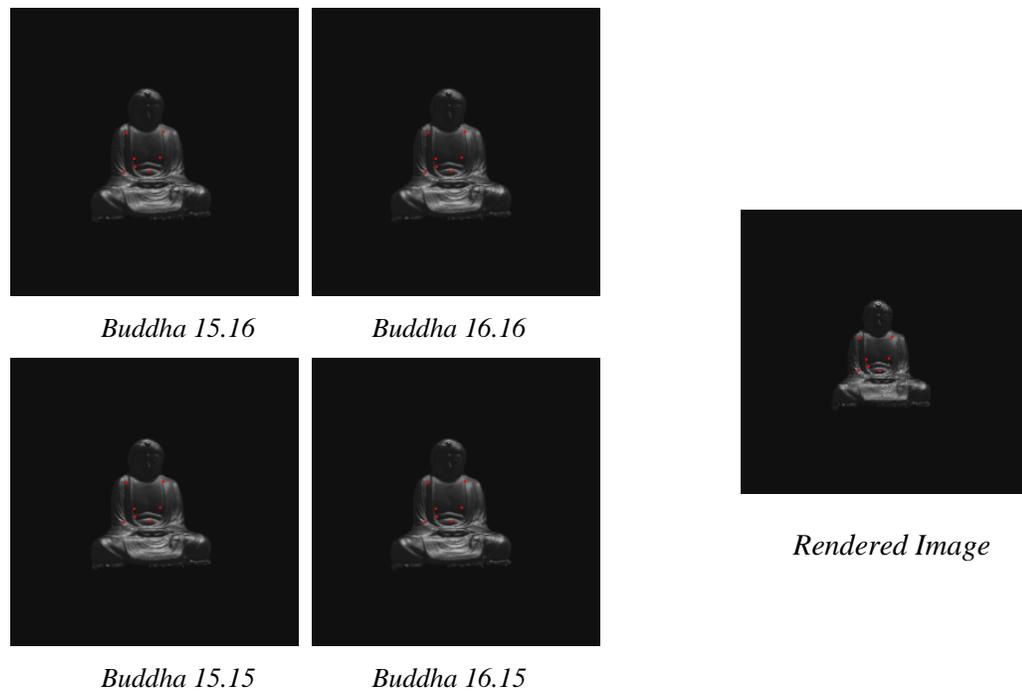


Figure 4.26 The results of the algorithm to find the matched points located on the *focal* plane.

After determination of these points, the homography between the focal plane and the image plane (Figure 4.3) is estimated by a model fit algorithm [101]. In the proposed approach, we utilize a search by randomly picking 4 correspondences and estimating a homography between the selected pairs. Then, the total re-projection error is determined by applying the estimated homography to all of the pairs. This operation is iterated for a number of times and the model, which results in the minimum error, is chosen, as the homography between the *focal* plane and the *image* plane.

4.6.2 Robustness Results

The estimated projective transformation between the focal plane and image plane is applied to the original watermark, W , to compute the rendered watermark W_{ren} . Then, the normalized correlation between the rendered watermark and the high-pass filtered rendered image is computed. Since there can be some small shifts in the calculation of W_{ren} due to the homography estimation, the normalized correlation is computed by using the magnitude coefficients of 2-D *Fourier* transformation of the images, which are invariant to the translations in the pixel locations [10]. The results for the rendered images for different imagery camera positions and orientations are presented in Figure 4.27.

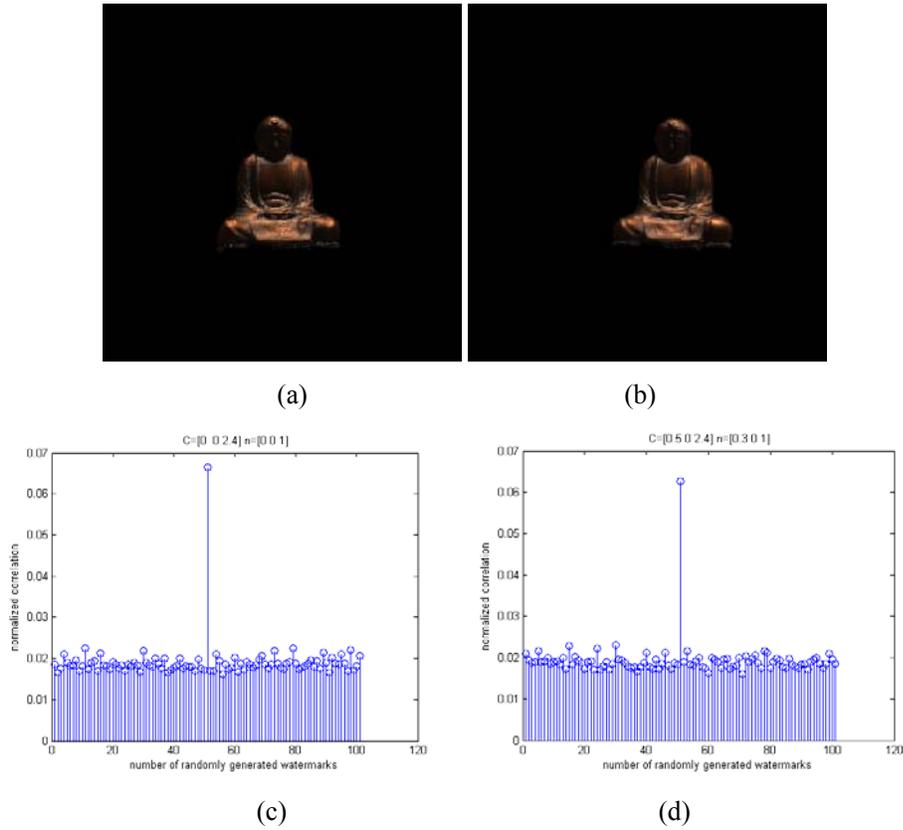


Figure 4.27 Rendered Image: **(a)** Camera position = $[0 \ 0 \ 2.4]$. Image normal = $[0 \ 0 \ 1]$ **(b)** Camera position = $[0.5 \ 0 \ 2.4]$. Image normal = $[0.3 \ 0 \ 1]$. **(c), (d)** Normalized correlation results (Eq. 4.2) for the images in (a) and (b).

4.7 Conclusions

In this chapter, a novel problem, denoted as *free-view watermarking*, is introduced. The specific problems in imperceptibility and robustness requirements for *free-view watermarking* are presented. Assuming that the position and orientation for the imagery view is known, the proposed method extracts the watermark successfully from an arbitrarily rendered image. In order to extend the method for the case of an unknown imagery camera position and orientation, the variations on the watermark pattern during nearest neighborhood- and bilinear interpolation-based IBR are analyzed. Based on the analysis, the relation between the original watermark pattern and the watermark

component on the rendered image is obtained as different forms of projective planar transformation. The solution of the problem is achieved by utilizing the determined projective planar transformation. The embedded watermark is detected successfully for any unknown imagery camera position and orientation. The proposed method mainly focuses on the watermark insertion and extraction for the static scenes consisting of only one object, which forms the fundamental case in the development of IBR technology. The encouraging simulation results promise not only a novel method, but also a new direction for the watermarking research.

CHAPTER 5

3D-2D PROJECTIVE INVARIANT WATERMARKING FOR 3D OBJECTS

This chapter presents a novel method within a different class in 3D watermarking, namely 3D-2D, which embeds a watermark into the 3-D representation of the object and detects the watermark from the 2-D projection (image) of this model. This class of 3D watermarking algorithms is particularly challenging, since 3-D information might not be directly available at the detector. The method proposed in this chapter is based on projective invariance. In other words, the method alters 3-D properties of a model such that the invariant measures extracted from the 2D projection are the same, for all possible 2-D (image) projections. In particular, a well-known projective invariant, *5-point cross-ratio* [106] is used during watermark embedding stage. The advantages of the proposed method compared to previous studies are also examined in the following sections.

The chapter begins with an overview of a few studies in 3D-2D watermarking. Section 5.2 briefly summarizes key results of projective invariants necessary for this chapter. Section 5.3 describes the proposed algorithm and Section 5.4 provides the experimental results with the proposed method. Section 5.5 concludes with a summary and discussion of these results.

5.1 Previous Work on 3D-2D Watermarking

The previous attempts on 3D-2D watermarking consist of only a few methods. As the pioneering research, the method in [46] embeds the watermark into the apparent contour of a 3D object. In particular, this scheme extracts the 3D silhouette of the object for a number of viewing angles and insert watermark into the extracted silhouette for each viewing angle to obtain the 3D watermarked object [46]. During the detection step, given

a 2D test view, the method extracts its 2D contour and finds the most suitable silhouette among the previous watermarked examples to be able to detect the watermark. The correlation between the extracted contour and the most suitable silhouette shows whether the tested 2D view is watermarked or not.

Although, the aforementioned method [46] is a representative of 3D-2D watermarking, some important steps of this method is still incomplete. For example, the optimum way for the selection of the best viewing angles and determining the closest silhouette to the given test image is not addressed in the proposed method. In addition, the capability of silhouettes to represent the visual content of a 3D object should be questioned in the method, since the objects that have the same geometrical structure, but different textures might yield identical silhouettes.

In another method for 3D-2D watermarking [45], the watermark is embedded into the texture information of a 3D object. The watermark detection step consists of extracting the watermark from the recovered texture by using 2D projections (images) of the object [45]. This method assumes that the original 3D object is available at the detector. The core part of the proposed method in [45] is the estimation of the perspective projection between the original 3D object in the detector and the available 2D projection.

In order to estimate the projective transformation, minimum six point correspondences between the 3D surface and the given 2D projection should be accurately determined [107]. For obtaining these point correspondences, the method first generates an approximate 2D projection of the 3D model onto the given 2D projection and then performs a simple block matching between the two views to find the point correspondences. Since the approximate view is generated with a known estimate of the projection matrix, the point correspondences in the approximate view are equivalent to that of the 3D surface. This procedure yields the required point correspondences between the 3D surface and the given 2D projection in order to update the projection matrix. Then, the resulting projection matrix is used to generate the new approximate view and the procedure is repeated iteratively for a number of times until the mean distortion between the estimated and real projections is small enough. The disadvantage of the

proposed method [45] is due to the manual selection of six point correspondences during the initialization to estimate the projective transformation.

This chapter proposes an alternative method within 3D-2D watermarking, which is based on projective invariants. The cross-ratio of 5 coplanar points¹ is utilized as a projective invariant to embed the watermark. The advantage of this approach is that the detector is not required to know or infer the exact projective mapping from the 3D model to the 2D image. Rather, it is sufficient for the detector to infer the point correspondences between model and image points prior to calculating the cross-ratio. The next section briefly summarizes key properties of projective invariants.

5.2 Projective Invariants

Perspective projection is mostly used to model the image formation process that occurs in (pin-hole) cameras. Hence, it is also commonly used during the rendering of 3D models². Projective invariants are properties of geometrical configurations which are unchanged under perspective transformation. Such invariants have received considerable attention from the computer vision research community and various configurations of point-sets, line-set [106] and curves [107] are quite well studied in object recognition applications [109].

Among these geometrical configurations, point-set invariants are considered to be very amenable to watermarking, although lines and curves might also be handled in the future. Projective invariants for point configurations are defined for (i) points on a line, (ii) points on a plane, and (iii) points in an arbitrary position in 3D space. The basic

¹ Note that recent paper by Pantuwong [108] described the use of 4-point invariants for image watermarking to achieve robustness against planar projective transformation. The framework in this thesis is completely different. While the method in [108] utilizes 4 point projective invariant to achieve watermark synchronization in the images distorted by planar projective transformation, the main focus of this thesis is to protect a 3D model by means of inserting a watermark into the projective invariants of the model and extracting the watermark directly from these invariants in any 2D projection of the model.

² Other projective transformations can also be used to approximate the image formation process, e.g. parallel orthographic projection. Projective invariants also exist for this and other image formation models. It is straightforward to generalize this work to such models, as well.

projective invariant for point configurations is the *cross-ratio of four collinear points* [110]. For the given 4 collinear points in Figure 5.1, their cross-ratio is defined as,

$$cr(x_1, x_2, x_3, x_4) = \frac{\overline{x_3 x_1} \cdot \overline{x_4 x_2}}{\overline{x_3 x_2} \cdot \overline{x_4 x_1}} \quad (5.1)$$

where $\overline{x_i x_j}$ is the signed *Euclidian* distance between x_i and x_j [110].

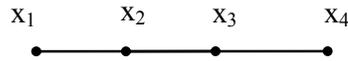


Figure 5.1 Four collinear points

Another projective invariant for coplanar points can also be defined for five points on the same plane. *5-point cross-ratio* is obtained by first selecting one of the five points and joining this point to each of the other four points on the plane [106]. If an arbitrary line is intersected with the formed four lines, then the cross-ratio of the set of four intersection points (y_1, y_2, y_3, y_4 , as in Figure 5.2) is the projective invariant of the original five points (x_1, x_2, x_3, x_4, x_5 , as in Figure 5.2). It should be noted that the cross-ratio is independent from the arbitrary line.

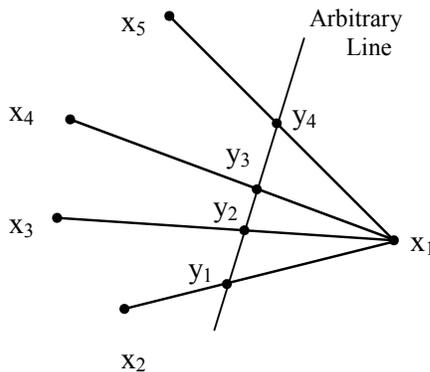


Figure 5.2 Five coplanar points and the resulting 5-point cross-ratio

Finally, a projective invariant for the points in *arbitrary* positions in 3D space can also be defined for six points in a similar way, as in the coplanar five point case [106]. If two points are selected as the base points, then for any other point, there exists a unique plane passing through that point and these two base points. The other four points give rise to four planes all containing the line joining these two base points. If an arbitrary line in 3D space, which is skew to the line joining the two points, is intersected with these four planes, then the cross-ratio of the four intersection points is a projective invariant of the six original points in 3D space [106].

Considering the projective invariants for these different point configurations, the appropriate invariant to utilize in 3D watermarking application is the one for the 5 coplanar points. It has been shown that no projective invariant of arbitrary point sets for a 3D object can be computed from a single view of the 3D object [111]. In other words, cross-ratio of six points in arbitrary positions in 3D space is not suitable to use in a watermarking application by utilizing only a single projection of a 3D object. Regarding with the cross-ratio of the 4 collinear points, a 3D object usually might not contain 4 collinear points. In addition, the number of collinear points on a 3D object may not be sufficient to represent the visual significance of a 3D object in a watermarking application.

5.3 Proposed 3D-2D Watermarking Method

The proposed approach utilizes the cross-ratio of five points in order to embed imperceptible information into a 3D model. In general, the method aims to alter the cross-ratio of five coplanar points on a 3D model, such that the cross-ratio extracted from any 2D projection is equal to that of the 3D model.

Considering these requirements, the watermark embedding procedure should consist of the following steps:

- Identifying five coplanar feature points on the 3D object
- Changing the location of the feature points on the model in order to change the cross-ratio to a value that will represent the embedded information.

Similarly, the watermark detection procedure consists of these steps:

- Identifying the same five co-planar points in the 2D image or video
- Calculating the associated cross-ratio to extract the information due to the value of this ratio

The overall structure of the proposed watermarking method is given in Figure 5.3. These steps are explained in detail in the preceding sections.

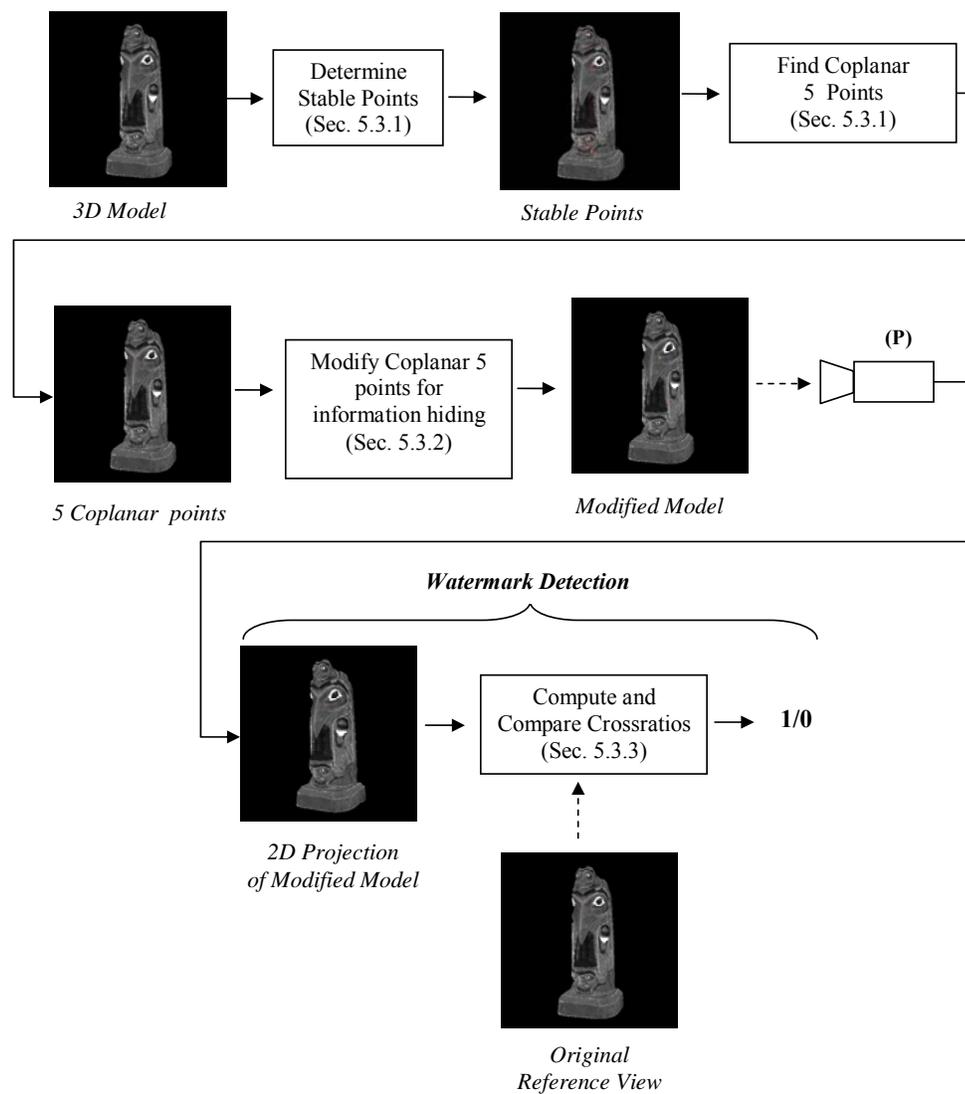


Figure 5.3 Overall structure of the watermark insertion and detection process

5.3.1 Identification of Coplanar Five Points at the Embedder

Identification of the five coplanar points is strictly required both at the watermark embedder and the detector. At the embedder, the complete 3D model is available. However, this situation may not be possible at the detector depending on the application.

The most trivial method for identifying five co-planar points on a 3D model is simply to intersect the model with a plane and extract point features which lie on the boundaries of the resulting intersection. Obviously, there is infinitely many planes for the intersection and numerous potential feature points.

For the selection of five co-planar points, the key consideration is the stability of the points, i.e. the points should remain visible between views, as well easy to detect and identify over a wide range of viewing angles. Unfortunately, determining the stability of 3D feature points from their 2D projections is quite difficult to assess, since many perspective projections must be generated and the visibility of the 3D feature point must be confirmed for each view. Hence, a reverse strategy is applied in which the features are detected in different 2D projections of the 3D model and checked for their repeatability.

The first step in such a procedure is to find the stable points which remain visible over a range of viewing angles. For this aim, the following strategy is proposed:

- A range for the angles is selected and divided into equal sized angles (see Figure 5.4).
- 2D views of the model for all these angle samples within the range are rendered to generate images (Figure 5.5).
- A feature extraction algorithm (Scale Invariant Feature Transform, *SIFT* [105]) is applied for each view to produce a candidate set of feature points.
- A reference view is selected for the given angle range and the feature points for each view in the angle range are matched with the ones in the reference view by using a matching algorithm [105] (Figure 5.6).
- The feature points on the reference view which have matches in *all* other views are assumed to be *stable* points of the model for that angle range (Figure 5.7)

In this procedure, SIFT algorithm in [105], which is invariant to scaling, rotation and changes in the illumination, is selected for feature point extraction and matching due to its frequently reported high performance.

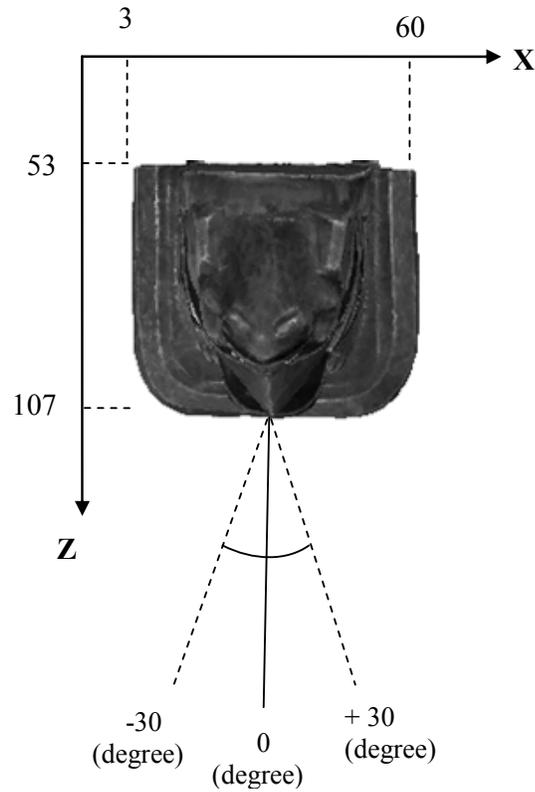


Figure 5.4 Selected angle range for *Totem* textured object [112]. (Top view of the object. *Y* axis is towards the outside of the paper plane.). Angle range 1: $[0,30]$ degree; sample angles (deg.): $[0,5,10,15,20,25,30]$; ref. angle: 20 deg. ; angle range 2: $[0,-30]$ degree; sample angles (deg.): $[-5,-10,-15,-20,-25,-30]$; ref. angle: -20 deg.



(a) 0 deg.



(b) 10 deg.



(c) 20 deg. (reference view)



(d) 30 deg.

Figure 5.5 2D views of *Totem* object for the viewing angles (0, 10, 20, 30) degrees in the angle range (0, 30) degree shown in Figure 5.3



(a) 0 deg.



(b) 10 deg.



(c) 20 deg. (reference view)



(d) 30 deg.

Figure 5.6 Matched feature points between the reference view (c) and other views.



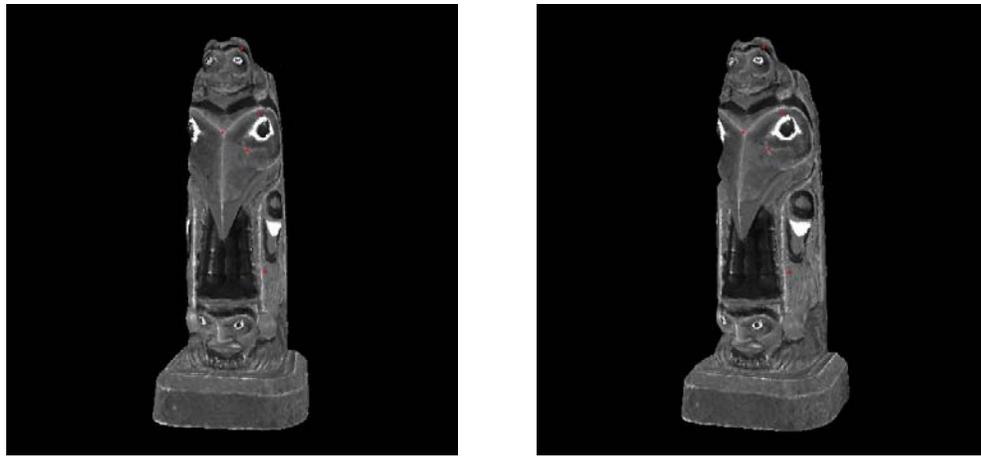
Figure 5.7 Stable points of 3D object for the corresponding angle range. (i.e. feature points on the reference view which have matches in all of the other views in the corresponding angle range.)

After the determination of the stable feature points, the next step is the identification of 5 coplanar points on the 3D model. Although it is possible to use more sophisticated methods to select five coplanar points from the matched feature points, a relatively simple approach is proposed:

- Five feature points of the reference view is randomly selected from the feature set and their cross-ratio is computed.
- Cross-ratio of the corresponding five points in another sample view in the angle range is also computed.
- If the cross ratios are approximately equal to each other, it is further checked whether they belong to the same plane in 3D coordinates. Coplanarity is checked as follows:

- Using the 3D coordinates of each pixel in the model, 10 different planes, which must occur from any 3 points out of 5 points, the plane which gives minimum average distance of the remaining 2 points is selected.
- If the average distance is less than a threshold, then selected five points are assumed to be on the same plane.

An example for the selected five points is shown in Figure 5.8



(a) 10 deg.

(b) 20 deg. (Ref. view)

Figure 5.8 Five coplanar points (shown in red color) for the angle range 1 shown in Figure 5.4. The cameras are located at (77, 109, 338) and (121, 109, 326), respectively.

5.3.2 Modification of the Feature Point Locations by the Embedder

While modifying the positions of the feature points, some restrictions should be considered, so that the same points could still be detected after the modification. The first restriction is to preserve the intensity values in the neighborhood of the feature points. Therefore, while modifying the position of a feature point, the locations of the pixels in its neighborhood should also be changed by the same amount. Another restriction applies for the direction of the displacement during the point location change. All the corresponding 3D locations of the neighborhood pixels should be moved in the same

direction, which must be parallel to the plane, where the five selected points are located, in order to preserve the coplanarity of these points after positional modification. These restrictions can be extended by also considering the curvature characteristics of the surface around a selected feature point and imposing the invisibility requirement for the distortion at the projected images.

In this study, only the first two restrictions are considered. Hence, one of the five coplanar points on the projected image is selected and the corresponding nodes for that feature point and its neighbor pixels are determined on the available mesh model. In fact, considering the invisibility requirement, the selection of the feature point could be more constrained, based on the smoothness around the feature points, which remains as a future direction. Then, all the corresponding nodes of the mesh model are displaced in a direction parallel to the plane where the five points are located. In this manner, the texture for each triangle in the mesh model is preserved during the displacement of the triangle nodes in the model. An example showing the changes in the location of a feature point is given in Figure 5.9 for the viewing direction of 20 degrees. Another example is given for the viewing direction of -10 degrees in Figure 5.10.

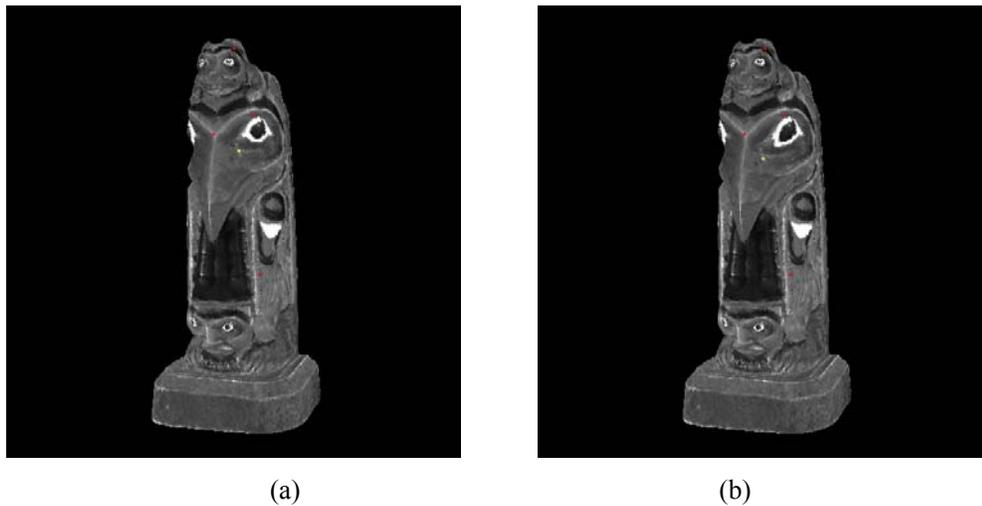


Figure 5.9 Feature point whose location is changed is shown in yellow color. The images (a) before and (b) after point location modification. (viewing angle= 20 degrees.)

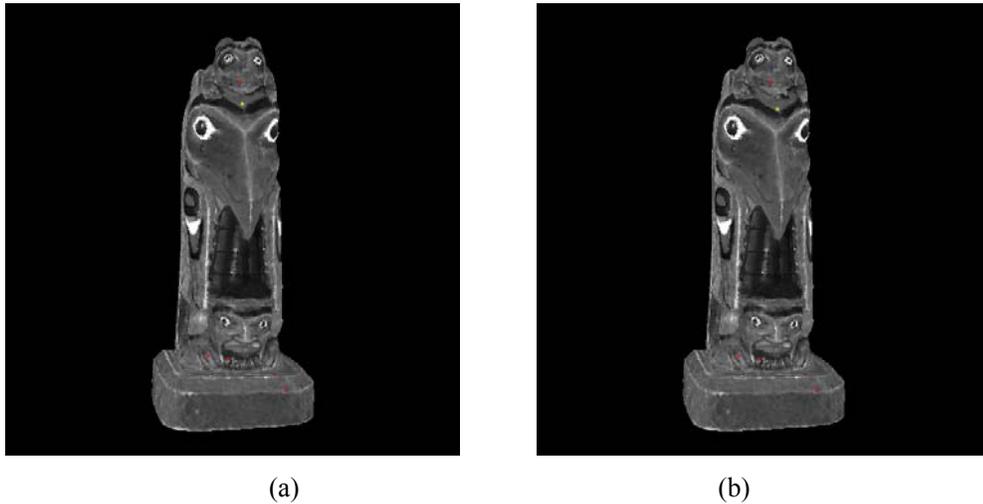


Figure 5.10 Another example for the modification on the location of coplanar five points for the angle range 0- (-30) degree. The images (a) before and (b) after point location modification. (viewing angle = -10 degrees.)

5.3.3 Identification at the Detector

Based on the data available at the extractor, there are several cases that should be considered separately. For different applications, only the following data could be available at the detector:

- A single 2D projection of the watermarked model
- Two (or more) 2D projections of the watermarked model
- A single 2D projection of the watermarked model and the original 3D model

The first case, a single 2D (projection) image, is the most difficult scenario, due to the fact that it is unclear for the detector the way to determine the coplanar five points after applying the feature extractor on this image. One solution to this problem might be tagging the five coplanar points in a projection invariant manner. For example, in some circumstances, it might be possible to assign a specific intensity value to all five pixels. Another option is to modify the local neighborhood around each of these five points to a

predetermined value. Obviously, such a tagging procedure has limitations, especially due to producing perceptible artifacts. This case is not considered further in this thesis.

The second case might occur more frequently; for example, computer graphic and animations create a sequence of images to produce a video, i.e. multiple 2D projections are available. In this case, five coplanar points on the 3D object might be obtained by comparing only the cross ratios of five points in two different projections of the object, without an explicit necessity for the 3D information [106]. However, the main problem is to identify the utilized feature points during watermarking apart from all the other coplanar points. A possible strategy might be distorting the coplanarity of feature points other than the ones used for watermarking during the embedding stage. The disadvantage of such an approach is creating visible artifacts, especially on the objects consist of planar surfaces. Another solution might be smoothing the texture and intensity gradients around the coplanar feature points other than the ones used for watermarking during watermark embedding stage, in order to force the extractor to find only five coplanar feature points during detection. The drawback is again possible visible distortions on 3D objects containing high contrast textures. Due to aforementioned problems, a generic solution for this case seems also very challenging.

The third case corresponds to an informed detector. A number of applications might support the informed detection, e.g. traitor tracing. However, if the 3D model is available, it is necessary to determine the correspondence between the 2D image and 3D model, which might be difficult. In fact, the correspondence problem for 3D-2D is less mature, compared to 2D-2D correspondence problem [45]. Therefore, a slightly modified version for such a detector is proposed, which utilizes the 2D projections of the original 3D object from a number of viewing angles, instead of using the 3D object itself.

In the proposed method, the detector contains the projection of original 3D object for the reference viewing angle in each angle range. For instance, the reference viewing angles for the angle ranges (0, 30) and (0,-30) degrees for *Totem* object in Figure 5.4, are chosen as 20 and -20 degrees, respectively. The detector knows a priori the positions of the feature points on the reference views that were used during watermarking.

Given a test image, the detector first determines the closest reference view to the test image which gives the maximum number of matched feature points with this image. Then, the feature points used for watermarking on the closest reference view are matched with the ones on the test data. Note that the matching algorithms [105] usually use the similarities between the intensity changes around the feature points during matching process. Therefore, the intensity values around the modified feature point are preserved during the watermark embedding stage by only changing the location of the mesh vertices around the corresponding feature point, without modifying the texture, as mentioned before. Hence, the matching between the projections of the watermarked and original model in the detector is achieved successfully as illustrated in Figure 5.11. (See Figure 5.9 for a comparison.)



(a) Original projection (20 deg., ref. view)



(b) Watermarked projection (10 deg.)



(c) Watermarked projection (20 deg.)



(d) Watermarked projection (30 deg.)

Figure 5.11 Matched feature points between the feature points used for watermarking on the reference view and the ones on different projections of watermarked model.

5.4 Simulation Results

During the experiments, one-bit information is embedded into the 3D model which indicates whether the projection of the 3D model is watermarked or not. For this purpose, only the location of one feature point among five coplanar points on the 3D model is modified during watermark embedding stage. Then, a projection of the watermarked model is generated and the cross ratios of the five points on the watermarked projection and the reference view in the detector are compared. These tests are repeated for all the angle samples in the specified angle range.

The experimental procedure is as follows:

- Given the original 3D model, determine the angle ranges and specify a reference viewing angle for each angle range (Figure 5.4).
- For the angle range, find the stable points on the 3D object (Figure 5.5, 5.6, 5.7).
- Identify five coplanar points among the stable points (Figure 5.8).
- Modify the location of one feature point among the five coplanar points (Figure 5.9)
- For each of the angle sample in the angle range, generate the 2D projection of the modified (watermarked) 3D model.

Then, during detection,

- For each watermarked projection, match the feature points on the reference view in the detector to the feature points on a test image (Figure 5.11)
- Compute the cross ratio of matched five points on the tested image.

The cross ratio results for the angle range of 0-30 degrees for *Totem* object (see Figure 5.3) are plotted in Figure 5.12 both for the original and modified models. For each of the angle sample, which are shown with crosses in the graph, feature points are obtained at the modified locations on the generated 2D views. While the cross ratio shows a small variation with respect to the different viewing angles, the significant gap between the two graphs for the original and modified model is observable. The cross-ratio results for the

angle range between 0 to -30 degrees are given in Figure 5.13. The results indicate that the proposed method achieves watermarking of 3D models by using cross-ratio of five points and identifies quite well, whether the projection of a 3D model is watermarked or not.

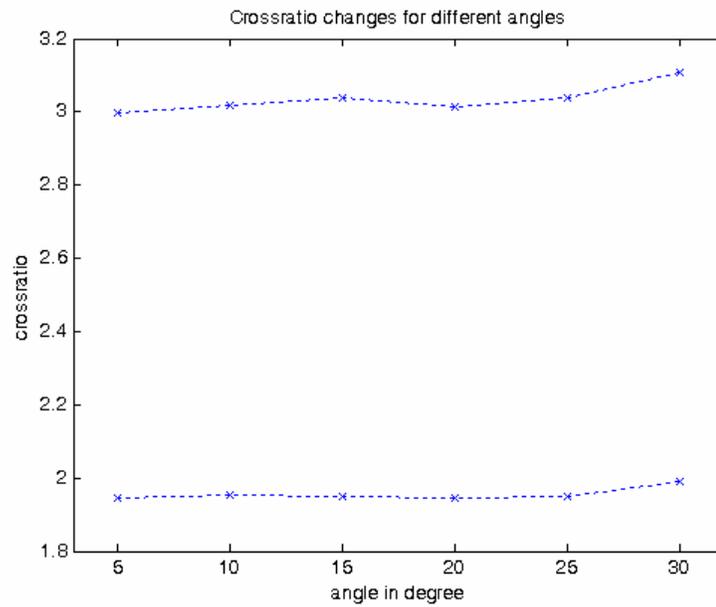


Figure 5.12 Cross ratio changes between 0 – 30 degree for the projections of original and modified model. Note that cross ratio is unitless. See (5.1).

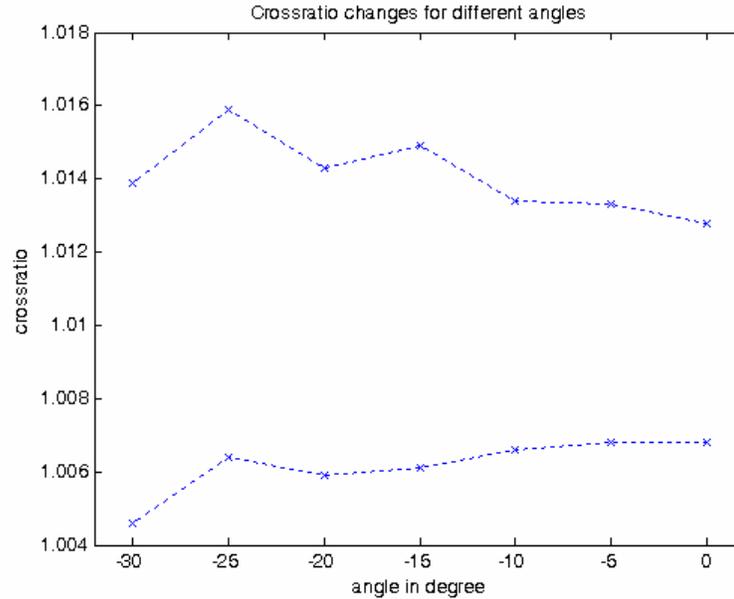


Figure 5.13 Cross ratio changes between 0 – (-30) degree for the projections of original and modified model

5.5 Conclusions and Future Directions

In this chapter, the utilization of projective invariants for copyright protection of 3D objects by watermarking is presented. One bit information is embedded into the 3D model by modifying the location of a feature point among five coplanar points on the same model. This information could be extracted from any 2D projection of the 3D model by comparing the cross-ratio value with that of the original.

Considering the quite limited studies in the literature and vast area of applications, such as the right management problems in computer animation, computer gaming, and virtual and augmented realities in 3D-2D watermarking category, the use of projective invariants for watermarking forms an alternative direction for watermarking research. Although, in principle, the ability of projective invariants for information embedding is justified in this study, the future directions might include more sophisticated solutions.

In particular, detection of 5 coplanar points might be achieved automatically in a more efficient way. Moreover, the case, in which the absence of five coplanar features points, should also be addressed. A possible solution might be intersecting the 3D object with a plane and performing some intended perturbations in the intersection contours to generate such coplanar feature points.

A detailed analysis on the selection of the displacement direction, while modifying the location of feature points, could also be performed. Preserving connectivity of the nodes on the triangle meshes and preserving surface curvature should be taken into account in this step. It is unfortunate that there is no detailed research in the literature about the visibility of the distortions on a 3D object. In addition, the invisibility problems in 2D projections of a 3D model, while modifying the 3D model, should be also investigated.

Finally, the maximum amount of information that a cross-ratio could carry requires more attention. In this study, only one point among five points is changed to embed a single bit of information. Considering the other points and different cross ratio levels, the maximum number of bits that can be embedded into a 3D object might be examined, as a future research effort.

CHAPTER 6

SUMMARY and CONCLUSIONS

This thesis proposes new techniques in two main areas of watermarking, namely, video and 3D watermarking. For each area, the specific requirements are presented, the state-of-the-art methods are examined, a categorization among these methods is achieved and novel and superior techniques are developed, considering the specific requirements of each area.

Regarding video watermarking, a new method which solves the usual flickering type of distortions in existing video watermarking schemes by utilizing the temporal sensitivity of human visual system (HVS) is developed. The experimental results on robustness, imperceptibility and capacity of the methods clearly indicate that the proposed method is superior to the compared two other well-known methods [20], [33] from the literature. Based on the simulation results, one might conclude that integrating temporal characteristics of HVS into a watermarking scheme increases the robustness against common temporal signal processing operations, in terms of bit-error-rate and mainly solves the imperceptibility problem in video watermarking.

It should be noted that the proposed method is applied to the video frames between shot boundaries in order to utilize the temporally stationary signals. As an extension, the proposed method can be improved by using not only the temporal contrast but also the temporal masking characteristics of HVS, which corresponds to a significant decrease in the sensitivity of HVS at these shot boundaries. Moreover, the method can be further improved by embedding the watermark into the chromatic components, as well as the luminance component of the video.

In the thesis, 3D watermarking categorization is achieved based on the fundamental components of 3D representation techniques, namely 3D-3D, 3D-2D and 2D-2D watermarking. The thesis presents a detailed literature survey on 3D-3D watermarking.

Although most of the 3D watermarking methods in the literature belong to this category, the literature has still lack of such a complete survey for 3D-3D watermarking. Therefore, rather than investigating new problems and methods in this relatively mature category compared to the other two new categories, a rigorous state-of-the-art analysis is presented.

The presented survey points out that 3D-3D geometry watermarking still have some specific problems, compared to image and video watermarking. These problems might be summarized as the lack of a unique 3D scene representation, standardization for the coding schemes and benchmarking tools on 3D geometry watermarking. In addition, the invisibility of distortions on the 3D models due to watermarking should also be investigated by the vision researches.

An important consequence of the presented overview is that the current 3D-3D geometry watermarking methods are mostly proposed for 3D model-based applications, such as VRML-type of internet applications. On the other hand, during the recent years, the research efforts on the promising applications of 3D representations, such as 3DTV or FTV, have considerably increased. In this scenario, the right management problems for such 3D representations is expected to become more critical. Considering the fact that the 3D geometry-based representations also constitute a suitable framework for the upcoming 3DTV technologies, some suitable extensions of the mentioned 3D-3D geometry watermarking methods could be good candidates for the solution of the copyright problem and other related problems, such as authentication, content labeling, and content protection, in these new application areas. However, the presented overview has indicated that the extension of the existing watermarking methods for a 3DTV application should require more sophisticated research efforts on standardization, benchmarking, invisibility, and finally, real time necessity, which is not handled in any of the current 3D-3D geometry watermarking schemes.

One of the main contributions of this thesis for the watermarking research is achieved within 2D-2D watermarking category. A novel problem is introduced for watermarking research, as a solution to the copyright protection of multi-view video content for the emerging *Free View Televisions*, in which TV-viewers freely select the viewing position

and angle for the transmitted multi-view video. It should be noted that the related the research results on FTVs has already reached to an acceptable maturity both in terms of standardization and the realization of such systems. Recently, ISO and ITU bodies have delivered a standardization on multi-view coding and Free-view TV is expected to be the next goal for standardization. In conclusion, it is highly possible that the copyright problems for such technologies become an unavoidable reality in the upcoming years. In this aspect, the proposed watermarking method in this thesis is particularly important for presenting the basic problems and requirements, as well as proposing a novel scheme for its solution.

The proposed free-view watermarking method mainly focuses on the watermark insertion and extraction stages for the static scenes, consisting of only a single object which forms the fundamental case in the development of IBR technology. As a future research direction, the extension of the solution for dynamic scenes with multiple objects could be examined, in which different rendering and interpolation techniques are being used. In addition, it should be noted that the other main approaches for FTVs, which is based on depth image-based rendering [113], should also be investigated as a future direction.

In the last category of 3D watermarking, the thesis presents 3D-2D watermarking scenario which embeds a watermark into the 3-D representation of the object and detects the watermark on the 2-D projection (image) of the model. The thesis proposes a novel solution based on projective invariants. Considering the limited studies in the literature, but a large area of applications, such as the right management problems in computer animation, computer gaming, and virtual and augmented realities in 3D-2D watermarking category, the use of projective invariants for watermarking constitutes an alternative direction for the watermarking research. In fact, the related research on projective invariants for object recognition applications in computer vision indicates the efficient utilization of projective invariants and may form a good background for watermarking research in this category [109].

Although, in principle, the ability of the projective invariants for information embedding is justified via simulations in the thesis study, the future directions might include more sophisticated solutions. One of the achievements of this thesis is also addressing these

future directions, as well as presenting the basic requirements in this category of 3D watermarking.

In general, while most of the 3D watermarking techniques in the literature belong to *3D-3D watermarking* group, *3D-2D* and *2D-2D watermarking* have received limited attention from the watermarking research. However, once a 3D model is used in a computer animated video in any application, such as animations, cartoons, simulators, its projection is usually more precious than the 3D model itself. In addition, considering the increasing research and progress in virtual view synthesis and its applications, as free-view TV, it is highly expected that *3D-2D* and *2D-2D* watermarking will be more attractive for the implicit copyright protection of 3D representations. The exceptional contributions of this thesis are to present the basic requirements and problems, and propose pioneering novel solutions to these new areas of 3D watermarking.

APPENDIX A

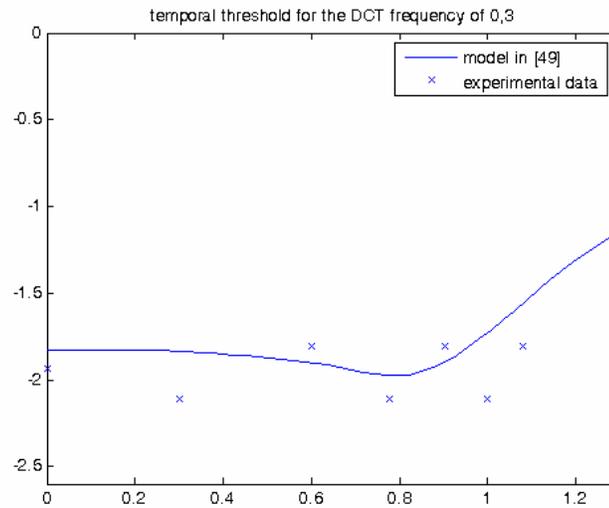
The visual experiments in [49], [50] are known to be conducted in an environment, in which the display frame rate is 120 Hz, display resolution is 32 pixels/degree, mean luminance is 50 cd/m² and viewing is binocular with natural pupils from a distance of 91.5 cm. The mean luminance of the display is set to 50 cd/m² and the amplitudes of the sinusoidal patterns are increased accordingly, in the units of cd/m².

In order to validate the given model and familiarize on temporal contrast thresholds, the authors also perform similar visual tests, where computer-generated visual patterns of 8-bit resolution (only Y-component) and mean intensity of 128 are used as a visual stimulus on a computer screen. The frame-rate of the visual pattern is set to 25 frames/sec. Therefore, temporal contrast thresholds can only be a multiple of $1/128$ and the continuous temporal frequencies can be a multiple of $1/25$ Hz, which remains to be the main difference compared to the set-up in [49]. Each visual pattern for spatial and temporal frequencies consists of 16 frames. During these tests, the display-rate and color temperature of the monitor is set to 75 Hz and 9300°K, respectively. The display resolution is 32 pixels/degree and viewing is binocular with natural pupils from a distance of 90 cm, similar to [49]. Lighting condition is the standard room lighting with one lighted lamp resulting about 200 lux illumination on the screen [114].

Visual threshold estimates are obtained by a similar *forced-choice stair-case procedure* [115]. The subjects are forced to answer, either *Yes* or *No*, whether the temporal variation in the modulated target is distinguishable, beginning from a target with zero contrast. When the response is negative, the contrast of the sinusoid is increased in steps of $1/128$ until the target becomes visible to the subject. In each step, the visual pattern are presented only once to the subject. A total of 15 viewers [114] took part in the tests. The subjects consist of 7 male and 7 female second year university students, and a 44 years old electrical technician, who have no experience on image processing and subjective visual tests. The contrast, where the ratio of the positive answers is just greater than 50 %, is assumed to be the contrast threshold for that spatial and temporal frequency.

The resulting experimental data is shown in Figure A.1, A.2 and A.3 for the DCT frequencies of {0,3}, {0,7}, {3,3}, and temporal frequencies of 1, 2, 4, 6, 8, 10, 12 Hz., respectively. Since the video frame rate is 25 Hz, the temporal frequency can only be up to 12.5 Hz in the tested case to avoid aliasing; hence, the data after this temporal frequency is not utilized. Based on Figure A.1, the results in [49] and these simulations verify each other in a good extend. The visual patterns created for the experiments are available at:

http://snap-mmrg.eee.metu.edu.tr/paper_ieee/visual_patterns



(c)

Figure A.1 Temporal Contrast Thresholds as a function of temporal frequency (Hz) for the spatial DCT frequency of {0,3}. Points are the threshold estimates as a result of the experiments. The solid line shows the proposed model in [49].

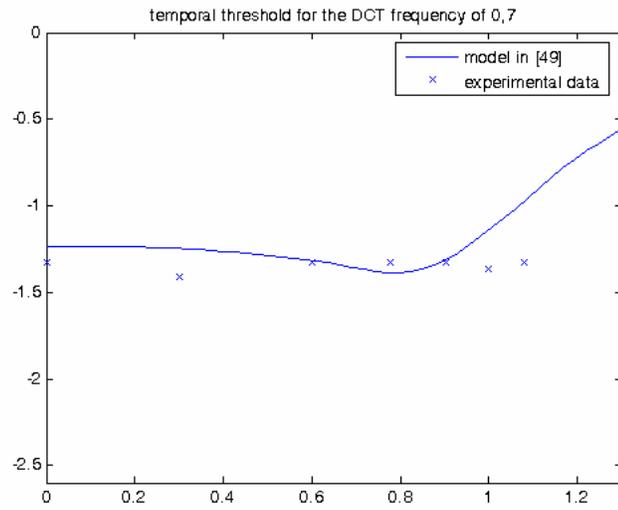


Figure A.2 Temporal contrast thresholds as a function of temporal frequency (Hz) for the spatial DCT frequency of {0,7}.

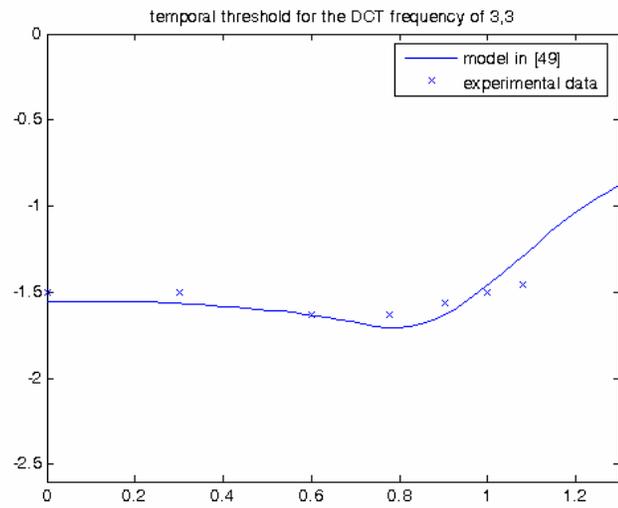


Figure A.3 Temporal contrast thresholds as a function of temporal frequency (Hz) for the spatial DCT frequency of {3,3}.

APPENDIX B

A general configuration for two cameras viewing a planar surface is given in Figure B.1.

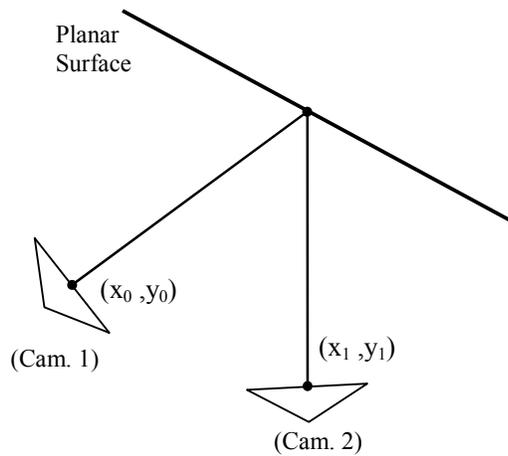


Figure B.1 A general configuration for two cameras viewing a planar surface

The relation between the coordinates of the two camera views is given as planar projective transformation [101]:

$$\begin{bmatrix} x_1 \\ y_1 \\ 1 \end{bmatrix} = H \begin{bmatrix} x_0 \\ y_0 \\ 1 \end{bmatrix}. \quad (\text{B.1})$$

In (B.1), H is a 3×3 matrix corresponding to the planar projective transformation between the (homogeneous) camera coordinates shown in Figure B.1. The matrix H contains 9 entries, but is defined only up to a scale. Thus, total number of degrees of freedom in a planar projective transformation is 8. Given the two camera views related with each other with homography, at least 4 different correspondences between two camera views

are needed to find the homography matrix. Note that each correspondence accounts for 2 constraints corresponding to x and y components [101].

In Figure B.1, if only the first camera view and its homography relation with the second camera are given, then an approximate view can be generated for the second camera by simply assigning the intensity of a pixel in the first image to the pixel of the second image whose coordinates is the homography mapping of the coordinates of the pixel in the first image.

In fact, the procedure to generate an arbitrary view in LFR with nearest neighbor interpolation is similar. If the virtual camera is on the camera plane, then all the light rays (pixels) for the virtual camera will be generated from the nearest neighbor camera view with respect to the intersection of light rays with the focal plane as illustrated in Figure B.2. Hence, the relation between the virtual view and its nearest neighbor camera view is concluded as planar projective transformation.

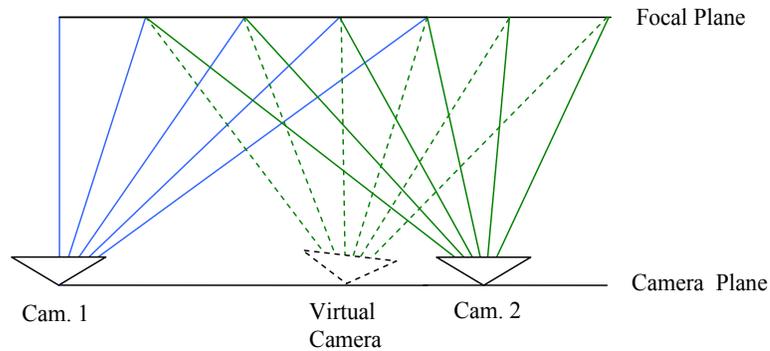


Figure B.2 The pixels of the virtual view are generated from the pixels of nearest neighbor camera view (Cam.2), as shown in green color in the figure.

REFERENCES

- [1] I. J. Cox, M. L. Miller, J. A. Bloom, *Digital Watermarking*, Academic Press, 2002.
- [2] I. J. Cox, M. L. Miller and J. A. Bloom, “Watermarking Applications and their properties”, Proc. of Int. Conf. on Information Technology: Coding and Computing, pp. 6–10, 2000.
- [3] G. C. Langelaar, I. Setyawan and R. L. Lagendijk, “Watermarking Digital Image and Video Data”, IEEE Signal Processing Magazine, vol. 17, pp. 20–46, Sept. 2000.
- [4] G. Doërr and J.-L. Dugelay, “A Guide Tour of Video Watermarking”, in Signal Processing: Image Communication, vol. 18, no. 4, pp. 263–282, Apr. 2003.
- [5] T. Liu, Z.-D. Qiu, “The survey of digital watermarking-based image authentication techniques”, Proc. of IEEE 6’th International Conference on Signal Processing, vol. 2, pp. 1556–1559, Aug. 2002.
- [6] A. Yilmaz and A. A. Alatan, “Error Concealment of Video Sequences by Data Hiding”, Proc. of IEEE Int. Conf. on Image Processing, vol. 2, pp. 679–682, Sept. 2003.
- [7] I. J. Cox, J. Killian, T. Leighton and T. Shamon, “Secure spread spectrum watermarking for multimedia”, IEEE Trans. on Image Processing, vol. 12, no. 6, pp. 1673–1687, Dec. 1997.
- [8] M. D. Swanson, M. Kobayashi, and A. H. Tewfik, “Multimedia Data-Embedding and Watermarking Technologies”, Proceedings of the IEEE, vol. 86, no. 6, pp. 1064–1087, June 1998.
- [9] R. B. Wolfgang, C. I. Podilchuk and E. J. Delp, “Perceptual Watermarks for Image and Video”, Proceedings of the IEEE, vol. 87, no. 7, pp.1108–1126, July 1998.
- [10] M. Maes, T. Kalker, J.-P. M. G. Linnartz, J. Talstra, G. F. G. Depovere, and J. Haitsma, “Digital Watermarking for DVD Video Copy Protection”, IEEE Signal Processing Magazine, vol. 17, no. 5, pp. 47–57, Sept. 2000.
- [11] P. Termont, L. De Strycker, J. Vandewege, J. Haitsma, T. Kalker, M. Maes, G. Depovere, A. Langell, C. Alm, P. Norman, “Performance measurements of a

- real-time digital watermarking system for broadcast monitoring”, Proc. of IEEE Int. Conf. on Multimedia Computing and Systems, vol. 2, pp. 220–224, June 1999.
- [12] M. Maes, T. Kalker, J. Haitsma, G. Depovere, “Exploiting shift invariance to obtain a high payload in digital image watermarking”, IEEE Int. Conf. on Multimedia Computing and Systems, vol. 1, pp. 7–12, June 1999.
- [13] Q. Sun, D. He and Q. Tian, “A Secure and Robust Authentication Scheme for Video Transcoding”, IEEE Trans. on Circuits and Systems for Video Technology, vol. 16, no.10, pp. 1232–1244, Oct. 2006.
- [14] K. Su, D. Kundur, and D. Hatzinakos, “Statistical invisibility for collusion-resistant digital video watermarking”, IEEE Trans. on Multimedia, vol. 7, no. 1, pp. 43–51, Feb. 2005.
- [15] R. G. van Schyndel, A. Z. Tirkel and C. F. Osborne, “A digital watermark”, in Proc. of IEEE Int. Conf. on Image Processing, vol. 2, pp. 86–90, Nov. 1994.
- [16] S. Burgett, E. Koch, and J. Zhao, “Copyright labeling of digitized data”, IEEE Commun. Mag., vol. 36, no. 3, pp. 94–100, Mar. 1998.
- [17] C.-T. Hsu, J.-L. Wu, “DCT-based watermarking for video”, IEEE Trans. on Consumer Electronics, vol. 44, pp. 206–216, Feb. 1998.
- [18] M. Ejima and A. Miyazaki, “A wavelet based Watermarking for Digital Images and Video”, Proc. of Int. Conf. on Image Processing, vol.3, pp. 678–681, Sept. 2000.
- [19] H. Inoue, A. Miyazaki, T. Araki, T. Katsura, “A Digital Watermark Method Using the Wavelet Transform for Video Data”, Proc. of IEEE International Symposium on Circuits and Systems, vol. 4, pp. 247–250, June 1999.
- [20] F. Deguillaume, G. Csurka, J. O’Ruanaidh, T. Pun, “Robust 3D DFT Video Watermarking”, Security and Watermarking of Multimedia Contents, Volume 3657 of SPIE Proceedings, pp. 113–124, Jan. 1999.
- [21] G. C. Langelaar, R. L. Lagendijk and J. Biemond, “Real time labeling of MPEG-2 compressed video”, J. Visual Commun. Image Representation, vol. 9, no.4, pp. 256–270, Dec. 1998.
- [22] C. S. Lu, J. R. Chen, and K. C. Fan, “Real-time frame-dependent video watermarking in VLC domain”, Signal Processing: Image Communication, vol. 20, no. 7, pp. 624–642, 2005.

- [23] S. Biswas, S. R. Das, E. M. Petriu, “An Adaptive Compressed MPEG-2 Video Watermarking Scheme”, *IEEE Trans. on Instrumentation and Measurement*, vol. 54, no. 5, Oct. 2005.
- [24] S. Sakazawa, Y. Takishima and Y. Nakajima, “H.264 Native video watermarking method”, *Proc. of IEEE International Symposium on Circuits and Systems*, Page(s):4 pp., May 2006.
- [25] G. C. Langelaar and R. L. Lagendijk, “Optimal Differential Energy Watermarking of DCT Encoded Images and Video”, *IEEE Trans. on Image Processing*, vol. 10, no.1, pp.148–158, Jan. 2001.
- [26] I. Setyawan, R. L. Lagendijk, “Extended Differential Energy Watermarking (XDEW) algorithm for low bit rate video watermarking”, *Proc. of the Seventh Annual Conference of the Advanced School for Computing and Imaging*, 2001.
- [27] J. Zhang, J. Li, L. Zhang, “Video watermarking technique in motion vector”, in *XIV Brazilian Symp. on Computer Graphics and Image Processing*, pp. 179–182, Oct. 2001.
- [28] Y. Bado, N. Laurent, J.-L. Dugelay, “Watermarking video, hierarchical embedding in motion vectors”, *Proc. of IEEE Int. Conf. on Image Processing*, vol. 2, pp. 739–742, Sept. 2003.
- [29] M. Kutter, F. Jordan and T. Ebrahimi, “Proposal of a Watermarking Technique for Hiding/Retrieving Datas in Compressed and Decompressed Video”, in *ISO/IEC/JTC1/SC29/WG11/MPEG97/M2281*, 1997.
- [30] W. Zhu, Z. Xiong and Y.-Q. Zhang, “Multiresolution watermarking for Images and Video”, *IEEE Trans. on Circuits and Systems for Video Technology*, vol. 9, no. 4, pp. 545–550, June 1999.
- [31] X. Wu, W. Zhu, Z. Xiong, Y.-Q. Zhang, “Object-based multiresolution watermarking of images and video,” *Proc. of IEEE International Symposium on Circuits and Systems*, vol. 1, pp. 212–215, May 2000.
- [32] P. Campisi and A. Neri, “Video watermarking in the 3D-DWT domain using perceptual masking”, *Proc. of IEEE Int. Conf. on Image Processing*, vol. 1, pp. 997–1000, Sept. 2005.
- [33] F. Hartung and B. Girod, “Watermarking of Uncompressed and Compressed Video”, *Signal Processing*, vol. 66, no. 3 (Special issue on Watermarking), pp. 283–301, May 1998.

- [34] A. Piva, R. Cadelli and A. De Rosa, "A DWT based Object Watermarking System for MPEG-4 Video Streams", Proc. of IEEE Int. Conf. on Image Processing, vol. 3, pp. 5–8, Sept. 2000.
- [35] T.-Y. Chung, M.-S. Hong, Y.-N. Oh, D.-H. Shin, S.-H. Park, "Digital watermarking for copyright protection of MPEG2 Compressed video", IEEE Trans. on Consumer Electronics, vol. 44, no. 3, pp. 895–901, Aug. 1998.
- [36] P. Mas and B. Macq, "A new Video-Object Watermarking Scheme Robust to Object Manipulation", Proc. of IEEE Int. Conf. on Image Processing, vol. 3, pp. 526–529, Oct. 2001.
- [37] S. Suthaharan, S. W. Kim, S. Sathanathan, H. K. Lee and K. R. Rao, "Perceptually Tuned Robust Watermarking Scheme Digital Video using Motion Entropy Masking", Proc. of IEEE Int. Conf. on Consumer Electronics, pp. 104–105, June 1999.
- [38] A. Koz and A. A. Alatan, "Foveated Image Watermarking", Proc. of IEEE Int. Conf. on Image Processing, vol. 3, pp. 657–660, June 2002.
- [39] M. D. Swanson, B. Zhu and A. H. Tewfik, "Multiresolution Scene-Based Video Watermarking using perceptual Models", IEEE Journal on Selected Areas in Communications, vol. 16, no. 4, pp. 540–550, May 1998.
- [40] J. Haitzma and T. Kalker, "A watermarking Scheme for Digital Cinema", Proc. of IEEE Int. Conf. on Image processing, vol.2, pp. 487–489, Oct. 2001.
- [41] Y. Zhao and R. L. Lagendijk, "Video Watermarking Scheme Resistant to Geometric Attacks", Proc. of IEEE Int. Conf. on Image Processing, vol. 2, pp. 145–148, Sept. 2002.
- [42] O. Benedens, "Geometry-based watermarking of 3d models", IEEE Computer Graphics and Applications, vol. 19, pp 46–55, Jan. 1999.
- [43] R. Ohbuchi, H. Masuda and M. Aono, "Watermarking 3D polygonal models", Proc. ACM Multimedia '97, Nov. 1997.
- [44] A. G. Bors, "Watermarking Mesh-Based Representations of 3D Objects Using Local Moments", IEEE Trans. on Image Processing, vol. 15, no.3, pp. 687–701, Mar. 2006.
- [45] E. Garcia and J. L. Dugelay, "Texture-based watermarking of 3D video objects", IEEE Trans. on Circuits and Systems for Video Technology, vol. 13, no. 8, pp. 853–866, Aug. 2003.

- [46] J. Bennour and J. L. Dugelay, "Protection of 3D Object Through Silhouette Watermarking", Proc. of 31st IEEE Int. Conf. on Acoustics, Speech, and Signal Processing, vol. 2, May 2006.
- [47] M. Tanimoto, "FTV (Free Viewpoint Television) Creating Ray-Based Image Engineering", Proc. of IEEE Int. Conf. on Image Processing, vol. 2, pp. 25–28, Sept. 2005.
- [48] L. Onural, H. M. Ozaktas, E. Stoykova, A. Gotchev, J. Watson, "An Overview of the Holographic Display Related Tasks within the European 3D TV project", Proc. of SPIE vol 6187 Photon Management II, pp. 61870T-1 - 61870T-10, Apr. 2006.
- [49] A. B. Watson, J. Hu and J. F. McGowan, "DVQ: A Digital Video Quality Metric based on Human Vision", Journal of Electronic Imaging, 10(1), pp. 20–29, 2001.
- [50] A. B. Watson. *Temporal sensitivity*. In Boff, K., Kaufmann, L. & Thomas, J. (eds.), Handbook of Human Perception and Performance, 1, Chapter 6. New York: John Wiley and Sons, 1986.
- [51] A. B. Watson, "DCT quantization matrices visually optimized for individual images", Proc. of SPIE on Human Vision, Visual Processing, and Digital Display IV, vol. 1913, pp. 202–216, 1993.
- [52] B. Zhu, A. H. Tewfik, Ö. N. Gerek, "Low Bit Rate Near-Transparent Image Coding", in Proc. of the SPIE Int. Conf. on Wavelet Apps. for Dual Use, vol. 2491, pp. 173–184, 1995.
- [53] G. E. Legge and J. M. Foley, "Contrast Masking in Human Vision", J. Opt. Soc. Am., 70(12), pp. 1458–1471, 1980.
- [54] A. B. Watson, M. Taylor and R. Borthwick, "Image quality and entropy masking", in Proc. of Human Vision, Visual Processing, and Digital Display VIII, SPIE, 3016, pp. 2–12, 1997.
- [55] S. Suthaharan, S. W. Kim, S. Sathanathan, H. K. Lee, and K. R. Rao, "Perceptually Tuned Robust Watermarking Scheme Digital Video using Motion Entropy Masking", Proc. of the Int. Conf. on Consumer Electronics, pp. 104–105, June 1999.
- [56] Z. Wang and A. C. Bovik, "Embedded Foveation Image Coding", IEEE Trans. on Image Processing, vol. 10, No. 10, pp. 1397–1410, Oct. 2001.

- [57] C. I. Podilchuk and W. Zeng, "Image-Adaptive Watermarking Using Visual Models", *IEEE Journal of Selected Areas in Communications*, vol. 16, no. 4, pp. 525–539, May 1998.
- [58] A. A. Alatan, A. N. Akansu, W. Wolf, "Multi-Modal Dialogue Scene Detection using Hidden Markov Models for Content-based Multimedia Indexing", *Int. Journal on Multimedia Tools and Applications*, vol. 14, no.2, pp. 137–151, June 2001.
- [59] J. G. Proakis, *Digital Communications*, Second Edition, Chapter 5, "Efficient signaling with coded waveforms", pp. 396–399, 1995.
- [60] P. Termont, L. De Stuycker, J. Vandewege, M. Op de Beeck, J. Haitsma, T. Kalker, M. Maes, G. Depovere, "How to achieve robustness against scaling in a real-time digital watermarking system for broadcast monitoring", *Proc. of IEEE Int. Conf. on Image Processing*, vol. 1, pp. 407–410, Sept. 2000.
- [61] S. Pereira and T. Pun, "Fast robust template matching for affine resistant image watermarking", in *International Workshop on Information Hiding*, ser. Lecture Notes in Computer Science. Berlin, Germany: Springer-Verlag, vol. LNCS 1768, pp. 200–210, 1999.
- [62] L. Boney, A. I. Tewfik, and K. I. Hamdy, "Digital Watermarks for Audio Signals", *Proc. of EUSIPCO 96, Vol. III, VIII European Signal Processing Conf.*, pp. 1697–1700, 1996.
- [63] M. Pollefeys et al., "3D Structure from Images - SMILE 2000", Springer Verlag, LNCS 2080, pp. 164, July 2000.
- [64] H. S. Song, N. I. Cho, J. Kim, "Robust Watermarking of 3D Mesh Models", *Proc. of IEEE Int. Conf. on Image Processing*, pp.332–335, Dec. 2002.
- [65] S.-H. Lee, T.-S. Kim, S.-J. Kim, Y. Huh, K.-R. Kwon, K.-I. Lee, "3D Mesh Watermarking Using Projection onto Convex Sets", *Proc. of IEEE Int. Conf. on Image Processing*, vol. 3, pp. 1577–1580, Oct. 2004.
- [66] J. F. Delaigle, "Protection of Intellectual Property of Images by perceptual Watermarking", PhD thesis, Universite Catholique de Louvain, Sept. 2000.
- [67] R. Ohbuchi, H. Masuda and M. Aono, "Watermarking three dimensional polygonal models through geometric and topological modifications", *IEEE Journal on Selected Areas in Communication*, vol. 16, no. 4, pp. 551–560, 1998.

- [68] Stanford Computer Graphics Laboratory, *The Stanford 3D Scanning Repository*: <http://graphics.stanford.edu/data/3Dscanrep>, Last Access Date: 27.08.2007.
- [69] S. Kanai, H. Date, and T. Kishinami, “Digital Watermarking for 3D Polygons using Multiresolution Wavelet Decomposition”, Proc. Sixth IFIP WG 5.2 GEO-6, pp. 296–307, Dec. 1998.
- [70] R. Ohbuchi, S. Takahashi, T. Miyazawa and A. Mukaiyama, “Watermarking 3D Polygonal Meshes in the Mesh Spectral Domain”, in Proc. Graphics Interface 2001, pp. 9–17, 2001.
- [71] M. Ashourian and R. Enteshary, “A new masking method for spatial domain watermarking of three-dimensional triangle meshes”, TENCON 2003. Conference on Convergent Technologies for Asia-Pacific Region, vol. 1, pp:428–431, Oct. 2003.
- [72] Z.-Q. Yu, H. H. S. Ip and L. F. Kowk, “Robust watermarking of 3D polygonal models based on vertex scrambling”, in Proc. of Computer Graphics International, pp. 254–257, July 2003.
- [73] S. Zafeiriou, A. Tefas and I. Pitas, “A Blind Robust Watermarking Scheme for Copyright Protection of 3D Mesh Models”, Proc. of IEEE Int. Conf. on Image Processing, vol. 3, pp. 1569–1572, Oct. 2004.
- [74] A. Kalivas, A. Tefas and I. Pitas, “Watermarking of 3D models using principal component analysis”, Proc. of IEEE Int. Conf. on Acoustics, Speech, and Signal Processing, vol. 5, pp. 676–679, 2003.
- [75] T. Harte and A. G. Bors, “Watermarking 3D Models”, Proc. of IEEE Int. Conf. on Image Processing, pp. 661–664, vol. 3, June 2002.
- [76] A. G. Bors, “Watermarking 3D Shapes Using Local Moments”, Proc. of IEEE Int. Conf. on Image Processing, vol. 2, pp. 729–732, Oct. 2004.
- [77] P. Daras, D. Zarpalas, D. Tzovaras and M. G. Strintzis, “Watermarking of 3D models for Data Hiding”, Proc. of IEEE Int. Conf. on Image Processing, vol. 1, pp. 47–50, Oct. 2004.
- [78] P. Daras, D. Zarpalas, D. Tzovaras, D. Simitopoulos and M.G. Strintzis: “Combined Indexing and Watermarking of 3D Models using the Generalized 3D Radon Transform”, Multimedia Security Handbook, B. Fuhr and D. Kirovski Eds, CRC Press, pp 733–758, ISBN 0-8493-2773-3, Dec. 2004.

- [79] M. M. Yeung and B.-L. Yeo, "Fragile watermarking of 3D objects", Proc. of Int. Conf. on Image Processing, vol. 2, pp. 442–446, Oct. 1998.
- [80] H.-Y. S. Lin, H.-Y. M. Liao, C.-S. Lu and J.-C. Lin, "Fragile Watermarking for Authenticating 3D Polygonal Meshes", Proc. 16th IPPR Conf. on CVGIP, pp. 298–304, 2003.
- [81] K.-R. Kwon, S.-G. Kwon, S.-H. Lee, T.-S. Kim, K.-I. Lee, "Watermarking for 3D polygonal meshes using normal vector distributions of each patch", Proc. of IEEE Int. Conf. on Image Processing, vol. 2, pp. 499–502, Sept. 2003
- [82] E. Praun, H. Hoppe and A. Finkelstein "Robust mesh watermarking," In Proc. Of SIGGRAPH 99, pp. 69–76, 1999.
- [83] R. Ohbuchi, A. Mukaiyama and S. Takahashi, "A Frequency Domain Approach to Watermarking 3D Shapes", in Proc. EuroGraphics, vol. 21, no 3, 2001.
- [84] G. Louizis, A. Tefas and I. Pitas, "Copyright protection of 3D images using watermarks of specific spatial structure", Proc. of IEEE Int. Conf. on Multimedia and Expo, vol. 2, pp. 557–560, Aug. 2002.
- [85] R. A. Patrice and M. Benoit, "Blind Watermarking of 3D Meshes Using Robust Feature Points Detection", Proc. of Int. Conf. on Image Processing, vol. 1, pp. 693–696, Sept. 2005.
- [86] K. Yin, Z. Pan, J. Shi, D. Zhang, "Robust mesh watermarking based on multiresolution processing", Computers and Graphics, vol. 25, pp. 409–420, 2001.
- [87] F. Uccheddu, M. Corsini and M. Barni, "Wavelet-Based Blind Watermarking of 3D Models", Int. Multimedia Conference, Proc. of the 2004 Workshop on Multimedia and Security, pp 143–154, 2004.
- [88] E. Corsini, E. D. Gelasca, T. Ebrahimi, M. Barni, "Watermarked 3D Mesh Quality Assesment", IEEE Trans. on Multimedia, vol. 9, no. 2, pp. 247–256, Feb. 2007.
- [89] H. Hoppe, "Progressive Meshes", Proc. of ACM SIGGRAPH 96, pp. 99–108, Aug. 1996.
- [90] F. Cayre, P. Rondao-Alface, F. Schmitt, B. Macq, H. Maitre, "Application of spectral decomposition to compression and watermarking of 3D triangle mesh geometry", Signal Processing-Image communications, vol. 18, no. 4, pp. 309–319, Apr. 2003.

- [91] L. Li, D. Zhang, Z. Pan, J. Shi, K. Zhou, K. Ye, “Watermaking 3D mesh by spherical parameterization”, *Computer and Graphics* 28, pp. 981–989, 2004.
- [92] D. Cotting, T. Weyrich, M. Pauly, M. Gross, “Robust Watermarking of Point Sampled Geometry”, *Proc. of Int. Conf. on Shape Modeling and Applications*, pp. 233–242, June 2004.
- [93] Y. Wu, X. Guan, M. S. Kankanhalli, Z. Huang, “Robust invisible watermarking of volume data using the 3D DCT ”, *Proc. of Computer Graphics International*, pp. 359–362, July 2001.
- [94] E. J. Stollnitz, A. D. DeRose and D. H. Salesin, “Wavelets for Computer Graphics: Theory and Applications”, Morgan Kaufman, 1996.
- [95] I. Guskov, W. Sweldens and P. Schröder, “Multiresolution signal processing for meshes”, *Proc. SIGGRAPH ’99*, pp. 49–56, 1999.
- [96] C. Zhang and T. Chen “A survey on image based rendering-representation, sampling and compression”, *Signal Processing: Image Communications*, vol. 19 pp. 1–28, Jan. 2004.
- [97] C. Fehn, P. Kauff, O. Schreer, R. Schafer, “Interactive virtual view video for immersive TV applications”, *Proc. of IBC’01*, vol. 2, pp. 53–62, Sept. 2001.
- [98] M. Levoy, P. Hanrahan, “Light field rendering”, *Computer Graphics (SIGGRAPH’96)*, pp. 31–42, Aug. 1996.
- [99] Stanford Computer Graphics Laboratory, *The Stanfords Light Field Archive*: <http://graphics.stanford.edu/software/lightpack/lifs.html>, Last Access Date: 27.08.2007.
- [100] Carnegie Mellon University, Electrical and Computer Engineering, Light Field Archive of Advanced Multimedia Processing Laboratory: <http://amp.ece.cmu.edu/projects/MobileCamArray/>, Last Access Date: 27.08.2007.
- [101] R. Hartley and A. Zisserman, “Multiple View Geometry in Computer Vision”, Second Edition, Cambridge University Press, pp.32–37, 2003.
- [102] Z. Zhang, R. Deriche, O. Faugeras and Q.-T. Leang, “A Robust Technique for Matching Two Uncalibrated Images through the Recovery of the Unknown Epipolar Geometry”, INRIA, Report No. 2273, 1994.
- [103] C. Harris and M. Stephens, “A Combined Corner and Edge Detector”, *Proc. of the 4th Alvey Vision Conf.*, pp. 147–151, 1988.

- [104] M. A. Fischler and R. C. Bolles, “Random Sample Consensus: A Paradigm for Model Fitting with Applications to Image Analysis and Automated Cartography”, *Communications of the ACM*, vol. 24, pp. 381–395, June 1981.
- [105] D. G. Lowe, “Distinctive image features from scale-invariant keypoints”, *International Journal of Computer Vision*, 60, 2, pp. 91–110, 2004.
- [106] R. Hartley, “Projective Reconstruction and Invariants from Multiple Images”, *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 16, no. 10, pp:1036–1041, Oct. 1994.
- [107] I. Weiss, “Projective Invariants of Shape”, *Proc. of IEEE Computer Society Conf. on Computer Vision and Pattern Recognition*, pp. 291–297, June 1988.
- [108] N. Chotikakamthorn, N. Pantuwong and W. Yawai, “Projective Invariant Digital Image Watermarking Technique Using Four Co-Planar Feature Points”, *Proc. of IEEE Int. Conf. on Image Processing*, vol. 1, pp. 1005–1008, Sept. 2005.
- [109] J. L. Mundy and A. Zisserman (Editors), *Geometric Invariance in Computer Vision*, MIT Press Cambridge, MA, USA, 1992.
- [110] R. Hartley and A. Zisserman, *Multiple View Geometry in Computer Vision*, 2nd Edition, Cambridge University Press, Chapter 2, pp. 44–45, 2003
- [111] J. B. Burns, R. S. Weiss and E. M. Riseman, “The non existence of general-case view invariants”, in *Geometric Invariance for Computer Vision*, J. L. Mundy and A. Zisserman, Eds. Boston, MA: MIT Press, pp 120–134, 1992.
- [112] Mesh Viewer, <http://mview.sourceforge.net/>, Last Access Date: 27.08.2007.
- [113] C. Fehn, “A 3D-TV approach Using Depth Image Based Rendering”, *Proc. of 3rd IASTED Conf. on Visualization, Imaging and Image Processing*, pp. 482–487, Sept. 2003.
- [114] Recommendation ITU-R BT.500-11, “Methodology for the Subjective Assessment of the quality of television pictures”, Geneva, June 2002.
- [115] G. B. Wetherill and H. Levitt, “Sequential estimation of points on a psychometric function”, *Br. J. Math. Stat. Psychol.*, vol. 18, pp. 1–10, 1965.

VITA

Alper Koz was born in Kahramanmaraş, Turkey, in 1978. He received the B.S. and M.S degree in Electrical and Electronics Engineering, from Middle East Technical University, Ankara, Turkey in 2000 and 2002, respectively. Since August 2000, he is a research and teaching assistant in the same department. He participated in the projects in the area of image and video watermarking supported by The Scientific and Technological Research Council of Turkey from 2000 to 2004. Since September 2004, he is also a researcher in 3DTV project, Network of Excellence funded by the European Commission 6th Framework Programme. His research interests include image, video and 3D watermarking, image based representation and rendering.

His publications are as follows:

Book Chapter

- Alper Koz, George Triantafyllidis and A. Aydın Alatan, *3D Watermarking: Techniques and Directions*, in *Three-Dimensional Television: Capture, Transmission, and Display*, Editors: Haldun M. Ozaktas and Levent Onural, Springer, Heidelberg, 2007.

Journal Papers

- Alper Koz and A. Aydın Alatan, “Oblivious Spatio-Temporal Watermarking of Digital Video by exploiting Human Visual System”, accepted for publication in IEEE Transactions on Circuits, Systems and Video Technology, (July 2007).
- Aljoscha Smolic, Karsten Mueller, Nikolce Stefanoski, Joern Ostermann, Atanas Gotchev, Gozde B. Akar, Georgios Triantafyllidis, Alper Koz, “3D TV: Coding Algorithms for 3D TV-A Survey”, accepted for publication in special issue on multi-view video and 3D TV, IEEE Transactions on Circuits, Systems and Video Technology. Tentative publication date: October 2007.
- Alper Koz, Cevahir Çığla, A. Aydın Alatan, “Free View Watermarking for Free-View Television”, submitted to IEEE Transactions on Image Processing, August 2007.

Pending Patent

- “Watermark Detection Method for Free View Televisions”; Alper Koz, Cevahir Çıgla, A. Aydın Alatan; Filed October 3, 2006.

Conference Papers

- Alper Koz, Cevahir Çıgla and A. Aydın Alatan, “Watermarking for Light Field Rendering”, European Signal Processing Conference (EUSIPCO), September 3-7 2007, Poznan, Poland.
- Evlambios E. Apostolidis, Alper Koz and George Triantafyllidis, “Watermarking Tests for Free-View Point Television”, 3D TV Conference, 2007, Kos Island, Greece.
- Alper Koz, Cevahir Çıgla and A. Aydın Alatan, “Free-View Watermarking for Free-View Television”, IEEE International Conference on Image Processing, October 8-11 2006, Atlanta, USA.
- Alper Koz and A. Aydın Alatan, “Oblivious Video Watermarking Using Temporal Sensitivity of HVS”, IEEE International Conference on Image Processing, September 11-14 2005, Genova, Italy.
- Alper Koz and A. Aydın Alatan, “Temporal Watermarking of Digital Video”, 4th European Workshop on Image Analysis for Multimedia Interactive Services (WIAMIS), April 2003, London, UK.
- Alper Koz and A. Aydın Alatan, “Video Watermarking Using Temporal Sensitivities of Human Visual System”, *Workshop on Transmitting, Processing and watermarking Multimedia Contents '2003*, Bordeaux, FRANCE.
- Alper Koz and A. Aydın Alatan, “Foveated Image Watermarking”, IEEE International Conference on Image Processing, September 2002, Rochester, NY.
- Alper Koz, Cevahir Çıgla ve A. Aydın Alatan, “Eşyazım Tabanlı Sanal Kamera Konumu Tahmini ile Görüntü Tabanlı Gerçeklemelerde Gizli Damgalama”, IEEE 15. Sinyal İşleme ve İletişim Uygulamaları Kurultayı, 11-13 Haziran 2007, Eskişehir. (Turkish)
- Alper Koz ve A. Aydın Alatan, “İnsan Görme Sisteminin Zamansal Duyarlılığına Dayalı Alındısız Video Damgalama”, IEEE 12. Sinyal İşleme ve İletişim Uygulamaları Kurultayı, 28-30 Nisan 2004, Kuşadası. (Turkish)

- Alper Koz ve A. Aydın Alatan, “Zamansal Kontrast Duyarlılığına Dayalı Video Damgalama”, 11. Sinyal İşleme ve İletişim Uygulamaları Kurultayı, 18-20 Haziran 2003, İstanbul. (Turkish)
- Alper Koz ve A. Aydın Alatan, “Odaklanmaya Dayalı Görünmez Damgalama Metodu,” 10. Sinyal İşleme ve İletişim Uygulamaları Kurultayı,” c. 2, s. 703-708, 12-14 Haziran 2002, Pamukkale, Denizli.(Turkish)