Steganalytic Methods for 3D Objects

Zhenyu Li

Doctor of Philosophy

University of York

Computer Science

January 2018

Abstract

This PhD thesis provides new research results in the area of using 3D features for steganalysis. The research study presented in the thesis proposes new sets of 3D features, greatly extending the previously proposed features. The proposed steganlytic feature set includes features representing the vertex normal, curvature ratio, Gaussian curvature, the edge and vertex position of the 3D objects in the spherical coordinate system. Through a second contribution, this thesis presents a 3D wavelet multiresolution analysis-based steganalytic method. The proposed method extracts the 3D steganalytic features from meshes of different resolutions. The third contribution proposes a robustness and relevance-based feature selection method for solving the cover-source mismatch problem in 3D steganalysis. This method selects those 3D features that are robust to the variation of the cover source, while preserving the relevance of such features to the class label. All the proposed methods are applied for identifying stego-meshes produced by several steganographic algorithms.

Contents

List of Tables

List of Figures

Acknowledgements

I would like to express my sincere gratitude to my supervisor Dr. Adrian G. Bors for his support and encouragement for my study, research and academic writing during the past three years. I have learnt a lot of research and writing skills from him. I am grateful to him for arranging my research visit to LIRMM, France and funding me to some flagship international conferences. I am also indebted to his regular meetings, constructive suggestions and helpful corrections on the writing that helped me finish this thesis in a timely manner. It is a very pleasant experience to work with him.

My gratitude also goes to my assessor Prof. Richard Wilson for the helpful discussions in the TAP meetings and his precise assessment of my work. Thanks are also due to all my colleagues in the CVPR group who shared a lot of happy time with me and provided me kind help. I am very grateful to Prof. William Puech for providing me the opportunity to visit LIRMM for one month. The collaboration with the researchers in LIRMM is a treasurable experience to me. I would like to acknowledge the scholarship received from Zhengzhou Institute of Information Science and Technology.

Many thanks are due to my friends in UK for their help and support during my PhD study. I would also like to thank the friends I met in Holgate Hall and International Cafe who gave me a lot of encouragement and made my life more enjoyable.

Lastly and most importantly, I wish to thank all my family members for their support and love. My very special gratitude goes to my mother Xinxia Lyu and my father Can Li to whom I owe everything I am today. I am so grateful to them for always encouraging me to pursue higher education and helping me with my emotional intelligence throughout my entire life.

Declaration

I declare that this thesis is a presentation of original work, which I undertook at the University of York during 2015 - 2018, and I am the sole author. This work has not previously been presented for an award at this, or any other, University. All sources are acknowledged as References.

Some parts of this thesis have been published in conference proceedings and journals; where items were published jointly with collaborators, the author of this thesis is responsible for the material presented here.

Conference Papers

- **Zhenyu Li** and Adrian G. Bors. **3D mesh steganalysis using local shape features**. Proceedings of 41th International Conference on Acoustics, Speech and Signal Processing (ICASSP), pages 2144-2148. IEEE, 2016.

- **Zhenyu Li** and Adrian G. Bors. **Selection of robust features for the cover source mismatch problem in 3D steganalysis**. Proceedings of 23rd International Conference on Pattern Recognition (ICPR), pages 4251-4256. IEEE, 2016.

- **Zhenyu Li**, Sébastien Beugnon, William Puech, and Adrian G. Bors. **Rethinking the high capacity 3D steganography: Increasing its resistance to steganalysis**. Proceedings of 19th International Conference on Image Processing (ICIP), pages 510-514. IEEE, 2017.

Journal Papers

- **Zhenyu Li** and Adrian G. Bors. **Steganalysis of 3D objects using statistics of local feature sets**. Information Sciences, volume 415-416, pages 85-99, 2017.

Chapter 1

Introduction

Each day, billions of digital images, video, audio or 3D objects are shared through the Internet. Based on these digital files, some secret messages can be hidden in plain sight using the technique known as "information hiding". Information hiding is the practice of concealing information within media data, whilst causing no perceptible effect on the given object, so that no one would suspect there is a hidden message in the object. An object which has been embedded with hidden information is called stego-object, while the original object, *i.e.* without any information embedded in, is called cover-object. Research in 3D steganography started more than 17 years ago, shortly after research in image watermarking developed. Nowadays, there are several 3D steganographic approaches.

The word steganography originates from two Greek words, steganos ($\sigma\tau\epsilon\gamma\alpha\upsilon\acute{o}\varsigma$) and graphein ($\gamma\rho\acute{\alpha}\varphi\epsilon\iota\upsilon$), meaning "covered or concealed" and "writing", respectively. Steganography hides the information into the innocuous cover-objects, obtaining the stego-objects with hidden messages. One of the famous example of the steganography in the real life is using an invisible ink to write on the paper. The message will disappear after the paper is dry and appear once the paper is close to the heat. Whilst the modern steganography usually uses the digital files to hide the secret messages and send through the Internet. Compared to cryptography, steganography is concerned with concealing the fact that a secret message is being sent, whereas cryptography is about protecting the contents of a message. While cryptography changes completely the data, by encoding it according to a code, steganography does not apparently alter the cover-data, but aims to hide the information unnoticed.

Digital watermarking technique is similar to the steganography, which can hide a certain amount of information through a code, called watermark, in the digital files. There are two kinds of digital watermarking algorithms, robust watermarking and fragile watermarking. The goal of robust watermarking is to make the watermark robust against attacks, such as denoising, compression and remodulation, so the watermark can then be used as evidence for copyright protection. Since the focus of robust watermarking is on the robustness, the embedding capacity would be rather low, because we are restricted when embedding additional bits of information by the requirement for robustness. Meanwhile, the fragile watermarking is considered for the authentication applications, in another word, it is used to identify whether the object has been tampered with or not. So the fragile watermark has to be sensitive to any modifications of the stego-objects. In term of the capacity, some fragile watermarking algorithms have an embedding capacity as high as the steganography.

Since steganography can provide a covert communication channel for two parties, it can be used maliciously by unlawful organisations, becoming a threat to the modern society. For example, steganography was utilized by the terrorists to communicate, according to a report published by BBC in 2013 [Gardner, 2013]. Only recently, steganography is started being used by malware operators to develop covert communication channels between infected computers and their command and control servers, as reported in the Black Hat Europe 2015 [Bureau and Dietrich, 2015]. The modifications on the objects produced by the information hiding algorithms are too small to be noticed with the naked eye. Nevertheless, through a technique, called steganalysis we can identify whether a message was embedded or not in a given media object.

Steganalysis represents the method for identifying whether a message was hidden in a certain media, using steganography or not. Steganalysis becomes a more and more important research topic, because the potential threat of steganography being used for malicious purposes is on the increase under the current security threats. Steganalysis employs feature extraction algorithms from a specific media, which are then used for training a classifier in order to distinguish cover-objects from stego-objects.

Following the development of steganography, in diverse media such as audio signals, images and video, steganalytic algorithms have been developed for these media as well.

Steganalysis for detecting the hidden information in images was studied in [Ker, 2005, Chen and Shi, 2008, Fridrich and Kodovský, 2012], in audio [Liu et al., 2009, Ren et al., 2017] and in video [Cao et al., 2012, Wang et al., 2014]. Unlike in the case of images or video, which contain information structured on regular lattices, 3D objects are defined by their geometry. In this thesis we consider mesh based representations or 3-D objects, encoding the geometry of their surfaces. Such representations correspond to meshes, where vertices are connected through edges and polygons.

Despite the development of new 3D information hiding algorithms during the past two decades, 3D steganalysis received little interest from the research community. The first paper about the 3D steganalysis was published in 2014 [Yang and Ivrissimtzis, 2014]. The 3D steganalytic approach proposed in [Yang and Ivrissimtzis, 2014] uses a 208-dimensional feature set, YANG208, and quadratic classifier to detect the stego-objects embedded by six information hiding algorithms. However, the performance of the steganalyzers trained in [Yang and Ivrissimtzis, 2014] still needs further improvement.

The Cover Source Mismatch (CSM) problem is a barrier when attempting to apply the steganalytic approaches in the real world, and this is represented by the scenario that the objects used for training the steganalyzer may be originated from a different cover source than the one used for hiding the information [Ker et al., 2013] by the steganographier.

In our study, we would like to concentrate on improving the performance of 3D steganalysis in order to be used in a practical situation.

In 3D steganalysis, specific features are extracted from the surface of the 3D object, and their characteristic statistical parameters are then fed into a classifier in order to distinguish cover-objects from stego-objects. In 3D steganalysis, geometric features of stego-objects and cover-objects are modelled statistically. The geometric features locally capture the shape of the object, such as the vertex position, the face normal, the dihedral angle between two adjacent faces, and so on. Based on this idea, we propose a more comprehensive local feature set, considering some new geometric features, for instance, the vertex normal, the Gaussian curvature, the curvature ratio, and features from the spherical coordinates of the vertex, for steganalysis. The new local features would more effectively capture the slight distortions caused by the information embedding, thus increasing the detection accuracy of

the steganalyzer.

The 3D wavelet transforms provide a set of tools for the multi-scale analysis of mesh-based representations of surfaces. Some steganographic methods have been proposed for hiding information in the 3D wavelet domain and such information is hardly detectable by the existing 3D steganalyzers.

In this study we propose to extract a set of 3D wavelet features from the initial resolution triangle mesh, the lower resolution one and the higher resolution one. During the experiments, we compare the proposed wavelet feature set with other steganalytic features when detecting the stego-objects embedded by various information hiding algorithms.

When the steganalyzer is applied in the real world, it has to address the CSM problem, which is due to the fact that the cover source used for the training of the steganalyzer is different from the data used in the testing. This is a challenging problem, which greatly restricts the applications on steganalyzers in the real situations.

After analyzing the techniques used to mitigate the CSM problem in the image steganalysis, we propose to select those features which are robust to the variation of the cover source but sensitive to the embedding changes. So we propose a feature selection algorithm considering both the feature's robustness to the variation of the cover source and their relevance to the class label.

The remainder of the thesis is organized as follows: Chapter 2 gives a detailed review of the research literature related to the work presented in this thesis. Chapter 3 presents the proposed the local feature set used for 3D steganalysis and provides experimental results when detecting information hidden by six different steganographic algorithms. Then, Chapter 4 presents the 3D wavelet analysis-based feature set for the steganalysis of the 3D triangle meshes and evaluates the results of the proposed methodology in the context of information hidden by several steganographic algorithms. Furthermore, Chapter 5 describes the robustness and relevance-based feature selection algorithm for solving the cover source mismatch problem in 3D steganalysis. Chapter 6 concludes this thesis and points to several problems that deserve further research.

Chapter 2

Literature Review

Steganography and steganalysis have been considered for many types of data, including audio, images and video. In this thesis we consider the steganalysis of 3D graphical objects and models. Firstly, we outline the main approaches in 3D steganography and information hiding. Then existing steganalytic algorithms for digital images, video and 3D meshes are introduced. Finally, the research studies about the cover source mismatch problem in image steganalysis are discussed as well.

2.1 3D information hiding methods

The 3D information hiding methods are classified into three categories: embedding in the spatial domain, those embedding in a transform domain, and those algorithms embedding in the vertex ordering in the mesh representation. In the following, we first discuss the requirements for information hiding and then introduce the 3D embedding methods for each category.

2.1.1 Requirements for information hiding

Embedding information in media has several requirements, including the imperceptibility of changes produced, security, high bit capacity and robustness. The most important is that the embedded information should not produce visible changes. The security of the embedded information is also a very important issue. The use of information hiding should not be

easily detected by steganalysis and the embedded messages should be extracted only by the owners of a secret key, in the case of private communication channels, or by using a public key otherwise. The bit capacity is also an important feature, representing the amount of bits embedded in a certain media. Another requirement is that of robustness, which measures how well the embedded messages can survive attacks, such as media compression, cropping, smoothing, noise addition, vertex reordering and so on.

In fact, the steganographiers have to find a balance between these factors. Higher capacity usually means more modifications, which will affect the invisibility and security negatively. Meanwhile, while aiming to increase the robustness to various attacks, the information hiding may produce visible changes in the media, while the data capacity may suffer as well. Additionally, seeking to increase the security will influence all the other factors, because security changes require additional data modifications.

2.1.2 Embedding information into spatial domain

Ohbuchi *et al.* [Ohbuchi et al., 1997] published the first paper on 3D information hiding describing two methods. The two methods use different geometrical primitives for embedding: Triangle Similarity Quadruple (TSQ) and Tetrahedral Volume Ratio (TVR). The TSQ algorithm first finds a macro embedding unit in a triangle mesh which consists of four neighboring triangles with one triangle surrounded by the other three. During the procedure, it avoids the triangles already used for embedding. Then, for each embedding unit, a quadruple $\{subscript, mark, data1, data2\}$ is embedded into the four triangles by displacing vertices by small amount. It repeats these steps until the entire message is embedded. Meanwhile, the TVR algorithm finds a sequence of triangles based on the spanning tree of vertices on the triangular mesh. Then, a sequence of tetrahedrons which are subtended by two triangles adjacent to the same edge is generated. The tetrahedron with the largest volume is selected as the first one in the sequence while its volume is considered as the common denominator. While the volumes of other tetrahedrons are used for numerators. The message is embedded by changing the ratio of the volumes for the tetrahedrons, after displacing the vertices which are contained in those selected for the numerator.

Cayre and Macq [Cayre and Macq, 2003] proposed an information hiding algorithm which

includes two steps: firstly it selectes a list of triangles from the mesh that contains the payload driven by the secret key; then, each so-called admissible triangle in the list is modified or or is left unchanged according to the binary symbol it conveys, which is called a Macro Embedding Procedure (MEP). In the first step, the list of triangles starts with the triangle determined on the basis of a specific geometric characteristic, such as the triangle area or the intersections of the principal axes by considering the Principal Component Analysis (PCA) of the mesh. The next triangle in the list is either the first or the second neighbored triangle in clockwise order, depending on the key. The edge of the triangle which is shared with its predecessor is named the entry edge. The entry edge is divided into two subsets of intervals, for embedding either the symbol 0 or 1, and this decomposition is an application of Quantization Index Modulation (QIM) [Chen and Wornell, 2001] to 3D meshes. Then, the vertex opposite to the entry edge is used to embed information by displacing it such that the vertex's projection on the entry edge is located within a certain interval in order to embed the hidden bit. Wang and Cheng [Wang and Cheng, 2005] modified the method of Cayre and Macq and proposed a Multi-Level Embedding Procedure (MLEP) to embed information by sliding, extending or rotating the triangle. The sliding method is very similar to the method proposed in [Cayre and Macq, 2003]. In the extending approach, QIM is applied by displacing a vertex of the triangle so that the height of the triangle is located within a certain interval according to the message's bit. Finally, they apply the same concept to the rotating level by embeding messages in the degree of the dihedral angle.

Chao *et al.* [Chao et al., 2009] proposed a Multi-Layer Steganography (MLS) for 3D objects. Firstly, this method defines a vertex embedding order by traversing the triangles in the mesh using the same method as in [Cayre and Macq, 2003]. When considering a single layer, this method applies PCA on the 3D object and finds the two extreme vertices according to their locations with respect to the first principal axis. Then they cut this axis into two-state region sets, used to embed the message bit of 0 or 1. Next, it projects the x-coordinates of all the vertices onto this axis. The location of a vertex is changed or not, defining a certain projection on the principal axis of the object, according to the bit to be embedded. This method can be easily extended to multiple layers. When implementing this method, we found that not all the bits embedded by this method are retrievable and some

are lost.

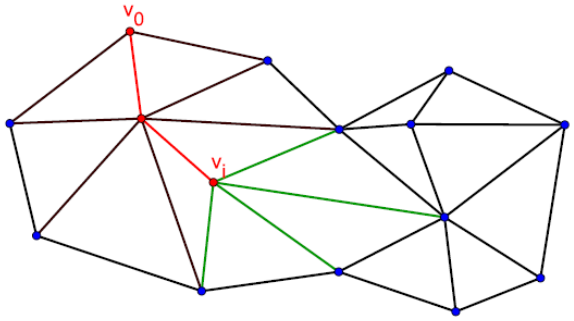


Figure 2.1: Using the Hamiltonian path for ordering the vertices from the mesh surface. The vertices and edges in red are covered in the constructed path, the green edges are compared in order to find the next vertex in the path. This figure was reproduced after Figure 2 (b) from [Itier and Puech, 2017]

The 3D steganographic algorithm, Hamiltonian Path Quantization (HPQ), proposed in [Itier and Puech, 2017] utilizes a synchronization technique to guarantee that the order of the embedded data is the same during the embedding and extraction stages. It builds a Hamiltonian path over the complete graph of the vertices in the 3D object without using the connectivity information. The Hamiltonian path is an approach for ordering all the vertices from the 3D object, starting from a specific vertex, chosen by a secret key. For each step, as illustrated in Figure 2.1, the algorithm chooses the nearest neighbor v_{i+1} of the current vertex v_i. The message is embedded by changing the relative position of a vertex v_{i+1} with respect to its predecessor v_i once the vertex v_{i+1} is added to the Hamiltonian path $\mathbf{P}_n$. In order to embed a large bit capacity, the vertex v_{i+1} is displaced along three coordinates in the Spherical Coordinate System (SCS) which originates in the location of the previously chosen vertex. The algorithm applies QIM by splitting each section of the path into intervals, controlled by the quantization parameter Δ. A given interval is subdivided into s sub-intervals which correspond to different embedded bits. The vertex is then moved to a specific location within the interval, according to the embedded information. The new position of the vertex is converted back to the Cartesian coordinate system after the embedding.

A recent study [Li et al., 2017] shows that a variation of the HPQ algorithm, named HPQ-R, which changes the vertex only in the radial distance coordinate of the SCS can increase its resistance to steganalysis to a large extent. The study from [Li et al., 2017] analyses the

influence of the parameter Δ and shows that a smaller value for this parameter leads to a smaller distortion in the 3D stego-shapes. By increasing the resistance to steganalysis we increase the protection of the information stored into the 3D objects.

Watermarking algorithms have better robustness but lower embedding capacity than the steganographic algorithms mentioned above. For example, two watermarking algorithms proposed in [Cho et al., 2007] are based on modifying the Mean or the Variance of the distribution of the vertices' Radial distance coordinates in the Spherical coordinate system, named MRS and VRS. Both algorithms transform the meshes from the Cartesian coordinate system to spherical coordinate system. Then they divide the radial distances coordinates of the vertices into a set of histogram bins according to their values. The mean or variance of the elements in each bin is adjusted to a certain range, which indicates the embedded watermark bit, by modifying the radial distance coordinates of the vertices. During the extraction stage, for each bin, extracted following the same procedure as for the embedding, if the mean or variance of the bin's elements is larger than the preset threshold, then the embedded bit is '1', but otherwise it is '0'.

Based on a similar idea used in [Cho et al., 2007], Yang *et al.* [Yang et al., 2017b] proposed a Steganalysis-Resistant Watermarking (SRW) algorithm. It also uses the spherical coordinate system to represent the positions of the vertices. In the SRW algorithm, the heights of the histogram bins containing the vertices' radial distance coordinates are changed so that certain information is embedded. For instance, after the displacement of the vertices belonging to the related bins, the height of the $(k+1)$th bin should be higher than that of the kth bin, so that bit '0' is embedded. The stego-meshes watermarked by the SRW algorithm are harder to be identified by the steganalyzer, when compared to those watermarked by the MRS or VRS algorithms.

2.1.3 Embedding information into transform domain

There are a number of information hiding methods in one of the transform domains of 3D meshes. Ohbuchi *et al.* [Ohbuchi et al., 2001] proposed the first 3D watermarking method in the spectral domain of 3D meshes in 2001. Firstly, spectral analysis is applied to the mesh in order to obtain the spectral coefficients of the mesh. Then, it repeatedly embeds

the same information a certain number of times in order to increase the resistance of the watermark against additive random noise. The information is embedded by altering the spectral coefficients based on a known stego-key and the modulation amplitude parameter. After embedding the messages, an inverse transformation converts the watermarked spectral coefficients back into the original mesh. Nevertheless, this is a non-blind watermarking algorithm, which means it needs both the cover mesh and the watermarked one during the watermark extraction. Moreover, some of the bits embedded by this method are lost during the extraction.

Luo and Bors [Luo and Bors, 2008] proposed a blind watermarking method using spectral coefficients. It is known that the low frequency spectral coefficients correspond to large scale features and the high frequency ones correspond to the detailed information [Karni and Gotsman, 2000]. To avoid the visible changes, their approach embeds the watermark into the middle and high frequency range of the spectral coefficients. This method firstly splits the coefficients of the upper 85% of the coefficients into a number of bins. Each set of coefficients forms a point cloud whose shape is analyzed by using PCA. Then the cloud of 3D points is "squashed" when embedding a bit of 1 and "inflated" to a well defined sphere for a bit of 0. This watermarking approach is blind so that it is able to retrieve the watermark without the cover mesh. During the extraction, the spectral coefficients are extracted and grouped into sets in the same way used during the embedding. Finally, the information is retrieved by calculating the ratio between the largest variance and the smallest variance of the point cloud formed by the watermarked coefficients.

3D wavelet multiresolution analysis, introduced in [Lounsbery et al., 1997], allows the mesh to be represented at various resolutions. 3D wavelet analysis leads to various applications, including filtering [Abdul-Rahman et al., 2013], mesh compression [Payan and Antonini, 2006, Kammoun et al., 2012], subdivision [Shao et al., 2014], as well as information hiding [Date et al., 1999, Uccheddu et al., 2004, Kim et al., 2005, Kim et al., 2006, Wang et al., 2008, Zaid et al., 2015].

In the following we outline the 3D wavelet decomposition. Figure 2.2 illustrates one iteration of the lazy wavelet decomposition, proposed in [Lounsbery et al., 1997], as a simple implementation of the 3D wavelet decomposition. The decomposition is applied on meshes

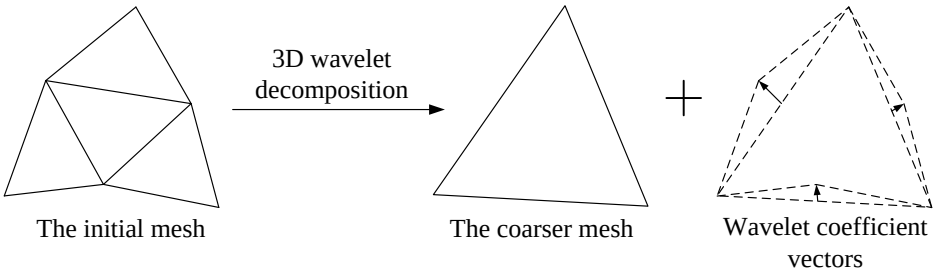


Figure 2.2: The illustration of one iteration of the lazy wavelet decomposition on the triangle meshes.

consisting of triangles. In the following for illustrative purposes we consider four triangles, including one triangle in the centre surrounded by the other three triangles, as shown in Figure 2.2. During the 3D wavelet decomposition, the three vertices, defining the central triangle, are removed, while the remaining three vertices would form a new triangle. Consequently, following the 3D wavelet decomposition, the four original triangles are transformed into a single triangle, part of a coarser mesh. At the same time, three Wavelet Coefficient Vectors (WCVs) are obtained as the vectors from the midpoints of the new edges to the vanished vertices. The new edges in the larger triangle are called the support edges for the WCVs. The new larger triangle preserves the basic shape formed by the former four triangles and the WCVs encode the local details of the 3D shape. This decomposition propagates on the whole mesh generating a coarser mesh together with a number of WCVs. This decomposition can then continue recursively. The limitation of this wavelet decomposition approach is that it requires the high resolution mesh to be a semi-regular mesh. The mathematical formulation defining the 3D wavelet decomposition can be found in [Lounsbery et al., 1997].

A 3D wavelet-based watermarking algorithm was proposed by Kanai *et al.* [Date et al., 1999], which modifies the ratio between the norm of a WCV and the length of its support edge. However, this 3D watermarking method is non-blind, which significantly reduces its potential for application. Uccheddu *et al.* [Uccheddu et al., 2004] proposed a blind 3D wavelet-based watermarking algorithm that embeds information by changing the position of the WCV's terminal point, according to a watermarking map generated by a secret key. Kim *et al.* [Kim et al., 2005] proposed a robust watermarking algorithm which hides information by changing the WCVs' norms. Another 3D wavelet-based robust watermarking algorithm

was also proposed by Kim *et al.* [Kim et al., 2006], which applies a technique similar to the one used in [Cho et al., 2007] in order to embed information into the histogram of WCVs' norms.

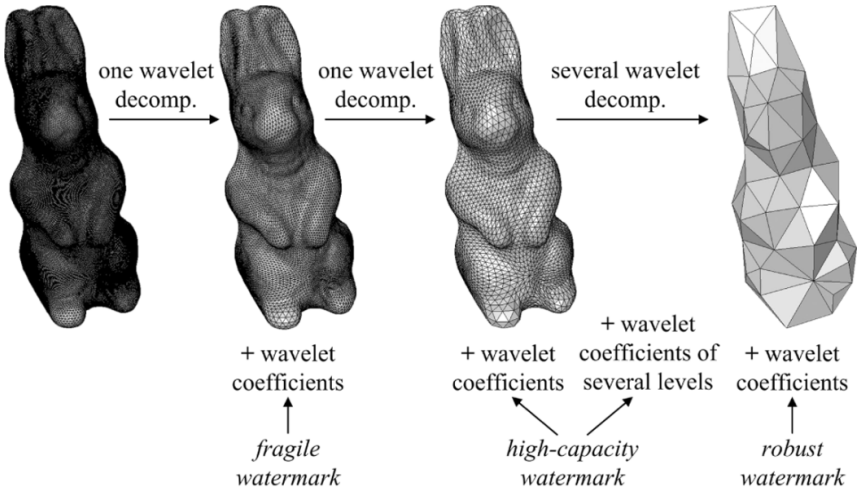


Figure 2.3: The illustration of the hierarchical watermarking framework proposed in [Wang et al., 2008]. This figure corresponds to Figure 2 from [Wang et al., 2008]

Multiple iterations of 3D wavelet decomposition of the triangle mesh is the foundation of the hierarchical watermarking method. At each iteration of the 3D wavelet decomposition, the mesh is decomposed into a coarser mesh representation and a set of WCVs. The 3D wavelet decomposition continues recursively, until the mesh obtained cannot be decomposed any further. A hierarchical 3D watermarking methodology based on the wavelet transform was developed by Wang *et al.* [Wang et al., 2008], which includes three different algorithms, each one enforcing one of the following requirements: robustness, high-capacity and fragility for authentication. The framework of this method is illustrated in Figure 2.3. A robust watermark is embedded by modifying the norms of the WCVs associated with the lowest resolution level of the mesh. Then, a denser mesh is reconstructed based on the lowest resolution mesh and its corresponding watermarked WCVs. In order to embed a high bit capacity watermark code, at each iteration, the WCVs are indexed according to the lengths of their support edges for synchronization. A controlling parameter is defined as $p = l_{av}/\epsilon_{hc}$, where l_{av} is the average length of the WCVs' support edges, and ϵ_{hc} is a specific constant. The watermark is embedded by adjusting the residuals of the WCVs' norms divided by the parameter p, into a certain sequence. Finally, the fragile watermark is embedded before

the last iteration of the mesh reconstruction. In the case of fragile watermark embedding, an identical symbol is embedded by two different embedding approaches. One approach is based on the quantization of the angle between the WCV and its support edge using the M-symbol scalar Costa scheme [Eggers et al., 2003], which depends on a quantization step Δ_θ. In the second embedding approach, the symbol is embedded by adjusting the norm-length ratio of the WCV and its support edge. Finally, the hierarchically watermarked mesh is reconstructed to the size of the cover-object.

2.1.4 Embedding through changing the ordering of the vertices

Changing the ordering of vertices in the mesh representation, using permutations, has been used for 3D steganography in certain approaches. In this case by the mesh representation we mean the order or list in which the vertices and faces are stored in the files representing the 3D object.

In [Cheng and Wang, 2006], Cheng and Wang used the vertex representation order, face representation order and the face index order, with respect to a conventionally defined vertex or face ordering, in order to embed secret information. The algorithm firstly produces a specific ordering of the vertices which is geometrical invariant and which is then used as a reference order for embedding. The vertex ordering in the sequence is used to embed the information. A similar embedding idea is applied to the face representation order. When embedding information by modifying the face index, a triangular face is corresponding to one of three states according to the type of face index order. The algorithm uses two triangles as a unit for embedding, resulting into nine states, which can embed three bits of information.

A high capacity steganographic method, proposed in [Bogomjakov et al., 2008], also embeds information by modifying the order of the vertex. It firstly computes a reference ordering of faces and vertices, as in the Edgebreaker mesh compression algorithm [Rossignac, 1999]. In the second stage, the message is encoded as a permutation of the mesh vertices and faces relative to their reference order. During the extraction, the message is decoded by comparing the representation order to the reference one. During the following year, Huang *et al.* [Huang et al., 2009] improved the embedding capacity of the permutation steganography proposed in [Bogomjakov et al., 2008].

As for the advantages of the steganographic methods in the representation domain, first of all, they would not cause any distortions, because they do not actually change the coordinates of the vertices at all. Moreover, the embedding capacity of these methods is very high, resulting into more than 10 Bits Per Vertex (BPV). However, some fatal flaws limit the application of these steganographic methods. The third party can obtain the embedded messages if they find the reference order which is generated by public algorithms. In addition, these methods are fragile to the vertex reordering attack. If the order of the vertices is changed, the recipient can no longer extract the message correctly.

There are a few 3D steganographic algorithms using the connectivity of the mesh as the carrier, for instance, the algorithm proposed in [Amat et al., 2010]. These algorithms do not change the geometry of the mesh surface, so no distortion is produced to the 3D objects. Nevertheless, their weakness consists in the fact that they are vulnerable to the remeshing attack.

2.2 Image and video steganalysis

In the following we discuss some approaches adopted for image and video steganalysis. Image steganalysis has received significant attention from the academic community since 1998 [Johnson and Jajodia, 1998]. The early steganalytic approaches for digital images usually would just detect the changes produced by a particular information hiding algorithm by considering some flaws of the embedding algorithm. For example, Westfeld and Pfitzmann [Westfeld and Pfitzmann, 1999] presented a chi-square test to detect the changes of the Discrete Cosine Transform (DCT) coefficients caused by the JSteg [Upham, 1997] embedding algorithm. Fridrich *et al.* [Fridrich et al., 2000] proposed a steganalytic approach based on the observation that the Least Significant Bit (LSB) encoding would increase the number of unique colors in the true-color images.

A concept of calibration was introduced in [Fridrich et al., 2002a] as a method of using a reference image in order to improve features' sensitivity to embedding changes while reducing image-to-image variations. Based on the histograms of low frequency DCT coefficients of the original image and those of the reference image, the steganalytic approach

proposed in [Fridrich et al., 2002a] can estimate the length of the secret message embedded by the steganographic algorithm F5 [Westfeld, 2001]. Later, some higher-order features for Joint Photographic Experts Group (JPEG) image steganalysis were proposed in [Fridrich, 2004] which considers the absolute differences between the statistics that extracted from the original images and those of the reference image as the steganalytic features. Image calibration was used in image steganalysis in many approaches[Ker, 2005, Pevný and Fridrich, 2007, Kodovskỳ and Fridrich, 2012].

A Markov process-based steganalytic approach used to identify the changes produced by the JPEG steganography was proposed in [Shi et al., 2006]. In this approach, differences between 2-D arrays of JPEG coefficients corresponding to horizontal, vertical and oblique directions between the cover-images and stego-images, are modelled as a Markov process. The elements from the transition probability matrices are then considered as features in the classifier used as steganalyzer. A truncation technique is used to limit the range of the elements in the transition probability matrix in order to reduce the dimensionality of the feature vectors. In another approach [Chen and Shi, 2008], the interblock correlation among the JPEG coefficients are used to improve the steganalysis performance. The idea of modelling the differences between pixels by higher-order Markov chains was proposed in [Pevny et al., 2010]. This approach calculates the difference between the adjacent pixels along different directions and obtains the difference arrays. Then, it models the difference arrays by both first-order and second-order Markov processes. The elements in the corresponding transition probability matrices are considered as a 686-dimensional feature set, called the Subtractive Pixel Adjacency Model (SPAM).

Many high-dimensional features are proposed for image steganalysis in the case of adaptive steganographic algorithms, such as Highly Undetectable steGO (HUGO) [Pevnỳ et al., 2010], Wavelet Obtained Weights (WOW) [Holub and Fridrich, 2012], Uniform Embedding Distortion (UED) [Guo et al., 2012], UNIversal WAvelet Relative Distortion (UNIWARD) [Holub and Fridrich, 2013]. For example, the steganalytic approach proposed in [Kodovskỳ and Fridrich, 2011] considers using the co-occurrence matrices of selected DCT coefficient pairs, together with the features proposed in [Chen and Shi, 2008] and [Pevný and Fridrich, 2007], as a 48,600-dimensional feature set. Another high-dimensional feature set, Spatial

Rich Models (SRM) [Fridrich and Kodovský, 2012], is based on the co-occurrence matrices of neighboring samples from the truncated and quantized noise residuals of the image obtained by using linear and non-linear high-pass filters. Some stgeanalytic approaches such as [Tang et al., 2014, Denemark et al., 2014] improved the SRM features by considering the knowledge of the embedding change probabilities, according to the distortion functions used by the adaptive steganographic algorithms. Song *et al.* [Song et al., 2015] proposed a steganalytic approach for JPEG compressed images which extracts the histogram features from the residuals of the image obtained by using the 2D Gabor filters with different scales and orientations.

Machine learning tools used for image steganalysis are also very important for the performance of the steganalyzer. Fisher Linear Discriminant (FLD) and Support Vector Machine (SVM) were among the machine learning methods used in the early studies of image steganalysis [Farid, 2002, Lyu and Farid, 2002, Pevný and Fridrich, 2007]. However, because the dimensionality of the steganalytic feature set has significantly increased, the curse of dimensionality led to a high complexity of training and the degradation of the generalisation ability of the steganalyzer. In order to solve this problem, the framework of using an ensemble of base FLD classifier for image steganalysis was proposed in [Kodovskỳ and Fridrich, 2011, Kodovskỳ et al., 2012]. During the training of this framework, multiple base classifiers as FLDs are trained over various subsets of the feature set. Then, the base classifiers are combined by taking a majority vote of their decisions, which means that, for each instance, the class chosen by most number of base classifier is the ensemble decision. Since the dimensionality of the feature subset is much lower than that of the whole set, the complexity of training a base classifier is much lower. Furthermore, the combination of multiple base classifiers can produce a stronger classifier ensemble with better generalization ability. Consequently, the FLD ensemble has been the most popular machine learning tool used in the area of image steganalysis [Fridrich and Kodovský, 2012, Tang et al., 2014, Denemark et al., 2014, Song et al., 2015].

Deep learning methods have recently been applied to image steganalysis [Tan and Li, 2014, Qian et al., 2015, Xu et al., 2016, Yang et al., 2017a, Ye et al., 2017, Wu et al., 2017] and some deep learning approaches achieved better performance than other steganalytic

approaches. It is noted that the structure of the Convolutional Neural Network (CNN) used for steganalysis is quite different from that of the CNN used in the computer vision tasks, such as image recognition and classification. For instance, the weights in the first layer of the CNN proposed in [Ye et al., 2017] are initialized with the high-pass filter set used in SRM, which aims to suppress the image content. Meanwhile, the output of the first layer of the CNN is processed by a new activation function called the Truncated Linear Unit (TLU), which truncates the output to a fixed range. Furthermore, the knowledge of the selection channel is exploited to boost the steganalysis performance for the adaptive steganographic algorithms. The CNN-based steganalytic approach, proposed in [Ye et al., 2017], provides better performance than SRM [Fridrich and Kodovský, 2012], maxSRM [Denemark et al., 2014] and FLD ensemble classifier when detecting the stego-images embedded by certain adaptive steganography.

Most of the video steganalytic approaches aim to detect the motion vector modification produced by the video steganographic methods. The motion vectors are the results of block-based motion estimation during video coding, which are commonly exploited by the video steganography. Su *et al.* [Su et al., 2011] proposed a video steganalytic method based on the statistical analysis of the neighboring motion vectors in both spatial and temporal domain. Later, Cao *et al.* [Cao et al., 2012] applied the calibration technique to video steganalysis and proposed the features characterizing the changes of the motion vectors during recompression in order to differentiate the stego-video from the cover-video.

The works from [Wang et al., 2014] and [Ren et al., 2014] pointed out that the modifications of the motion vectors may destroy the local optimality which is the aim of the motion estimation in video compression. Following the same framework proposed in [Wang et al., 2014], Zhang *et al.* [Zhang et al., 2017] improved the method of checking the local optimality of motion vectors by considering both distortion and bit estimation associated with motion vectors.

While 3D objects can be represented in various ways, their most usual data representation is by means of meshes. Such irregular representations, modelling complex 3D objects, are very different from the regular structural arrays representing digital images or video signals. Consequently, the existing image and video steganalytic algorithms cannot be successfully

applied to 3D objects. In the following, we would introduce the approaches adopted for 3D steganalysis so far.

2.3 3D steganalysis

There are much fewer steganalytic approaches for 3D objects than for images and video signals. The first steganalytic algorithm for 3D meshes was proposed in [Yang and Ivrissimtzis, 2014]. This 3D steganalytic algorithm is based on the features of 3D meshes and by using machine learning, for distinguishing stego-objects from cover-objects. During 3D steganalysis, both cover- and stego-objects are smoothed using one Laplacian smoothing iteration. Then, the geometric features such as the vertex position and norm in Cartesian and Laplacian coordinate systems [Yang and Ivrissimtzis, 2010], the dihedral angle of edges and face normals, are extracted from the original mesh and the smoothed one. It calculates the absolute differences between the features from the original mesh and those from the smoothed mesh. The feature vectors used for steganalysis are the four statistical moments of the logarithm of the absolute differences between the object and its smoothed counterpart. Meanwhile, the histograms of the differences between the features corresponding to the original objects and their smoothed counterparts are formed and used for extracting the steganalytic features. Finally, this steganalytic approach uses quadratic discriminate analysis to train the classifiers for separating the stego-objects, produced by several steganographic algorithms from their corresponding cover-objects. The experimental results show that the steganalyzers trained as quadratic classifiers achieve high detection rates for certain 3D information hiding algorithms.

More recently, Yang *et al.* [Yang et al., 2014, Yang et al., 2017b] proposed a new steganalytic algorithm, specifically designed for the robust 3D watermarking algorithm, MRS, proposed in [Cho et al., 2007]. During steganalysis, the number of bins, K, used in the watermarking algorithm is estimated using exhaustive search. For each K, the steganalytic algorithm classifies the bins into two clusters using a standard clustering algorithm fitting the data with a mixture of two Gaussian distributions. The estimate of K corresponds to that which maximizes the Bhattacharyya distance between the two clusters. Then it uses

a normality test to decide if the bins of the mesh can be modeled by a single Gaussian, in which case the mesh would not contain any hidden information. Otherwise, the distribution is bimodal and consequently the mesh is watermarked. The limitation of this algorithm is that it is only effective for the information embedded by the MRS algorithm and would not be useful when the mesh is embedded by other information hiding algorithms than MRS.

Kim *et al.* [Kim et al., 2017] extended the approach from [Li and Bors, 2016], which is presented in this thesis in Chapter 3, and proposed to use some additional features such as the edge normal, mean curvature and total curvature as supplement to LFS52 and formed LFS64 for 3D steganalysis. The improvement shown for LFS64 with respect to LFS52 is limited to the steganalysis of the information embedded by certain information hiding algorithms and the associated experiments are explained in the experimental part of Chapter 4.

2.4 The cover source mismatch problem

The Cover Source Mismatch (CSM) problem consists of the realistic scenario that the objects used for training a steganalyzer may be originated in a cover source that is different from those which the steganographier actually used for hiding information [Ker et al., 2013]. A thorough study of the implications for the CSM paradigm in image steganalysis was addressed during the "Break Our Steganographic System" (BOSS) contest [Bas et al., 2011]. The mismatch between the training set and testing set caused many difficulties to the participants in this contest [Bas et al., 2011, Fridrich et al., 2011, Gul and Kurugollu, 2011]. In the machine learning community, the methods of domain adaptation [Patel et al., 2015, Long et al., 2015] and transfer learning [Pan and Yang, 2010, Long et al., 2014] were studied in order to learn a target classifier using labeled data from a different distribution. More specifically, in image steganalysis, the CSM problem was addressed by considering the following aspects: the training sets, the feature set and the machine learning methods used for steganalysis.

In principle, in order to increase the generalisation ability of the steganalyzer we have to increase the diversity of the training set. In the case of digital images, the study [Kodovsky et al., 2014] mitigated the CSM's impact by training the steganalyzers over a mixture of

images taken by different cameras and by using a diversity of JPEG compression quality factors. Xu *et al.* [Xu et al., 2015] constructed an image set for training steganalyzer by selecting the samples from a large data set of images taken from various sources from the Internet. Meanwhile, they removed repetitions of images in order to reduce the redundancy of the training set.

The feature space is also a factor related to the impact of CSM paradigm. Gul and Kurugollu [Gul and Kurugollu, 2011] proposed a feature selection algorithm, in the context of the BOSS contest, which calculates the correlation between a feature vector and the embedding rate as the criterion for selecting the features. Pasquet *et al.* [Pasquet et al., 2014] proposed to use the ensemble classifier enabled with a feature selection mechanism in order to address the CSM problem. The feature selection was considered by evaluating the importance of each feature in the learning process in [Chaumont and Kouider, 2012]. A feature condensing method, called Calibrated Least Squares (CLS) was proposed in [Pevnỳ and Ker, 2013] to make the high dimensional feature sets compatible with the anomaly detector employed for steganalysis. A method to mitigate the CSM due to changes in the features of the cover image was presented in [Ker and Pevny, 2014]. This approach normalizes the cover features for all steganographiers by subtracting the centroid of their joint distribution.

Other research studies addressing the CSM problem in images aim to find a classifier that would be robust to the variation between training and testing data. In [Lubenko and Ker, 2012] it was shown that simple classifiers, such as the FLD ensemble and the Online Ensemble Average Perceptron (OEAP) had better performances than other more complex classifiers, when faced with the cover source mismatch problem. To mitigate the mismatch due to various changes in steganalytic features, Ker and Pevnỳ [Ker and Pevny, 2014] used an ensemble of classifiers which gave more weight to those classifiers that were robust to the changes in the steganalytic features. A similar weighting strategy for improving the FLD ensemble's performance in the CSM paradigm was presented in [Xu et al., 2015].

2.5 Summary

In this chapter, we revised the main 3D information hiding algorithms, the various approaches adopted for image and video steganalysis, the 3D steganalytic methods and the studies on solving the cover source mismatch problem in steganalysis. The 3D information hiding algorithms are categorized into three groups according to the domain they embed message in, namely, the spatial domain, the transform one and the vertex ordering domain. The basic ideas and the encoding procedure of the 3D embedding algorithms are introduced. Afterwards, the research studies on image and video steganalysis are briefly reviewed from two aspects, the statistical features and the machine learning tools used for steganalysis. The 3D steganography has received a significant attention during the last 20 years, almost similar to the one received by image and video steganography. Nevertheless, while there are several studies of image and video steganalysis, 3D steganalysis is still in its infancy. The cover source mismatch problem has recently been studied in the case of image steganalysis but not for 3D object steganalysis. This PhD thesis includes research studies for proposing new feature sets for 3D steganalysis. Moreover, it analyses for the first time, the efficiency of 3D steganalysis under the cover source mismatch paradigm.

Chapter 3

Local Geometry-Based Feature Set for 3D Steganalysis

3.1 Introduction

From the literature review about the 3D steganalysis, it is clear that the research field of 3D steganalysis is still in its infancy. 208 features have been proposed for 3D steganalysis in [Yang and Ivrissimtzis, 2014]. In this chapter, we propose to use, a new set of features derived from the local geometry of the 3D surface, for 3D steganalysis. The proposed features are used in combination with some of the features proposed in [Yang and Ivrissimtzis, 2014], while other features proposed before are dropped. We propose to use the statistics of the Gaussian curvature and the curvature ratio in order to capture the changes of the surface's curvatures. Furthermore, the vertex normal is also considered for finding the changes in the orientation of polygon faces containing a certain vertex. We also propose to use the statistics of spherical coordinates, and the length of the edge defined in spherical coordinates, for 3D steganalysis. The spherical coordinate system is often used for hiding information in 3D objects, but it has not been previously used in 3D steganalysis. The statistics of sets of 3D features are then fed into machine learning algorithms, for example, the Fisher Linear Discriminant (FLD) ensemble [Kodovskỳ et al., 2012], which was successfully used for image steganalysis.

In the experimental part, we consider 354 3D objects from the Princeton Mesh Seg-

mentation project [Chen et al., 2009] database for training and testing the proposed 3D steganalyzer. This database contains a large variety of shapes, covering the human postures, animals, tools and so on. The stego-objects are generated by applying six different embedding algorithms, which embed information into spatial or transform domain of the 3D objects.

With the aim of evaluating the proposed 3D steganalytic features, we use the proposed steganalytic features extracted from 260 pairs of cover- and stego-objects, produced by a certain information hiding algorithm, in order to train a steganalyzer. Then, we test the performance of the trained steganalyzer when identifying the stego-objects from the other 94 pairs of cover- and stego-objects.

This chapter was published in [Li and Bors, 2017]. The rest of this chapter is organized as follows: The description of the 3D steganalysis framework formulated in this study is provided in Section 3.2. The 3D feature set, used by the steganalyzer is presented in detail in Section 3.3. The experimental results are provided in Section 3.4, while the summary of this chapter are outlined in Section 3.5.

3.2 3D steganalysis framework

In this section, we provide a brief introduction of the 3D steganalysis framework. The steganalysis framework is treated as a machine learning problem, consisting of training and testing stages. The training of the steganalyzer has the following processing steps: preprocessing, feature extraction and learning, as illustrated in Figure 3.1. The result of these processing steps consists of a parameter set discriminating between the 3D objects carrying hidden information and those that are not. The testing stage includes the same preprocessing and feature extraction steps as the training stage, while applying the parameters learnt during the training on the features extracted from various sets of test objects.

Firstly, during the preprocessing stage, Laplacian smoothing is applied on all the graphical objects. The idea of 3D object smoothing was borrowed from image steganalysis, where it was observed that the difference between the stego-image and its smoothed version is more significant than the difference between the cover-image and its corresponding smoothed ver-

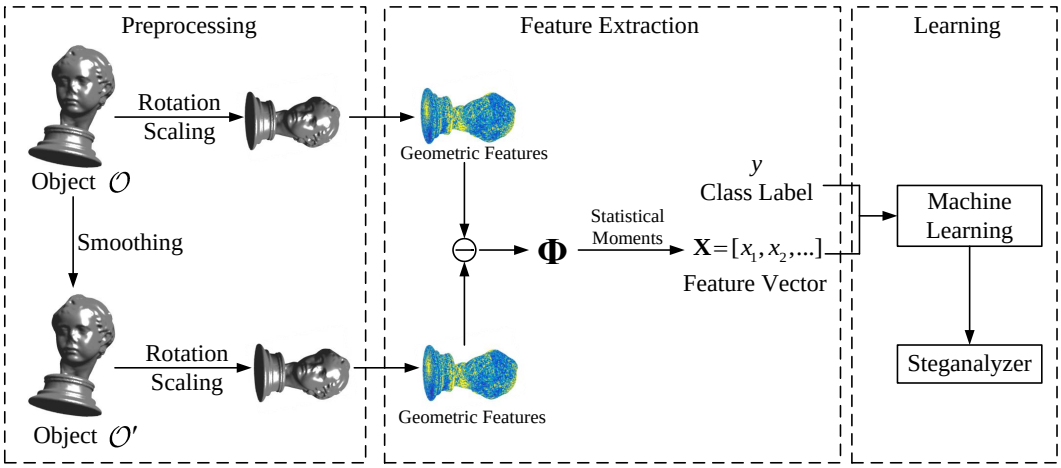


Figure 3.1: The 3D steganalysis framework based on learning from statistics of the local feature sets and classification by means of machine learning methods.

sion [Fridrich et al., 2002b, Kodovsky and Fridrich, 2009]. Similarly, it is expected that the difference between a mesh and its smoothed version is larger for a stego mesh than for a cover mesh. In most 3D watermarking algorithms, the changes produced to the stego-object, following the watermark embedding, can be associated to noise-like changes. Consequently, when smoothing a cover mesh, the resulting modifications will be smaller than those obtained when smoothing its corresponding stego mesh. We consider Laplacian smoothing for the object $\mathcal{O}$, resulting in its smoothed version $\mathcal{O}'$. Then the rotation and scaling are used to normalize the objects such that their size is constrained within a cube with sides of one, in order to eliminate the perturbation on the features, caused by the variation of the size in the objects from the training set.

Features, characterizing the local geometry of 3D objects are extracted after the preprocessing stage. In Section 3.3 we propose to use a new set of 3D features for steganalysis. Then, $\mathbf{\Phi}$ represents the absolute difference between the geometric feature extracted from the object $\mathcal{O}$ and its smoothed version $\mathcal{O}'$. In order to statistically model the difference, we consider the first four statistical moments, representing the mean, variance, skewness and kurtosis, of the logarithm of $\mathbf{\Phi}$ as the feature vector $\mathbf{X}$, which is then used as an input to a machine learning algorithm along with the class label y for the object $\mathcal{O}$.

In the supervised learning phase, a classifier is trained over the extracted feature vectors and the corresponding class labels of the objects. The classifier, which is also known as the

steganalyzer, separates the feature space defining the stego-objects from that of the cover-objects. In this research study we propose to use the FLD ensemble [Kodovskỳ et al., 2012] for training the steganalyzer.

3.3 Local geometry-based feature set

3D watermarking and steganographic methods are specifically designed to embed information in a way that does not visibly alter the surface of the objects [Luo and Bors, 2011, Bors and Luo, 2013]. 3D steganalysis aims to find computationally such changes, separating the stego-objects from the cover-objects. Depending on the specific algorithm used, such changes could be randomly distributed on the surface of the 3D mesh [Bors, 2006] or they could be specifically located in certain regions of the object [Alface et al., 2007]. Artefacts produced in objects, following the information hiding embeddings, could be assimilated to low level protuberances on mesh surfaces and consequently could be identified by feature detection algorithms. In the following we outline some 3D local features which can be used for identifying whether objects have been watermarked or not. Such feature detectors range from very simple vertex displacement measurements to algorithms that take into account the local neighbourhoods and measure specific shape characteristics.

Let us assume that we have the shape of a 3D object, considered as a cover mesh $\mathcal{O} = \{V, F, E\}$, containing the vertex set $V = \{v_i | i = 1, 2, \ldots, |V|\}$, where $|V|$ represents the number of vertices in the object $\mathcal{O}$, its face set F, and its edge set E, respectively. We define the neighbourhood $\mathcal{N}(v_i)$ of a vertex v_i as $\{v_j \in \mathcal{N}(v_i) | e_{(i,j)} \in E\}$, where $e_{(i,j)}$ is the edge connecting vertices v_i and v_j.

3.3.1 Preprocessing

As already mentioned in Section 3.2, the preprocessing of the 3D objects is an essential phase for extracting the steganalytic features. One iteration Laplacian smoothing is firstly applied to the 3D object $\mathcal{O}$, which updates the vertex v_i into v_i' as follows, [Taubin, 1995]:

$$v_i' \leftarrow v_i + \frac{\lambda}{\sum_{v_j \in \mathcal{N}(v_i)} w_{ij}} \sum_{v_j \in \mathcal{N}(v_i)} w_{ij}(v_j - v_i), \tag{3.1}$$

where λ is a scale factor and w_{ij} are the weights defined as:

$$w_{ij} = \begin{cases} 1 & \text{if } v_j \in \mathcal{N}(v_i) \\ 0 & \text{otherwise} \end{cases} \tag{3.2}$$

.

The value of λ is found empirically, such that the object is appropriately smoothed. The object is afterwards aligned according to its first and second principal axes, given by the Principal Component Analysis (PCA). Afterwards, the object is scaled to fit inside a cube of sides equal to 1.

3.3.2 The YANG40 features

The 40-dimensional feature vector YANG40 contains the most effective features from YANG208, used in [Yang and Ivrissimtzis, 2014], which correspond to the statistics of features evaluated from the vertices, edges and faces that make up the given meshes. For YANG40 we remove certain features, which provide lower performance, from YANG208 and abandon the strategy used in [Yang and Ivrissimtzis, 2014] which treats the vertices with valence less, equal, or greater than six separately for the sake of reducing the dimensionality.

Let us denote by $\mathbf{\Phi}$, the feature set representing differences between the object $\mathcal{O}$ and its smoothed version $\mathcal{O}'$. The first six components of $\mathbf{\Phi}$ represent the absolute distance, measured along each coordinate axis x, y, z between the locations of vertices of the meshes $\mathcal{O}$ and $\mathcal{O}'$ after being normalized and aligned, in both the Cartesian and Laplacian coordinate systems [Yang and Ivrissimtzis, 2010]:

$$\begin{aligned} \phi_1(i) &= |v_{x,c}(i) - {v'}_{x,c}(i)|, \\ \phi_2(i) &= |v_{y,c}(i) - {v'}_{y,c}(i)|, \\ \phi_3(i) &= |v_{z,c}(i) - {v'}_{z,c}(i)|, \end{aligned} \tag{3.3}$$

$$\begin{aligned}\phi_4(i) &= |v_{x,l}(i) - v'_{x,l}(i)|,\\ \phi_5(i) &= |v_{y,l}(i) - v'_{y,l}(i)|,\\ \phi_6(i) &= |v_{z,l}(i) - v'_{z,l}(i)|,\end{aligned} \tag{3.4}$$

where $v_{x,c}(i)$ and $v_{x,l}(i)$ represent the x-coordinate of v_i in Cartesian and Laplacian coordinate systems, respectively, $i = 1, 2, \ldots, |V|$. The Laplacian coordinates of the object are the results of the Cartesian coordinates multiplied by the *Kirchhoff* matrix [Bollobás, 2013] of the object. Next, we evaluate the changes produced in the Euclidean distance between vertex locations and the centre of the object, representing the vertex norms. The absolute differences between the vertex norms of pairs of corresponding vertices in the meshes $\mathcal{O}$ and $\mathcal{O}'$ are calculated as:

$$\phi_7(i) = |\|\mathbf{v}_c(i)\| - \|\mathbf{v}'_c(i)\|| \tag{3.5}$$

$$\phi_8(i) = |\|\mathbf{v}_l(i)\| - \|\mathbf{v}'_l(i)\|| \tag{3.6}$$

where $\|\mathbf{v}_c(i)\|$, $\|\mathbf{v}_l(i)\|$, represent the vector norms of v_i in Cartesian and Laplacian coordinates, respectively, for $i = 1, 2, \ldots, |V|$.

Another feature evaluates the local mesh surface variation by calculating the changes in the orientations of faces adjacent to the same edge. This is measured by the absolute differences between the dihedral angles of neighbouring faces, calculated in the plane perpendicular on the common edge $\{e_i \in \mathcal{E} | i = 1, 2, \ldots, |E|\}$, where $|E|$ represents the number of edges of the object $\mathcal{O}$:

$$\phi_9(i) = |\theta_{e_i} - \theta'_{e_i}|, \tag{3.7}$$

where the calculation of the dihedral angle $\theta_{e(i)}$ is illustrated in Figure 3.2.

Changes in the local surface orientation are measured by calculating the angle between the surface normals $\vec{N}_{f_i}$, $f_i \in \mathcal{F}$, of the faces from the object $\mathcal{O}$, and their correspondents $\vec{N}_{f'_i}$, $f'_i \in \mathcal{F}'$, from the smoothed object $\mathcal{O}'$:

$$\phi_{10}(i) = \arccos \frac{\vec{N}_{f_i} \cdot \vec{N}_{f'_i}}{\left\|\vec{N}_{f_i}\right\| \cdot \left\|\vec{N}_{f'_i}\right\|} \tag{3.8}$$

where $i = 1, 2, \ldots, |F|$. The 40-dimensional feature vector YANG40 represents the first four

statistical moments, representing the mean, variance, skewness and kurtosis of the logarithm of the ten vectors $\{\phi_i | i = 1, 2, \ldots, 10\}$, described above.

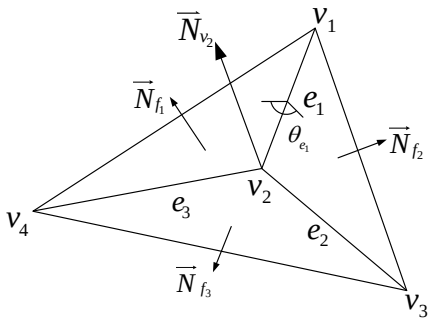


Figure 3.2: Dihedral angles and vertex-based normals for representing local geometry properties of the surface.

3.3.3 The vertex normal and curvature features

In the following we propose to use some additional 3D features. These features model localized geometrical properties of the 3D shapes, and following extensive experimentation, they have shown to be efficient for steganalysis. The vertex normal is the weighted sum of the normals of all faces that contain the vertex [Max, 1999]. A vertex normal is illustrated in Figure 3.2 and is computed as:

$$\vec{N}_{v_i} = \sum_{f_j} \frac{A(f_j) \cdot \vec{N}_{f_j}}{\|e_{(v_i,1)}\|^2 \cdot \|e_{(v_i,2)}\|^2} \tag{3.9}$$

where f_j represents the jth face that contains the vertex v_i, $A(f_j)$ represents its area, $e_{(v_i,1)}$ and $e_{(v_i,2)}$ are the two edges containing v_i in the face f_j. The weighting scheme used here was proposed in [Max, 1999], which produces more accurate vertex normal estimates than other weighting approaches. The equal weighted approach is not the most appropriate when the faces surrounding a vertex vary significantly in size. Some other approaches for estimating surface normal are described in [Jin et al., 2005].

The change between two vertex normals is calculated as a dot product:

$$\phi_{11}(i) = \arccos \frac{\vec{N}_{v_i} \cdot \vec{N}_{v_i'}}{\left\|\vec{N}_{v_i}\right\| \cdot \left\|\vec{N}_{v_i'}\right\|} \tag{3.10}$$

where $\vec{N}_{v'_i}$ is the normal for a vertex from the smoothed object $\{v'_i \in \mathcal{O}' | i = 1, 2, \ldots, |V|\}$.

Next we consider the local shape curvatures, calculated according to the Gaussian curvature and the curvature ratio formula used in [Rugis and Klette, 2006]. Most natural objects have shapes with many curvatures all over their surface. Steganographic algorithms would tend to embed changes that may influence the local curvatures. In differential geometry, the two principal curvatures of a surface are provided by the eigenvalues of the shape operator, calculated at the location of a vertex using the vertices from its first neighbourhood. Such curvatures measure how the local surface bends by different amounts in the orthogonal directions at that point. The Gaussian curvature is defined as:

$$K_G = K_1 K_2, \tag{3.11}$$

where K_1 is the minimum principal curvature and K_2 is the maximum principal curvature at a given point [Rusinkiewicz, 2004]. A special case is that of singularity in the shape operator, when we have a linear dependency in one direction or in both. In this case we have locally a planar region, which is characterized by a linear relationship among its coordinates and consequently by zero curvature. In our study we found that the curvature ratio proposed in [Rugis and Klette, 2006], defined as

$$K_r = \frac{\min(|K_1|, |K_2|)}{\max(|K_1|, |K_2|)}, \tag{3.12}$$

is effective to be used as a feature when training steganalyzers. The Gaussian curvature from equation (3.11) and the curvature ratio from equation (3.12) have been shown to be sensitive to very small mesh modifications and have been used to model 3D shape characteristics in various applications [Tombari et al., 2013, Vieira et al., 2016]. The two principal curvatures are evaluated at the location of each vertex in the object $\mathbf{v}(i) \in \mathcal{O}$ and for its corresponding vertex from the smoothed object $\mathbf{v}'(i) \in \mathcal{O}'$. Their absolute differences represent the features ϕ_{12} and ϕ_{13} used in the proposed set of features:

$$\phi_{12}(i) = |K_G(v_i) - K_G(v'_i)|, \tag{3.13}$$

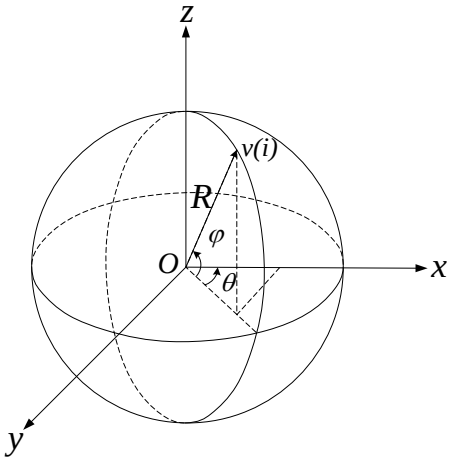


Figure 3.3: The spherical coordinate system, where R is the radial distance of vertex v_i, θ and φ are its azimuth angle and elevation angle, respectively.

$$\phi_{13}(i) = |K_r(v_i) - K_r(v_i')|, \tag{3.14}$$

for $i = 1, 2, \ldots, |V|$.

3.3.4 The spherical coordinates features

Spherical coordinates provide a straight forward representation for most graphical objects in characterizing the distance from the centre and the location of each vertex on a sphere. Certain 3D watermarking methods, such as those from [Cho et al., 2007, Yang et al., 2017b], specifically embed changes into spherical coordinates. We convert the 3D objects from the Cartesian coordinate system to the spherical coordinate system, considering the centre of the object as its reference.

The spherical coordinate system specifies a point in the 3D space by a radius and two angles and the link to the Cartesian coordinate system is given by:

$$\begin{aligned} v_x &= R\cos(\varphi)\cos(\theta) \\ v_y &= R\cos(\varphi)\sin(\theta) \\ v_z &= R\sin(\varphi) \end{aligned} \tag{3.15}$$

where $\mathbf{v} = (v_x, v_y, v_z)$ represents the Cartesian coordinates of the vertex, and (R, θ, φ) its spherical coordinates, representing R, the Euclidean norm from a fixed origin, θ, the azimuth

angle, while φ is the elevation angle, as illustrated in Figure 3.3. We compute the absolute differences of the spherical coordinates of all vertices, $\{(R(i), \theta(i), \varphi(i))$ between the original object $\mathcal{O}$ and the smoothed object $\mathcal{O}'$ in the spherical coordinate system:

$$\begin{aligned} \phi_{14}(i) &= |\theta(i) - \theta'(i)|, \\ \phi_{15}(i) &= |\varphi(i) - \varphi'(i)|, \\ \phi_{16}(i) &= |R(i) - R'(i)| \end{aligned} \tag{3.16}$$

where $i = 1, 2, \ldots, |V|$. The centre of the spherical coordinate system is O, representing the centre of the 3D object calculated by averaging all the vertices in the object, as shown in Figure 3.3.

We also use the statistics of the edges, defined in the spherical coordinate system. In this case, the lengths of the edges are defined by the differences in the spherical coordinates of the two vertices that define the edge ends:

$$\begin{aligned} K_\theta(e_{(i,j)}) &= |\theta(i) - \theta(j)|, \\ K_\varphi(e_{(i,j)}) &= |\varphi(i) - \varphi(j)|, \\ K_R(e_{(i,j)}) &= |R(i) - R(j)| \end{aligned} \tag{3.17}$$

where $e_{(i,j)}$ is the edge connecting vertices v_i and v_j, and $e_{(i,j)} \in E$. The corresponding features extracted from both the original object and its smoothed version are

$$\begin{aligned} \phi_{17}(i) &= |K_\theta(i) - K'_\theta(i)|, \\ \phi_{18}(i) &= |K_\varphi(i) - K'_\varphi(i)|, \\ \phi_{19}(i) &= |K_R(i) - K'_R(i)| \end{aligned} \tag{3.18}$$

where, for example, $K_\theta(i)$ is obtained from the ith edge of the original object, while $K'_\theta(i)$ from its corresponding edge in the smoothed object, for $i = 1, 2, \ldots, |E|$, $|E|$ is the total number of edges in object $\mathcal{O}$.

It was observed that most histograms of the features, such as those mentioned above, are highly skewed towards smaller values, resulting in an almost exponential distribution.

It is known that by using the log transform we can transform such distributions in order to make them more symmetric. In order to introduce evenness in the distribution of the features, we apply the logarithm for all features. Then, we consider the first four statistical moments, representing the mean, variance, skewness and kurtosis, of the logarithm of all vertex normals, Gaussian curvatures, curvature ratios, and the spherical coordinate features calculated as indicated above, as it was done in the case of the feature set YANG40, presented in Section 3.3.2. Another option for modelling the features' distribution is by fitting the distribution with a function defined by certain parameters. However, there is no a priori justification for selecting a particular distribution for modelling such features.

In this way we define a vector $\mathbf{X}$ of $19 \times 4 = 76$ dimensions, which we call LFS76. The first four moments capture almost entirely the statistical characteristics of the distribution of the features, representing their centre and the deviation from the centre, as indicated by the mean and variance, respectively. The degree of symmetry in the logarithm of feature values is indicated by the skewness, while the level of peakedness and the presence of specific values in the statistical distribution is indicated by the kurtosis, representing the fourth statistical moment.

A subset of the proposed feature set, LFS52, was used in [Li and Bors, 2016]. That feature set did not include the 24-dimensional feature vector extracted in the spherical coordinate system of 3D objects. A higher dimensional feature set, used in [Yang and Ivrissimtzis, 2014], is represented by the 208-dimensional vector defined as YANG208. This feature set considers separately the statistics of the first eight features described above, distinctly on vertex sets with valences less, equal, or greater than six. Moreover, YANG208 feature set considers the histogram differences of the ten features defined in Section 3.3.2, as well. The features described in this section are mainly local and are centred on either the vertices or the edges or the faces forming the 3D meshes of the objects. During the experimental study we have tested other features, such as the Heat Kernel Signature (HKS) [Sun et al., 2009], representing larger regions of 3D objects. We obtained 400-dimensional steganalytic features based on HKS of the object by using the first four statistical moments to model the first 100 frequencies of the HKS. The detection errors of the FLD ensemble classifiers trained over the HKS-based features for three information hiding algorithms are shown in Table 3.1. The

training of the steganalyzer and parameters for the embedding are identical to the settings given in Section 3.4.5. It is shown that the HKS-based feature is not suitable for steganalysis according to the poor results from Table 3.1. This is because the information embedding only produces slight distortions locally, rather than changing the global shape of the object, so the HKS-based feature cannot capture such changes.

Table 3.1: Median values and the standard deviations of the detection errors for the steganalysis results of three information hiding algorithms when using the FLD ensemble classifier trained over the HKS-based features for 30 different splits of the training/testing sets.

Feature set	Information hiding methods		
	SRW	MRS	WHC
HKS	0.5000($\pm$0.0061)	0.5000($\pm$0.0065)	0.5000($\pm$0.0056)

The feature set described in this section is used as an input to a machine learning classifier. The machine learning classifier has two stages. During the first stage, it learns the feature spaces characterizing the stego-objects and the cover-objects respectively, and estimates the boundary between the two classes. Then, during a test stage, the parameters, learnt during the initial stage, are used for identifying new stego-objects, which have not been used during the training stage. A machine learning classifier is expected to provide a good generalization. For this study we are using the Fisher Linear Discriminant (FLD) ensamble, and Quadratic Discriminant Analyser (QDA), as classifiers for 3D steganalysis. The quadratic discriminant fits multivariate normal densities with covariance estimates [Krzanowski, 2000] and was used for 3D steganalysis in [Yang and Ivrissimtzis, 2014]. The FLD ensemble classifier was successfully used in image steganalysis [Denemark et al., 2016, Kodovskỳ et al., 2012, Yu et al., 2016].

3.4 Experimental results

In the following we provide the experimental results for the proposed 3D steganalytic methodology, when detecting the stego-objects obtained by six different information hiding methods. For the experimental data set we consider 354 3D objects represented as meshes which are part of the Princeton Mesh Segmentation project [Chen et al., 2009] database, which is a

combination of the databases of AIM@SHAPE[1], FOCUS K3D[2] projects and shapes from the Watertight Models Track of the SHape REtrieval Contest 2007 [Giorgi et al., 2007]. The shapes of ten objects from this database are shown in Figure 3.4.

Figure 3.4: 3D objects used in the steganalytic tests.

3.4.1 The 3D information hiding methods and their parameter settings

During the tests we consider detecting the 3D stego-objects obtained by hiding information using six different embedding algorithms: the Steganalysis-Resistant Watermarking (SRW) method proposed in [Yang et al., 2017b]; the two blind robust watermarking algorithms based on modifying the Mean or the Variance of the distribution of the vertices' Radial distance coordinates in the Spherical coordinate system, denoted as MRS and VRS, from [Cho et al., 2007]; the Wavelet-based High Capacity (WHC) watermarking method and Wavelet-based FRagile (WFR) watermarking method proposed in [Wang et al., 2008]; the Multi-Layer Steganography (MLS) provided in [Chao et al., 2009]. The embedded information is a pseudorandom bit stream which simulates the secret messages or watermarks hidden by the steganographier.

During the generation of the stego-objects using SRW method from [Yang et al., 2017b], we consider multiple values for the parameter K which determines the number of bins for the histogram of the radial distances for all vertices. According to [Yang et al., 2017b], the upper bound of the embedding capacity is $\lfloor (K-2)/2 \rfloor$. In our experiments we set the parameter $K \in \{32, 64, 96, 128\}$ and thus obtain multiple sets of stego-objects. Another parameter in

[1] http://cordis.europa.eu/ist/kct/aimatshape_synopsis.htm
[2] http://cordis.europa.eu/fp7/ict/content-knowledge/projects-focus-k3d_en.html

the watermarking method from [Yang et al., 2017b] is n_{thr} which controls the robustness of the embedding method. In order to keep the distortion of the embedding to a relatively low level, we set the parameter n_{thr} as 20. If the smallest number of the elements in the bins from the objects is less than 20, we would choose n_{thr} equal to the smallest nonzero number of the elements in the bins. Examples of stego-objects obtained using SRW method are shown in Figure 3.7(a) and Figure 3.8(a), where $K = 128$.

For MRS and VRS watermarking methods from [Cho et al., 2007], we consider various values for the watermark strength, such as $\alpha \in \{0.02, 0.04, 0.06, 0.08, 0.1\}$, while fixing the incremental step size to $\Delta k = 0.001$ and the message payload as 64 bits. Larger values of strength can increase the robustness of the watermark, but also enlarge the extent of the embedding modifications. An example of a stego-object obtained using MRS method is shown in Figure 3.7(f), where the watermark strength factor is set as $\alpha = 0.04$.

In both WHC and WFR watermarking methods, proposed in [Wang et al., 2008], the information is embedded in the wavelet coefficient vectors obtained just after one wavelet decomposition of the original mesh, but the modifications are made in different ways for each of these algorithms. During the watermark embedding by WHC, the wavelet coefficient vectors' norms are firstly divided by the parameter p. Then the resulting residues, representing the differences from the rounding error, are changed accordingly in order to generate a particular permutation which carries the watermark. The parameter p is obtained by dividing the average edge length for the entire object by the control parameter ϵ_{hc}, which is set to the values from the set {50, 100, 500, 1000}. When using WFR to embed information, the angle between the wavelet coefficient vector and its associated edge is changed, where Δ_θ is the quantization step used to establish the codebook. To investigate the influence of parameter Δ_θ on the steganalysis results, we set it to the values from the set $\{\pi/6, \pi/4, \pi/3, \pi/2\}$. The other parameters involved in WHC and WFR are all exactly set to the values suggested in [Wang et al., 2008].

When using MLS method from [Chao et al., 2009], we increase the number of layers from 2 to 10, with a step of 2, and we consider the number of intervals as 10000. Increasing the number of embedding layers in this steganographic method corresponds to increasing the payload capacity. During the embedding, all the vertices in the mesh are used as payload

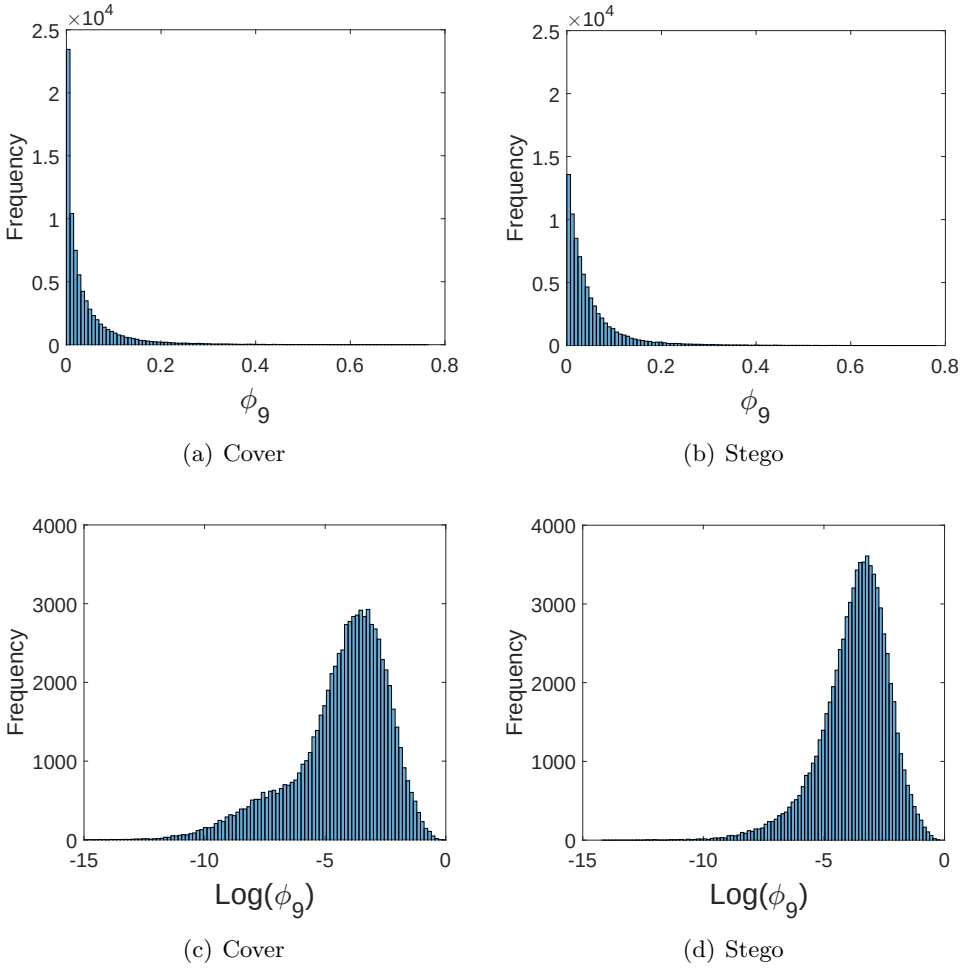
carriers, except for three vertices which are used as references for the extraction process. A stego-object obtained using MLS method is shown in Figure 3.7(k), where the number of layers is 10.

3.4.2 Feature extraction

The steganalytic features are extracted from the cover-objects and the corresponding stego-objects obtained after embedding the information by using the six steganographic algorithms mentioned above. During the preprocessing, we first apply one iteration of Laplacian smoothing on both cover-objects and stego-objects, by setting the scale factor $\lambda = 0.2$. We consider the proposed feature set LFS76, discussed in Section 3.3 and compare their results against YANG208, proposed in [Yang and Ivrissimtzis, 2014], its simplified version, called YANG40, and the feature set LFS52, which was proposed in our previous work [Li and Bors, 2016]. We also consider the feature sets combining LFS52 and the features defining the Vertices' Spherical coordinates, VS12, representing the mean, variance, skewness and kurtosis of ϕ_{14}, ϕ_{15} and ϕ_{16} from equation (3.16), the combination of LFS52 and the features defining the Edge length in the Spherical coordinate system, ES12, representing the mean, variance, skewness and kurtosis of ϕ_{17}, ϕ_{18} and ϕ_{19} from equation (3.18).

Figures 3.5(a) and (b) show the histograms of the dihedral angle feature ϕ_9, calculated according to equation (3.7), for the cover-object and stego-object, respectively, for the object "Head statue", shown in Figure 3.7(f). The histograms of the logarithm of ϕ_9 are shown in Figures 3.5(c) and (d), for the cover-object and stego-object, respectively. Figures 3.6(a) and (b) show the histograms of the vertex normal feature ϕ_{11} calculated according to equation (3.10), while Figures 3.6(c) and (d) show the corresponding histogram of logarithms for the cover-object "Horse" shown in Figure 3.4, and its corresponding stego-object embedded by MRS method from [Cho et al., 2007]. From these figures, we can observe that following the application of the logarithm, the distributions of feature components ϕ_9 and ϕ_{11} become similar to normal distributions, where it is easier to model the differences between the geometric feature of the cover-object and stego-object using the first four statistical moments of mean, variance, skewness and kurtosis.

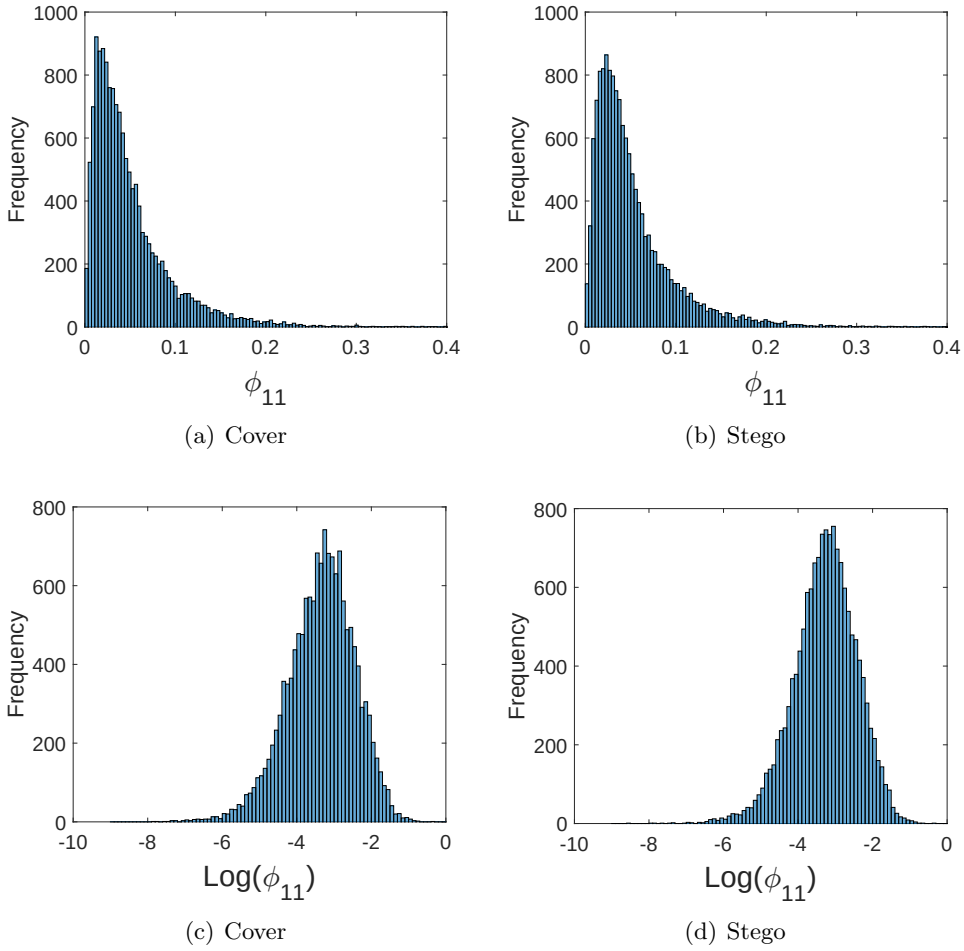
Figures 3.7(a) (f) and (k) show the stego-objects embedded by the information hiding



(a) Cover (b) Stego

(c) Cover (d) Stego

Figure 3.5: Histograms of dihedral angles feature ϕ_9 and its logarithm of the cover, in (a) and (c), and stego versions, in (b) and (d), of the "Head statue" object from the database from [Chen et al., 2009].



(a) Cover (b) Stego

(c) Cover (d) Stego

Figure 3.6: Histograms of vertex normal feature ϕ_{11} and its logarithm of the cover, in (a) and (c), and stego versions, in (b) and (d), of the "Horse" object from the database from [Chen et al., 2009].

methods, SRW [Yang et al., 2017b], MRS [Cho et al., 2007] and MLS [Chao et al., 2009], respectively. Figure 3.8(a) represents the stego-object obtained by using SRW method. From the second to the fifth columns of Figures 3.7 and 3.8 illustrate the absolute differences of the features between the cover-object and its corresponding stego-object, namely, vertex normals ϕ_{11}, the curvature ratios ϕ_{13}, the azimuth angles ϕ_{14} and the radial distances ϕ_{16}, depicted on the stego-objects. From these figures it can be observed that each feature identifies specific differences between the cover-object and stego-object, which usually does not overlap with those identified by the others.

3.4.3 Training the steganalyzers

The steganalyzers are trained as binary classifiers implemented using two methods: the Quadratic Discriminant Analysis (QDA) and the Fisher Linear Discriminant (FLD) ensemble. The FLD ensemble consists of a set of base learners trained uniformly on a randomly selected feature subset of the whole training data. The dimensionality of the random subspace and the number of base learners are found by minimizing the Out-Of-Bag (OOB) error, representing an estimate of the testing error calculated on bootstrap samples of the training set, [Duda et al., 2012]. Compared to the SVM classifier, the FLD ensemble can provide a comparable high accuracy, but with a relatively low computational cost. On the other hand, in the case of FLD ensemble, it is much easier to find the optimal tuning parameters. For more technical details of the FLD ensemble, we refer to the literature [Cogranne and Fridrich, 2015, Kodovskỳ et al., 2012]

For each steganalyzer, we split the 354 pairs of cover-object and stego-object into 260 pairs, used for training, and 94 pairs for testing. We consider 30 different splits of the given 3D object database, into the training and testing data sets. Two different assessment measures are used: the first one is the median value of the detection errors which are the sums of false negatives (missed detections) and false positives (false alarms) from all 30 trials, while the other one is the median value of the area under the Receiver Operating Characteristic (ROC) curves of the detection results, evaluated over the 30 splits of the data into training and testing sets.

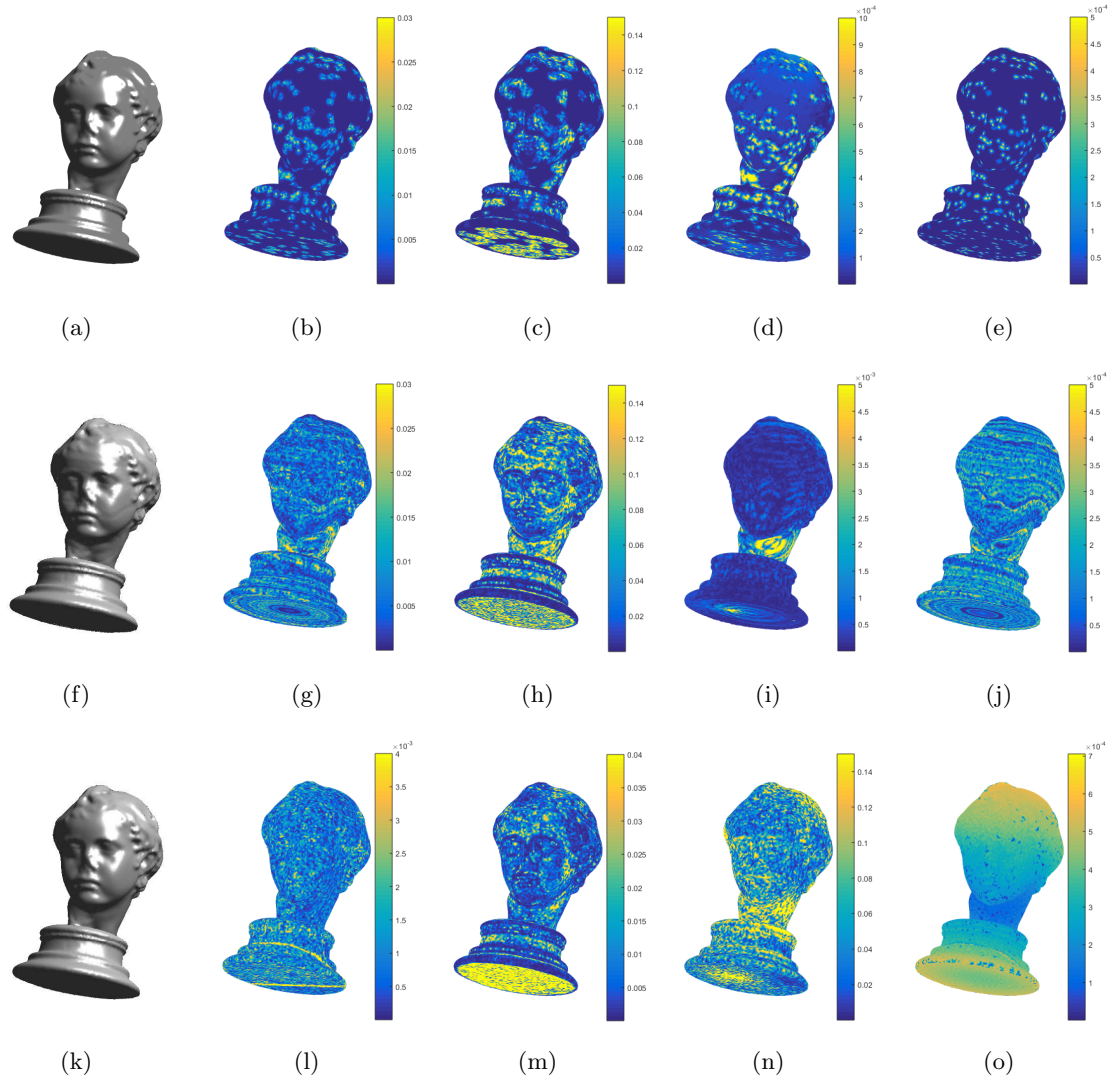
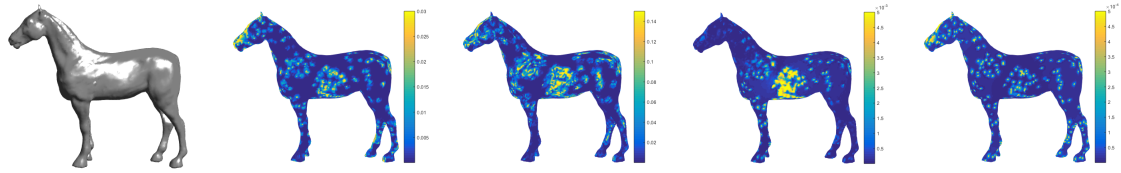


Figure 3.7: Stego-objects and the visualization of differences in the detection of features used for steganalysis. (a), (f) and (k) are the stego-objects obtained after using the information hiding algorithms, SRW [Yang et al., 2017b], MRS [Cho et al., 2007] and MLS [Chao et al., 2009], respectively; (b), (g) and (l) show the absolute differences of vertex normals ϕ_{11} between those stego-objects and their corresponding cover-object, respectively; (c), (h) and (m) for the curvature ratios ϕ_{13}; (d), (i) and (n) for the azimuth angle ϕ_{14}; (e), (j) and (o) for the radial distance ϕ_{16}.



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

Figure 3.8: Stego-object and the visualization of differences in the detection of features used for steganalysis. (a) The stego-object obtained after using SRW algorithms described in [Yang et al., 2017b]; (b) The absolute differences of vertex normals ϕ_{11} between the stego-objects and their corresponding cover-object; The absolute differences in (c) for the curvature ratios ϕ_{13}; (d) for the azimuth angle ϕ_{14}; (e) for the radial distance ϕ_{16}.

3.4.4 Statistical steganalysis study

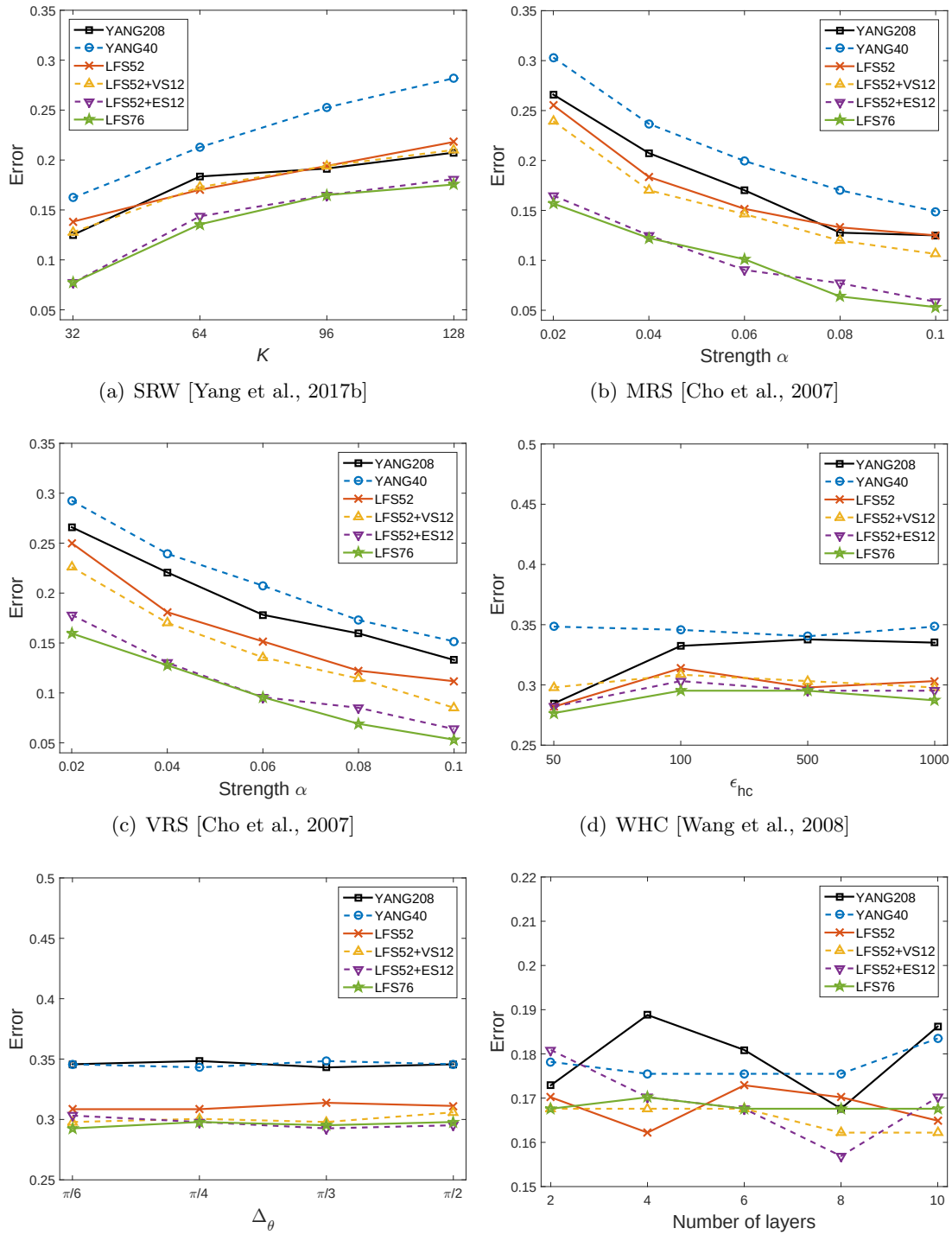
In the following we test the FLD ensemble and QDA steganalyzers on 3D objects with information embedded by six different information hiding algorithms. Figure 3.9 shows the detection errors for the six information hiding methods, SRW [Yang et al., 2017b], MRS [Cho et al., 2007], VRS [Cho et al., 2007], WHC [Wang et al., 2008], WFR [Wang et al., 2008] and MLS [Chao et al., 2009], using the FLD ensemble classifier, trained with the six combinations of feature sets, formed as mentioned above. It can be seen from Figure 3.9 that the LFS76 shows best performance among the six combinations for most of the cases. The improvement of the efficiency of LFS76, compared to YANG208, is quite evident for all the six embedding algorithms. The advantage of LFS76 over LFS52 is more obvious when detecting the watermarks embedded by SRW, MRS and VRS methods, than when detecting those embedded by WHC, WFR and MLS methods. This is because WHC, WFR and MLS methods do not produce changes in the spherical coordinate system, and consequently the feature sets VS12 and ES12 are less useful. When considering the feature sets VS12 and ES12, it appears that the combination of LFS52 and ES12 achieves better performance than that of LFS52 and VS12, considered individually, which means that the ES12 features are more efficient than VS12.

We can observe from Figure 3.9(a) that as the value of K increases, the detection error for SRW [Yang et al., 2017b] tends to increase as well. This happens because a larger K will lead to a smaller range size of each bin of the radial coordinate histogram. Consequently, if some vertices need to be changed in order to embed the information, the displacement of each

vertex will be smaller due to the smaller range size of each bin. So even when the embedding capacity of the SRW is higher when K is larger, the distortion of the 3D surface caused by the embedding is less significant, resulting in a higher detection error of the steganalyzer. From Figures 3.9(b) and (c) we can observe that as the watermarking strength of MRS and VRS [Cho et al., 2007] increases, all steganalyzers provide better detection accuracy. This is due to the fact that more significant changes are produced in the 3D object surface, which is caused by watermarks that have stronger embedding parameters. For WHC [Wang et al., 2008] method, the detection error shown in Figure 3.9(d) increases slightly when ϵ_{hc} ranges from 50 to 100, but remains stable afterwards. In Figure 3.9(e), the detection error for WFR [Wang et al., 2008] method does not have obvious changes when the parameter Δ_θ varies, which indicates that the parameter Δ_θ does not influence significantly the embedding distortion of the object. It can be observed from Figure 3.9(f) that the detection error for MLS [Chao et al., 2009] does not decline when the embedding capacity increases. The reason for this is that, according to the multi-layer embedding framework applied in [Chao et al., 2009], the distortions produced to the objects are well controlled during the embedding.

Figure 3.10 shows the detection errors for the six information hiding methods, using the QDA classifier, trained with the various feature sets. The trends of the detection errors for the six embedding algorithms depicted in Figure 3.10 are similar to those shown in Figure 3.9, but the performance of the QDA classifier is not as good as the FLD ensemble classifier in general.

In the following, we discuss the detectability of the watermarks embedded by the six information hiding algorithms by the steganalyzers trained with the LFS76 feature set. When $K = 128$, the payload of SRW is close to 64 bits, which is the payload of MRS and VRS as well in our experiments. From Figures 3.9(a)-(c) we can see that SRW has a lower detectability than those of MRS and VRS, which was reported in [Yang et al., 2017b] as well. Since the procedure and embedding domains of MRS and VRS are very similar, it is not surprising that their watermark detectability is close to each other. WHC and WFR also have quite similar watermark detectability ratios, but which are lower than those of the other four information hiding algorithms considered during the experiments. This is because LFS76 does not include the features from the wavelet domain, in which modifications of 3D objects



(a) SRW [Yang et al., 2017b]

(b) MRS [Cho et al., 2007]

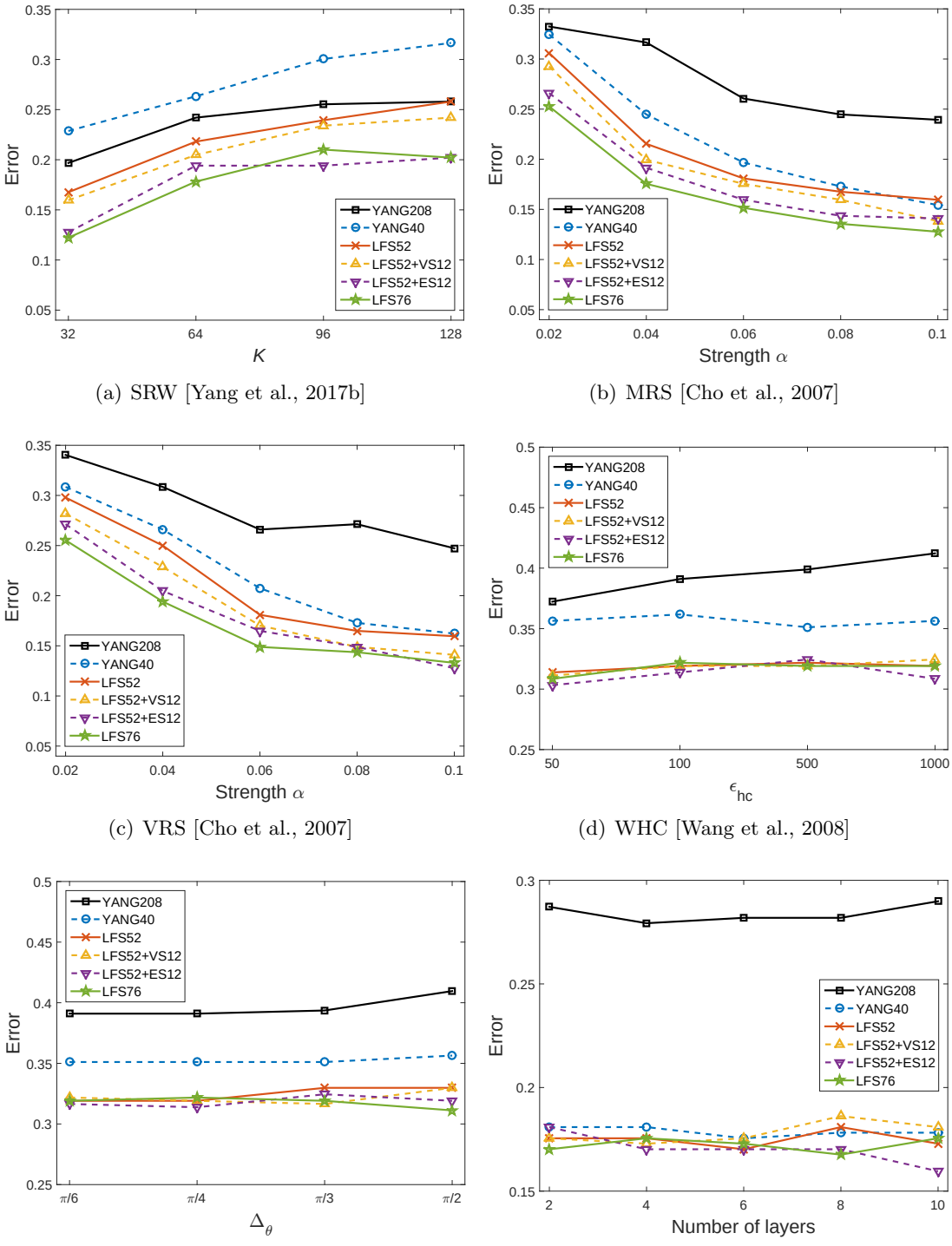
(c) VRS [Cho et al., 2007]

(d) WHC [Wang et al., 2008]

(e) WFR [Wang et al., 2008]

(f) MLS [Chao et al., 2009]

Figure 3.9: Median value of detection errors of the steganalyzers trained as FLD ensemble classifiers on the testing set over 30 independent splits for the six information hiding methods with different values for the embedding parameters.



(a) SRW [Yang et al., 2017b]

(b) MRS [Cho et al., 2007]

(c) VRS [Cho et al., 2007]

(d) WHC [Wang et al., 2008]

(e) WFR [Wang et al., 2008]

(f) MLS [Chao et al., 2009]

Figure 3.10: Median value of detection errors of the steganalyzers trained as QDA classifiers on the testing set over 30 independent splits for the six information hiding methods with different values for the embedding parameters.

are made by WHC and WFR. The detectability of changes embedded by MLS algorithm is moderate among the six information hiding methods, according to the Figure 3.9, but the payload of MLS can be much higher than the other methods, equal to approximately 10 times the number of vertices in the object. If the payload of MLS information hiding algorithm is decreased to 64 bits per object, the stego-objects embedded by MLS would be much harder to detect.

It can be observed that the issue of robustness against malicious attacks aiming to remove the watermark is not taken into account here. However, the developers of steganographic schemes ideally would like that their scheme to be robust not only against steganalytic detection, but perhaps equally importantly against malicious attacks for watermark removal. The study of improving the robustness of the steganography is important.

3.4.5 Analysing the efficiency of features for steganalysis

In order to investigate the contribution of different categories of features from the set LFS76 to the steganalysis, we use the relevance between the feature vectors and the class label in order to assess each feature's efficiency. The measurement of the relevance is addressed by using the Pearson correlation coefficient,

$$\rho(x_i, y) = \frac{cov(x_i, y)}{\sigma_{x_i}\sigma_y} \tag{3.19}$$

where x_i is the ith feature of a given feature set, $\mathbf{X} = \{x_i | i = 1, 2, \ldots, N\}$, and N is the dimensionality of the input feature, y is the class label indicating whether the class corresponds to a cover-object or a stego-object, cov represents the covariance and σ_{x_i} is the standard deviation of x_i. The Pearson correlation coefficient is well known as a measure of the linear dependence between two variables [Hall, 1999]. Then we set $|\rho(x_i, y)|$ for assessing the relevance, where $|\rho(x_i, y)| = 1$ indicates a highly linear relationship between the feature and the class label, corresponding to a better discriminant ability of that feature.

The analysis is conducted on the features extracted from the 354 cover-objects used above and for the six sets of corresponding stego-objects which are produced by the six watermarking and steganographic algorithms, SRW, MRS, VRS, WHC, WFR and MLS,

respectively. We set the parameter K in SRW algorithm from [Yang et al., 2017b] as 128. For the two watermarking methods, MRS and VRS, from [Cho et al., 2007], in order to find a balance between the watermarking strength and its detectability, we set the watermarking strength as 0.04 and embed a payload of 64 bits. For WHC and WFR methods, we set the parameters as $\epsilon_{hc} = 100$ and $\Delta_\theta = \pi/3$, which are the same as the settings in [Wang et al., 2008]. In the case when using MLS method from [Chao et al., 2009], we consider ten layers of embedding.

We split the features from the set LFS76 into 10 categories according to their representations of the local shape geometry: 1, the vertex coordinates in the Cartesian coordinate system (ϕ_1, ϕ_2 and ϕ_3); 2, the vertex norm in the Cartesian coordinate system (ϕ_7); 3, the vertex coordinates in the Laplacian coordinate system (ϕ_4, ϕ_5 and ϕ_6); 4, the vertex norm in Laplacian coordinate system (ϕ_8); 5, the face normal (ϕ_{10}); 6, the dihedral angle (ϕ_9); 7, the vertex normal (ϕ_{11}); 8, the curvature (ϕ_{12} and ϕ_{13}); 9, the vertex coordinates in the spherical coordinate system (ϕ_{14}, ϕ_{15} and ϕ_{16}); 10, the edge length in the spherical coordinate system (ϕ_{17}, ϕ_{18} and ϕ_{19}).

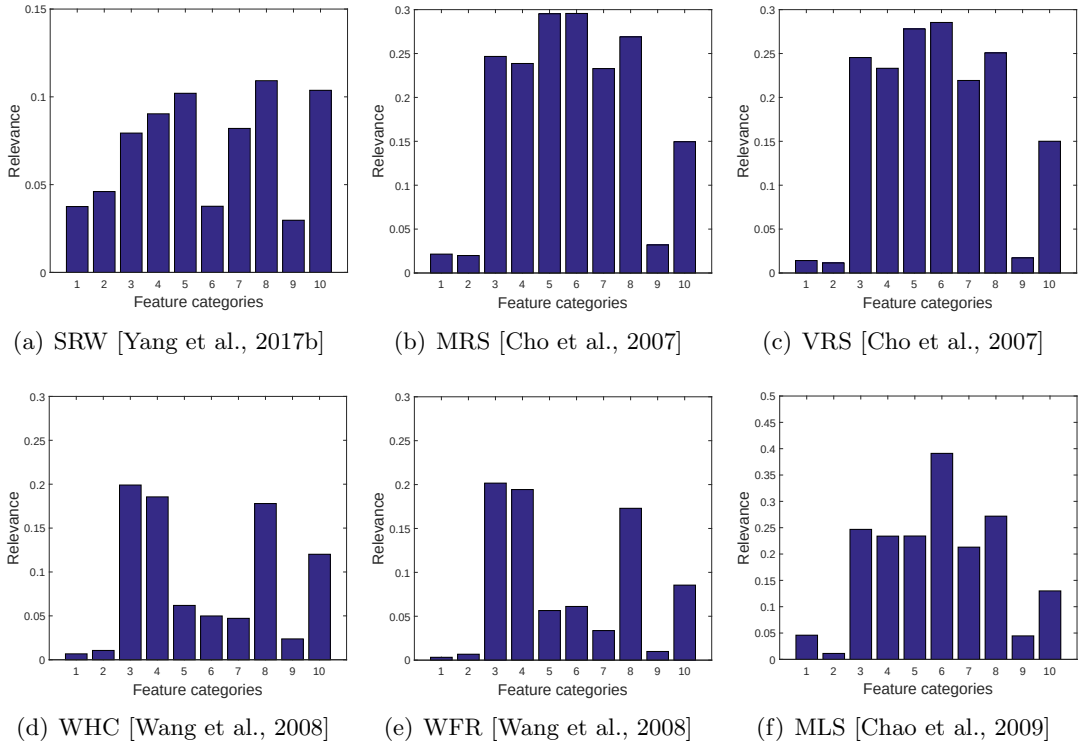
The relevance for all features from LFS76, is calculated according to equation (5.1), and the averaged relevances of the features in each category are shown in Figure 3.11. From Figure 3.11 we can observe that the new proposed features, represented by labels 7, 8 and 10, have relatively high relevance to the class label. More specifically, in Figure 3.11(a), the features characterizing the local curvature (label 8) achieve the highest relevance. The relevance of the proposed ES12 feature, represented by label 10, is higher than that of the proposed VS12 represented by label 9. This implies that the efficiency of ES12 is higher than that of VS12, which is also reflected in the results shown in the Figures 3.9 and 3.10. Comparing the formulation of VS12 and ES12, it is noted that two adjacent vertices are taken into account when extracting the ES12 features. However, the vertices in the object are considered individually in the case of VS12. So ES12 is better able of capturing the distortion in the local region caused by the embedding modifications than VS12. However, the VS12 probably detects hidden information in the 3D object, which ES12 cannot identify, resulting in better performance of LFS76 than the combination of LFS52 and ES12 as shown in Figures 3.9 and 3.10. Meanwhile, in Figure 3.11, the vertex normal feature (label 7) and

the face normal feature (label 5) have a similar level of relevance, which is because the vertex normal is dependent on face normal in equation (3.9).

It is interesting that the relevance of the dihedral angle feature (label 6) shows a high relevance to the class label in the cases of MRS, VRS and MLS, but shows much lower relevance when the stego-objects are generated by SRW, WHC and WFR methods. This happens because almost all vertices from a mesh are only slightly changed by MRS, VRS and MLS methods, while such changes are scattered among the vertices in the case of SRW method from [Yang et al., 2017b], as it can be observed from Figures 3.7(b) (g) and (l). Similarly, when considering WHC and WFR, the modifications are made after only one wavelet decomposition, so only half of the vertices are likely to be modified, preserving the dihedral angles to some extent. It is noticed that the features representing the vertex coordinates and the norm in the Laplacian coordinate system (labels 3 and 4) have much higher relevance than those representing the vertex coordinates and the norm in the Cartesian coordinate system (labels 1 and 2). This is because, according to [Yang and Ivrissimtzis, 2010], the Laplacian coordinates of a vertex are calculated from the position of the vertex and its adjacent vertices, which capture the geometrical information of a larger region than the Cartesian coordinates for each vertex.

In the following, we increase gradually the feature set used for training the steganalyzer, from YANG40 to LFS52, by adding either VS12 or ES12 to LFS52, and eventually to the LFS76 feature set, and then compare with YANG208 feature set. YANG40 includes the features represented by labels 1-6 in Figure 3.11. Features represented by labels 1-8 form LFS52 , while labels 1-10 correspond to LFS76. VS12 and ES12 are represented by labels 9 and 10, respectively. We employed the FLD ensemble and QDA as the machine learning algorithms to train the steganalyzers when the information was hidden into 3D objects by the six information hiding methods mentioned above. The performance of the feature sets are evaluated by the area under the ROC curves of the corresponding steganalyzers. A larger area under the ROC curve means that the classifier has a better detection accuracy.

Tables 3.2 and 3.3 provide the median values of the area under the ROC curves for the steganalytic methods when using six combinations of feature sets for 30 independent splits of the training/testing set. It can be seen from Table 3.2 that the areas under the ROC



(a) SRW [Yang et al., 2017b] (b) MRS [Cho et al., 2007] (c) VRS [Cho et al., 2007]

(d) WHC [Wang et al., 2008] (e) WFR [Wang et al., 2008] (f) MLS [Chao et al., 2009]

Figure 3.11: The relevance between the features and the class label, for cover-objects (0) or stego-objects (1), where the stego-objects are generated by the six information hiding methods, SRW, MRS, VRS, WHC, WFR and MLS, respectively. The meaning of the category labels are: 1, the vertex coordinates in Cartesian coordinate system; 2, the vertex norm in Cartesian coordinate system; 3, the vertex coordinates in Laplacian coordinate system; 4, the vertex norm in Laplacian coordinate system; 5, the face normal; 6, the dihedral angle; 7, the vertex normal; 8; the curvature; 9, the vertex coordinates in spherical coordinates system; 10, the edge length in spherical coordinate system.

Table 3.2: Median values of the area under the ROC curves for the steganalysis results of the six information hiding algorithms when using the FLD ensemble classifier. The best results are shown in bold.

Feature sets	Information hiding methods					
	SRW	MRS	VRS	WHC	WFR	MLS
YANG208	0.8781	0.8745	0.8748	0.7403	0.7320	0.9138
YANG40	0.7782	0.8167	0.8076	0.7048	0.7233	0.9207
LFS52	0.8621	0.8857	0.8902	0.7633	0.7732	0.9254
LFS52+VS12	0.8617	0.8999	0.9037	0.7676	0.7845	**0.9297**
LFS52+ES12	**0.9064**	0.9496	0.9414	0.7770	0.7905	0.9184
LFS76	0.9032	**0.9544**	**0.9466**	**0.7858**	**0.7963**	0.9269

Table 3.3: Median values of the area under the ROC curves for the steganalysis results of the six information hiding algorithms when using the QDA classifier. The best results are shown in bold.

Feature sets	Information hiding methods					
	SRW	MRS	VRS	WHC	WFR	MLS
YANG208	0.8035	0.7871	0.7888	0.6651	0.6730	0.8482
YANG40	0.7485	0.8573	0.8395	0.7118	0.7285	0.8834
LFS52	0.8228	0.8770	0.8518	0.7600	0.7394	0.8697
LFS52+VS12	0.8351	0.8886	0.8721	0.7467	0.7530	0.8697
LFS52+ES12	0.8702	0.8797	0.8829	0.7551	**0.7594**	**0.8909**
LFS76	**0.8871**	**0.8907**	**0.8895**	**0.7610**	0.7590	0.8692

curves of the steganalyzers increase with the addition of new features, such as the vertex normal and the curvature features, to the YANG40 feature set. After adding VS12 and ES12 to LFS52 feature set, the LFS76 feature set achieves the best performance in most cases. For SRW and MLS, the combination of LFS52 and ES12 and that of LFS52 and VS12 give the best performance, respectively, which also justifies the importance of VS12 and ES12 features. But in general, the combination of LFS52 and ES12 has a better performance than that of LFS52 and VS12, indicating the higher efficiency of ES12 when compared to that of VS12, consistently reflected in the results provided in Figure 3.11. The upward trend in the area under the ROC curves along with the addition of new features can be identified in the results provided in Table 3.3 as well. According to these results, the FLD ensemble classifier provides better results than the QDA, used in [Yang and Ivrissimtzis, 2014].

3.5 Conclusion

In this chapter, we propose to use the statistics of a new set of shape features as inputs for 3D steganalyzers. The features proposed in this chapter are used in combination with some of the features proposed in [Yang and Ivrissimtzis, 2014]. We analyse various combinations of local features used for 3D steganalysis by evaluating their relevance to the class label and by testing their performance in 3D steganalysis. The first four statistical moments of 3D feature sets are used for training steganalyzers by two machine learning methods, namely, the Quadratic Discriminant Analysis (QDA) and the Fisher Linear Discriminant (FLD)

ensemble. Afterwards, the steganalyzers based on the parameters learned during the training, are used for differentiating the stego-objects from the cover-objects. The experimental results show that the proposed 3D feature set provides the best results for the steganalysis when the information is hidden in 3D objects by six different 3D information hiding algorithms. The detection errors for the 3D wavelet-based information embedding algorithms, such as WHC and WFR, are higher than those for the other embedding algorithms. In the next chapter we use the multi-scale analysis of 3D wavelets for estimating a new set of 3D features, in order to improve the 3D steganalysis results.

Chapter 4

3D Wavelet Multiresolution Analysis-Based Features for Steganalysis

4.1 Introduction

While the experimental results from Chapter 3 have improved the 3D steganalysis results in detecting the embedding changes made by various steganographic methods, those embedded by some other methods are hard to identify. The changes embedded by 3D wavelet-based steganographic methods are not well detected by the existing 3D steganalysis features, discussed in Chapter 3. In this chapter we use 3D Wavelet Multiresolution Analysis (WMA) in order to propose a new set of features for steganalysis.

Inspired by the way how the information is embedded into the 3D wavelet domain of meshes, we propose a 3D steganalytic approach based on the 3D wavelet multiresolution analysis. 3D wavelet analysis provides a decomposition of the given mesh surface into a mesh of lower resolution and a set of 3D Wavelet Coefficient Vectors (WCVs). Moreover, a mesh subdivision is considered to the given mesh, producing a mesh of higher resolution and a set of 3D WCVs. The features described in this chapter are derived from the relationships between consecutive resolution representations of the mesh. Such features depend on the

geometrical properties of the initial mesh, those of the lower resolution mesh and the higher resolution mesh, as well as of the WCVs relating the three meshes.

The rest of this chapter is organized in the following way. Section 4.2 presents the details of how to extract the 3D wavelet feature set based on multiresolution meshes. Section 4.3 provides the experimental results of the proposed feature set, as well as comparisons of the efficiency of the proposed feature set with other steganalytic features. The summary of this chapter is given in Section 4.4.

4.2 Multiresolution analysis of meshes using 3D wavelets

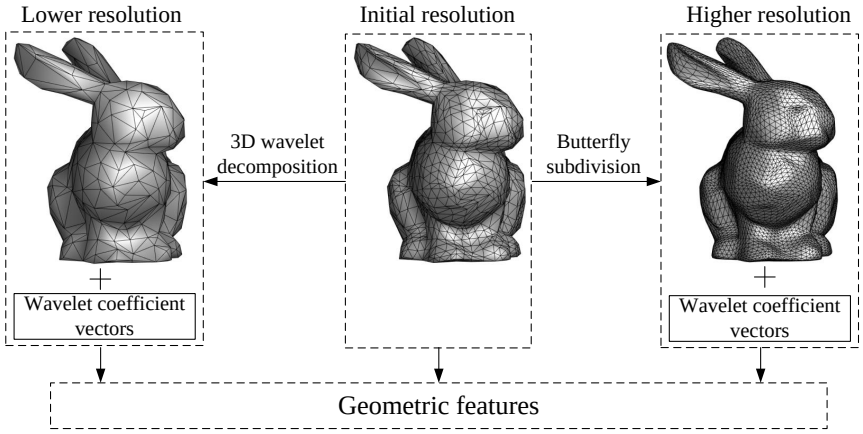


Figure 4.1: Generating the multiresolution meshes using 3D wavelet decomposition and Butterfly subdivision.

In this section we provide a short outline of 3D wavelet analysis methodology and how this can be applied for extracting features useful for steganalysis. In the area of 3D signal processing, the multiresolution analysis of 3D meshes usually works in two ways: (i) the wavelet coefficient vectors are also encoded as part of the mesh representation and thus, from the coarse mesh one can retrieve the exact original. (ii) the wavelet coefficient vectors are not encoded as part of the mesh. Our study considers the case (ii), which means that we assume that we do not have the knowledge of the wavelet coefficient vector as part of the given mesh at the start of the multiresolution analysis.

Figure 4.1 illustrates how the original object $\mathcal{O}$ is decomposed into a lower resolution mesh $\mathcal{O}^l$, shown to left, using the 3D lazy wavelet decomposition [Lounsbery et al., 1997], and a

set of Wavelet Coefficient Vectors (WCVs). The same mesh is subdivided, as shown in the right side of Figure 4.1, into a higher resolution mesh $\mathcal{O}^h$ and the WCVs using the Butterfly scheme [Dyn et al., 1990]. Geometric features are generated using the initial resolution mesh, the lower resolution mesh, the higher resolution mesh, and the corresponding WCVs, resulting from the processes of downscaling or upscaling the 3D object through the 3D wavelet mesh analysis. The same processing steps, such as decomposition, subdivision and calculation of the geometric features, are applied to the smoothed mesh $\mathcal{O}'$ produced during the preprocessing stage. Then, the differences between the geometric features extracted from the original mesh $\mathcal{O}$ and the smoothed mesh $\mathcal{O}'$ are represented by the vector $\mathbf{\Phi}$. Finally, the first four statistical moments of the logarithm of $\mathbf{\Phi}$ are used as a feature vector for training the 3D steganalyzers in order to detect the 3D stego-meshes. In the following we explain in more detail how to extract features for steganalysis using 3D wavelet analysis.

4.2.1 Extracting the features using edge flipping

The first two geometric features, extracted from the initial mesh, are the edge vector and flipped edge vector. The edge vectors $\{\mathbf{e}_{(i,j)}\}$ represent the vectors from the vertex v_i to the vertex v_j, where v_i and v_j are adjacent in the initial resolution mesh. An example of edge vector is illustrated in Figure 4.2 as $\mathbf{e}_{(2,3)}$. Each flipped edge vector $\{\mathbf{e}^*_{(i,j)}\}$ is connecting the opposite vertices, from the triangles which are sharing the associated edge vector $\{\mathbf{e}_{(i,j)}\}$. For example, the vector $\mathbf{e}^*_{(2,3)}$ from Figure 4.2, connecting two vertices v_1 and v_4, is the flipped edge vector of the edge vector $\mathbf{e}_{(2,3)}$. The direction of the flipped edge vector is from the vertex with a lower index to the one with a higher index as shown in Figure 4.2.

4.2.2 Geometric features extracted from the lower resolution mesh

The lower resolution mesh is obtained after one iteration of the 3D lazy wavelet decomposition, which is illustrated in Figure 4.3. In this figure, four triangles $\triangle v_4v_5v_6, \triangle v_4v_6v_8, \triangle v_6v_7v_8$ and $\triangle v_4v_8v_9$ from the initial resolution mesh are merged into a single larger triangle $\triangle v_5v_7v_9$ as a part of a coarser mesh. The vertices, v_4, v_6 and v_8 in Figure 4.3, are removed in the process of downscaling the mesh, and they correspond to the terminal points for the three WCVs, $\mathbf{w}^l_{(5,9)}$, $\mathbf{w}^l_{(5,7)}$ and $\mathbf{w}^l_{(7,9)}$. The subscripts of $\mathbf{w}^l_{(i,j)}$ represent the two vertices v_i and v_j

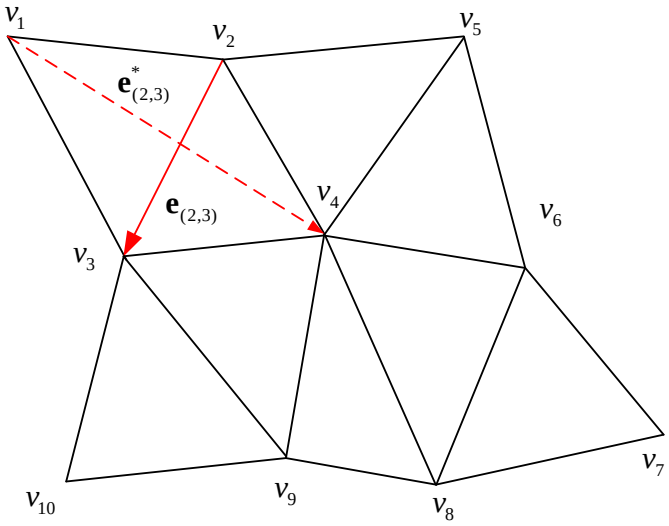


Figure 4.2: Extracting the edge vectors and their flipped counterparts from the mesh of initial resolution.

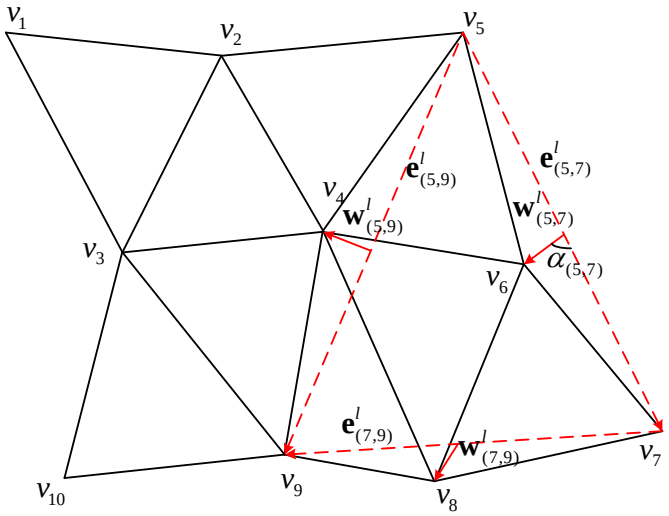


Figure 4.3: The illustration of the 3D wavelet decomposition for a mesh from its initial resolution to a lower resolution.

of the WCV's support edge $\mathbf{e}^l_{(i,j)}$ from the lower resolution mesh. Meanwhile, the initial point of the WCV, $\mathbf{w}^l_{(5,9)}$, is the midpoint of its support edge $\mathbf{e}^l_{(5,9)}$ in the lower resolution mesh. So each edge vector in the lower resolution mesh is associated with one WCV. The WCV, $\mathbf{w}^l_{(i,j)}$, and the edge vector, $\mathbf{e}^l_{(i,j)}$, in the lower resolution mesh are considered as components of the proposed 3D wavelet feature vector used for steganalysis.

Two other features are considered from the lower resolution mesh. One is the angle between the WCV and its support edge vector in the lower resolution mesh, defined as

$$\alpha_{(i,j)} = \arccos \frac{\mathbf{w}^l_{(i,j)} \cdot \mathbf{e}^l_{(i,j)}}{\left\|\mathbf{w}^l_{(i,j)}\right\| \cdot \left\|\mathbf{e}^l_{(i,j)}\right\|}, \tag{4.1}$$

where i and j are the indexes of two adjacent vertices in the lower resolution mesh. For example, as illustrated in Figure 4.3, $\alpha_{(5,7)}$ is the angle between the WCV, $\mathbf{w}^l_{(5,7)}$, and its support edge vector, $\mathbf{e}^l_{(5,7)}$. The other one is the ratio between the Euclidean norm of the WCV and that of its support edge vector in the lower resolution mesh, defined as

$$\rho^l_{(i,j)} = \frac{\left\|\mathbf{w}^l_{(i,j)}\right\|}{\left\|\mathbf{e}^l_{(i,j)}\right\|}. \tag{4.2}$$

Since these two geometric features are used to carry information payloads by various 3D wavelet watermarking methods such as those proposed in [Wang et al., 2008, Kanai et al., 1998], it is straight forward to use them as geometric features for steganalysis.

One important issue about the 3D wavelet decomposition is that the steganalyst lacks the knowledge of how the steganographier groups the triangle faces when embedding the information into the 3D shape. In fact, the grouping of the triangle faces determines the generation of the WCVs. For instance, in Figure 4.3, if the four triangles $\triangle v_2v_3v_4, \triangle v_3v_4v_9, \triangle v_4v_8v_9$ and $\triangle v_3v_9v_{10}$ would merge into a larger triangle $\triangle v_2v_8v_{10}$ in the lower resolution mesh, we would actually not obtain WCVs such as $\mathbf{w}^l_{(5,9)}$ and $\mathbf{w}^l_{(7,9)}$. By not knowing this information, we can have a mismatch of the 3D wavelet decompositions which would degrade the performance of the steganalysis. In order to avoid this problem in 3D wavelet decomposition, we apply all the possible grouping options for the triangle faces in the given neighbourhood, generating all the possible WCVs together with their support edge vectors in the lower res-

olution meshes, as well as the other geometric features. When calculating the geometric features from the lower resolution meshes, we find all the groups of four neighboring triangles, including one triangle in the centre, surrounded by the other three triangles, that can be merged into a larger triangle during the wavelet decomposition. Since each triangle from a fully connected mesh can be the central triangle used in 3D wavelet decomposition, we can obtain $|F|$ different groups of four neighboring triangles, where $|F|$ is the number of the triangles from the initial resolution mesh. For each of these groups, we apply the wavelet decomposition and calculate the geometric features presented above in this section. Meanwhile, we remove the duplicated geometric features, such as WCVs and edge vectors. Finally, the obtained features, considering all the possible grouping options in the wavelet decomposition, form the geometric features from the lower resolution mesh.

4.2.3 Geometric features extracted from the higher resolution mesh

When transforming the given mesh into a higher resolution mesh, each triangle from the initial resolution mesh is subdivided into four smaller triangles by inserting three vertices, each corresponding to one of the edges of the initial resolution triangle. In the higher resolution mesh, each newly inserted vertex is adjacent to the two ends of the support edge of the initial resolution triangle, and it is also adjacent to the other newly inserted vertices. As illustrated in Figure 4.4, the vertices, $v_{11}, v_{12}, v_{13}, v_{14}$ and v_{15} are added to the local mesh in order to produce the higher resolution mesh following the 3D wavelet transformation. Since the subdivision is based on the Butterfly scheme [Dyn et al., 1990], the position of the newly added vertex is computed from eight vertices which define a neighbourhood resembling the shape of a butterfly. For example, the position of the vertex v_{13} associated to edge vector $\mathbf{e}_{(3,4)}$ in Figure 4.4 is given by

$$v_{13} = \frac{1}{2}(v_3 + v_4) + \frac{1}{4}(v_2 + v_9) - \frac{1}{8}(v_1 + v_5 + v_8 + v_{10}). \tag{4.3}$$

The WCV from the higher resolution mesh, denoted as $\mathbf{w}^h_{(i,j)}$, is the vector from the midpoint of the support edge $\mathbf{e}_{(i,j)}$ in the initial resolution mesh to the newly added vertex. For example, as shown in Figure 4.4, the WCVs, $\mathbf{w}^h_{(2,3)}$, $\mathbf{w}^h_{(2,4)}$ and $\mathbf{w}^h_{(3,4)}$, are associated to

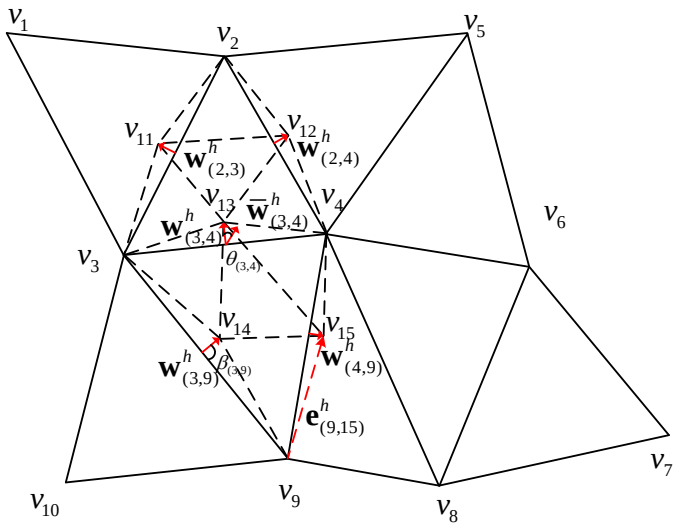


Figure 4.4: The illustration of the 3D wavelet subdivision for the mesh from the initial resolution to a higher resolution.

the support edge vectors, $\mathbf{e}_{(2,3)}$, $\mathbf{e}_{(2,4)}$ and $\mathbf{e}_{(3,4)}$, from the initial resolution mesh. We use the WCVs from the higher resolution mesh as one of the geometric features for 3D steganalysis.

Consistent with the rules of extracting geometric features from the initial and lower resolution meshes, the edge vector in the higher resolution mesh, $\mathbf{e}^h_{(i,j)}$, is also considered as a geometric feature. The edge vector, $\mathbf{e}^h_{(9,15)}$, illustrated in Figure 4.4, is an example of the edge vector in the higher resolution mesh.

Two geometric features, $\beta_{(i,j)}$ and $\rho^h_{(i,j)}$, which are similar to $\alpha_{(i,j)}$ and $\rho^l_{(i,j)}$ from the lower resolution mesh, are obtained from the higher resolution mesh. $\beta_{(i,j)}$ is the angle between the WCV $\mathbf{w}^h_{(i,j)}$, and its support edge vector, $\mathbf{e}_{(i,j)}$, in the initial resolution mesh, which is calculated in a similar way by equation (4.1). For instance, $\beta_{(3,9)}$ shown in Figure 4.4 is the angle between $\mathbf{w}^h_{(3,9)}$ and $\mathbf{e}_{(3,9)}$. $\rho^h_{(i,j)}$ represents the ratio between the Euclidean norm of the WCV, $\mathbf{w}^h_{(i,j)}$, and that of its support edge vector, $\mathbf{e}_{(i,j)}$, from the initial resolution mesh.

In order to capture the relationship between each WCV and its neighboring WCVs, we consider the average of the neighbouring WCVs for each given WCV from the higher resolution mesh. We define that two WCVs are neighbors only when their terminal points are adjacent in the higher resolution mesh. The set of the neighboring WCVs of WCV, $\mathbf{w}^h_{(i,j)}$, is denoted as $\mathcal{N}(\mathbf{w}^h_{(i,j)})$. Then the average neighboring WCVs of $\mathbf{w}^h_{(i,j)}$ is calculated

as,

$$\bar{\mathbf{w}}^h_{(i,j)} = \frac{1}{|\mathcal{N}(\mathbf{w}^h_{(i,j)})|} \sum_{\mathbf{w}^h_{(k,l)} \in \mathcal{N}(\mathbf{w}^h_{(i,j)})} \mathbf{w}^h_{(k,l)}. \tag{4.4}$$

In Figure 4.4, $\bar{\mathbf{w}}^h_{(3,4)}$ is the averaged neighboring WCV for $\mathbf{w}^h_{(3,4)}$. The vector of the difference between the WCV and its averaged neighboring WCV is considered as a geometric feature,

$$\mathbf{w}^h_{(i,j)}{}' = \mathbf{w}^h_{(i,j)} - \bar{\mathbf{w}}^h_{(i,j)}. \tag{4.5}$$

Meanwhile, another geometric feature is the angle between the WCV and its averaged neighboring WCV,

$$\theta_{(i,j)} = \arccos \frac{\mathbf{w}^h_{(i,j)} \cdot \bar{\mathbf{w}}^h_{(i,j)}}{\left\|\mathbf{w}^h_{(i,j)}\right\| \cdot \left\|\bar{\mathbf{w}}^h_{(i,j)}\right\|}. \tag{4.6}$$

For each WCV in the higher resolution mesh, we consider the mean and variance of the angles between the WCV and its neighboring WCVs as geometric features, which are given by

$$\mu^M_{(i,j)} = \frac{1}{|\mathcal{N}(\mathbf{w}^h_{(i,j)})|} \sum_{\mathbf{w}^h_{(k,l)} \in \mathcal{N}(\mathbf{w}^h_{(i,j)})} \delta_{(k,l)}, \tag{4.7}$$

$$\mu^V_{(i,j)} = \frac{1}{|\mathcal{N}(\mathbf{w}^h_{(i,j)})|} \sum_{\mathbf{w}^h_{(k,l)} \in \mathcal{N}(\mathbf{w}^h_{(i,j)})} \left(\delta_{(k,l)} - \mu^M_{(i,j)}\right)^2. \tag{4.8}$$

where $\delta_{k,l}$ is the angle between $\mathbf{w}^h_{(i,j)}$ and its neighboring WCV, $\mathbf{w}^h_{(k,l)}$,

$$\delta_{(k,l)} = \arccos \frac{\mathbf{w}^h_{(i,j)} \cdot \mathbf{w}^h_{(k,l)}}{\left\|\mathbf{w}^h_{(i,j)}\right\| \cdot \left\|\mathbf{w}^h_{(k,l)}\right\|} \tag{4.9}$$

With respect to the WCV and its neighboring WCVs, we also consider the mean and variance of the absolute differences between the Euclidean norms of WCV and its neighboring

WCVs, namely,

$$\kappa^M_{(i,j)} = \frac{1}{|\mathcal{N}(\mathbf{w}^h_{(i,j)})|} \sum_{\mathbf{w}^h_{(k,l)} \in \mathcal{N}(\mathbf{w}^h_{(i,j)})} \left| \|\mathbf{w}^h_{(i,j)}\| - \|\mathbf{w}^h_{(k,l)}\| \right|, \tag{4.10}$$

$$\kappa^V_{(i,j)} = \frac{1}{|\mathcal{N}(\mathbf{w}^h_{(i,j)})|} \sum_{\mathbf{w}^h_{(k,l)} \in \mathcal{N}(\mathbf{w}^h_{(i,j)})} \left(\|\mathbf{w}^h_{(k,l)}\| - \kappa^M_{(i,j)} \right)^2. \tag{4.11}$$

4.2.4 From the geometric features to the feature vectors

Table 4.1: A list of the proposed geometric features based on the 3D wavelet multiresolution analysis.

	Notation	Geometrical representation	Type	Resolution
1	$\mathbf{e}_{(i,j)}$	Edge vector	Vector	Initial
2	$\mathbf{e}^*_{(i,j)}$	Flipped edge vector	Vector	Initial
3	$\mathbf{w}^l_{(i,j)}$	WCV	Vector	Lower
4	$\mathbf{e}^l_{(i,j)}$	Edge vector	Vector	Lower
5	$\alpha_{(i,j)}$	Angle between $\mathbf{w}^l_{(i,j)}$ and $\mathbf{e}^l_{(i,j)}$	Scalar	Lower
6	$\rho^l_{(i,j)}$	Ratio between $\|\mathbf{w}^l_{(i,j)}\|$ and $\|\mathbf{e}^l_{(i,j)}\|$	Scalar	Lower
7	$\mathbf{w}^h_{(i,j)}$	WCV	Vector	Higher
8	$\mathbf{e}^h_{(i,j)}$	Edge vector	Vector	Higher
9	$\beta_{(i,j)}$	Angle between $\mathbf{w}^h_{(i,j)}$ and $\mathbf{e}_{(i,j)}$	Scalar	Higher
10	$\rho^h_{(i,j)}$	Ratio between $\|\mathbf{w}^h_{(i,j)}\|$ and $\|\mathbf{e}_{(i,j)}\|$	Scalar	Higher
11	$\bar{\mathbf{w}}^h_{(i,j)}$	Averaged neighboring WCV	Vector	Higher
12	$\mathbf{w}^h_{(i,j)}{}'$	Difference between $\mathbf{w}^h_{(i,j)}$ and $\bar{\mathbf{w}}^h_{(i,j)}$	Vector	Higher
13	$\theta_{(i,j)}$	Angle between $\mathbf{w}^h_{(i,j)}$ and $\bar{\mathbf{w}}^h_{(i,j)}$	Scalar	Higher
14	$\mu^M_{(i,j)}$	Mean of the angles between WCV and its neighboring WCVs	Scalar	Higher
15	$\mu^V_{(i,j)}$	Variance of the angles between WCV and its neighboring WCVs	Scalar	Higher
16	$\kappa^M_{(i,j)}$	Mean of the differences between the norms of WCV and its neighboring WCVs	Scalar	Higher
17	$\kappa^V_{(i,j)}$	Variance of the differences between the norms of WCV and its neighboring WCVs	Scalar	Higher

All the proposed features derived based on the 3D wavelet transformation are listed in Table 4.1, where their notations are indicated as well as their geometrical representation and the mesh resolution level used for their calculation. The geometric features of the original

mesh and the smoothed one are extracted simultaneously and subtracted from each others.

For the scalar geometric features, the differences between the geometric feature from the original mesh, denoted as g_t, and that from the smoothed mesh, g'_t, is given by their absolute differences

$$\phi_t = |g_t - g'_t|. \tag{4.12}$$

where t is the index of the geometric feature.

Meanwhile, the difference between the vectorial geometric features from the original mesh, $\mathbf{g}_t$, and from the smoothed mesh, $\mathbf{g}'_t$ are calculated in four different ways. Firstly, the absolute differences are calculated for features defined in the Cartesian coordinate system, such as

$$\begin{aligned} \phi_{t_1} &= |\mathbf{g}_{t,x} - \mathbf{g}'_{t,x}|, \\ \phi_{t_2} &= |\mathbf{g}_{t,y} - \mathbf{g}'_{t,y}|, \\ \phi_{t_3} &= |\mathbf{g}_{t,z} - \mathbf{g}'_{t,z}|, \end{aligned} \tag{4.13}$$

where $\mathbf{g}_{t,x}$ represents the x-component of the vector $\mathbf{g}_t$ in the Cartesian coordinate system. Secondly, the norm of the difference between vectors $\mathbf{g}_t$ and $\mathbf{g}'_t$ is calculated as

$$\phi_{t_4} = \left\|\mathbf{g}_t - \mathbf{g}'_t\right\|, \tag{4.14}$$

and we consider the absolute differences between the norms of the two vectors, namely,

$$\phi_{t_5} = |\,\|\mathbf{g}_t\| - \left\|\mathbf{g}'_t\right\|\,|. \tag{4.15}$$

Finally the angle between the two vectors, $\mathbf{g}_t$ and $\mathbf{g}'_t$ is considered as well,

$$\phi_{t_6} = \arccos\frac{\mathbf{g}_t \cdot \mathbf{g}'_t}{\|\mathbf{g}_t\| \cdot \|\mathbf{g}'_t\|}. \tag{4.16}$$

It can be observed in Table 4.1 that 8 of the proposed geometric features are vectors while the other 9 are scalars. The differences between the geometric features from the original mesh and those of the smoothed one are summarized into a set of $8 \times 6 + 9 = 57$

elements, $\mathbf{\Phi} = \{\phi_t | t = 1, 2, .., 57\}$. Then, the logarithm is used to transform each entry of $\mathbf{\Phi}$ in order to enforce the evenness of the feature distributions and we make up the corresponding empirical probability density functions. Finally, we consider the first four statistical moments, representing the mean, variance, skewness and kurtosis, of $\{\lg(\phi_t) | \phi_t \in \mathbf{\Phi}\}$ as the $57 \times 4 = 228$ dimensional feature vector $\mathbf{X}$. The feature vector $\mathbf{X} = [x_1, x_2, ..., x_{228}]$ and the class label y, corresponding to the mesh $\mathcal{O}$, are used as the inputs to a machine learning algorithm to train the 3D steganalyzer. The proposed 228-dimensional 3D Wavelet Feature Set is labeled as WFS228.

It can be observed from the Table 4.1 that more features are extracted from the higher resolution mesh than those from the lower resolution mesh. This happens because the uncertainty of the grouping of the triangles, during the implementation of the 3D wavelet decomposition, makes it difficult to find the neighboring WCVs of a certain WCV in the lower mesh. Then, the geometric features that would represent the information of the neighboring WCVs, listed as 11-17 in Table 4.1, are not possible to be extracted from the lower resolution mesh. Besides, the features extracted from the higher resolution mesh may have linear dependencies to some extent, because the location of the vertex in the higher resolution mesh is based on a linear combination of the vertices in the Butterfly neighborhood in the original resolution mesh.

4.3 Experimental results

In the following experiments we evaluate the performance of the proposed WFS228 feature set by detecting the information embedded by eight 3D information hiding algorithms, while providing comparisons with the performance provided by four other 3D steganalytic feature sets. The 354 cover-meshes used during the experiments are from the Princeton Mesh Segmentation project [Chen et al., 2009] database, which include various shapes representing human forms, animals, statues, tools and so on.

The proposed feature set WFS228 is extracted from the cover-meshes and the corresponding stego-meshes when embedded with information by various 3D embedding algorithms. The embedded information is a pseudorandom bit stream which simulates the secret

messages or watermarks hidden by the steganographier. During the preprocessing stage, we firstly apply one iteration of Laplacian smoothing on both cover-meshes and stego-meshes, by setting the scale factor as $\lambda = 5$. The 3D steganalytic features are extracted as described in Section 4.2. We consider the proposed feature set WFS228, and compare its results against other existing 3D steganalytic feature sets such as, YANG208 [Yang and Ivrissimtzis, 2014], LFS52 [Li and Bors, 2016], LFS64 [Kim et al., 2017] and LFS76 [Li and Bors, 2017]. We also consider the feature set combining LFS76 and the proposed WFS228. The parameters used for the calculation of YANG208, LFS52, LFS64 and LFS76 are identical to those used in Chapter 3.

The steganalyzers are trained using the Fisher Linear Discriminant (FLD) ensemble which is broadly used for image steganalysis [Denemark et al., 2016, Kodovskỳ et al., 2012, Pevnỳ and Ker, 2015, Yu et al., 2016, Tang et al., 2016, Tan et al., 2017]. The FLD ensemble includes a number of base learners trained uniformly on the randomly selected feature subsets of the whole training data. The FLD ensemble uses the majority voting to combine the results of all base learners, but achieves much higher accuracy than any individual base learner. For more details of the FLD ensemble, we refer to the literature [Kodovskỳ et al., 2012, Cogranne and Fridrich, 2015]. The contribution of this study consists in identifying the appropriate feature set for 3D steganalysis and for this reason we do not test other machine learning algorithms for discriminating the stego-objects from the cover-objects.

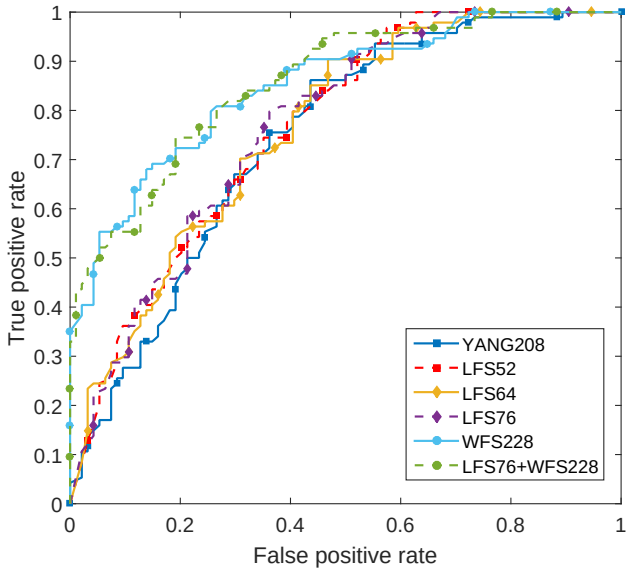
For each steganalyzer, we split the 354 pairs of cover-mesh and stego-mesh into 260 pairs for training and 94 pairs for testing. The steganalysis results are assessed by calculating two measurements: one is the median value of the areas under the Receiver Operating Characteristic (ROC) curves of the testing results, the other is the median value of the detection errors, both evaluated over 30 different splits of the data into training and testing sets. The ROC curve is created by plotting the true positive rate against the false positive rate at various threshold settings. The larger area under the ROC curve represents higher accuracy of the testing results. The detection error is the sum of false negatives (missed detections) and false positives (false alarms).

4.3.1 Steganalysis of two wavelet-based information hiding algorithms

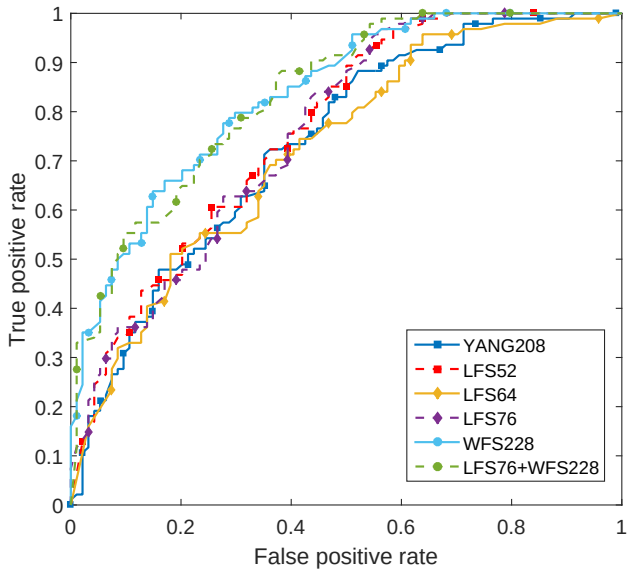
We first test the steganalytic features' performance when detecting the secret information embedded by the Wavelet-based High Capacity (WHC) watermarking method and Wavelet-based FRagile (WFR) watermarking method proposed in [Wang et al., 2008]. In order to compare the distortion produced by these two embedding algorithms, the information is hidden into the meshes after one iteration of 3D wavelet decomposition during the implementation of both algorithms, individually, rather than hierarchically as in [Wang et al., 2008]. Besides, aiming to investigate the influence of parameters on the steganalysis results, the control parameter ϵ_{hc} in WHC is set to the values {50, 100, 500, 1000}. When using WFR to embed information, the quantization step Δ_θ is set to the values $\{\pi/6, \pi/4, \pi/3, \pi/2\}$. The other parameters involved in WHC and WFR are all identical to the values from [Wang et al., 2008].

Figure 4.5 provides the ROC curves for the 3D steganalysis results when the information was embedded by WHC ($\epsilon_{hc} = 100$) and WFR ($\Delta_\theta = \pi/3$) for one trial when using the FLD ensembles trained on various 3D steganalytic feature sets. Instead of presenting many ROC curves, Figure 4.6 gives the median value of the area under the ROC curves of the detection for the stego-meshes carrying the information hidden by WHC and WFR, when varying the values of the parameters, ϵ_{hc} and Δ_θ in 30 trials.

It is obviously from both Figure 4.5 and 4.6 that the proposed feature set, WFS228, and the combination of LFS76 and WFS228 have better performance than the other 3D steganalytic feature sets. Among the existing feature sets, LFS76 shows relatively better performance than YANG208, LFS52 and LFS64, but not as good as WFS228. It is deduced that the 24-dimensional features extracted from spherical coordinate of the mesh contribute to the positive result provided by LFS76 when compared to LFS52. It is noted that, although LFS64 also includes LFS52 as a subset, its performance is usually worse than the latter, except in the case of WHC with $\epsilon_{hc} = 50$. It implies that the additional features in LFS64, such as the edge normal, mean curvature and total curvature are more efficient, when the targeted information hiding algorithms introduce higher distortions on the shape, because the setting of $\epsilon_{hc} = 50$ for WHC leads to higher distortions than the other values.



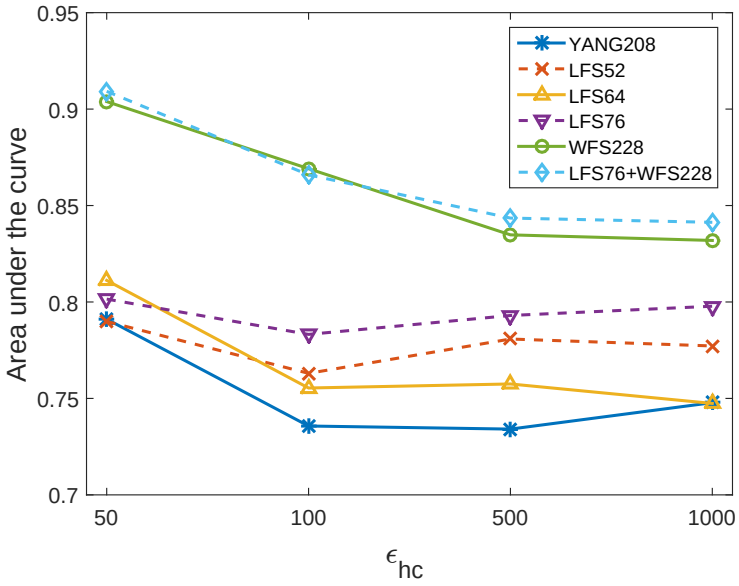
(a) WHC [Wang et al., 2008]



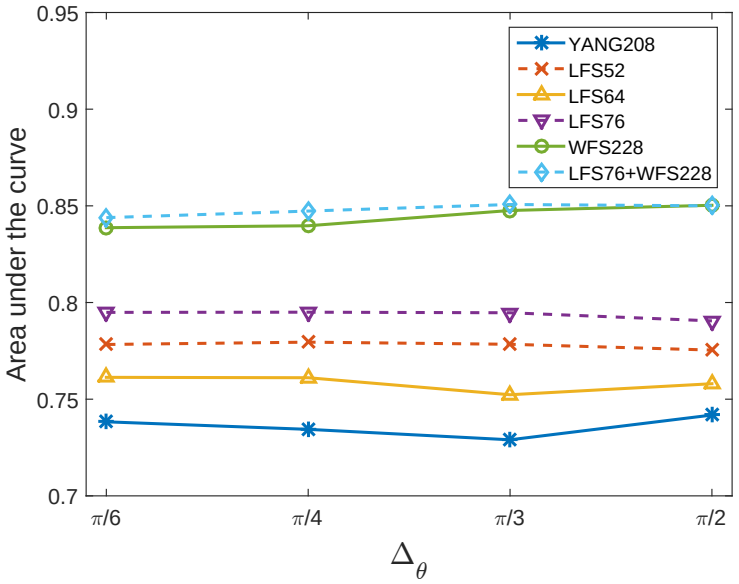
(b) WFR [Wang et al., 2008]

Figure 4.5: ROC curves of the steganalysis of WHC ($\epsilon_{hc} = 100$) and WFR ($\Delta_\theta = \pi/3$) for one trial using the FLD ensembles trained on various 3D steganalytic feature sets.

When considering the influence of the parameter ϵ_{hc} on WHC, it is shown that when ϵ_{hc} increases, the accuracies of the steganalyzers trained using the feature set WFS228 and the combination of LFS76 and WFS228, would decline. This happens because the increase of ϵ_{hc} leads to a smaller quantization step during embedding, and subsequently a lower level of embedding distortion. However, given the results from Figure 4.6(b), the parameter Δ_θ



(a) WHC [Wang et al., 2008]

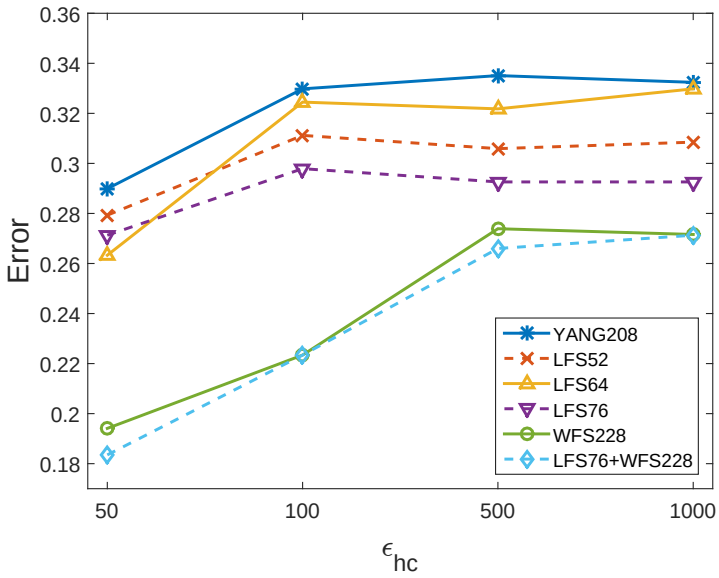


(b) WFR [Wang et al., 2008]

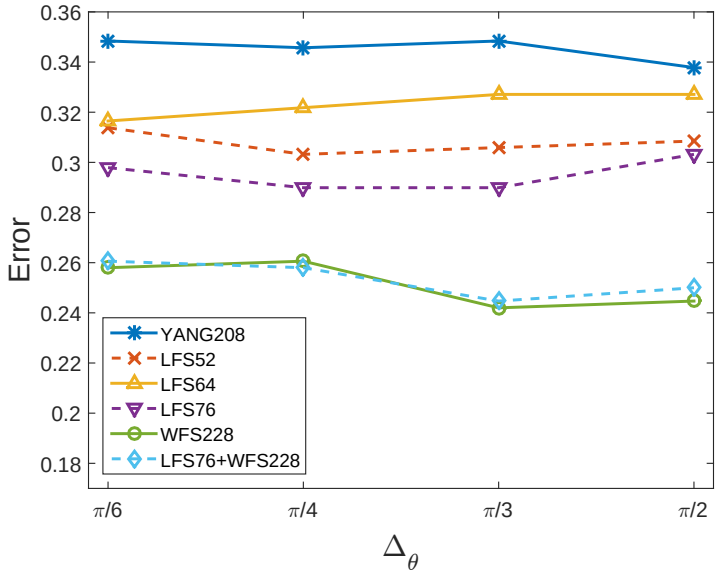
Figure 4.6: Median values of the area under the ROC curves of the detection results of the steganalyzers on the testing set over 30 independent splits for WHC and WFR when considering various values of the parameters.

does not affect the embedding distortion of WFR. When considering the resistance to 3D steganalysis, WFR is slightly better than WHC when the parameter ϵ_{hc} is low.

Figure 4.7 presents the median value of the detection errors of the stego-meshes embedded by WHC and WFR in 30 trials. The results are similar to the ones presented in Figure 4.6, which also indicates that proposed feature set, WFS228, and the combination of LFS76 and



(a) WHC [Wang et al., 2008]



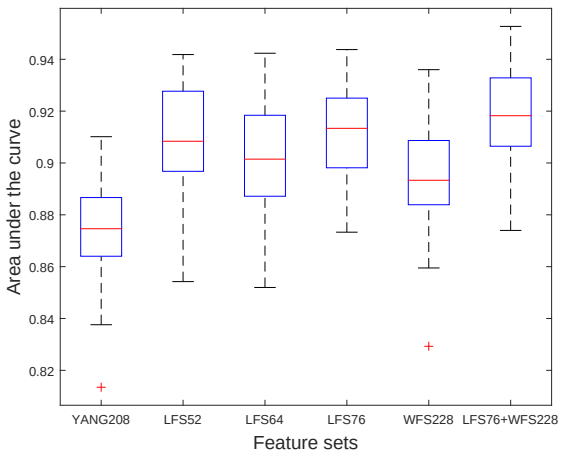
(b) WFR [Wang et al., 2008]

Figure 4.7: Median values of the detection errors of the steganalyzers on the testing set over 30 independent splits for WHC and WFR when considering various values of the parameters.

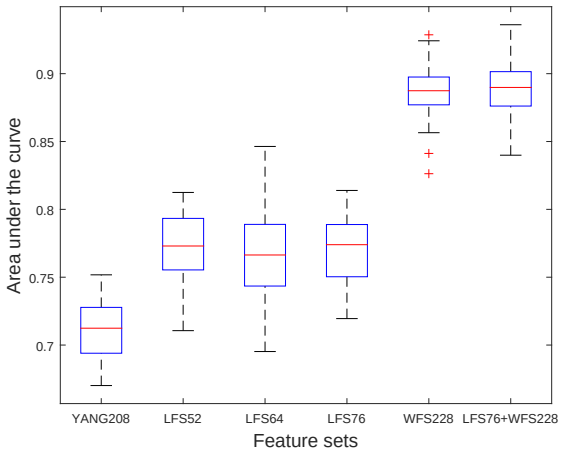
WFS228 achieve the best performance among various feature sets.

4.3.2 Steganalysis of six 3D embedding algorithms

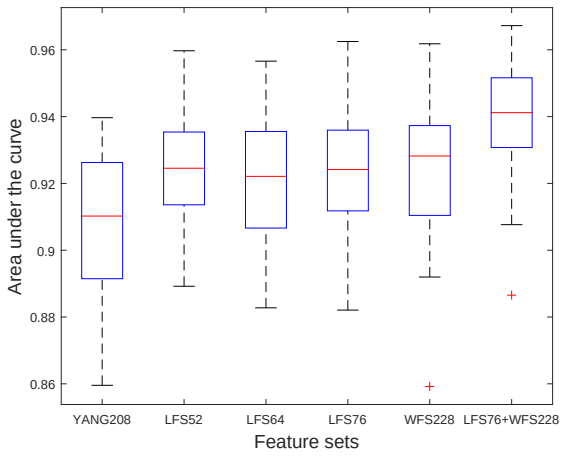
In the following we consider identifying the 3D stego-meshes produced by hiding information using six different embedding algorithms: the 3D steganography from [Itier and Puech, 2017]



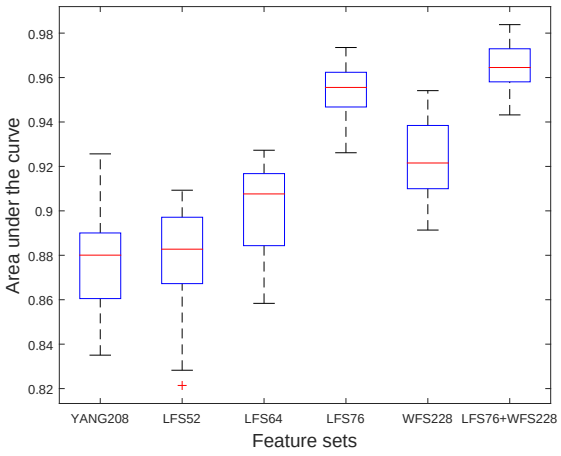
(a) HPQ [Itier and Puech, 2017]



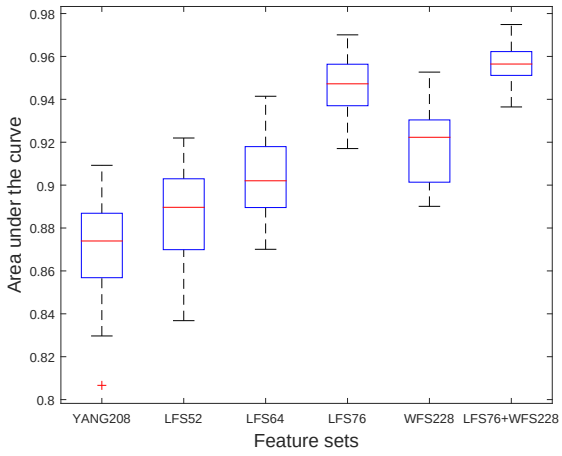
(b) HPQ-R [Li et al., 2017]



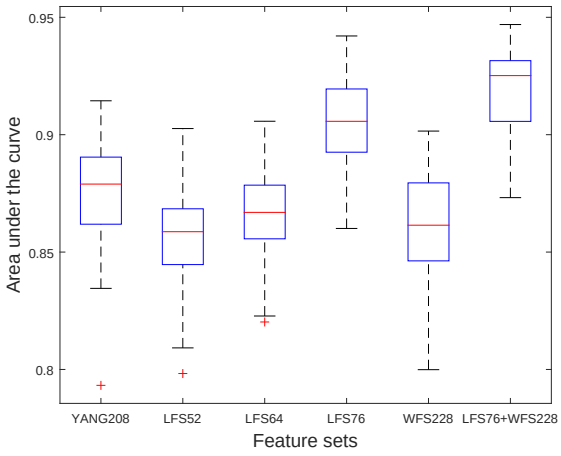
(c) MLS [Chao et al., 2009]



(d) MRS [Cho et al., 2007]

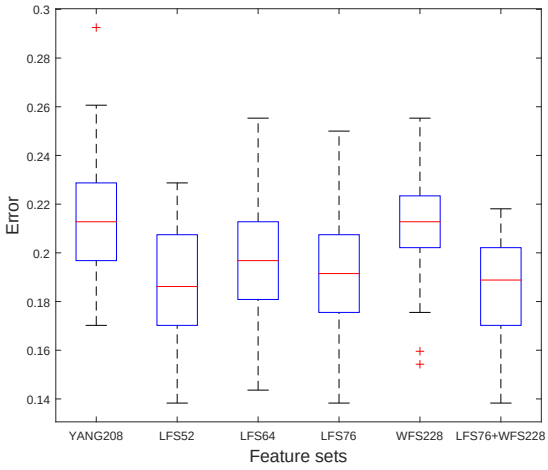


(e) VRS [Cho et al., 2007]

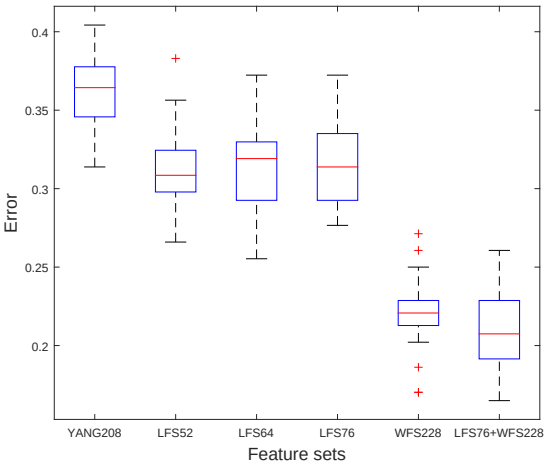


(f) SRW [Yang et al., 2017b]

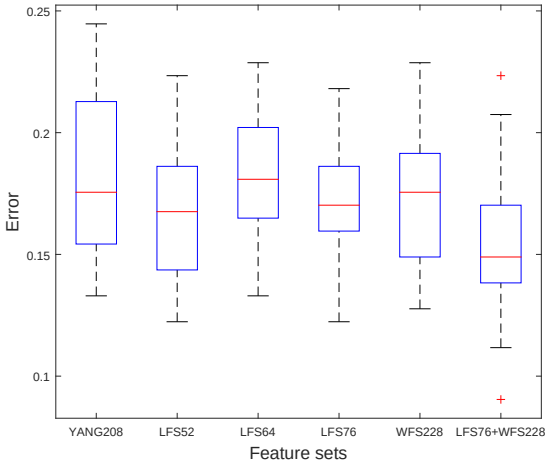
Figure 4.8: Box plots showing the confidence intervals for the area under the ROC curves of the detection results of steganalyzers trained when testing over 30 independent splits for the six 3D information embedding algorithms.



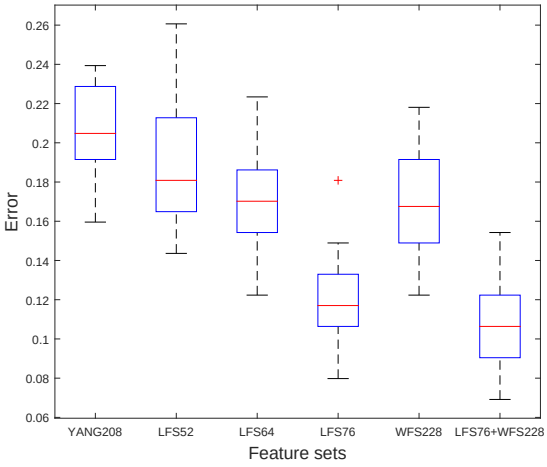
(a) HPQ [Itier and Puech, 2017]



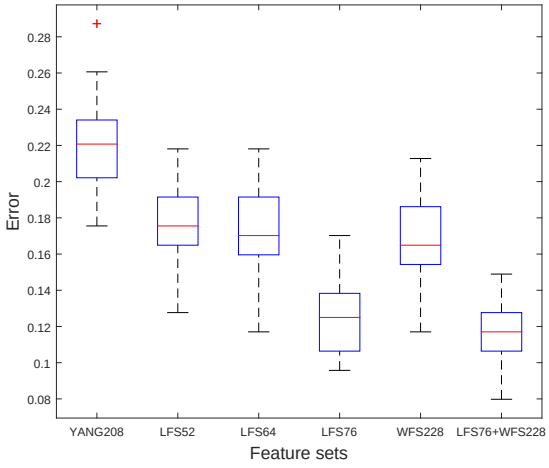
(b) HPQ-R [Li et al., 2017]



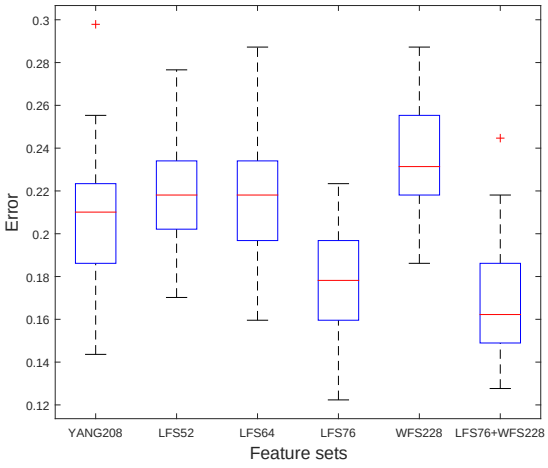
(c) MLS [Chao et al., 2009]



(d) MRS [Cho et al., 2007]



(e) VRS [Cho et al., 2007]



(f) SRW [Yang et al., 2017b]

Figure 4.9: Box plots showing the confidence intervals for the detection errors of steganalyzers trained when testing over 30 independent splits for the six 3D information embedding algorithms.

based on the Hamiltonian Path Quantization (HPQ) and its improved version having a high resistance to steganalysis, HPQ-R [Li et al., 2017]; the Multi-Layer Steganography (MLS) provided in [Chao et al., 2009]; two blind robust watermarking algorithms based on modifying the Mean or the Variance of the distribution of the vertices' Radial distances in the Spherical coordinate system, denoted as MRS and VRS, from [Cho et al., 2007] and the Steganalysis-Resistant Watermarking (SRW) method proposed in [Yang et al., 2017b].

For both 3D steganographic algorithms, HPQ and HPQ-R, the interval parameter is set as $\Delta = 1 \times 10^{-7}$ and the relative payload is 24 Bits Per Vertex (BPV), which was suggested in [Li et al., 2017]. When using MLS method from [Chao et al., 2009], we set the number of layers to 10, and consider the number of intervals as 10000. During the information embedding, all the vertices in the mesh are used as payload carriers, except for three vertices which are used as references for the extraction process. So the relative payload embedded by MLS is 10 BPV.

For MRS and VRS watermarking methods from [Cho et al., 2007], we consider $\alpha = 0.04$ for the watermark strength, while fixing the incremental step size to $\Delta k = 0.001$ and the message payload as 64 bits. During the generation of the stego-meshes using the SRW method from [Yang et al., 2017b], we set the parameter $K = 128$ which determines the number of bins in the histogram of the radial distances for all vertices. According to [Yang et al., 2017b], the upper bound of the embedding capacity is $\lfloor (K-2)/2 \rfloor$ bits. The parameter that controls the watermarking robustness of SRW is n_{thr}, which is set at 20. If the smallest number of elements in the bins from the objects is less than 20, we would choose the smallest nonzero number of elements in the bins as n_{thr}.

Figure 4.8 provides the box plots with the results for the area under the ROC curves of the detection results for the steganalyzers when testing over 30 independent data set splits when considering the information embedded by the above-mentioned six 3D embedding algorithms. Moreover, the corresponding detection errors for the steganalyzers are shown in the box plots in Figure 4.9. We can observe from Figures 4.8 and 4.9 that the combination of LFS76 and WFS228 achieves the best performance for the steganalysis of the stego-meshes embedded by these six 3D information hiding algorithms. It is important that in the context of the newly proposed HPQ-R, WFS228 shows much better performance than other existing

3D steganalytic feature sets. This happens because the WFS228 feature set can capture the displacement of the vertices made by HPQ-R particularly along the direction of the mesh edges, while the other steganalytic features fail to capture such changes. When aiming to identify the stego-meshes produced by HPQ, MRS, VRS and SRW, WFS228 feature set does not indicate a better performance than LFS76. Nevertheless, the WFS228 feature set extracts supplementary information about the existence of hidden information into 3D objects, when compared to LFS76, resulting in a better performance for the combination of LFS76 and WFS228 feature sets when compared to using just LFS76.

We can observe that LFS64 provides slightly worse performance than LFS52 with respect to the steganalysis of the stego-meshes embedded by HPQ, HPQ-R and MLS, but better performance in the cases of MRS, VRS and SRW. We infer that the additional 12-dimensional features in LFS64 proposed in [Kim et al., 2017] are more effective when detecting the higher level embedding distortions, because MRS, VRS and SRW are robust watermarking algorithms and usually produce higher distortions than the 3D steganographic algorithms, such as HPQ, HPQ-R and MLS. This also explains why LFS64 has better performance than LFS52 when detecting the stego-object embedded by WHC with $\epsilon_{hc} = 50$.

4.3.3 Efficiency analysis of the proposed features

In the following we provide the efficiency analysis of the proposed features when they are grouped in various categories according to the type of geometric features that they are characterizing. The efficiency of the features is assessed by the relevance between the feature vectors and the class label, which is calculated as the Pearson correlation coefficient

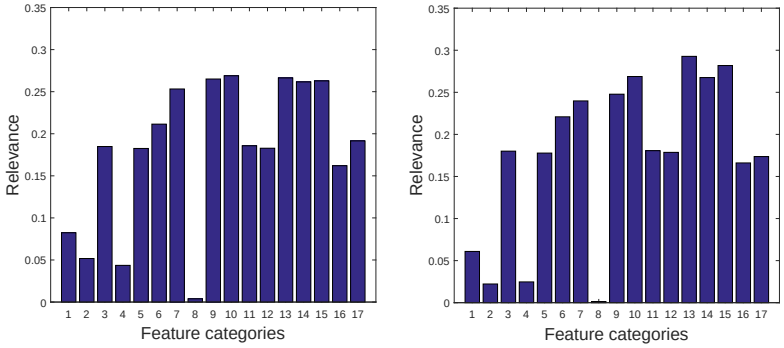
$$\rho(x_i, y) = \frac{cov(x_i, y)}{\sigma_{x_i}\sigma_y} \tag{4.17}$$

where x_i is the ith feature of a given feature set, and y is the class label indicating whether the mesh is a cover-mesh or a stego-mesh, cov represents the covariance and σ_{x_i} is the standard deviation of x_i. The Pearson correlation coefficient is a measure of the linear dependence between two variables [Hall, 1999], and it is often used in order to assess the feature's efficiency in the classic feature selection algorithms [Guyon and Elisseeff, 2003, Yu

and Liu, 2004]. The assumption made is that a higher linear dependence, or a higher relevance, between the feature and the class label can result in a better discriminant ability of that feature. So the relevance $\rho(x_i, y)$ is an estimation of the ith feature's efficiency for 3D steganalysis.

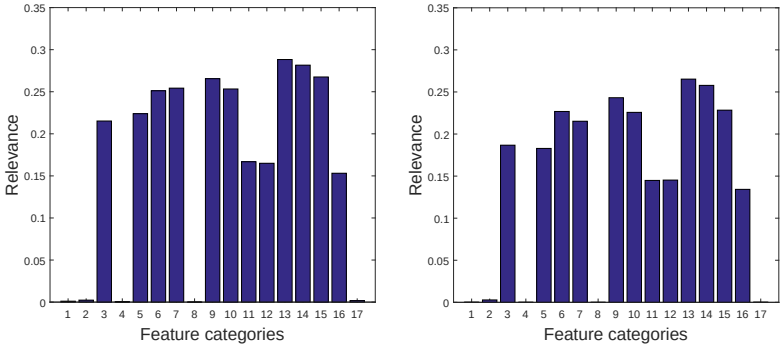
We split the features from the set WFS228 into 17 categories according to their representations of the geometric features listed in Table 4.1. Firstly, the relevance of each feature from the set WFS228 is calculated according to equation (5.1). Then, the relevance of each category is obtained by averaging the relevances for the features belonging to that category. The analysis is based on the features from the 354 cover-meshes and their corresponding stego-meshes obtained by embedding information when using the eight information hiding algorithms mentioned previously in this section. With respect to the setting of the parameters of the 3D watermarking algorithms, we set the parameters in WHC and WFR as $\epsilon_{hc} = 100$ and $\Delta_\theta = \pi/3$. The parameters in the other six algorithms, HPQ, HPQ-R, MLS, MRS, VRS and SRW, are the same with the ones used in the previous experiments. Since various embedding algorithms produce different kinds of distortions, the relevance of the feature may vary when the stego-meshes are produced by different information hiding algorithms.

The relevance of different categories of features in WFS228 are shown in Figure 4.10. The index of the categories correspond to the index of the geometric features from Table 4.1. It can be observed from Figure 4.10 that the features from categories 5, 6, 9, 10, 13, 14 and 15 have relatively higher relevance with the class label. These features are all characterizing the relationships between the WCV and its support edge or the relationships between the WCV and its neighbouring WCVs in the higher resolution mesh. Besides, the features from categories 3 and 7, characterizing the geometry of the WCVs, show a stable relevance with the class label among all eight cases. However, the features from categories 1, 2, 4 and 8, representing the edge vectors or flipped edge vectors of meshes in three resolution levels, show a lower relevance than the average. We infer that this is happening because the modifications made by the embedding algorithms are not changing directly the edges in different resolution meshes, so the corresponding features are not so efficient for steganalysis as the others. However, when we train the steganalyzers without the features from categories 1, 2, 4 and



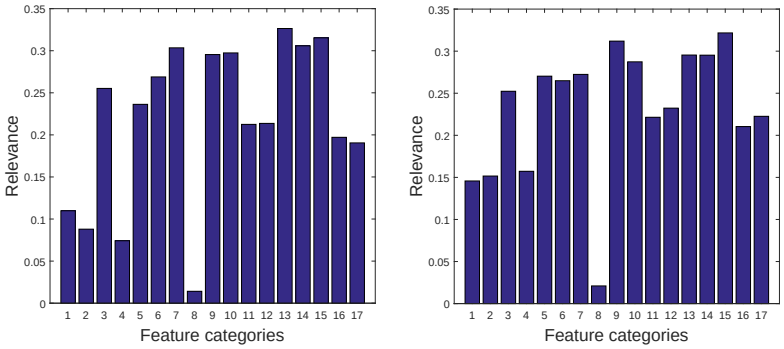
(a) WHC [Wang et al., 2008]

(b) WFR [Wang et al., 2008]



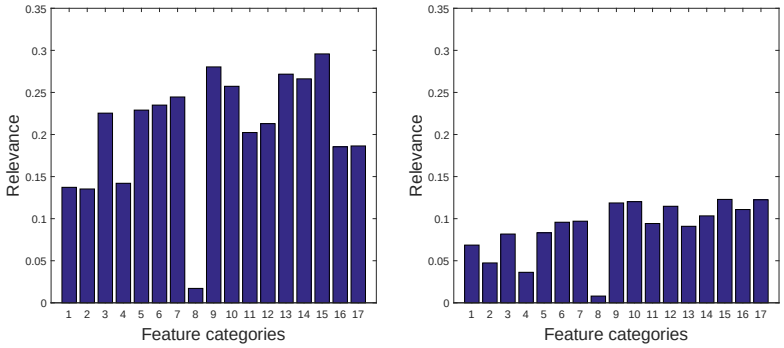
(c) HPQ [Itier and Puech, 2017]

(d) HPQ-R [Li et al., 2017]



(e) MLS [Chao et al., 2009]

(f) MRS [Cho et al., 2007]



(g) VRS [Cho et al., 2007]

(h) SRW [Yang et al., 2017b]

Figure 4.10: The relevance between the features and the class label, where the stego-meshes are obtained using eight information hiding methods, WHC, WFR, HPQ, HPQ-R, MLS, MRS, VRS and SRW, respectively. The category labels correspond with the index of the geometric features from Table 4.1.

8, the steganalysis results degrade to some extent, which means that all these features are contributing to the 3D steganalysis as well. When we categorize the features according to the four statistical moments that they represent, the features that represent higher order moments, such as skewness and kurtosis, have higher relevance than those representing the mean and variance. However, the features representing skewness and kurtosis are not robust to the variation of the cover source, which is going to be discussed in Chapter 5.

4.4 Conclusion

In this research chapter, we propose a steganalytic approach for triangle meshes based on the 3D wavelet multiresolution analysis. A number of geometric features are extracted from the original mesh and its smoothed counterpart, by considering three levels of resolution including the given resolution, one lower level of resolution and another of higher resolution. Characteristic geometric features are extracted from the original mesh and its smoothed version by taking into account the 3D wavelet decompositions and subdivisions. We consider both lower and high resolution scales obtained from their corresponding 3D wavelet decompositions and subdivisions. The first four moments of the distributions of the logarithms for the differences between the geometric features from the original mesh and those from its smoothed version are then used to form the proposed 228-dimensional steganalytic feature set, WFS228. Furthermore, the combination of LFS76 and the proposed WFS228 achieves better performance than other existing 3D steganalytic features when analyzing eight 3D embedding algorithms. An analysis of the efficiency for various feature categories, each grouping 3D wavelet features based on their geometry modeling properties, is undertaken as well, in this research study.

Chapter 5

Solving the Cover Source Mismatch Problem in 3D Steganalysis

5.1 Introduction

The feature sets proposed in Chapters 3 and 4 have shown good results in the 3D steganalysis experiments, when detecting embedding changes by several 3D steganographic algorithms. The previous experiments are carried out under the assumption that the cover source of the training data and the testing data are the same, however, this assumption is rarely true in the real world. Since it is very difficult to have the knowledge of the cover source used by the steganographiers, in the real world, the pre-trained steganalyzers are faced with the Cover Source Mismatch (CSM) problem, resulting in very poor detection accuracies in practice.

Firstly we discuss why the CSM would lead to poor detection accuracies of the steganalyzers. This happens because the steganalytic features are sensitive to the changes of the local geometrical and topological properties of mesh representations for the 3D objects. So the separation boundary for the classification of cover-objects and stego-objects, calculated for a specific set of cover-objects, may not be optimal for the objects from other cover sources, resulting in a poor classification accuracy for the steganalyzer. For example, the CSM problem in image steganalysis is caused by the fact that different brands or types of digital cameras may produce the images with different kinds of ISO noise, which would influence in a specific way the features used for training the image steganalyzer. So the hyperplane for the

classification of cover-images and stego-images trained over a certain cover source dataset may not be the optimal for the other cover sources, resulting in poor detection accuracies of the steganalyzers.

The steganalytic features are designed to be sensitive to the embedding changes which are rather small in order to be invisible, but non-sensitive to the global shapes of the objects. This is because the 3D steganographic algorithms embed unobservable changes into 3D objects, which are rather local in nature.

In order to solve the CSM problem, we need to find the features that are sensitive to the embedding changes, but robust to the variation of the cover sources. However, we are faced with a dilemma that the steganalytic features that are sensitive to the embedding changes are usually also sensitive to the variation of the cover sources, because the variation of the cover sources may lead to different types of local properties of the objects which are used for extracting the steganalytic features. The solution is to find a trade-off between the features' sensitivity to the embedding changes and their robustness to the variation of the cover sources. We propose to use the feature selection to achieve this trade-off.

The existing feature selection techniques used in pattern recognition applications can be grouped into three categories: wrapper methods, embedded methods and filter methods [Brown et al., 2012]. The wrapper methods use the training or validation error of a classifier to evaluate the utility of the candidate features, and an example is represented by the sequential feature selection algorithm [Pudil et al., 1994]. The embedded feature selection methods are based on properties specific to certain classification algorithms. For example, the method proposed in [Weston et al., 2001] selects the features that minimize bounds on the cross-validation error of the support vector machines. The filter methods define a scoring criterion to rank and select the features. The filter methods are independent of the classifier, so they are less likely to overfit than the wrapper or embedded feature selection methods which are classifier-dependent. Since the overfitting may cause more serious problem under the CSM scenarios, we choose to use a filter method to select the features to obtain a steganalyzer with better generalization ability.

In this chapter, we propose a feature selection algorithm that considers both the features' robustness to the variation of the cover source and their relevance to the class label, in

order to address the CSM problem in 3D steganalysis. Either the Pearson Correlation Coefficient (PCC) or Mutual Information Criterion (MIC) is used for measuring the feature's relevance to the class label in the algorithm. Meanwhile, the PCC is used for assessing the feature's robustness to the variation of the cover source. The 3D steganalysis framework which addresses the CSM problem is outlined in the diagram from Figure 5.1.

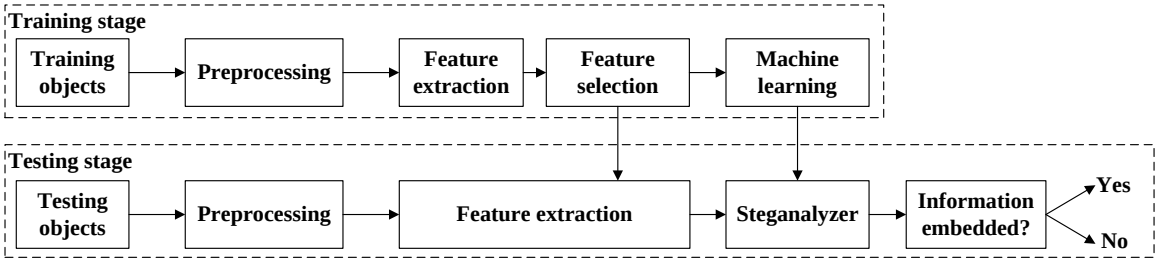


Figure 5.1: The 3D steganalysis framework based on statistical feature extraction and selection and machine learning methods.

The rest of this chapter is organized as follows: The definition of the features' robustness and relevance is presented in Section 5.2. The robustness and relevance-based feature selection algorithm and its pseudocode are presented in Section 5.3. The experiments that show the performance of the proposed method on solving the CSM problem in 3D steganalysis are provided in Section 5.4. The research proposed in this chapter is summarized in Section 5.5.

5.2 Definition of features' robustness and relevance

In the following we consider that we have a set of 3D objects $\mathcal{O}$, used as cover sources for training a steganalyzer. We use a data hiding algorithm for embedding information into the surface of these 3D objects, producing a set of stego-objects. A set of features is then extracted from both cover-set and stego-set of objects and the parameters characterizing their statistics are then used as inputs in a machine learning classifier to distinguish between the two classes of objects. The research studies from [Yang and Ivrissimtzis, 2014, Li and Bors, 2016] and [Li and Bors, 2017] have found several 3D features as being useful for 3D steganalysis. However, the sensitivity of these 3D features to the variation in the shape of the objects being analyzed varies from feature to feature. The steganalytic features that are more sensitive to the embedding changes contribute more to the performance of the ste-

ganalyzer. Nevertheless, these features would have a significant variation, outstripping their characteristic estimated distributions, when diversifying the cover source shapes. This ultimately leads to the degradation of the steganalyzer's performance under the CSM scenario. The solution of this dilemma in the CSM scenarios would be to find a trade-off between the features' sensitivity to the embedding changes and their robustness to the variation of the cover source. This is the motivation of the following feature selection method addressing the CSM problem in 3D steganalysis.

The proposed feature selection algorithm, called Robustness and Relevance-based Feature Selection (RRFS), presents a mechanism for choosing the features which will guarantee the steganalysis performance in the CSM scenarios. The key idea of the proposed algorithm is to find the features that are more robust to the variation of the cover source, while preserving a relatively high sensitivity to the embedding changes which is evaluated by their relevance to the class label. Naturally, two criteria are considered during the selection: the relevance of the features to the class label, and the robustness of the selected feature set to the variation of the cover source.

The feature selection algorithm proposed in this study belongs to the filter methods [Chandrashekar and Sahin, 2014], shown to be efficient when used for selecting input features in various machine learning algorithms. The filter methods are suitable to be applied in the cover source mismatch situations, because they can avoid the overfitting to the training data whilst being characterized by a better generalization during the testing stage [Guyon and Elisseeff, 2003].

In the proposed algorithm, the relevance of the features to the class label is estimated by using the Pearson Correlation Coefficient (PCC), calculated between the distribution of each feature and the corresponding objects' classes:

$$\rho(x_i, y) = \frac{cov(x_i, y)}{\sigma_{x_i}\sigma_y}, \tag{5.1}$$

where x_i is the ith feature of a given feature set, $\mathbf{X} = \{x_i | i = 1, 2, \ldots, N\}$, where N is the dimensionality of the input feature, y is the class label indicating whether the class corresponds to that of either being a cover-object or a stego-object, cov represents the covariance

and σ_{x_i} is the standard deviation of x_i. The Pearson correlation coefficient can capture the linear dependency between features and the label, with $|\rho(x_i, y)| = 1$ indicating a high degree of linearity while $\rho(x_i, y) = 0$ indicates a scattered dependency [Hall, 1999]. All features are ranked according to their relevance to the class label, calculated using equation (5.1), in descending order as:

$$|\rho(x_{i_1}, y)| > |\rho(x_{i_2}, y)| > \ldots > |\rho(x_{i_N}, y)|, \tag{5.2}$$

where $I = \{i_1, i_2, ...i_N\}$ is the feature index set.

In the following we also consider the Mutual Information Criterion (MIC) as a statistical measure of the relevance between each feature and the class label. MIC is known as a statistical measure of dependency between two variables. The mutual information between the ith feature, x_i, and the class label, y, is given by:

$$MI(x_i; y) = \sum_{x_i, y} p(x_i, y) \log \left(\frac{p(x_i, y)}{p(x_i)p(y)} \right), \tag{5.3}$$

where $p(x_i, y)$ is the joint probability distribution function of x_i and y, and $p(x_i)$ and $p(y)$ are the marginal probability distribution functions of x_i and y, respectively. Compared to the correlation coefficient, the mutual information is considered to be better in measuring the non-linear dependency between the variables [Li, 1990]. MIC was used in some classic feature selection methods, such as [Battiti, 1994, Fleuret, 2004, Lewis, 1992, Peng et al., 2005].

Features' robustness to the variation of the cover source is related to solving the CSM problem. Ideally, robust features should model the statistical characteristics that distinguish cover-objects and stego-objects even when these are different from those used during the training. In this study we assume that the testing dataset is different from the training one through some transformations which are controlled in the experimental setting of this study. If objects' features do not change much after applying various transformations to the cover-objects, they would be expected to provide similar steganalysis results to those achieved for the original cover-objects and stego-objects. Such features would have a strong robustness in the context of steganalyzers. In the following we consider certain changes to the surface of the objects and compare the features extracted before and after such changes.

The changes considered in this study are produced by mesh simplification and by adding noise to the mesh surface by significantly decreasing or increasing the local surface variation. Such changes can alter significantly the shape of 3D objects. We do not consider the mesh enhancement operations, such as remeshing and fairing, because these operations can make the mesh surface smoothed and the subsequent embedding modifications will be more easily detected. Then the Pearson correlation coefficient of the feature sets extracted before and after applying the changes to the 3D objects is calculated as:

$$\rho(x_i, x_{i,j}) = \frac{cov(x_i, x_{i,j})}{\sigma_{x_i}\sigma_{x_{i,j}}}, \tag{5.4}$$

where x_i and $x_{i,j}$ represent the ith feature extracted from the original set of cover-objects $\mathcal{O}$, used for training the steganalyzer, and from the objects obtained after applying specific transformations to the same cover source, $j = 1, 2, \ldots, M$, where M represents the number of transformations applied to the original set of cover-objects $\mathcal{O}$. This formula indicates how well correlated are the initial 3D features with those that are extracted after certain transformations. We normalize $|\rho(x_i, x_{i,j})|$ to the interval $[0, 1]$. The robustness is indicated by the average of the absolute values of the Pearson correlation coefficients obtained above, calculated for a specific feature i, for all $j = 1, ..., M$ transformations:

$$r_i = \frac{1}{M}\sum_{j=1}^{M} |\rho(x_i, x_{i,j})|, \tag{5.5}$$

where $i = 1, 2, \ldots, N$.

5.3 Robustness and relevance-based feature selection algorithm

The Robustness and Relevance-based Feature Selection (RRFS) algorithm starts with a preset number of N features as input. These N features, consist of several features that have been proposed for 3D steganalysis in previous studies [Yang and Ivrissimtzis, 2014, Li and Bors, 2016, Li and Bors, 2017]. The RRFS algorithm aims to find the most N' relevant

features which have relatively strong robustness to be used for a steganalyzer that addresses the CSM problem. N' features are selected after multiple passes through the features ranked according to their relevance, calculated using either equation (5.1) for PCC or (5.3) for MIC. During each pass, a subset of features $\mathcal{F}'$ with highest relevance is selected subject to the following conditions:

$$\mathcal{F}'|\{r_i > \theta_q, |\mathcal{F}'| < N'\} \tag{5.6}$$

where θ_q represents the threshold for the correlation corresponding to the q-th percentile of set $\{r_i|i = 1, 2, \ldots, N\}$, evaluating the robustness of the features. The features that are not robust enough are removed, and the RRFS algorithm reiterates, with considering the subset $\mathcal{F}'$ instead of N'. The trade-off between the robustness and the relevance of the features is controlled by a parameter τ. Initially, q is set as $100 - \tau$. After each iteration, if the cardinality of selected features $|\mathcal{F}'| < N'$, then we reduce the threshold to a value corresponding to a percentile of $q - \tau$, and repeat the feature selection by considering a new threshold $\theta_{q-\tau}$ instead of θ_q. When the parameter τ is closer to 0, the feature selection algorithm tends to select the more robust features. If τ increases, the features with higher relevance to the class label will be more probably selected by the proposed algorithm. The setting of the parameter τ is investigated in Section 5.4.2.

In this way with each iteration we add additional features to the set of selected features such that whilst increasing the feature set we preserve the generalization capability of the steganalyzer. Since the features are ranked according to their relevance in descending order, the features with higher relevance are first selected if their robustness is above the threshold θ_q. After each iteration, the threshold θ_q is gradually reduced, considering lower percentiles $q - \tau$ in stead of q , until the dimensionality of the selected features becomes equal to N'. These N' selected features are robust enough to the variation of cover source whilst having a relatively high relevance to the class label at the same time. The RRFS algorithm that uses PCC as the measure of the features' relevance to the class label is named RRFS-PCC and its pseudocode is provided in Algorithm 1. Instead of using PCC, the RRFS-MIC algorithm uses MIC to calculate the features' relevance as defined in equation (5.3). The description of RRFS-MIC is similar to that of Algorithm 1.

Algorithm 1: RRFS-PCC algorithm

Input:
Features extracted from the cover-objects and stego-objects used for training $\mathbf{X} = \{x_i | i = 1, 2, ..., N\}$;
Features extracted from other cover sources and corresponding stego-objects $\mathbf{X}_j = \{x_{i,j} | i = 1, 2, ..., N, j = 1, 2, ..., M\}$;
Class label y;
Step size parameter τ;
Dimensionality of the selected feature N'.
Output: Index of the selected feature subset $\mathcal{F}'$.

Compute the relevance of the features to the class label, $\rho(x_i, y) = \frac{cov(x_i,y)}{\sigma_{x_i}\sigma_y}$;
Compute the Pearson correlation coefficient of two feature sets, $\rho(x_i, x_{i,j}) = \frac{cov(x_i,x_{i,j})}{\sigma_{x_i}\sigma_{x_{i,j}}}$;
Normalize $|\rho(x_i, x_{i,j})|$ to [0,1];
Compute the robustness of the features to the variation of the cover source, $r_i = \frac{1}{M}\sum_{j=1}^{M} |\rho(x_i, x_{i,j})|$;
Sort the features by relevance $|\rho(x_i, y)|$ in the descending order and get the index $I = \{i_1, i_2, ...i_N\}$;
Initialize $q \leftarrow 100 - \tau$ and $\theta_q \leftarrow percentile(\{r_i | i = 1, 2, ...N\}, q)$;
while $|\mathcal{F}'| < N'$ **do**
    for $k \leftarrow i_1$ *to* i_N **do**
        if $(k \notin \mathcal{F}') \wedge (r_k > \theta_q) \wedge (|\mathcal{F}'| < N')$ **then**
            Add k *to* $\mathcal{F}'$;
        end
        $q \leftarrow q - \tau$;
        $\theta_q = percentile(\{r_i | i = 1, 2, ...N\}, q)$;
    end
end
Return $\mathcal{F}'$;

5.4 Experimental results

During the experimental results, we analyze the effect of the cover source mismatch in 3D steganalysis. We apply the RRFS algorithm to select a feature subset from a given larger feature set, when analyzing a large set of stego- and cover-objects, and test the performance of the selected feature subset within the context of CSM scenarios.

For the experimental data set we consider 354 3D objects represented as meshes which are part of the Princeton Mesh Segmentation project [Chen et al., 2009] database, which is also used in the experiments of Chapter 3 and Chapter 4. This database contains a large variety of shapes, representing the human body under a variety of postures, statues, animals, toys, tools and so on.

The stego-objects are generated by applying four information hiding algorithms: the 3D Multi-Layers Steganography (MLS) proposed in [Chao et al., 2009], the blind robust watermarking algorithms based on modifying the Mean of the distribution of the vertices' Radial distance coordinates in the Spherical coordinate system, denoted as MRS, from [Cho et al., 2007], the Steganalysis-Resistant Watermarking (SRW) method proposed in [Yang et al., 2017b] and the Wavelet-based High Capacity (WHC) [Wang et al., 2008] watermarking method. The embedded information is a pseudorandom bit stream which simulates the secret messages or watermarks hidden by the steganographier. In the case of MLS [Chao et al., 2009], the number of embedding layers is considered as 10 and the number of intervals is chosen as 10000. The relative payload ratio of each layer is nearly 1, except for three vertices used for extracting the code, which are not modified at all. The payload embedded by MRS from [Cho et al., 2007] is 64 bits and the watermarking strength is 0.04. We set the parameter $K = 128$ in SRW proposed in [Yang et al., 2017b] and the algorithm's upper bound of the embedding capacity is $\lfloor (K-2)/2 \rfloor$. The control parameter for WHC is $\epsilon_{hc} = 100$ and the other parameters are identical to the suggested values given in [Wang et al., 2008].

Similarly to the approach from [Li and Bors, 2016] we consider FLD ensembles [Cogranne and Fridrich, 2015, Kodovskỳ et al., 2012] as the machine learning based steganalyzer. The parameters for the FLD ensembles, such as the number of the base learner and the subspace dimensionality, are chosen as in [Kodovskỳ et al., 2012]. The classifiers' performance is

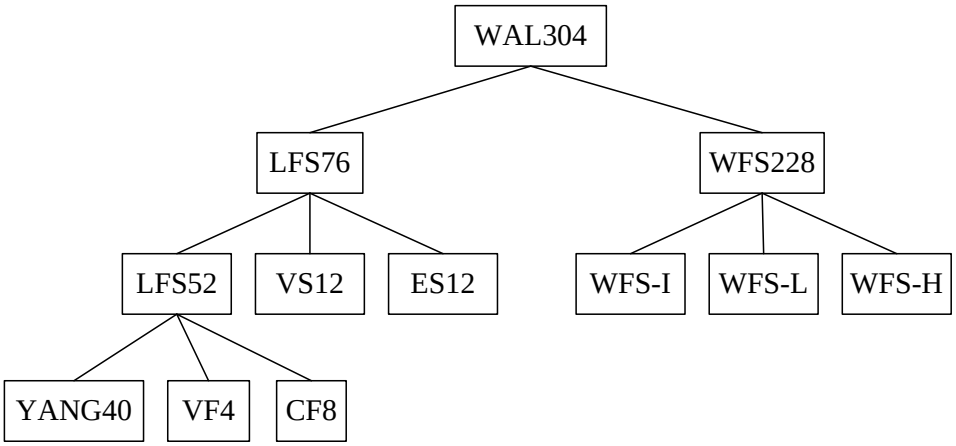


Figure 5.2: The feature sets that included in WAL304.

measured by the detection errors which are the sums of false negatives (missed detections) and false positives (false alarms).

The feature set, used as a base for selecting the most robust and relevant subset of features for steganalysis, is a 304-dimensional feature set, called WAL304, generated by combining two feature sets used for 3D steganalysis, LFS76 [Li and Bors, 2017] and WFS228 proposed in previous Chapters 3 and 4, respectively. The feature sets that included in WAL304 are illustrated in Figure 5.2 as a tree graph. The VF4 and CF8 are representing the 4-dimensional vertex formal features and 8-dimensional curvature feature in LFS52 [Li and Bors, 2016], respectively. The VS12 and ES12 are the spherical coordinate features added to LFS52, forming the LFS76. Meanwhile, WFS228 is the combination of Wavelet Feature Set from the Initial resolution mesh (WFS-I), Wavelet Feature Set from the Lower resolution mesh (WFS-L) and Wavelet Feature Set from the Higher resolution mesh (WFS-H).

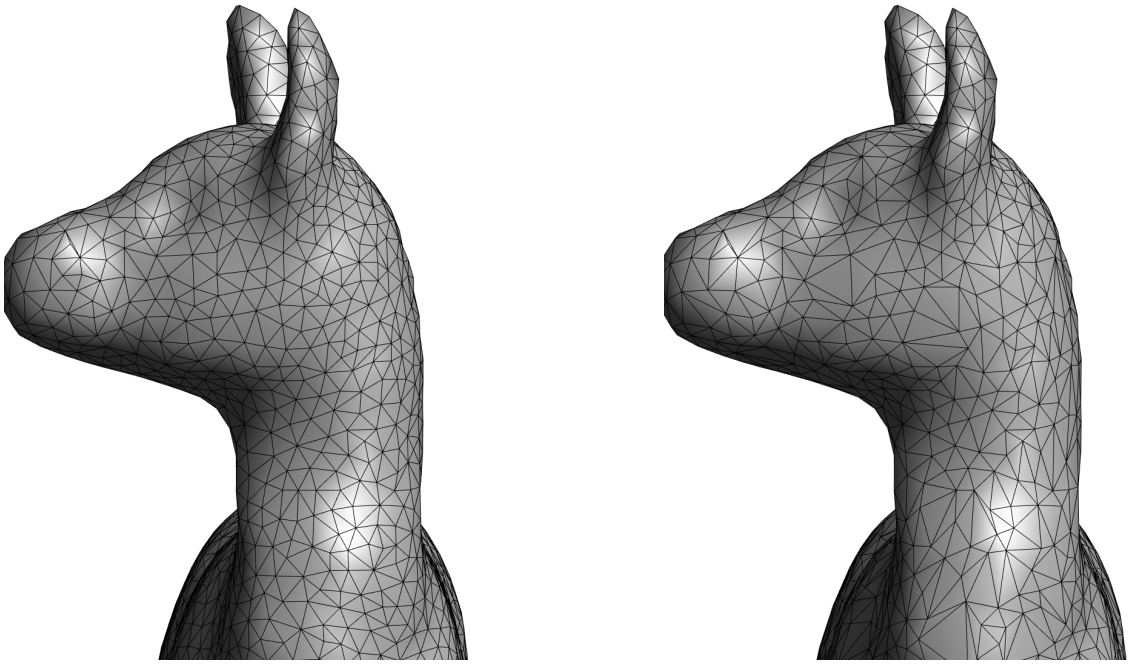
We consider the initial objects of the database as cover-objects and after watermarking these we obtain the stego-objects. In the experiments, the feature set WAL304, is initially extracted from the cover-objects and stego-objects. Then, in order to simulate the effect of CSM problem, we apply certain transformations, such as by adding noise or by mesh simplification, to the original cover-objects and we consider the transformed objects as cover-objects for information hiding. Feature sets are extracted from these transformed cover-objects and their corresponding stego-objects. For each kind of transformation, we assume four different levels of transformations going from superficial changes to more dramatic modifications ap-

plied to the surfaces of the objects, by either increasing the level of noise, through β, or the mesh simplification factor λ. Thus, during the calculation of the robustness, we have a set of $M = 8$ transformations applied to the original objects. In order to test the performance of the selected features in the context of the CSM scenario, we randomly select 260 cover-objects from the original cover source and the corresponding stego-objects for training the steganalyzer. The steganalyzers are trained over the feature subsets selected by the RRFS algorithm. Then we test the steganalyzer on the other 94 pairs of cover-objects and stego-objects originated from the transformed cover sources, which have not been used during the training.

5.4.1 The CSM problem

In the following, we analyze the steganalysis capability, when hiding information by means of four different information hiding algorithms. We consider both cases of 3D steganalysis under the CSM scenario and without it. In the case when testing the steganalytic algorithm, without considering the object transformations for the CSM paradigm, we utilize the whole WAL304 feature set from the 260 pairs of cover-objects and stego-objects for training the steganalyzer, while using the other 94 pairs of objects from the database for testing. The experiment is repeated 10 times with independent splits for the training and testing sets.

In order to generate multiple cover sources, we distort the original objects of the database by considering two different transformations to various extents: mesh simplification and noise addition. While the first transformation changes the local topology of the mesh, the latter one alters the roughness of the surface. These transformations can simulate the distortions of the meshes caused by using different 3D scanners when scanning the same object, because the 3D scanners may have different accuracies and precisions, and they may use different algorithms to create the 3D meshes. The likelihood of such mesh variations would increase even more in the case when scanning objects produced by 3D printers, because of the roughness of the surface in such objects. When creating new shapes by considering additive noise to the mesh surface of an original objects, we actually create a challenging problem for a 3D steganalyzer, because such distortions resemble those produced to the mesh when hiding information. Thus we would actually increase the uncertainty in separating the cover-objects from stego-objects.



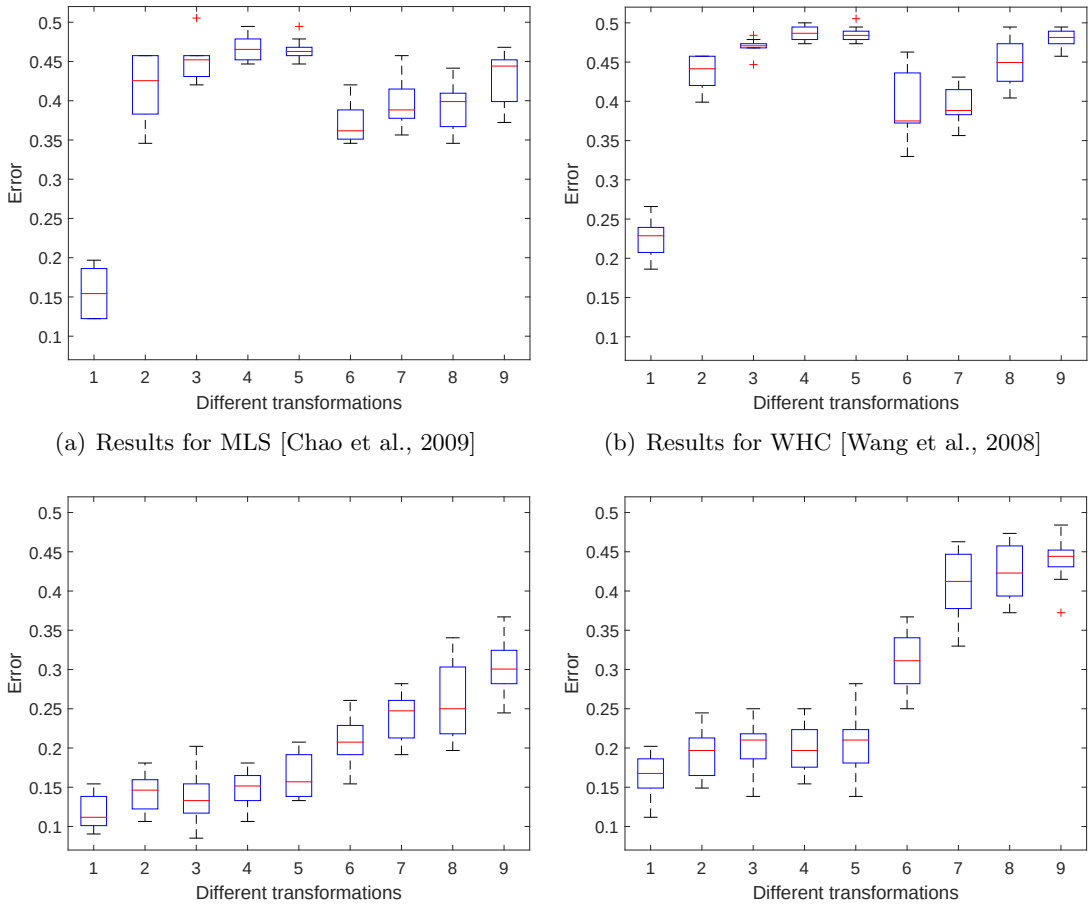
(a) Cover-object.

(b) Stego-object after mesh simplification with $\xi = 0.8$.

Figure 5.3: Example when using surface simplification on the cover-object to test the cover-source mismatch paradigm in 3D steganalyzers

The mesh simplification is performed using the MATLAB function *reducepatch*[1] which reduces the number of faces, while aiming to preserve the overall shape of the 3D object. The level of the simplification is controlled by the parameter $\xi \in \{0.98, 0.95, 0.9, 0.8\}$ which is interpreted as a fraction of the original number of faces. For example, if $\xi = 0.8$, then the number of the faces is reduced to 80% of their count from the original mesh. The close-up detail of one of the original 3D objects used in the experiments is shown in Figure 5.3(a), while its corresponding stego-object obtained by using MLS embedding algorithm after mesh simplification by a factor of $\xi = 0.8$, is shown in Figure 5.3(b). If we would have chosen smaller ξ values, the resulting meshes would have been dramatically changed, while addressing CSM problem in 3D steganalysis is about localized changes in the mesh surface. Besides, the mesh simplification algorithm used by *reducepatch* may produce particular artifacts, for example, it may result in the effect that the sizes of the triangles on the flat part of the simplified mesh would vary dramatically. When considering uniform noise addition, the amplitude of noise is

[1] http://uk.mathworks.com/help/matlab/ref/reducepatch.html



(a) Results for MLS [Chao et al., 2009]

(b) Results for WHC [Wang et al., 2008]

(c) Results for MRS [Cho et al., 2007]

(d) Results for SRW [Yang et al., 2017b]

Figure 5.4: Box plots showing the steganalysis detection errors for the information hiding methods proposed in [Chao et al., 2009, Wang et al., 2008, Cho et al., 2007, Yang et al., 2017b] when considering and without addressing the CSM challenge due to different 3D shape modifications. Label 1 represents the results without considering the CSM challenge during the training and testing. Labels 2 to 5 represent the results with the CSM due to additive noise at the levels of $\beta \in \{1 \cdot 10^{-5}, 2 \cdot 10^{-5}, 3 \cdot 10^{-5}, 5 \cdot 10^{-5}\}$. Labels 6 to 9 represent the results with the CSM due to mesh simplification at the level of $\xi \in \{0.98, 0.95, 0.9, 0.8\}$.

modulated by the parameter βD, with $\beta \in \{1 \cdot 10^{-5}, 2 \cdot 10^{-5}, 3 \cdot 10^{-5}, 5 \cdot 10^{-5}\}$, and D is the maximum distance between the projections of any two vertices on the first principal axis, obtained by applying the Principal Component Analysis (PCA) on the original 3D object. Since the size of the objects in the database may correspond to different scales, by using the parameter βD to control the amplitude of the noise we can obtain relative consistent effects by the additive noise on the original shapes. With the application of various levels of mesh simplification and noise addition, we can observe the performance of the steganalytic approaches under different levels of CSM scenarios.

Figure 5.4 depicts the box plots for the detection errors, indicating their variation from the mean, for the four information hiding algorithms without CSM (Label 1) and with CSM for labels 2-9, where the diversity of objects for testing the CSM problem is produced by shape transformations through adding noise, or by mesh simplification, each by considering 4 levels of induced distortions to the original shapes. We remark that in the case without CSM (Label 1), the training set did not contain the noisy or the simplified meshes. From Figures 5.4 (a) and (b) it can be observed that the CSM paradigm poses more challenges to steganalysis in the case of the changes embedded by the MLS [Chao et al., 2009] and WHC [Wang et al., 2008] steganographic algorithms than in the cases of the MRS [Cho et al., 2007] and SRW[Yang et al., 2017b], whose results are provided in Figures 5.4 (c) and (d), respectively. With respect to MRS and SRW, the CSM challenge due to the diversification of shapes through mesh simplification leads to the dampening of the hidden information detection accuracy. However, from these results, it can be observed that the CSM challenge due to the diversification of shapes through additive noise does not have much influence on the detection results. This happens because the added noise to the cover-object surface is actually smaller than the changes produced to the surface of 3D objects by the two watermarking algorithms.

5.4.2 The analysis for selecting the parameter τ in RRFS algorithm

The parameter τ controls the trade-off between the robustness and the relevance of the features during selection, as explained in the first paragraph from Section 5.3 and in the Algorithm 1. In the following experiment, we consider the steganalysis of stego-objects with

information embedded by the MLS algorithm, proposed in [Chao et al., 2009], whose steganalysis results were the poorest when considering the CSM assumptions according to the results provided in the previous section. The feature selection and training of the steganalyzer are following the same rules as described before. More specifically, during the feature selection stage, we set $\tau \in \{2, 10, 20, 30, 40, 50\}$, when using the RRFS-PCC algorithm. When τ is small, the algorithm gives more consideration to the robustness of the features to the variation of the cover source, while when τ is larger it gives more consideration to the feature's relevance to the class label. Because we consider that the robustness of the feature is very important for addressing the CSM problem, we tend to set a small value for τ. We consider increasing the number of selected features N' from 10 to 300, with steps of 10 at each iteration.

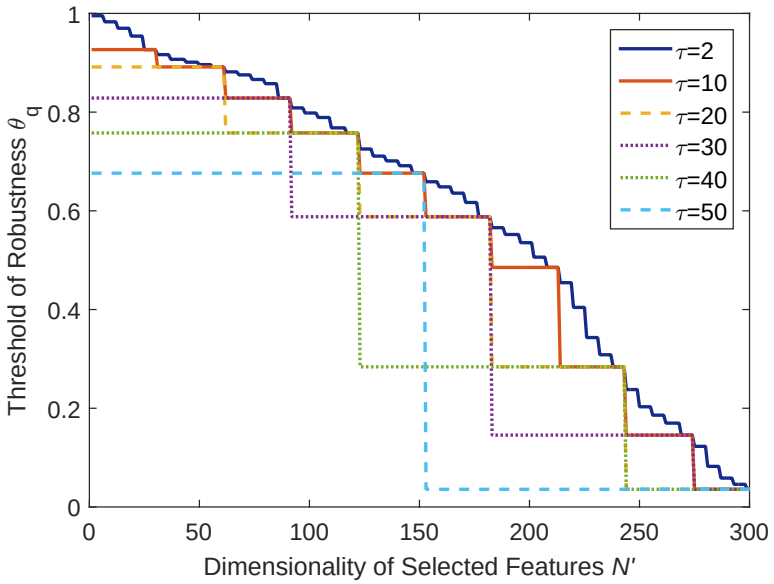


Figure 5.5: The variation for the threshold of the features' robustness θ_q when using the RRFS-PCC algorithm with $\Delta \in \{2, 10, 20, 30, 40, 50\}$ in order to select N' features over one split of data into training/testing sets.

In Figure 5.5, we present the variation of the threshold of the features' robustness θ_q, when using the RRFS-PCC algorithm with $\tau \in \{2, 10, 20, 30, 40, 50\}$ in order to select N' features over one split of data into training/testing sets. It shows that the threshold θ_q decreases according to the variation of the parameter τ, which controls the trade-off between the robustness and relevance. A larger area under the plot of the threshold of robustness θ_q

means more consideration is given to the features' robustness. So when $\tau = 2$, the selection of the features is mostly based on their robustness to the variation of the cover source.

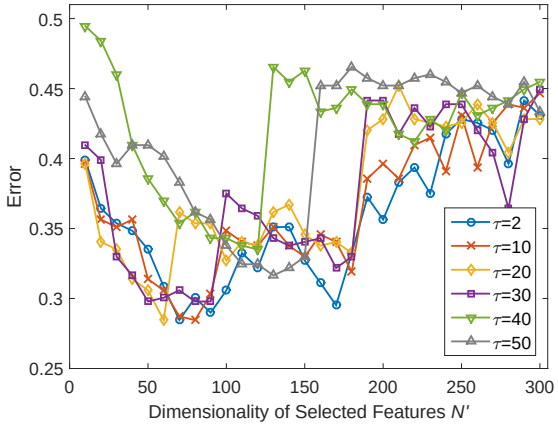
The testing of the steganalyzers is carried out by considering four 3D shape transformations, for testing the CSM scenario. The shape transformations are produced by additive noise with amplitude defined by $\beta \in \{1 \cdot 10^{-5}, 3 \cdot 10^{-5}\, 5 \cdot 10^{-5}\}$, and mesh simplification at the level of $\xi \in \{0.98, 0.9, 0.8\}$. The results are shown in Figure 5.6. From the plots in Figure 5.6 it can be observed that in the case of CSM due to noise addition, smaller values, such as $\tau \in \{2, 10, 20\}$ lead to a better performance of the steganalyzer. Since we have to consider the CSM scenarios due to both noise addition and mesh simplifications, we set $\tau = 10$ as a trade-off solution in the following experiments.

5.4.3 Comparison with other approaches

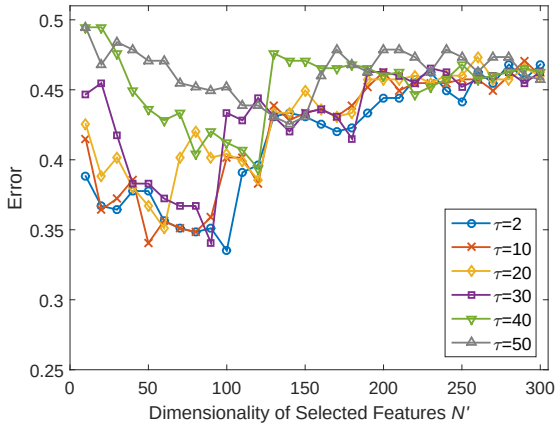
In the following, we compare the proposed feature selection algorithms for steganalysis, RRFS-PCC and RRFS-MIC, with filter feature selection algorithms used in pattern recognition, such as min-Redundancy and Max-Relevancy (mRMR) [Peng et al., 2005], Double Input Symmetrical Relevance (DISR) [Meyer and Bontempi, 2006], Conditional Mutual Information Maximization (CMIM) [Fleuret, 2004], Infinite Feature Selection (Inf-FS) [Roffo et al., 2015] and Infinite Latent Feature Selection (ILFS) [Roffo et al., 2017], which have shown very good generalization ability in a wide range of applications [Brown et al., 2012]. In addition, we also compare with a simplified version of our algorithm, Relevance based Feature Selection (RFS), which selects the features with higher relevance to the class label, measured by PCC, but without considering the robustness to the variation of cover source. We repeat the steganalysis experiments, using FLD ensembles for 10 different splits of data sets and then consider the median of the resulting errors as the final test results.

Figures 5.7, 5.8, 5.9 and 5.10 show the test results when using features selected by the proposed RRFS-PCC and RRFS-MIC algorithms compared with the other six feature selection algorithms. These results are obtained when considering the initial set of features as WAL304 for steganalysis under the CSM assumption, by considering the distortions caused by mesh simplification and uniform additive noise as in the previous section.

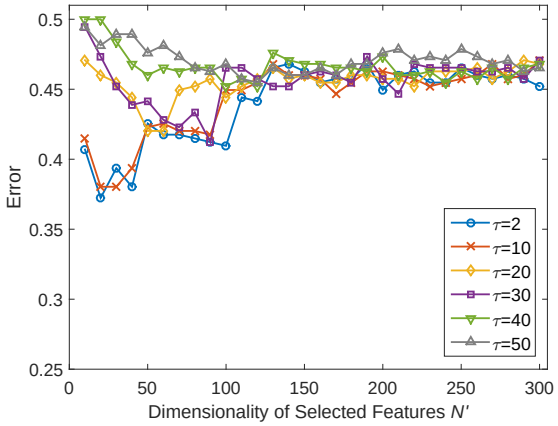
Figures 5.7 and 5.8 show the detection errors for the MLS and WHC algorithms, proposed



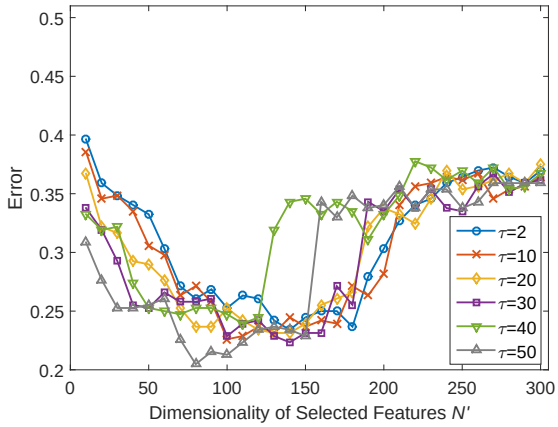
(a) Noise Level $\beta = 1 \cdot 10^{-5}$



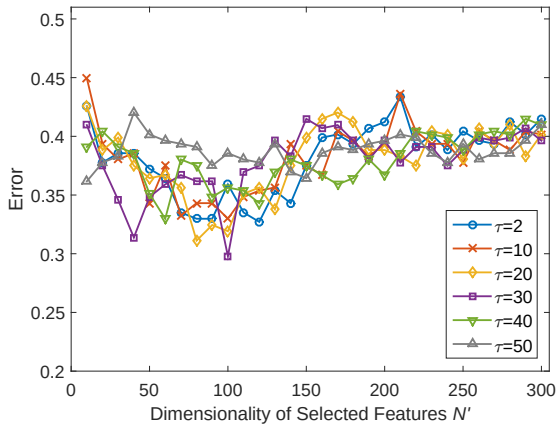
(b) Noise Level $\beta = 3 \cdot 10^{-5}$



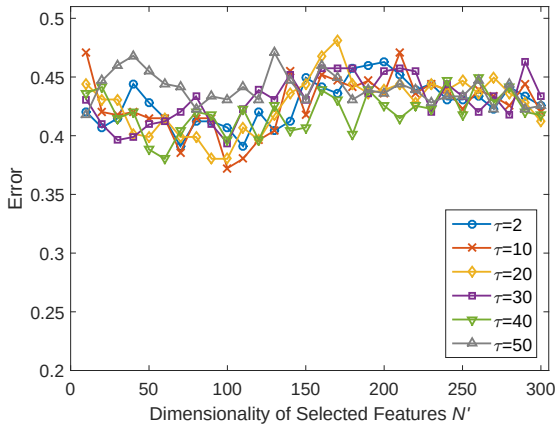
(c) Noise Level $\beta = 5 \cdot 10^{-5}$



(d) Simplification Level $\xi = 0.98$

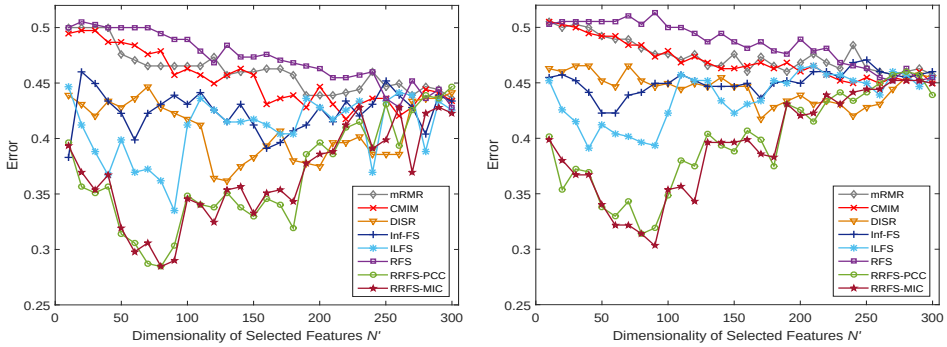


(e) Simplification Level $\xi = 0.9$

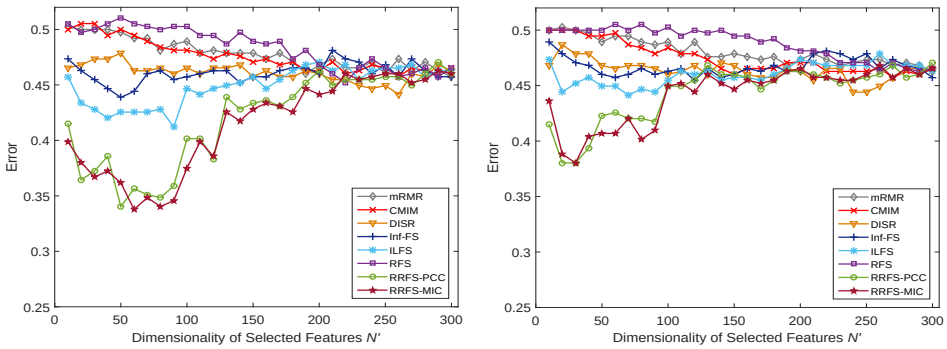


(f) Simplification Level $\xi = 0.8$

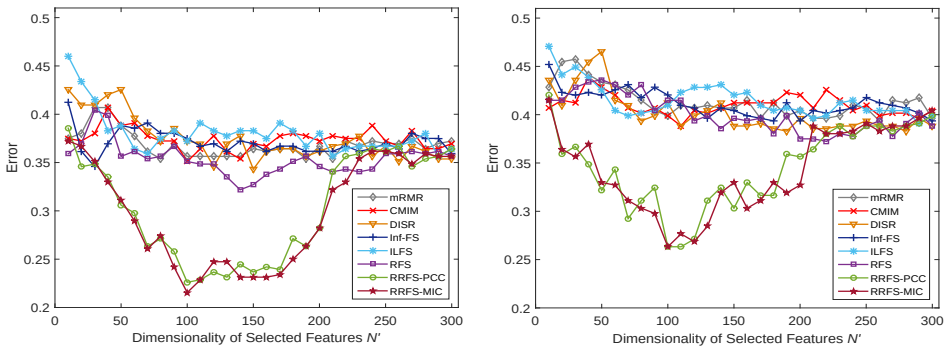
Figure 5.6: Median values of the detection errors for MLS [Chao et al., 2009] when the steganalyzers are trained over the feature subsets selected by the RRFS-PCC with $\tau \in \{2, 10, 20, 30, 40, 50\}$ in the CSM scenarios over 10 different splits of the training/testing set.



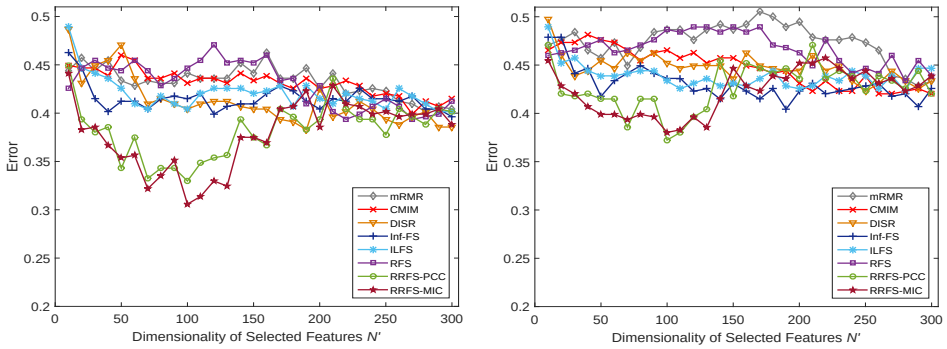
(a) Noise Level $\beta = 1 \cdot 10^{-5}$ (b) Noise Level $\beta = 2 \cdot 10^{-5}$



(c) Noise Level $\beta = 3 \cdot 10^{-5}$ (d) Noise Level $\beta = 5 \cdot 10^{-5}$

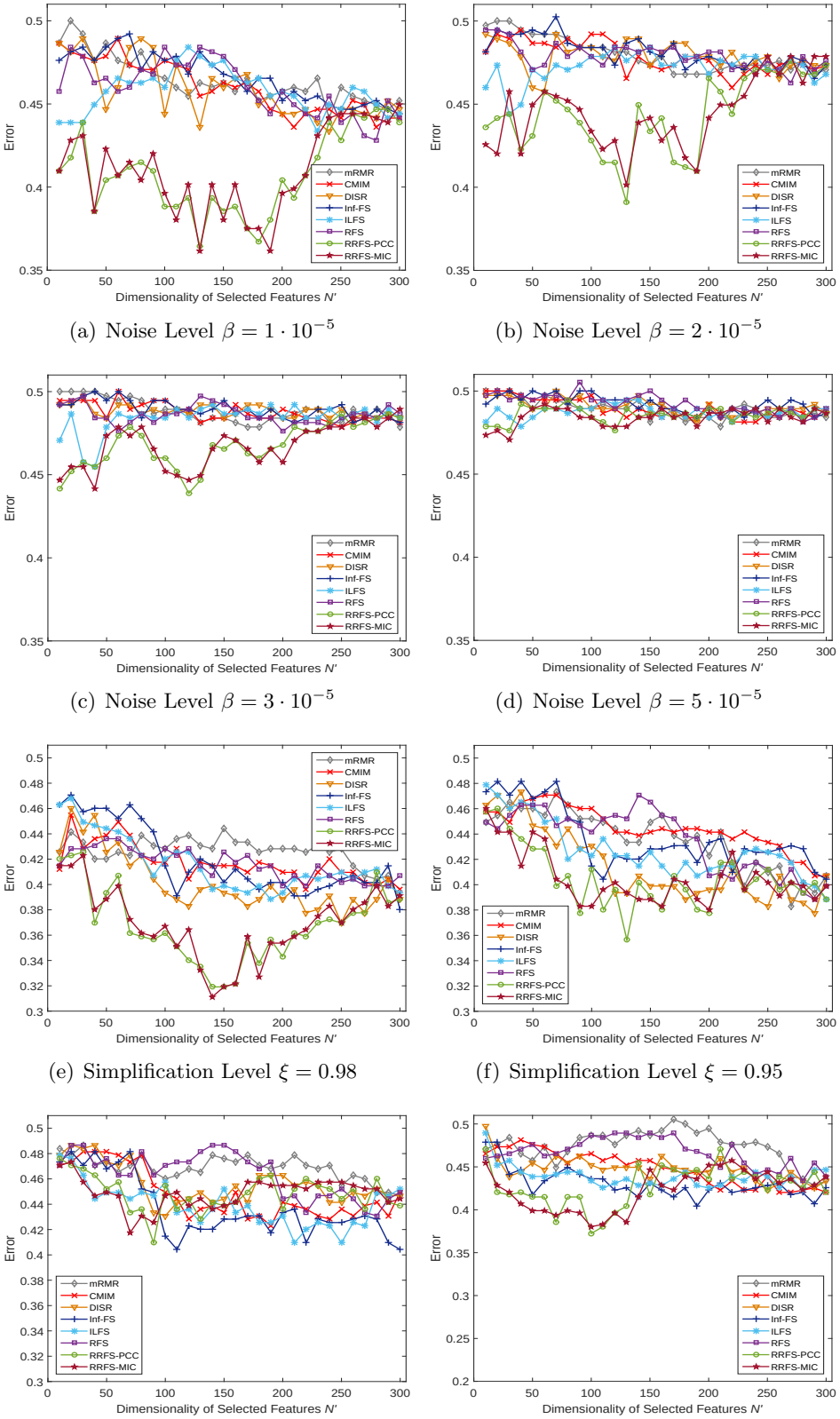


(e) Simplification Level $\xi = 0.98$ (f) Simplification Level $\xi = 0.95$



(g) Simplification Level $\xi = 0.9$ (h) Simplification Level $\xi = 0.8$

Figure 5.7: Median values of the detection errors when the information was hidden in 3D objects by the MLS algorithm, proposed in [Chao et al., 2009], using the steganalyzers trained over the feature subsets selected by different feature selection algorithms, where the results are calculated over 10 different splits of the training/testing sets.



(a) Noise Level $\beta = 1 \cdot 10^{-5}$

(b) Noise Level $\beta = 2 \cdot 10^{-5}$

(c) Noise Level $\beta = 3 \cdot 10^{-5}$

(d) Noise Level $\beta = 5 \cdot 10^{-5}$

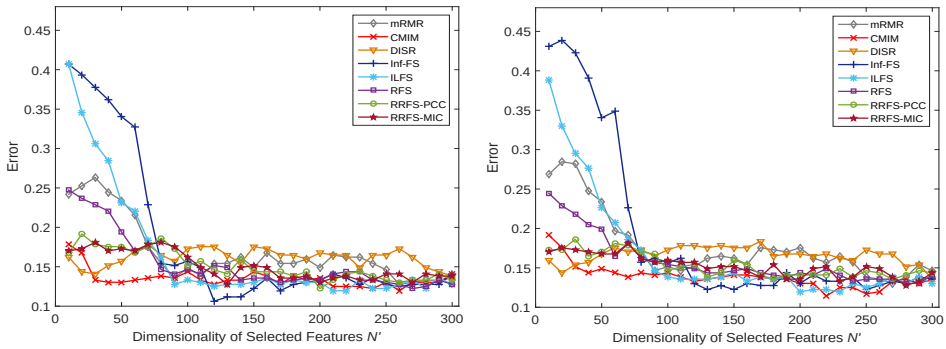
(e) Simplification Level $\xi = 0.98$

(f) Simplification Level $\xi = 0.95$

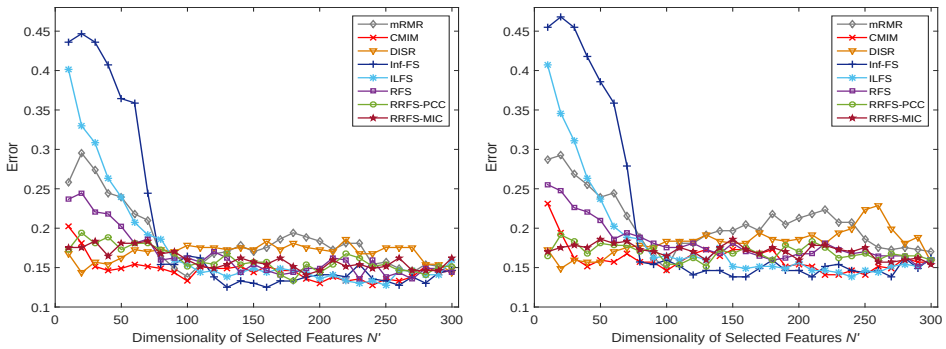
(g) Simplification Level $\xi = 0.9$

(h) Simplification Level $\xi = 0.8$

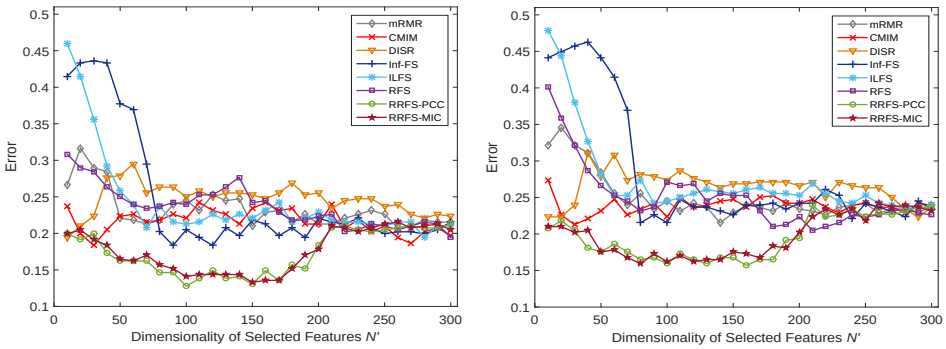
Figure 5.8: Median values of the detection errors when the information was hidden in 3D objects by the WHC algorithm, proposed in [Wang et al., 2008], using the steganalyzers trained over the feature subsets selected by different feature selection algorithms, where the results are calculated over 10 different splits of the training/testing sets.



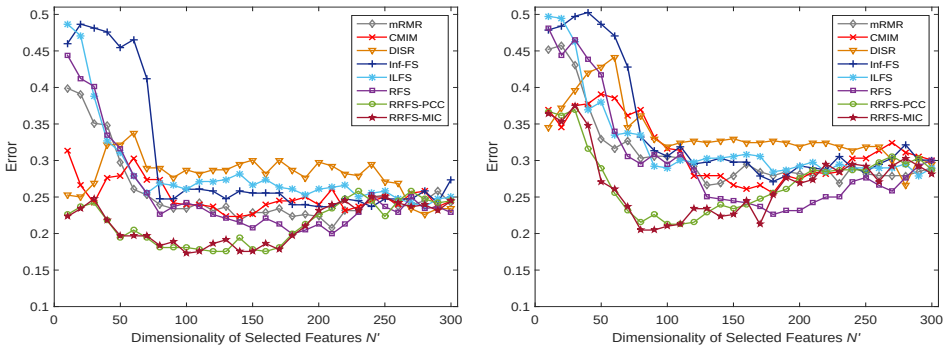
(a) Noise Level $\beta = 1 \cdot 10^{-5}$ (b) Noise Level $\beta = 2 \cdot 10^{-5}$



(c) Noise Level $\beta = 3 \cdot 10^{-5}$ (d) Noise Level $\beta = 5 \cdot 10^{-5}$

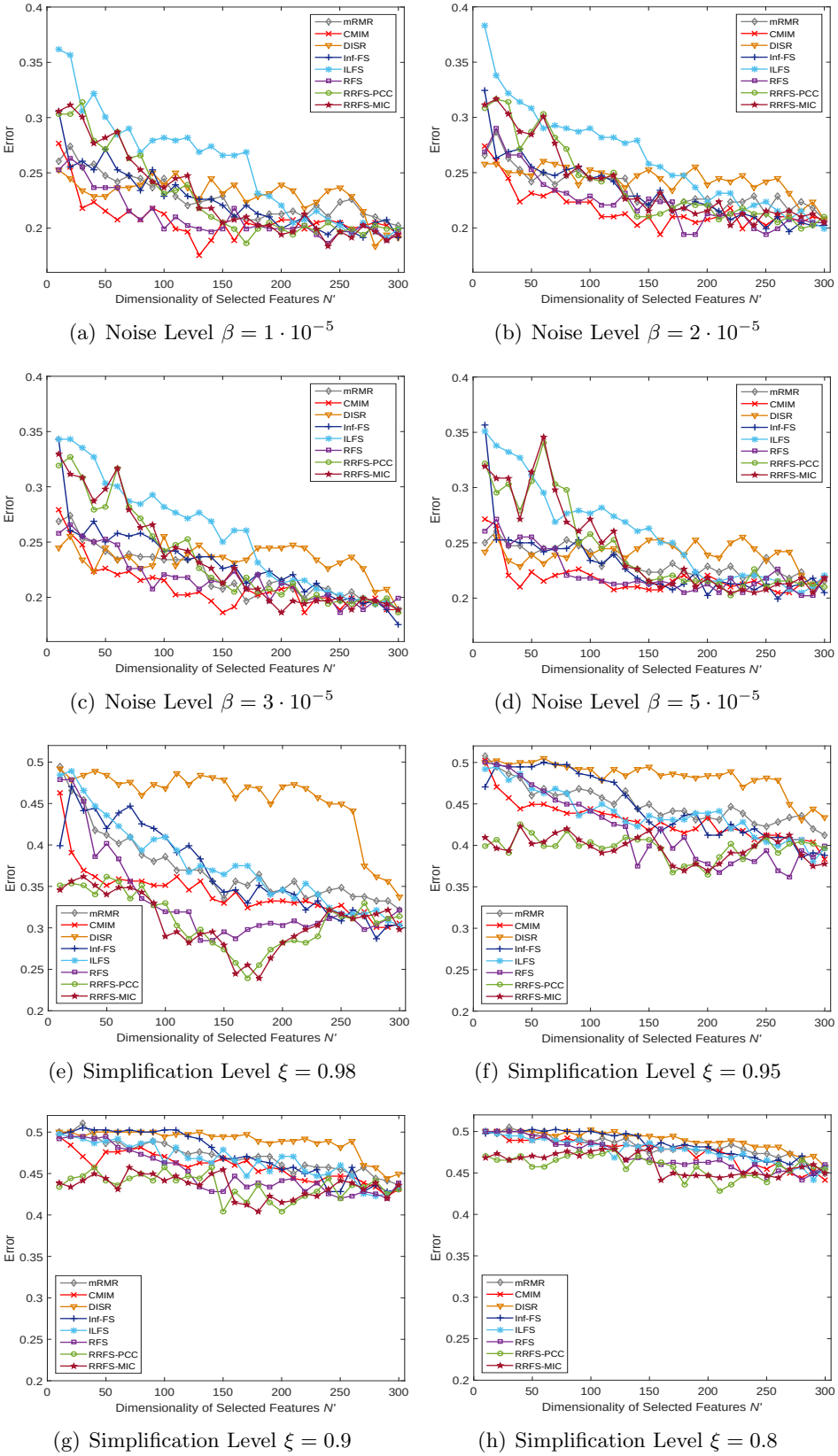


(e) Simplification Level $\xi = 0.98$ (f) Simplification Level $\xi = 0.95$



(g) Simplification Level $\xi = 0.9$ (h) Simplification Level $\xi = 0.8$

Figure 5.9: Median values of the detection errors when the information was hidden in 3D objects by the MRS algorithm, proposed in [Cho et al., 2007], using the steganalyzers trained over the feature subsets selected by different feature selection algorithms, where the results are calculated over 10 different splits of the training/testing sets.



(a) Noise Level $\beta = 1 \cdot 10^{-5}$

(b) Noise Level $\beta = 2 \cdot 10^{-5}$

(c) Noise Level $\beta = 3 \cdot 10^{-5}$

(d) Noise Level $\beta = 5 \cdot 10^{-5}$

(e) Simplification Level $\xi = 0.98$

(f) Simplification Level $\xi = 0.95$

(g) Simplification Level $\xi = 0.9$

(h) Simplification Level $\xi = 0.8$

Figure 5.10: Median values of the detection errors when the information was hidden in 3D objects by the SRW algorithm, proposed in [Yang et al., 2017b], using the steganalyzers trained over the feature subsets selected by different feature selection algorithms, where the results are calculated over 10 different splits of the training/testing sets.

in [Chao et al., 2009] and [Wang et al., 2008], respectively, under the CSM paradigm. As it can be observed from these figures, the RRFS-PCC and RRFS-MIC algorithms achieve rather similar results, which indicates that the dependence between the 3D features and the class label is relatively linear. The detection error of the steganalyzer using RRFS-PCC or RRFS-MIC usually decreases first and then increases as the dimensionality of the feature subset N' grows. This is because the earlier selected features have relatively strong robustness and high relevance, but the later selected ones are not that robust to the variation of the cover source. Meanwhile, when compared to the other feature selection algorithms, the RRFS-PCC and RRFS-MIC show better performance in most of the cases because lower minimum errors are achieved by using them during feature selection. The advantage of the RRFS-PCC and RRFS-MIC algorithms over other feature selection algorithms are larger when the CSM is due to lower levels of additive noise and mesh simplification. However, the optimal dimensionality of the feature subset is not consistent in all CSM cases. When addressing the CSM paradigm by assuming the same type of transformation, but of various levels of intensity, the optimal values of the feature subsets' dimensionality are rather close. For example, in the steganalysis of MLS under the CSM paradigm due to additive noise, the minimum errors are often obtained when the dimensionality of the selected feature subset is between 50 and 80. Nevertheless, as shown in Figure 5.7, when the CSM is due to mesh simplification, the optimal value of N' is usually around 100.

Figures 5.9 and 5.10 illustrate the steganalysis results when considering the watermarking methods, MRS and SRW, proposed in [Cho et al., 2007] and [Yang et al., 2017b], respectively, under the CSM paradigm. When considering the CSM due to additive noise, most of the feature selection algorithms show similar performance. As the dimensionality of the selected feature increases, the detection error decreases until it eventually becomes stable. This happens because the steganalyzers are not seriously influenced by the CSM due to additive noise when identifying stego-objects produced by MRS and SRW, which is validated in Figures 5.4(c) and (d). The RRFS-PCC and RRFS-MIC algorithms show better performance than the other algorithms under the CSM paradigm due to mesh simplification. In Figures 5.9(e)-(h), the detection errors using RRFS-PCC and RRFS-MIC are relatively constant, when the dimensionality of the feature subset N' is in the range between 80 and

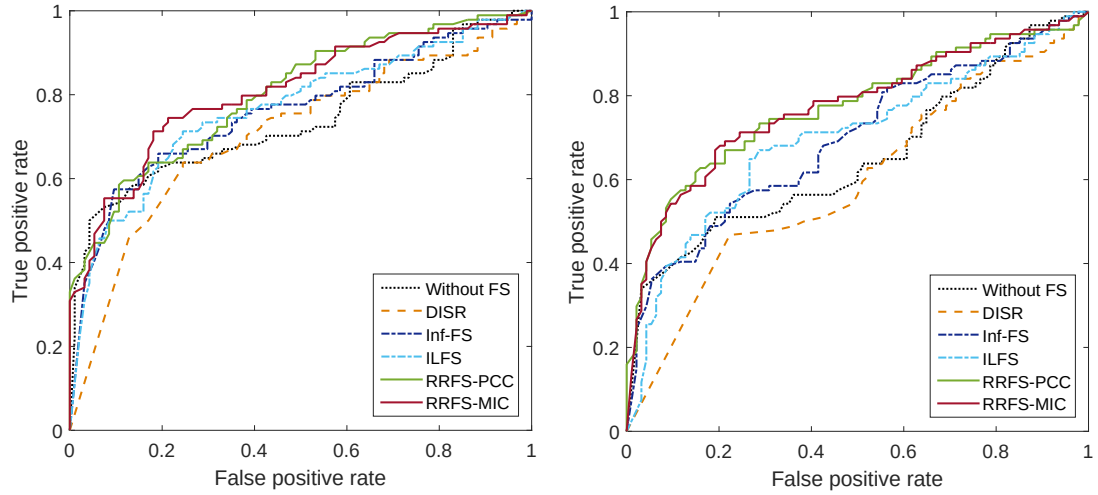
150. However, in Figures 5.10(e)-(h), the detection errors using RRFS-PCC and RRFS-MIC achieve the minimum when the dimensionality of the selected feature subset is between 160 and 200.

According to the Figure 5.7, 5.8, 5.9 and 5.10, the better results achieved by the RRFS algorithm when compared to the RFS indicates that considering the robustness of the features to the variation of cover source is essential when addressing the generalization of the steganalyzer under the CSM paradigm.

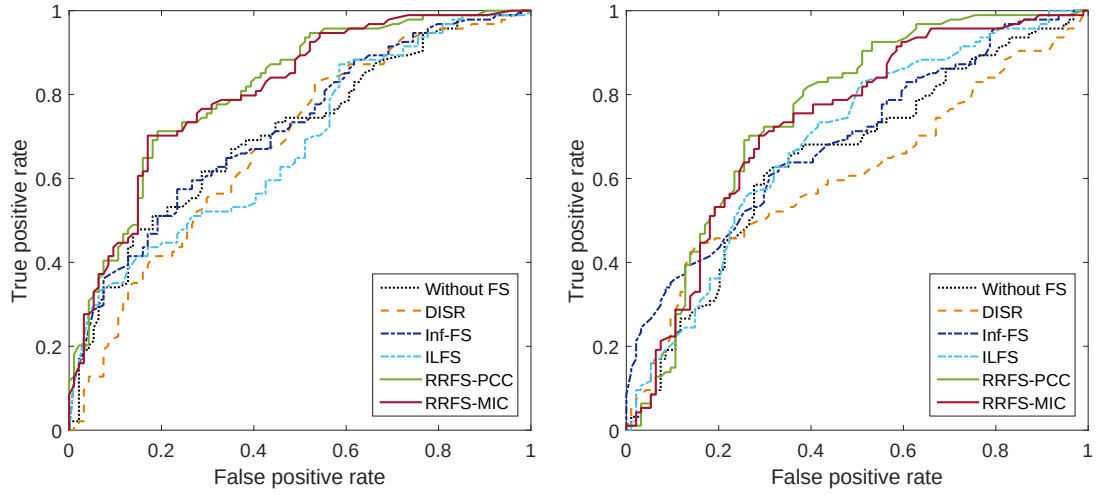
In the following we provide the Receiver Operating Characteristic (ROC) curves for the steganalysis results in the CSM scenarios after applying the feature selection algorithms or without considering Feature Selection (FS), in Figures 5.11, 5.12, 5.13 and 5.14. We consider detecting the stego-objects produced by MLS and WHC in the CSM by considering generating new cover-objects through additive noise with amplitude defined by $\beta \in \{1 \cdot 10^{-5}, 3 \cdot 10^{-5}\}$ and mesh simplification at the level of $\xi \in \{0.98, 0.9\}$. Since the steganalysis results of MRS and SRW tend to be rather poor under the CSM due to mesh simplification, we consider the CSM scenarios due to mesh simplification at the level of $\xi \in \{0.98, 0.9\}$ when detecting the stego-objects produced by MRS and SRW. When the steganalysis is carried out without feature selection, the whole feature set, WAL304, is used to train the steganalyzers. In this case we consider various feature selection algorithms, such as DISR, Inf-FS and ILFS, which have shown relatively good performance. In terms of the dimensionality of the selected feature subset, N', for all the feature selection algorithms, we set N' as 90, 130, 100 and 170, when detecting the stego-objects produced by MLS, WHC, MRS and SRW, respectively. The value of N' is decided according to the overall performance of the proposed feature selection algorithms in all CSM scenarios shown in Figures 5.7, 5.8, 5.9 and 5.10.

When detecting the stego-objects produced by the information hiding algorithms, in each case considered for the analysis under the CSM paradigm, the steganalysis results after using different feature selection algorithms or without FS are calculated based on one identical split of training/testing set. It can be observed from Figures 5.11, 5.12, 5.13 and 5.14 that the proposed RRFS-PCC and RRFS-MIC algorithms show better performance than the other feature selection algorithms in most of the case. Moreover, the proposed algorithms show improvement in the 3D steganalysis results, in the context of CSM problem, when compared

to using the whole feature set. However, the advantage of using feature selection over without using it is not very clear in the results shown in Figures 5.12 (d) and 5.14 (d).



(a) Noise Level $\beta = 1 \cdot 10^{-5}$ (b) Noise Level $\beta = 3 \cdot 10^{-5}$

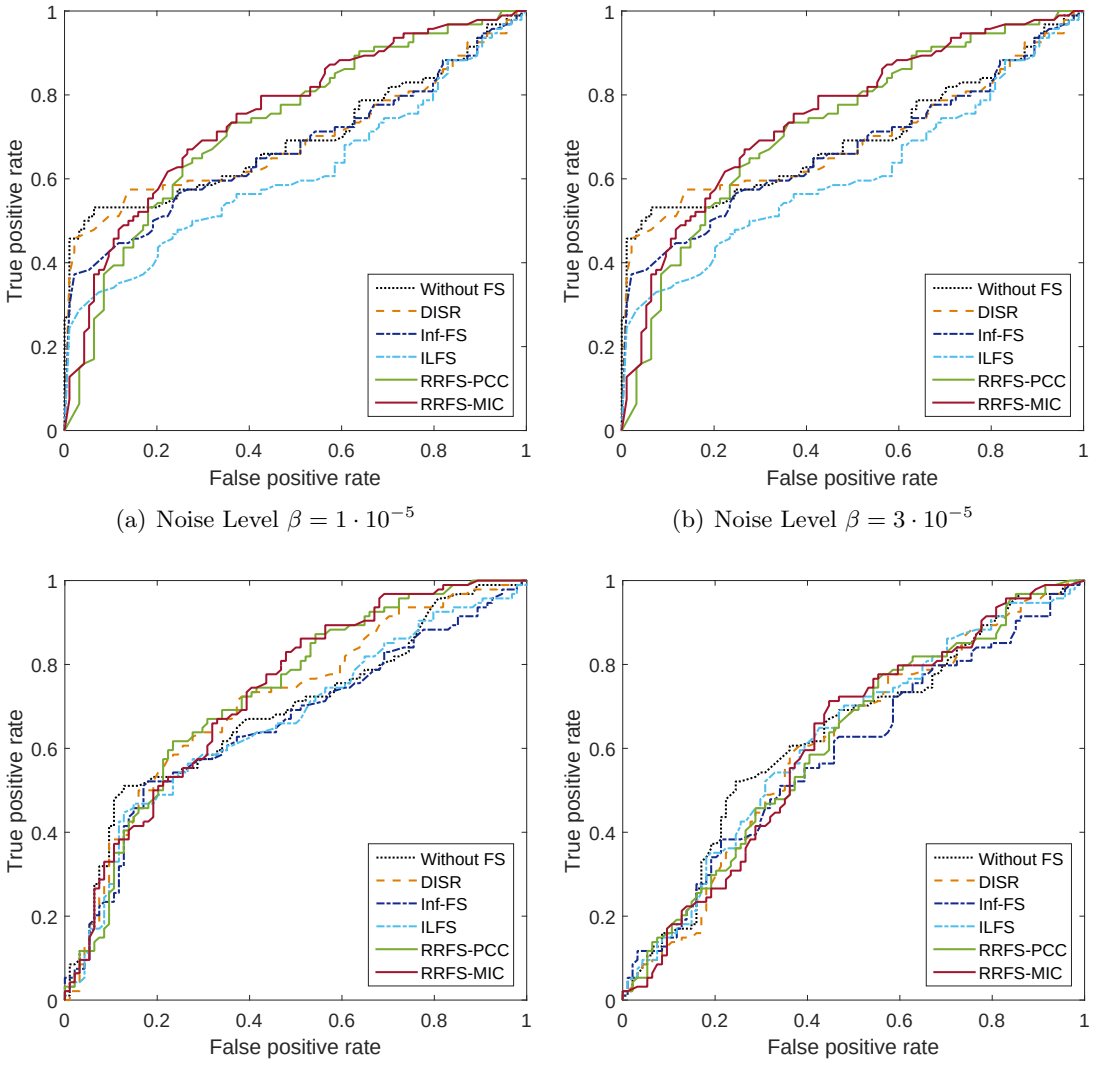


(c) Simplification Level $\xi = 0.98$ (d) Simplification Level $\xi = 0.9$

Figure 5.11: ROC curves for the steganalysis results when the information is hidden in 3D objects by the MLS under the CSM paradigm after applying the feature selection algorithms or without the Feature Selection.

5.4.4 Analyzing the selection of various categories of 3D features

In the following, we analyze the contribution of various categories of features that are selected by the proposed RRFS-PCC algorithm in the CSM scenarios of 3D steganalysis. Firstly, we categorize the steganalytic features according to their characteristics, as being either



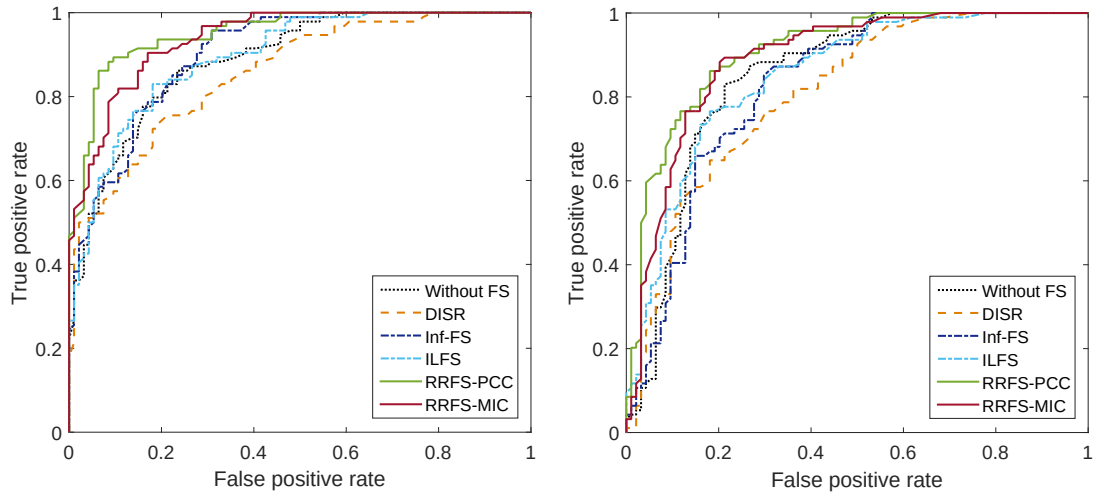
(a) Noise Level $\beta = 1 \cdot 10^{-5}$

(b) Noise Level $\beta = 3 \cdot 10^{-5}$

(c) Simplification Level $\xi = 0.98$

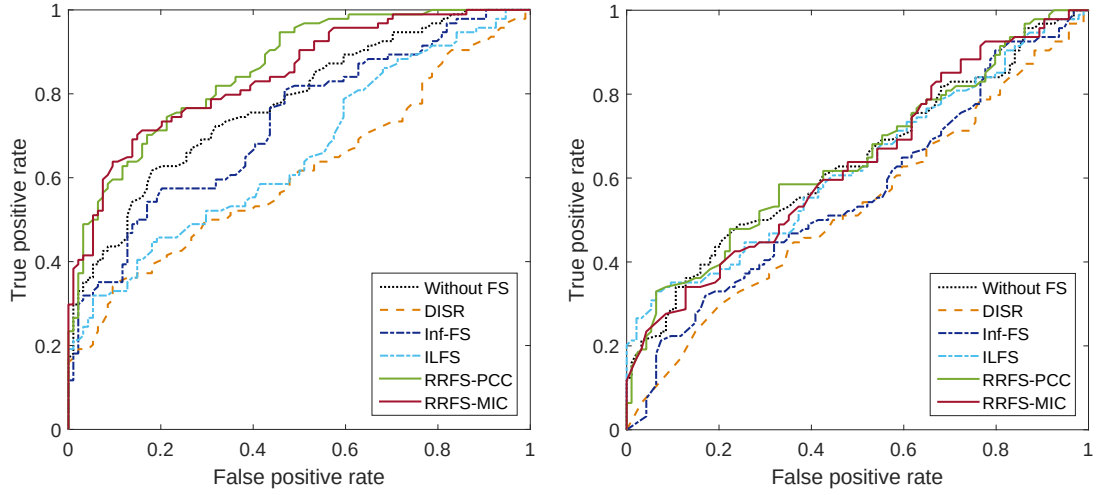
(d) Simplification Level $\xi = 0.9$

Figure 5.12: ROC curves for the steganalysis results when the information is hidden in 3D objects by the WHC under the CSM paradigm after applying the feature selection algorithms or without the Feature Selection.



(a) Simplification Level $\xi = 0.98$ (b) Simplification Level $\xi = 0.9$

Figure 5.13: ROC curves for the steganalysis results when the information is hidden in 3D objects by the MRS under the CSM paradigm after applying the feature selection algorithms or without the Feature Selection.



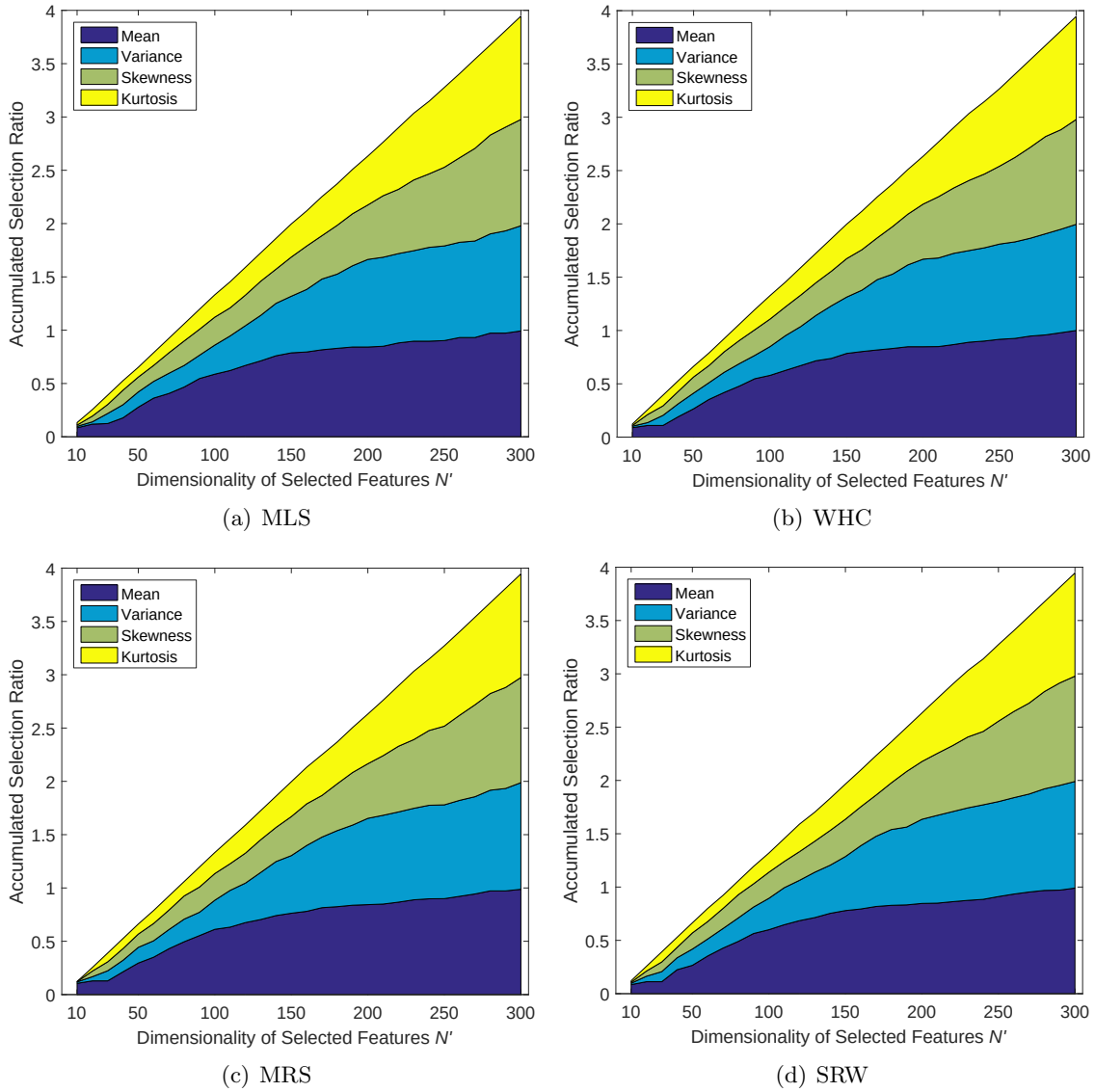
(a) Simplification Level $\xi = 0.98$ (b) Simplification Level $\xi = 0.9$

Figure 5.14: ROC curves for the steganalysis results when the information is hidden in 3D objects by the SRW under the CSM paradigm after applying the feature selection algorithms or without the Feature Selection.

statistic or geometrical in nature. Inside the former group of feature we define the features, according to the statistical moment they represent, as: mean, variance, skewness or kurtosis. In the latter group, we categorize the features by considering what kind of local geometry characteristic they reveal: LFS76 proposed in Chapter 3, and three subsets of WFS228 proposed in Chapter 4, which are Wavelet Feature Set from the Initial resolution mesh (WFS-I), Wavelet Feature Set from the Lower resolution mesh (WFS-L) and Wavelet Feature Set from the Higher resolution mesh (WFS-H). For each of these feature categories we calculate the percentage of the features being selected by the RRFS-PCC from the given pool of features when training the steganalyzers aiming to find the information hidden in 3D objects by the algorithms, MLS [Chao et al., 2009], WHC [Wang et al., 2008], MRS [Cho et al., 2007], and SRW [Yang et al., 2017b], under the CSM paradigm due to the additive noise and mesh simplification. The final selection ratio of every feature category is calculated as the average of 10 independent splits of the training/testing data.

Figures 5.15 and 5.16 depict the selection ratios of all feature categories when the dimensionality of the feature subset selected by RRFS-PCC algorithm varies from 10 to 300 with a step of 10, in the context of mitigating the CSM problem. As it can be observed from Figure 5.15 when N' is small, that the first order moments (means) of features are much more likely to be selected than their second order moments (variances), or other higher order moments of the features, such as their skewness and kurtosis. Then, the differences between the selection ratios of different feature categories declines as the N' increases. This result indicates that the mean-originated features are the most robust statistical feature, followed by the variance-originated ones, when considering the context of the CSM paradigm. The higher-order moments of the 3D shape data are more dramatically changed than the lower-order ones under the transformations considered for testing the CSM problem.

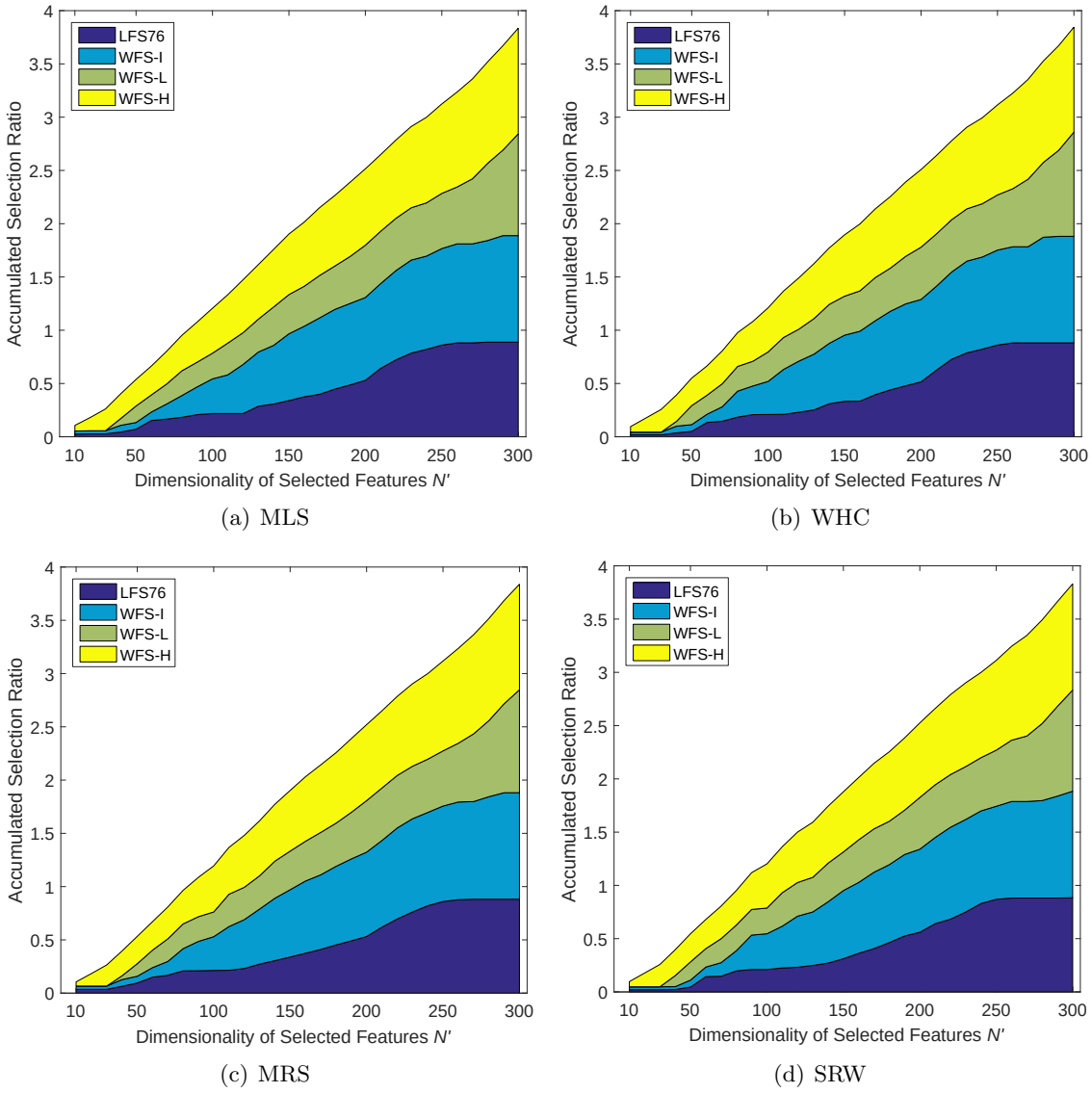
The selection ratios of the features of 4 geometrical categories are shown in Figure 5.16. It can be observed from Figure 5.16 that the RRFS-PCC algorithm most likely would select the WFS-H features when N' is ranged from 10 to 30, which implies that they have the strongest robustness. This is because the WFS-H features are extracted from the higher resolution mesh, where the influence of the transformation applied on the initial resolution meshes is weakened. The WFS-L features that extracted from the lower resolution mesh are



(a) MLS (b) WHC

(c) MRS (d) SRW

Figure 5.15: The accumulated selection ratios of the features, as being discriminative between the stego-objects, created by using the embedding methods, MLS [Chao et al., 2009], WHC [Wang et al., 2008], MRS [Cho et al., 2007] and SRW [Yang et al., 2017b] and their corresponding cover-objects, by using RRFS-PCC, under the specific CSM scenarios. The features correspond to the moments of the shape data they characterize, such as the mean, variance, skewness, kurtosis.



(a) MLS (b) WHC

(c) MRS (d) SRW

Figure 5.16: The accumulated selection ratios of the features, as being discriminative between the stego-objects, created by using the embedding methods, MLS [Chao et al., 2009], WHC [Wang et al., 2008], MRS [Cho et al., 2007] and SRW [Yang et al., 2017b] and their corresponding cover-objects, by using RRFS-PCC, under the specific CSM scenarios. The features correspond to the subsets of WAL304, such as LFS76, WFS-I, WFS-L, and WFS-H.

not selected until N' reaches 40. The lower resolution mesh consists of much less vertices and edges, which makes the WFS-L features more sensitive to the distortions on the mesh shape caused by the additive noise and simplification. The selection ratio of the LFS76 features increases fast when N' is 50 and 60, but then increases at a low speed. We believe it is because only a small number of the features in LFS76 are of enough robustness to be selected in the early selection stage. The selection ratio of the WFS-I features extracted from the initial resolution mesh rockets up since N' reaches 70 and becomes the highest one among the four when N' is between 100 and 200. It is because a larger population of the WFS-I features are of a moderate level of robustness than the other three groups of features.

5.5 Conclusion

This chapter proposes a solution for the cover source mismatch problem in the context of 3D steganalysis. According to the CSM paradigm, we consider that the objects investigated during the testing stage are significantly different from those used during the training. A feature selection algorithm, called the robustness and relevance-based feature selection, is proposed in this chapter. The proposed algorithm employs either the Pearson correlation coefficient or the Mutual Information Criterion in order to define the relevance of each feature to the class label. The robustness of the feature to the variations of the cover source is evaluated at the same time, leading to the selection of a robust feature subset. In this study we consider mesh simplification and additive noise for transforming the cover objects when testing the steganalyzer under the CSM paradigm. During the experimental analysis we consider four different information hiding methods, including a high capacity embedding method and a very recent method which embeds watermarks that cannot be detected through usual steganalytic methods. The proposed methodology is shown to choose a better feature set, than those considered by other studies, when addressing the CSM problem. In future research, it is necessary to improve the way of evaluating the features' robustness, avoiding the requirement of the pairwise relationship between the objects when calculating the Pearson correlation coefficient, in order to make the proposed feature selection algorithm suitable in a more universal CSM paradigm.

Chapter 6

Conclusion

This chapter summarizes the main contributions of this thesis, as well as its limitations and possible directions for future research.

6.1 Contributions

The research work reported in this thesis, extends the set of features used for 3D stegalysis. The new feature set, called LFS76, which is more discriminative and of lower dimensionality than an existing feature set for 3D steganalysis. Then, we proposed the 3D wavelet-based feature set, WFS228, in order to improve the steganalysis results for the 3D wavelet-based information hiding algorithms. Furthermore, we proposed a feature selection algorithm, considering both the features' relevance and robustness, in order to mitigate the negative effect of the cover source mismatch on the performance of the 3D steganalytic approaches. In the following we briefly outline the contributions presented in each chapter.

In Chapter 3, we presented a new 76-dimensional localised feature set, called LFS76, for 3D steganalysis. We reduced the dimensionality of the existing 3D steganalytic features, YANG208, from 208 to 40, obtaining the YANG40 features. In addition to these features, we proposed to consider the vertex normal, Gaussian curvature and curvature ratio as geometric features for 3D steganalysis. Moreover, the geometric features such as the vertex position and edge vectors represented in the spherical coordinate system are included in the LFS76 feature set. We tested the performance of the proposed LFS76 feature set by detecting

the stego-objects generated by six information hiding algorithms when the steganalyzers were trained using FLD ensemble or QDA classifiers. The experimental results show that the LFS76 was more discriminative than YANG208. We also analyzed the efficiency of the components of LFS76 using the measurement of the Pearson correlation coefficient between the feature vector and the class label in order to have a deeper understanding of the features used in 3D steganalysis.

In Chapter 4, we introduced the 228-dimensional 3D wavelet-based feature set, WFS228, for 3D steganalysis. We use the 3D wavelet multiresolution analysis on mesh representations of objects, producing a higher and a lower resolution mesh, respectively. A set of vectors relate each of these meshes with their higher resolution mesh, respectively. The proposed 3D wavelet features are extracted from the vectors modeling the transitions between the given mesh and its higher and lower resolution counterparts, respectively. Furthermore, we provided the way of deriving the steganalytic feature vectors from the geometric features. Consequently, we propose the WFS228 feature set which shows better performance than the existing steganalytic feature sets for the steganalysis of two 3D wavelet-based embedding algorithms. WFS228 also contributed to the improvement of the steganalysis for other six information hiding algorithms when training the steganalyzer over the combination of WFS228 and LFS76.

In Chapter 5, we presented the robust and relevance-based feature selection algorithm in order to address the cover source mismatch problem in 3D steganalysis. During the feature selection stage, we consider both the features' robustness to the variation of the cover source as well as the features' relevance to the class label. We also analyzed the trade-off between the features' robustness and relevance during selection. The variation of the shapes is produced by adding noise to the mesh and by mesh simplification. In addition, the proposed RRFS algorithm shows better performance than other feature selection algorithms when considering the CSM paradigm in 3D steganalysis. We have also studied the robustness and relevance of groups of features, with each group defined by a specific geometric property.

6.2 Limitations and future work

In this section, we discuss the limitations of the proposed methodology and provide some ideas for future work.

One limitation lies in the way how we model statistically the feature vectors. In this study, the steganalytic feature vectors are considered as the four statistical moments of the logarithms of the differences between the geometric features of the original mesh and its smoothed version. Nevertheless, the first four statistical moments do not contain the whole information contained in the data and a better statistical representation may be used.

Despite the good results achieved in 3D steganalysis, further improvement is necessary. It is shown in Chapter 4 that, although the proposed WFS228 feature set achieve better performance than the existing feature sets when detecting the embedding changes produced by the two 3D wavelet-based embedding algorithms, most of the detection errors for the two 3D wavelet-based embedding algorithms are still above 20%, which would require further improvements. Given the promising results obtained by using the deep learning methods in the image steganalysis, a 3D steganalytic approach based on the deep learning framework may provide a better performance. Nevertheless, there are challenges when using the deep learning for steganalysis. Deep learning would require a huge training set (of millions of training examples) and significantly more computation power.

In Chapter 5, the features' robustness to the variation of cover source is derived from the Pearson correlation coefficient between the features from the original cover source and those from the transformed cover source. The proposed approach requires a pairwise relationship between the objects in the original cover source and their transformation versions in the other cover source. However, if this kind of pairwise relationship does not exist between two completely different cover sources, the proposed method of calculating the features' robustness cannot be properly implemented. In order to improve over this limitation, a more universal method of statistical comparison could be developed with the aim of measuring the features' robustness to the variation of the cover source.

List of Symbols

$\mathbf{P}_n$	the Hamiltonian path
Δ	the interval parameter for HPQ
s	the number of sub-interval for HPQ
p	the threshold parameter for WHC
l_{av}	the average length of the WCVs' support edges
ϵ_{hc}	a controlling parameter for WHC
Δ_θ	a quantization step for WFR
K	the number of bins for SRW
$\mathcal{O}$	the original 3D object
$\mathcal{O}'$	the smoothed 3D object
$\mathbf{\Phi}$	the absolute difference between the geometric feature extracted from the object $\mathcal{O}$ and its smoothed version $\mathcal{O}'$
$\mathbf{X}$	the finally extracted feature vector
y	the class label of the object
V	the vertices set of the object $\mathcal{O}$
F	the face set of the object $\mathcal{O}$
E	the edge set of the object $\mathcal{O}$
$\mathcal{N}(v_i)$	the neighbourhood of vertex v_i
$e_{(i,j)}$	the edge connecting vertices v_i and v_j
v_i	the ith vertex in the object $\mathcal{O}$
v'_i	the ith vertex in the smoothed object $\mathcal{O}'$

λ	a scale factor used in the Laplacian smoothing
w_{ij}	the weights defined in the Laplacian smoothing
ϕ_i	the ith elements of $\mathbf{\Phi}$
$v_{x,c}(i)$	the x-coordinate of v_i in Cartesian coordinate systems
$v_{x,l}(i)$	the x-coordinate of v_i in Laplacian coordinate systems
$\|\mathbf{v}_c(i)\|$	the vector norm of v_i in Cartesian and Laplacian coordinates
$\|\mathbf{v}_l(i)\|$	the vector norms of v_i in Cartesian and Laplacian coordinates
e_i	the ith edge in E
$\theta_{e(i)}$	the dihedral angle corresponding to edge e_i
$\vec{N}_{f_i}$	the face normal of face f_i in the object $\mathcal{O}$
f_i	the ith face in F of the object $\mathcal{O}$
$\vec{N}_{f'_i}$	the face normal of face f'_i in the smoothed object $\mathcal{O}'$
f'_i	the ith face in F' of the smoothed object $\mathcal{O}'$
$\vec{N}_{v'_i}$	the vertex normal of v'_i in the smoothed object $\mathcal{O}'$
$\vec{N}_{v_i}$	the vertex normal of v_i in the object $\mathcal{O}'$
$A(f_j)$	the area of the jth face that containing vertex v_i
K_1	the minimum principal curvature
K_2	the maximum principal curvature
K_G	the Gaussian curvature
K_r	the curvature ratio
R	the radial distance in the spherical coordinate system
θ	the azimuth angle in the spherical coordinate system
φ	the elevation angle in the spherical coordinate system
$\mathbf{v}$	the Cartesian coordinates of the vertex v
v_x	the x-coordinate of the vertex v
v_y	the y-coordinate of the vertex v
v_z	the z-coordinate of the vertex v
$K_\theta(e_{(i,j)})$	the azimuth angle component of the edge $e_{(i,j)}$
$K_\varphi(e_{(i,j)})$	the elevation angle component of the edge $e_{(i,j)}$

$K_R(e_{(i,j)})$	the radial distance component of the edge $e_{(i,j)}$
n_{thr}	a robustness parameter for SRW
α	a robustness parameter for MRS and VRS
Δk	a incremental step for MRS and VRS
x_i	the ith feature of a given feature set $\mathbf{X}$
N	the dimensionality of the feature set $\mathbf{X}$
$\rho(x_i, y)$	the Pearson correlation coefficient between x_i and y
$cov(x_i, y)$	the covariance of x_i and y
σ_{x_i}	the standard deviation of x_i
$\mathcal{O}^l$	a lower resolution mesh of $\mathcal{O}$
$\mathcal{O}^h$	a higher resolution mesh of $\mathcal{O}$
$\mathbf{e}_{(i,j)}$	the edge vector of the initial mesh
$\mathbf{e}^*_{(i,j)}$	the flapping edge vector of the initial mesh
$\mathbf{w}^l_{(i,j)}$	the wavelet coefficient vector of the lower resolution mesh
$\mathbf{e}^l_{(i,j)}$	the edge vector of the lower resolution mesh
$\alpha_{(i,j)}$	the angle between $\mathbf{w}^l_{(i,j)}$ and $\mathbf{e}^l_{(i,j)}$
$\rho^l_{(i,j)}$	the ratio between $\|\mathbf{w}^l_{(i,j)}\|$ and $\|\mathbf{e}^l_{(i,j)}\|$
$\mathbf{w}^h_{(i,j)}$	the wavelet coefficient vector of the higher resolution mesh
$\mathbf{e}^h_{(i,j)}$	the edge vector of the higher resolution mesh
$\beta_{(i,j)}$	the angle between $\mathbf{w}^h_{(i,j)}$ and $\mathbf{e}_{(i,j)}$
$\rho^h_{(i,j)}$	the ratio between $\|\mathbf{w}^h_{(i,j)}\|$ and $\|\mathbf{e}_{(i,j)}\|$
$\bar{\mathbf{w}}^h_{(i,j)}$	the averaged neighboring WCV of the higher resolution mesh
${\mathbf{w}^h_{(i,j)}}'$	the difference between $\mathbf{w}^h_{(i,j)}$ and $\bar{\mathbf{w}}^h_{(i,j)}$
$\theta_{(i,j)}$	the angle between $\mathbf{w}^h_{(i,j)}$ and $\bar{\mathbf{w}}^h_{(i,j)}$
$\mu^M_{(i,j)}$	the mean of the angles between WCV and its neighboring WCVs of higher resolution mesh
$\mu^V_{(i,j)}$	the variance of the angles between WCV and its neighboring WCVs of higher resolution mesh

$\kappa^M_{(i,j)}$	the mean of the differences between the norms of WCV and its neighboring WCVs of higher resolution mesh
$\kappa^V_{(i,j)}$	the variance of the differences between the norms of WCV and its neighboring WCVs of higher resolution mesh
$\triangle$	a triangle
$\mathcal{N}(\mathbf{w}^h_{(i,j)})$	the set of the neighboring WCVs of $\mathbf{w}^h_{(i,j)}$
$\delta_{k,l}$	the angle between $\mathbf{w}^h_{(i,j)}$ and its neighboring WCV, $\mathbf{w}^h_{(k,l)}$
g_t	the tth scalar geometric feature from the original mesh
g'_t	the tth scalar geometric feature from the smoothed mesh
$\mathbf{g}_t$	the tth vectorial geometric feature from the original mesh
$\mathbf{g}'_t$	the tth vectorial geometric feature from the smoothed mesh
$\mathbf{g}_{t,x}$	the x-component of the vector $\mathbf{g}_t$ in the Cartesian coordinate system
lg	the common logarithm
I	the feature index set
$MI(x_i; y)$	the mutual information between the x_i and y
$x_{i,j}$	the ith feature extracted from the jth transformed cover source
M	the number of various transformation
$p(x_i, y)$	the joint probability distribution function of x_i and y
$p(x_i)$	the marginal probability distribution functions of x_i
q	the percentile rank
θ_q	the qth percentile of set $\{r_i \vert i = 1, 2, \ldots, N\}$
r_i	the robustness of the feature x_i
τ	the trade-off parameter between the robustness and the relevance of the features
$\mathcal{F}'$	the indexes of the selected feature subset
ξ	a simplification ratio parameter
β	a parameter controlling the level of additive noise
N'	the dimensionality of the selected features

Abbreviations

3D	Three-dimensional
BBC	British Broadcasting Corporation
BOSS	Break Our Steganographic System
BPV	Bits Per Vertex
CLS	Calibrated Least Squares
CMIM	Conditional Mutual Information Maximization
CNN	Convolutional Neural Network
CSM	Cover Source Mismatch
DCT	Discrete Cosine Transform
DISR	Double Input Symmetrical Relevance
ES	Edge length in the Spherical coordinate system
FLD	Fisher Linear Discriminant
FS	Feature Selection
HPQ	Hamiltonian Path Quantization
HPQ-R	Hamiltonian Path Quantization in Radial coordiante
HUGO	Highly Undetectable steGO
ILFS	Infinite Latent Feature Selection
Inf-FS	Infinite Feature Selection
ISO	International Organization of Standardization
JPEG	Joint Photographic Experts Group
LFS	Local Feature Set
LSB	Least Significant Bit

MEP	Macro Embedding Procedure
MIC	Mutual Information Criterion
MLEP	Multi-Level Embedding Procedure
MLS	Multi-Layer Steganography
mRMR	min-Redundancy and Max-Relevancy
MRS	Mean of Radial distances in Spherical coordinate system
MSER	Mean Square Error Ratio
OEAP	Online Ensemble Average Perceptron
OOB	Out-Of-Bag
PCA	Principal Component Analysis
PCC	Pearson Correlation Coefficient
QDA	Quadratic Discriminant Analysis
QIM	Quantization Index Modulation
RFS	Relevance-based Feature Selection
ROC	Receiver Operating Characteristic
RRFS	Robustness and Relevance-based Feature Selection
SCS	Spherical Coordinate System
SPAM	Subtractive Pixel Adjacency Model
SRM	Spatial Rich Models
SRW	Steganalysis-Resistant Watermarking
SVM	Support Vector Machine
TLU	Truncated Linear Unit
TSQ	Triangle Similarity Quadruple
TVR	Tetrahedral Volume Ratio
UED	Uniform Embedding Distortion
UNIWARD	UNIversal WAvelet Relative Distortion
VRS	Variance of Radial distances in Spherical coordinate system
VS	Vertices' Spherical coordinates
WAL	Wavelet And Local feature set
WCV	Wavelet Coefficient Vector
WFR	Wavelet-based FRagile

WFS Wavelet Feature Set

WFS-H Wavelet Feature Set from the Higher resolution mesh

WFS-I Wavelet Feature Set from the Initial resolution mesh

WFS-L Wavelet Feature Set from the Lower resolution mesh

WHC Wavelet-based High Capacity

WMA Wavelet Multiresolution Analysis

WOW Wavelet Obtained Weights

List of References

[Abdul-Rahman et al., 2013] Abdul-Rahman, H. S., Jiang, X. J., and Scott, P. J. (2013). Freeform surface filtering using the lifting wavelet transform. *Precision Engineering*, 37(1):187–202.

[Alface et al., 2007] Alface, P. R., Macq, B., and Cayre, F. (2007). Blind and robust watermarking of 3-D models: How to withstand the cropping attack? In *Proc. IEEE Int. Conf. Image Processing*, pages 465–468.

[Amat et al., 2010] Amat, P., Puech, W., Druon, S., and Pedeboy, J.-P. (2010). Lossless 3D steganography based on mst and connectivity modification. *Signal Processing: Image Communication*, 25(6):400–412.

[Bas et al., 2011] Bas, P., Filler, T., and Pevnỳ, T. (2011). Break our steganographic system: The ins and outs of organizing BOSS. In *Proc. Int. Workshop on Information Hiding, LNCS, vol. 6958*, pages 59–70.

[Battiti, 1994] Battiti, R. (1994). Using mutual information for selecting features in supervised neural net learning. *IEEE Transactions on Neural Networks*, 5(4):537–550.

[Bogomjakov et al., 2008] Bogomjakov, A., Gotsman, C., and Isenburg, M. (2008). Distortion-free steganography for polygonal meshes. *Computer Graphics Forum*, 27(2):637–642.

[Bollobás, 2013] Bollobás, B. (2013). *Modern graph theory*, volume 184. Springer Science & Business Media.

[Bors, 2006] Bors, A. G. (2006). Watermarking mesh-based representations of 3-D objects using local moments. *IEEE Transactions on Image Processing*, 15(3):687–701.

[Bors and Luo, 2013] Bors, A. G. and Luo, M. (2013). Optimized 3D watermarking for minimal surface distortion. *IEEE Transactions on Image Processing*, 22(5):1822–1835.

[Brown et al., 2012] Brown, G., Pocock, A., Zhao, M.-J., and Luján, M. (2012). Conditional likelihood maximisation: A unifying framework for information theoretic feature selection. *Journal of Machine Learning Research*, 13(Jan):27–66.

[Bureau and Dietrich, 2015] Bureau, P.-M. and Dietrich, C. (2015). Hiding in plain sight - advances in malware covert communication channels. In *Black Hat Europe*.

[Cao et al., 2012] Cao, Y., Zhao, X., and Feng, D. (2012). Video steganalysis exploiting motion vector reversion-based features. *IEEE Signal Processing Letters*, 19(1):35–38.

[Cayre and Macq, 2003] Cayre, F. and Macq, B. (2003). Data hiding on 3-D triangle meshes. *IEEE Transactions on Signal Processing*, 51(4):939–949.

[Chandrashekar and Sahin, 2014] Chandrashekar, G. and Sahin, F. (2014). A survey on feature selection methods. *Computers & Electrical Engineering*, 40(1):16–28.

[Chao et al., 2009] Chao, M.-W., Lin, C.-h., Yu, C.-W., and Lee, T.-Y. (2009). A high capacity 3D steganography algorithm. *IEEE Transactions on Visualization and Computer Graphics*, 15(2):274–284.

[Chaumont and Kouider, 2012] Chaumont, M. and Kouider, S. (2012). Steganalysis by ensemble classifiers with boosting by regression, and post-selection of features. In *Proc. IEEE Int. Conf. on Image Processing*, pages 1133–1136.

[Chen and Wornell, 2001] Chen, B. and Wornell, G. W. (2001). Quantization index modulation: A class of provably good methods for digital watermarking and information embedding. *IEEE Transactions on Information Theory*, 47(4):1423–1443.

[Chen and Shi, 2008] Chen, C. and Shi, Y. Q. (2008). JPEG image steganalysis utilizing both intrablock and interblock correlations. In *Proc. IEEE Int. Symposium on Circuits and Systems*, pages 3029–3032.

[Chen et al., 2009] Chen, X., Golovinskiy, A., and Funkhouser, T. (2009). A benchmark for 3D mesh segmentation. *ACM Transactions on Graphics*, 28(3):73:1–73:12.

[Cheng and Wang, 2006] Cheng, Y.-M. and Wang, C.-M. (2006). A high-capacity steganographic approach for 3D polygonal meshes. *The Visual Computer*, 22(9-11):845–855.

[Cho et al., 2007] Cho, J.-W., Prost, R., and Jung, H.-Y. (2007). An oblivious watermarking for 3-D polygonal meshes using distribution of vertex norms. *IEEE Transactions on Signal Processing*, 55(1):142–155.

[Cogranne and Fridrich, 2015] Cogranne, R. and Fridrich, J. (2015). Modeling and extending the ensemble classifier for steganalysis of digital images using hypothesis testing theory. *IEEE Transactions on Information Forensics and Security*, 10(12):2627–2642.

[Date et al., 1999] Date, H., Kanai, S., and Kishinami, T. (1999). Digital watermarking for 3-D polygonal model based on wavelet transform. In *Proc. ASME Design Engineering Technical Conf.*, pages 12–15.

[Denemark et al., 2014] Denemark, T., Sedighi, V., Holub, V., Cogranne, R., and Fridrich, J. (2014). Selection-channel-aware rich model for steganalysis of digital images. In *Proc. IEEE Int. Workshop on Information Forensics and Security*, pages 48–53.

[Denemark et al., 2016] Denemark, T. D., Boroumand, M., and Fridrich, J. (2016). Steganalysis features for content-adaptive JPEG steganography. *IEEE Transactions on Information Forensics and Security*, 11(8):1736–1746.

[Duda et al., 2012] Duda, R. O., Hart, P. E., and Stork, D. G. (2012). *Pattern Classification*. John Wiley & Sons.

[Dyn et al., 1990] Dyn, N., Levine, D., and Gregory, J. A. (1990). A butterfly subdivision scheme for surface interpolation with tension control. *ACM Transactions on Graphics*, 9(2):160–169.

[Eggers et al., 2003] Eggers, J. J., Bauml, R., Tzschoppe, R., and Girod, B. (2003). Scalar costa scheme for information embedding. *IEEE Transactions on Signal Processing*, 51(4):1003–1019.

[Farid, 2002] Farid, H. (2002). Detecting hidden messages using higher-order statistical models. In *Proc. IEEE Int. Conf. on Image Processing*, volume 2, pages II 905–II 908.

[Fleuret, 2004] Fleuret, F. (2004). Fast binary feature selection with conditional mutual information. *Journal of Machine Learning Research*, 5(11):1531–1555.

[Fridrich, 2004] Fridrich, J. (2004). Feature-based steganalysis for JPEG images and its implications for future design of steganographic schemes. In *Proc. Int. Workshop on Information Hiding*, pages 67–81.

[Fridrich et al., 2000] Fridrich, J., Du, R., and Long, M. (2000). Steganalysis of LSB encoding in color images. In *Proc. IEEE Int. Conf. on Multimedia and Expo*, volume 3, pages 1279–1282.

[Fridrich et al., 2002a] Fridrich, J., Goljan, M., and Hogea, D. (2002a). Steganalysis of JPEG images: Breaking the F5 algorithm. In *Proc. Int. Workshop on Information Hiding*, pages 310–323.

[Fridrich and Kodovský, 2012] Fridrich, J. and Kodovský, J. (2012). Rich models for steganalysis of digital images. *IEEE Transactions on Information Forensics and Security*, 7(3):868–882.

[Fridrich et al., 2011] Fridrich, J., Kodovskỳ, J., Holub, V., and Goljan, M. (2011). Breaking HUGO–the process discovery. In *Proc. Int. Workshop on Information Hiding, LNCS, vol. 6958*, pages 85–101.

[Fridrich et al., 2002b] Fridrich, J. J., Goljan, M., and Hoga, D. (2002b). Steganalysis of JPEG images: Breaking the F5 algorithm. In *Proc. Workshop of Information Hiding, LNCS, vol. 2578*, pages 310–323.

[Gardner, 2013] Gardner, F. (2013). How do terrorists communicate? BBC news, November 2, 2013.

[Giorgi et al., 2007] Giorgi, D., Biasotti, S., and Paraboschi, L. (2007). Shape retrieval contest 2007: Watertight models track. *SHREC competition*, 8(7).

[Gul and Kurugollu, 2011] Gul, G. and Kurugollu, F. (2011). A new methodology in steganalysis: Breaking highly undetectable steganograpy (HUGO). In *Proc. Int. Workshop on Information Hiding, LNCS, vol. 6958*, pages 71–84.

[Guo et al., 2012] Guo, L., Ni, J., and Shi, Y. Q. (2012). An efficient JPEG steganographic scheme using uniform embedding. In *Proc. IEEE Int. Workshop on Information Forensics and Security*, pages 169–174.

[Guyon and Elisseeff, 2003] Guyon, I. and Elisseeff, A. (2003). An introduction to variable and feature selection. *Journal of Machine Learning Research*, 3:1157–1182.

[Hall, 1999] Hall, M. A. (1999). *Correlation-based Feature Selection for Machine Learning*. PhD thesis, The University of Waikato.

[Holub and Fridrich, 2012] Holub, V. and Fridrich, J. (2012). Designing steganographic distortion using directional filters. In *Proc. IEEE Int. Workshop on Information Forensics and Security*, pages 234–239.

[Holub and Fridrich, 2013] Holub, V. and Fridrich, J. (2013). Digital image steganography using universal distortion. In *Proc. ACM workshop on Information Hiding and Multimedia Security*, pages 59–68.

[Huang et al., 2009] Huang, N.-C., Li, M.-T., and Wang, C.-M. (2009). Toward optimal embedding capacity for permutation steganography. *Signal Processing Letters, IEEE*, 16(9):802–805.

[Itier and Puech, 2017] Itier, V. and Puech, W. (2017). High capacity data hiding for 3D point clouds based on static arithmetic coding. *Multimedia Tools and Applications*, 76(24):26421–26445.

[Jin et al., 2005] Jin, S., Lewis, R. R., and West, D. (2005). A comparison of algorithms for vertex normal computation. *The Visual Computer*, 21(1-2):71–82.

[Johnson and Jajodia, 1998] Johnson, N. and Jajodia, S. (1998). Steganalysis of images created using current steganography software. In *Proc. Int. Workshop on Information Hiding*, pages 273–289.

[Kammoun et al., 2012] Kammoun, A., Payan, F., and Antonini, M. (2012). Sparsity-based optimization of two lifting-based wavelet transforms for semi-regular mesh compression. *Computers & Graphics*, 36(4):272–282.

[Kanai et al., 1998] Kanai, S., Date, H., and Kishinami, T. (1998). Digital watermarking for 3D polygons using multiresolution wavelet decomposition. In *Proc. Int. Workshop Geometric Modeling: Fundamentals and Applications*, volume 5, pages 296–307.

[Karni and Gotsman, 2000] Karni, Z. and Gotsman, C. (2000). Spectral compression of mesh geometry. In *Proc. Annual Conf. on Computer Graphics and Interactive Techniques*, pages 279–286.

[Ker, 2005] Ker, A. D. (2005). Steganalysis of LSB matching in grayscale images. *IEEE Signal Processing Letters*, 12(6):441–444.

[Ker et al., 2013] Ker, A. D., Bas, P., Böhme, R., Cogranne, R., Craver, S., Filler, T., Fridrich, J., and Pevnỳ, T. (2013). Moving steganography and steganalysis from the laboratory into the real world. In *Proc. ACM Workshop on Information Hiding and Multimedia Security*, pages 45–58.

[Ker and Pevny, 2014] Ker, A. D. and Pevny, T. (2014). A mishmash of methods for mitigating the model mismatch mess. In *Proc. IS&T/SPIE Electronic Imaging*, pages 90280I–90280I–15.

[Kim et al., 2017] Kim, D., Jang, H.-U., Choi, H.-Y., Son, J., Yu, I.-J., and Lee, H.-K. (2017). Improved 3D mesh steganalysis using homogeneous kernel map. In *Proc. Int. Conf. on Information Science and Applications*, pages 358–365.

[Kim et al., 2006] Kim, M.-S., Cho, J.-W., Prost, R., and Jung, H.-Y. (2006). Wavelet analysis based blind watermarking for 3-D surface meshes. In *Proc. Int. Workshop on Digital Watermarking*, pages 123–137.

[Kim et al., 2005] Kim, M.-S., Valette, S., Jung, H.-Y., and Prost, R. (2005). Watermarking of 3D irregular meshes based on wavelet multiresolution analysis. In *Proc. Int. Workshop on Digital Watermarking*, pages 313–324.

[Kodovskỳ and Fridrich, 2011] Kodovskỳ, J. and Fridrich, J. (2011). Steganalysis in high dimensions: Fusing classifiers built on random subspaces. In *Media Watermarking, Security, and Forensics III*, pages 7880 – 7880 – 13.

[Kodovskỳ et al., 2012] Kodovskỳ, J., Fridrich, J., and Holub, V. (2012). Ensemble classifiers for steganalysis of digital media. *IEEE Transactions on Information Forensics and Security*, 7(2):432–444.

[Kodovsky and Fridrich, 2009] Kodovsky, J. and Fridrich, J. J. (2009). Calibration revisited. In *Proc. ACM Workshop on Multimedia and Security*, pages 63–74.

[Kodovskỳ and Fridrich, 2012] Kodovskỳ, J. and Fridrich, J. J. (2012). Steganalysis of JPEG images using rich models. In *Media Watermarking, Security, and Forensics*, pages 8303 – 8303 – 13.

[Kodovsky et al., 2014] Kodovsky, J., Sedighi, V., and Fridrich, J. (2014). Study of cover source mismatch in steganalysis and ways to mitigate its impact. In *Proc. IS&T/SPIE Electronic Imaging*, pages 90280J–90280J–12.

[Krzanowski, 2000] Krzanowski, W. (2000). *Principles of multivariate analysis*. Oxford University Press.

[Lewis, 1992] Lewis, D. D. (1992). Feature selection and feature extraction for text categorization. In *Proc. Workshop on Speech and Natural Language*, pages 212–217.

[Li, 1990] Li, W. (1990). Mutual information functions versus correlation functions. *Journal of Statistical Physics*, 60(5-6):823–837.

[Li et al., 2017] Li, Z., Beugnon, S., Puech, W., and Bors, A. G. (2017). Rethinking the high capacity 3D steganography: Increasing its resistance to steganalysis. In *Proc. IEEE Int. Conf. on Image Processing*, pages 510–514.

[Li and Bors, 2016] Li, Z. and Bors, A. G. (2016). 3D mesh steganalysis using local shape features. In *Proc. IEEE Int. Conf. on Acoustics, Speech and Signal Processing*, pages 2144–2148.

[Li and Bors, 2017] Li, Z. and Bors, A. G. (2017). Steganalysis of 3D objects using statistics of local feature sets. *Information Sciences*, 415-416:85–99.

[Liu et al., 2009] Liu, Q., Sung, A. H., and Qiao, M. (2009). Temporal derivative-based spectrum and mel-cepstrum audio steganalysis. *IEEE Transactions on Information Forensics and Security*, 4(3):359–368.

[Long et al., 2015] Long, M., Cao, Y., Wang, J., and Jordan, M. (2015). Learning transferable features with deep adaptation networks. In *Proc. 32nd Int. Conf. on Machine Learning*, pages 97–105.

[Long et al., 2014] Long, M., Wang, J., Ding, G., Pan, S. J., and Philip, S. Y. (2014). Adaptation regularization: A general framework for transfer learning. *IEEE Transactions on Knowledge and Data Engineering*, 26(5):1076–1089.

[Lounsbery et al., 1997] Lounsbery, M., DeRose, T. D., and Warren, J. (1997). Multiresolution analysis for surfaces of arbitrary topological type. *ACM Transactions on Graphics*, 16(1):34–73.

[Lubenko and Ker, 2012] Lubenko, I. and Ker, A. D. (2012). Steganalysis with mismatched covers: Do simple classifiers help? In *Proc. ACM Workshop on Multimedia and Security*, pages 11–18.

[Luo and Bors, 2008] Luo, M. and Bors, A. G. (2008). Principal component analysis of spectral coefficients for mesh watermarking. In *Proc. IEEE Int. Conf. on Image Processing*, pages 441–444.

[Luo and Bors, 2011] Luo, M. and Bors, A. G. (2011). Surface-preserving robust watermarking of 3-D shapes. *IEEE Transactions on Image Processing*, 20(10):2813–2826.

[Lyu and Farid, 2002] Lyu, S. and Farid, H. (2002). Detecting hidden messages using higher-order statistics and support vector machines. In *Proc. Int. Workshop on Information Hiding*, pages 340–354.

[Max, 1999] Max, N. (1999). Weights for computing vertex normals from facet normals. *Journal of Graphics Tools*, 4(2):1–6.

[Meyer and Bontempi, 2006] Meyer, P. E. and Bontempi, G. (2006). On the use of variable complementarity for feature selection in cancer classification. In *Proc. Workshops on Applications of Evolutionary Computation*, pages 91–102.

[Ohbuchi et al., 1997] Ohbuchi, R., Masuda, H., and Aono, M. (1997). Embedding data in 3D models. In *Interactive Distributed Multimedia Systems and Telecommunication Services*, pages 1–10.

[Ohbuchi et al., 2001] Ohbuchi, R., Takahashi, S., Miyazawa, T., and Mukaiyama, A. (2001). Watermarking 3D polygonal meshes in the mesh spectral domain. In *Graphics Interface*, volume 2001, pages 9–17.

[Pan and Yang, 2010] Pan, S. J. and Yang, Q. (2010). A survey on transfer learning. *IEEE Transactions on Knowledge and Data Engineering*, 22(10):1345–1359.

[Pasquet et al., 2014] Pasquet, J., Bringay, S., and Chaumont, M. (2014). Steganalysis with cover-source mismatch and a small learning database. In *Proc. European Signal Processing Conference*, pages 2425–2429.

[Patel et al., 2015] Patel, V. M., Gopalan, R., Li, R., and Chellappa, R. (2015). Visual domain adaptation: A survey of recent advances. *IEEE Signal Processing Magazine*, 32(3):53–69.

[Payan and Antonini, 2006] Payan, F. and Antonini, M. (2006). Mean square error approximation for wavelet-based semiregular mesh compression. *IEEE Transactions on Visualization and Computer Graphics*, 12(4):649–657.

[Peng et al., 2005] Peng, H., Long, F., and Ding, C. (2005). Feature selection based on mutual information criteria of max-dependency, max-relevance, and min-redundancy. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 27(8):1226–1238.

[Pevny et al., 2010] Pevny, T., Bas, P., and Fridrich, J. (2010). Steganalysis by subtractive pixel adjacency matrix. *IEEE Transactions on Information Forensics and Security*, 5(2):215–224.

[Pevnỳ et al., 2010] Pevnỳ, T., Filler, T., and Bas, P. (2010). Using high-dimensional image models to perform highly undetectable steganography. In *Proc. Int. Workshop on Information Hiding*, pages 161–177.

[Pevný and Fridrich, 2007] Pevný, T. and Fridrich, J. (2007). Merging Markov and DCT features for multi-class JPEG steganalysis. In *Proc. SPIE Electronic Imaging*, pages 6505 – 6505 – 13.

[Pevnỳ and Ker, 2013] Pevnỳ, T. and Ker, A. D. (2013). The challenges of rich features in universal steganalysis. In *Proc. IS&T/SPIE Electronic Imaging*, pages 86650M–86650M–15.

[Pevnỳ and Ker, 2015] Pevnỳ, T. and Ker, A. D. (2015). Towards dependable steganalysis. In *Proc. SPIE/IS&T Electronic Imaging*, pages 94090I–94090I–14.

[Pudil et al., 1994] Pudil, P., Novovičová, J., and Kittler, J. (1994). Floating search methods in feature selection. *Pattern Recognition Letters*, 15(11):1119–1125.

[Qian et al., 2015] Qian, Y., Dong, J., Wang, W., and Tan, T. (2015). Deep learning for steganalysis via convolutional neural networks. In *Porc. SPIE/IS&T Electronic Imaging*, pages 9409 – 9409 – 10.

[Ren et al., 2017] Ren, Y., Yang, J., Wang, J., and Wang, L. (2017). AMR steganalysis based on second-order difference of pitch delay. *IEEE Transactions on Information Forensics and Security*, 12(6):1345–1357.

[Ren et al., 2014] Ren, Y., Zhai, L., Wang, L., and Zhu, T. (2014). Video steganalysis based on subtractive probability of optimal matching feature. In *Proc. ACM Workshop on Information Hiding and Multimedia Security*, pages 83–90. ACM.

[Roffo et al., 2017] Roffo, G., Melzi, S., Castellani, U., and Vinciarelli, A. (2017). Infinite latent feature selection: A probabilistic latent graph-based ranking approach. In *Proc. IEEE Int. Conf. on Computer Vision*, pages 1398–1406.

[Roffo et al., 2015] Roffo, G., Melzi, S., and Cristani, M. (2015). Infinite feature selection. In *Proc. IEEE Int. Conf. on Computer Vision*, pages 4202–4210.

[Rossignac, 1999] Rossignac, J. (1999). Edgebreaker: Connectivity compression for triangle meshes. *IEEE Transactions on Visualization and Computer Graphics*, 5(1):47–61.

[Rugis and Klette, 2006] Rugis, J. and Klette, R. (2006). A scale invariant surface curvature estimator. In *Advances in Image and Video Technology*, pages 138–147. Springer.

[Rusinkiewicz, 2004] Rusinkiewicz, S. (2004). Estimating curvatures and their derivatives on triangle meshes. In *Proc. Int. Symposium on 3D Data Processing, Visualization and Transmission*, pages 486–493.

[Shao et al., 2014] Shao, H.-C., Hwang, W.-L., and Chen, Y.-C. (2014). A backward wavelet remesher for level of detail control and scalable coding. In *Proc. IEEE Int. Conf. on Image Processing*, pages 5596–5600.

[Shi et al., 2006] Shi, Y. Q., Chen, C., and Chen, W. (2006). A Markov process based approach to effective attacking JPEG steganography. In *Proc. Int. Workshop on Information Hiding*, pages 249–264.

[Song et al., 2015] Song, X., Liu, F., Yang, C., Luo, X., and Zhang, Y. (2015). Steganalysis of adaptive JPEG steganography using 2D gabor filters. In *Proc. ACM Workshop on Information Hiding and Multimedia Security*, pages 15–23.

[Su et al., 2011] Su, Y., Zhang, C., and Zhang, C. (2011). A video steganalytic algorithm against motion-vector-based steganography. *Signal Processing*, 91(8):1901–1909.

[Sun et al., 2009] Sun, J., Ovsjanikov, M., and Guibas, L. (2009). A concise and provably informative multi-scale signature based on heat diffusion. In *Computer graphics forum*, volume 28, pages 1383–1392.

[Tan and Li, 2014] Tan, S. and Li, B. (2014). Stacked convolutional auto-encoders for steganalysis of digital images. In *Proc. Asia-Pacific Signal and Information Processing Association Annual Summit and Conf.*, pages 1–4.

[Tan et al., 2017] Tan, S., Zhang, H., Li, B., and Huang, J. (2017). Pixel-decimation-assisted steganalysis of synchronize-embedding-changes steganography. *IEEE Transactions on Information Forensics and Security*, 12(7):1658–1670.

[Tang et al., 2014] Tang, W., Li, H., Luo, W., and Huang, J. (2014). Adaptive steganalysis against WOW embedding algorithm. In *Proc. ACM workshop on Information Hiding and Multimedia Security*, pages 91–96.

[Tang et al., 2016] Tang, W., Li, H., Luo, W., and Huang, J. (2016). Adaptive steganalysis based on embedding probabilities of pixels. *IEEE Transactions on Information Forensics and Security*, 11(4):734–745.

[Taubin, 1995] Taubin, G. (1995). A signal processing approach to fair surface design. In *Proc. 22nd annual Conf. on Computer Graphics and Interactive Techniques*, pages 351–358.

[Tombari et al., 2013] Tombari, F., Salti, S., and Di Stefano, L. (2013). Performance evaluation of 3D keypoint detectors. *International Journal of Computer Vision*, 102(1-3):198–220.

[Uccheddu et al., 2004] Uccheddu, F., Corsini, M., and Barni, M. (2004). Wavelet-based blind watermarking of 3D models. In *Proc. ACM Workshop on Multimedia and Security*, pages 143–154.

[Upham, 1997] Upham, D. (1997). Jsteg. Available on the Internet. `ftp://ftp.funet.fi/pub/crypt/steganography/jpeg-jsteg-v4.diff.gz`.

[Vieira et al., 2016] Vieira, T., Martínez, D., Andrade, M., and Lewiner, T. (2016). Estimating affine-invariant structures on triangle meshes. *Computers & Graphics*, 60:83–92.

[Wang and Cheng, 2005] Wang, C.-M. and Cheng, Y.-M. (2005). An efficient information hiding algorithm for polygon models. In *Computer Graphics Forum*, volume 24, pages 591–600.

[Wang et al., 2008] Wang, K., Lavoué, G., Denis, F., and Baskurt, A. (2008). Hierarchical watermarking of semiregular meshes based on wavelet transform. *IEEE Transactions on Information Forensics and Security*, 3(4):620–634.

[Wang et al., 2014] Wang, K., Zhao, H., and Wang, H. (2014). Video steganalysis against motion vector-based steganography by adding or subtracting one motion vector value. *IEEE Transactions on Information Forensics and Security*, 9(5):741–751.

[Westfeld, 2001] Westfeld, A. (2001). F5a steganographic algorithm: High capacity despite better steganalysis. In *Proc. Int. Workshop on Information Hiding*, volume 2137, pages 289–302.

[Westfeld and Pfitzmann, 1999] Westfeld, A. and Pfitzmann, A. (1999). Attacks on steganographic systems. In *Proc. Int. Workshop on Information Hiding*, pages 61–76.

[Weston et al., 2001] Weston, J., Mukherjee, S., Chapelle, O., Pontil, M., Poggio, T., and Vapnik, V. (2001). Feature selection for svms. In *Advances in Neural Information Processing Systems*, pages 668–674.

[Wu et al., 2017] Wu, S., Zhong, S.-h., and Liu, Y. (2017). Residual convolution network based steganalysis with adaptive content suppression. In *Proc. IEEE Int. Conf. on Multimedia and Expo*, pages 241–246.

[Xu et al., 2016] Xu, G., Wu, H.-Z., and Shi, Y.-Q. (2016). Structural design of convolutional neural networks for steganalysis. *IEEE Signal Processing Letters*, 23(5):708–712.

[Xu et al., 2015] Xu, X., Dong, J., Wang, W., and Tan, T. (2015). Robust steganalysis based on training set construction and ensemble classifiers weighting. In *Proc. IEEE Int. Conf. on Image Processing*, pages 1498–1502.

[Yang et al., 2017a] Yang, J., Liu, K., Kang, X., Wong, E., and Shi, Y. (2017a). Steganalysis based on awareness of selection-channel and deep learning. In *Proc. Int. Workshop on Digital Watermarking*, pages 263–272.

[Yang and Ivrissimtzis, 2010] Yang, Y. and Ivrissimtzis, I. (2010). Polygonal mesh watermarking using Laplacian coordinates. *Computer Graphics Forum*, 29(5):1585–1593.

[Yang and Ivrissimtzis, 2014] Yang, Y. and Ivrissimtzis, I. (2014). Mesh discriminative features for 3D steganalysis. *ACM Transactions on Multimedia Computing, Communications, and Applications*, 10(3):27:1–27:13.

[Yang et al., 2014] Yang, Y., Pintus, R., Rushmeier, H., and Ivrissimtzis, I. (2014). A steganalytic algorithm for 3D polygonal meshes. In *Proc. IEEE Int. Conf. on Image Processing*, pages 4782–4786.

[Yang et al., 2017b] Yang, Y., Pintus, R., Rushmeier, H., and Ivrissimtzis, I. (2017b). A 3D steganalytic algorithm and steganalysis-resistant watermarking. *IEEE Transactions on Visualization and Computer Graphics*, 23(2):1002–1013.

[Ye et al., 2017] Ye, J., Ni, J., and Yi, Y. (2017). Deep learning hierarchical representations for image steganalysis. *IEEE Transactions on Information Forensics and Security*, 12(11):2545–2557.

[Yu et al., 2016] Yu, J., Li, F., Cheng, H., and Zhang, X. (2016). Spatial steganalysis using contrast of residuals. *IEEE Signal Processing Letters*, 23(7):989–992.

[Yu and Liu, 2004] Yu, L. and Liu, H. (2004). Efficient feature selection via analysis of relevance and redundancy. *Journal of Machine Learning Research*, 5(Oct):1205–1224.

[Zaid et al., 2015] Zaid, A. O., Hachani, M., and Puech, W. (2015). Wavelet-based high-capacity watermarking of 3-D irregular meshes. *Multimedia Tools and Applications*, 74(15):5897–5915.

[Zhang et al., 2017] Zhang, H., Cao, Y., and Zhao, X. (2017). A steganalytic approach to detect motion vector modification using near-perfect estimation for local optimality. *IEEE Transactions on Information Forensics and Security*, 12(2):465–478.