DEVELOPMENT OF EFFICIENT IMAGE STEGANOGRAPHY SYSTEM FOR 3D IMAGE MODELS

A thesis submitted

in partial fulfilment of the requirement for the award of degree of

Doctor of Philosophy

Submitted By

ASHISH GIRDHAR

(951503007)

Under the supervision of

DR. VIJAY KUMAR

(Assistant Professor)

COMPUTER SCIENCE AND ENGINEERING DEPARTMENT

THAPAR INSTITUTE OF ENGINEERING AND TECHNOLOGY

PATIALA-147004, PUNJAB INDIA.

October, 2019

Certificate

I, Ashish Girdhar, Regn. No. 951503007, hereby declare that the thesis entitled " **Development of Efficient Image Steganography System for 3D Image Models**" submitted to the Computer Science and Engineering Department at Thapar Institute of Engineering and Technology, Patiala, Punjab, India is an authenticated record of my own work for the award of degree of "Doctor of Philosophy" under the supervision of Dr. Vijay Kumar. This report has not been submitted to any other Institution for award of any other degree.

Girdhar

Ashish Girdhar

Regn. No. 951503007

Place: Patiala

Date: 29th October, 2019

This is to certify that the above statement made by the candidate is correct to best of my knowledge.

Verified by:

Vijay 29/10/19

Dr. Vijay Kumar

Assistant Professor,

Computer Science and Engineering Department,

Former : Thapar Institute of Engineering and Technology, Patiala

Present : National Institute of Technology, Hamirpur, H.P.

Abstract

Secure communication between two parties present at geographically different locations has become a matter of deep concern with information and technology progressing at such a fast pace. Steganography is the science of hiding secret message inside a harmless looking file in such a way that its presence is unnoticeable by human eye. Secret message to be hidden is considered a stream of binary bits. Secret message is sent by hiding it invisibly inside an innocuous-looking cover media. When the secret message is hidden in the cover file, stego-file is obtained as a result of embedding of secret message. A typical steganography system has two phases-embedding at message-sender side and extraction at message-receiver side. Extraction is termed as blind if cover file is not required for extraction of secret bits from stego-file. A steganography algorithm is called reversible if after extraction of secret message bits, original cover file is obtained. What cryptanalysis is to cryptography, steganalysis is to steganography. Thus, steganography should be done in such a manner that it remains an onerous task for steganalysts.

In 3D image steganography system, 3D mesh model is taken as the cover file and a binary stream of secret bits is hidden inside it. A 3D mesh model consists of vertices which join together to form edges. The edges enclose an area called face and these faces make surfaces. Vertices location is the secret bits hiding place in 3D image steganography. Reversible steganography having a blind extraction is a desirable feature. Also, rotation, scaling and translation of a 3D mesh model can be carried out by intruders which can destroy the hidden secret bits. Thus, it is important for a 3D steganography algorithm to safeguard the secret bits from these operations/attacks.

In this thesis, a mesh traversal algorithm is proposed. It is based on BFS (Breadth First Search) algorithm. Using the proposed mesh traversal algorithm, mesh vertices are visited and referenced in the order they have been visited. Hence, the steganography system withstands vertex reordering attack.

Proposed novel steganography system of data-hiding by difference shifting scheme is both reversible and has a blind extraction. In the proposed work, two connected vertices are taken for

embedding secret bits. One coordinate value-pair is picked for embedding of secret bits. This choice is made using the logistic map outcome so that secret bits are hidden randomly without forming a pattern. Logistic map is used for the first time in a 3D image steganography algorithm and thus this research work will arouse readers' interests to use one or more chaotic systems in steganography approaches.

The embedding algorithm hides secret bits inside 3D mesh model by novel difference shifting approach. Average and absolute difference values of two connected vertices are obtained. Difference is modified slightly in order to hide secret bits and new coordinates value are formed using modified average and modified difference values. Adaptive embedding is done while hiding secret bits so that the distortion to 3D cover model is minimum.

Secret information to be hidden inside the 3D mesh model is first encrypted before hiding. This adds an additional layer of security to the steganography system. Secret file to be embedded is image. A novel image encryption algorithm is proposed in the present work. The image encryption algorithm is based on the Lorenz-Rossler chaotic system and DNA cryptography on RGB images. Pixel values in RGB arrays and chaotic sequences are converted to DNA sequence. A total of six arrays (=3 from RGB + 3 from chaotic sequences) in DNA strands are thus obtained and performing operations on them, encrypted color image is obtained. Evaluation of proposed image encryption algorithm indicates a fairly good performance of the system.

In order to safeguard 3D stego-model from rotation, scaling and translation attacks, registration of gravity centre of 3D stego-model and two vertices is done. At the receiver side, if the gravity centre differs from that at the sender side, then 3D stego-model has been attacked. The effects of rotation, scaling and translation are reversed and secret bits are extracted. A trade-off between embedding capacity and distortion to 3D stego-model is obtained in this approach. Distortions are kept to minimum without compromising on embedding capacity. Reversible 3D image steganography techniques proposed in the literature are having very low embedding capacity. This reversible data hiding approach has decent embedding capacity and also withstands RST and vertex reordering attacks.

Other reversible data hiding techniques in 2D image steganography such as prediction error expansion can be modified and used for hiding secret bits in 3D mesh model. This novel steganography approach is intriguing, impelling and efficacious.

Acknowledgements

From registering in PhD to writing thesis a number of helpful hands came during different phases. I from core of my heart, would like to thank all of them.

I sincerely appreciate the almighty God for providing me this opportunity and granting me the capability to proceed successfully. A true homage to my Guru Ji Shri Swami Gurusharnanad Ji whose blessings has made me excel and efficacious throughout my life.

My absolute appreciation goes to my affable, ever supportive and humble supervisor, Dr. Vijay Kumar Assistant Professor, Computer Science and Engineering Department, Thapar Institute of Engineering and Technology for his capacious contribution, priceless guidance and motivation. I extremely appreciate the wonderful environment provided by him that made this thesis possible. It has been a boundless pleasure and experience to work under him as a student.

I am obliged to Head of Department, Prof. Maninder Singh and Prof. Inderveer Chana (Associate HOD), who made my learning a knowledgeable experience throughout my PhD. I am thankful to the Director, Dean (RSP) and the Management of Thapar Institute of Engineering and Technology, who provided me all the necessary resources and support. I am grateful to my doctoral committee members Prof. Kulbir Singh, Dr. Rinkle Rani, and Dr. Jhilik Bhattacharya for their productive suggestions during progress monitoring that ensured the right path of my study. I honestly thank the faculty and support staff of Computer Science and Engineering department for their continuous inspiration.

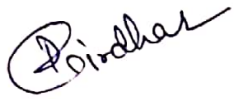
There are not enough words to thank my parents Mr. Ghan Sham Girdhar & Mrs. Veena Girdhar. Their unconditional love, care, moral support and encouragement helped me during different times. All that I am today, I owe to them. A special thanks to my chacha ji Mr. Sant Lal Girdhar for loving and supporting me in every situation like my father. He always provides me positivity and his way of caring makes me protected. This thesis would not have been completed without the support of my wife Himani Girdhar. She understood me in hard times and walked with me every step, during this happening journey. She inspired me to start it and her company made me to finish it joyfully. I would

like to sprinkle whole-hearted appreciation, gratitude and love to her. During this phase of my life, I received the best gift of life from God, my son Dhruv Girdhar. I would like to shower all my love on him. He made me happy by his delightful actions and charming smile in stressful situations.

A distinct thanks to my sister Sar Snehanand Ji whose blessings always help me in facing challenges in life. I would also like to thank my sisters and brother in laws, Mrs. Chanda Akash & Dr. Aakash Chanda, Mrs. Bhawna Arora & Mr. Vikas Arora for their concern, encouragement and love. A flower of love and thanks to nephews Mayukh & Shivansh, and niece Aaradhya & Ashima. I am indebted to my mother-in-law Dr. Meenu Kapur for her constant support, active interest and guidance. My in-laws Dr. Ravi Kapur, Dr. Indra Kapur, Dr. Aastha Gupta, Dr. Abhishek Gupta, Shivam Kapur, Aakriti, Pranav Kapur and Shubham Kapur deserve special mention for their continuous support and encouragement.

At last, I extend my sincere gratitude to all colleagues and friends, Dr. Prashant Singh Rana, Mr. Shatrughan Modi, Ms. Rajanpreet Kaur, Ms. Anika, Mr. Sahil Sharma, Dr. Seemu Sharma, Mr. Deep Chawla, Mr. Rakesh Suneja and Mr. Pradeep Singla for their support. Maybe, I have missed some names of my near and dear ones, I would like to thank all of them who helped me and supported me during different times in every possible manner.

Thank you all.



(Ashish Girdhar)

Contents

List of Figures

List of Tables

Chapter 1
Introduction

Due to advancement in hardware and software technologies, whole world has become a global village. A plethora of information is available on Internet and thus to everyone in all four corners of the world in just a few clicks. This led to availability of information to everyone in any part of the world. Thus, safeguarding the information in image from reaching wrong hands has become crucial. Protection of image can be done by two methods as shown in Fig. 1.1. Security systems are broadly classified into two main classes such as Cryptography and Information Hiding [1].

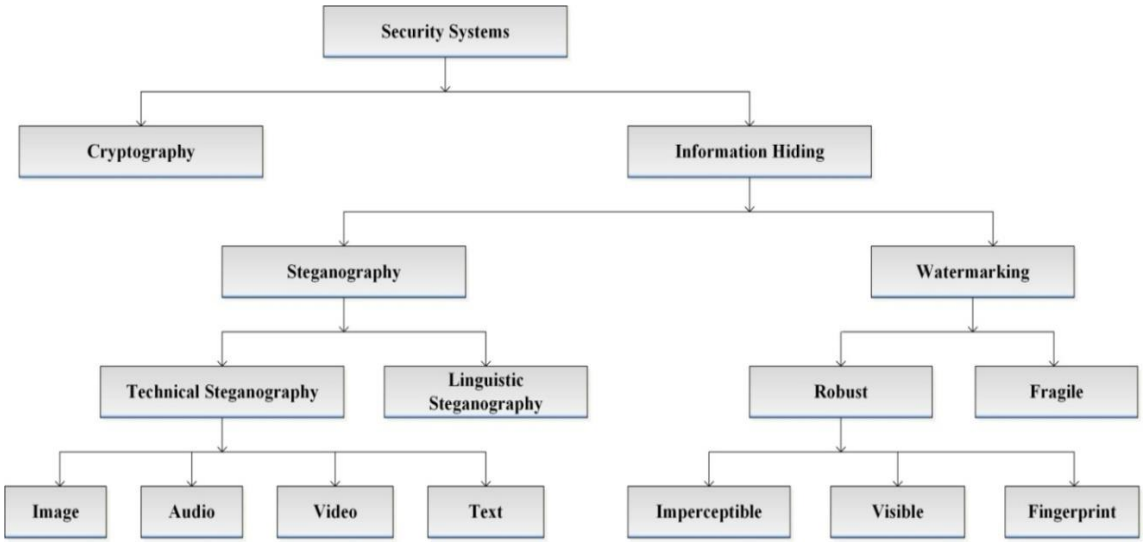


Fig. 1.1. Classification of security systems [1]

In cryptography systems, secret image is converted into some other form such that the intelligible property of image is destroyed. An image would be converted into some random arrangement of pixels. This method converts a meaningful image (called plain image) into some random image (called encrypted image) which does not convey any meaning and appears to be noise. Fig. 1.2 shows encrypted image of plain Lena image, encrypted Lena [2]. As can be seen, the encrypted image does not reveal anything about the original plain image. Thus, information contained in image is not revealed to the intruder.

(a) (b)

Fig. 1.2. (a) Plain Lena image(b) Encrypted Lena image

When the encrypted image is passed over internet from sender to receiver, it creates suspicion in the minds of eavesdropper. It arouses curiosity in them as to what important is being passed in form of random image. Thus, the main problem associated with cryptography systems is the jumbled representation of secret message. An intruder present over the communication channel might hamper the contents of jumbled message, if not understand it.

1.1 Information Hiding

The other method of sending secret message over a communication channel is information hiding. In this method, the secret message is hidden inside a naïve-looking image invisibly. This implies that the secret information is hidden inside a harmless media in such a way that it remains unperceivable to eyes of intruders. The harmless media used for hiding secret data is called cover media.

1.1.1 Types of Information Hiding

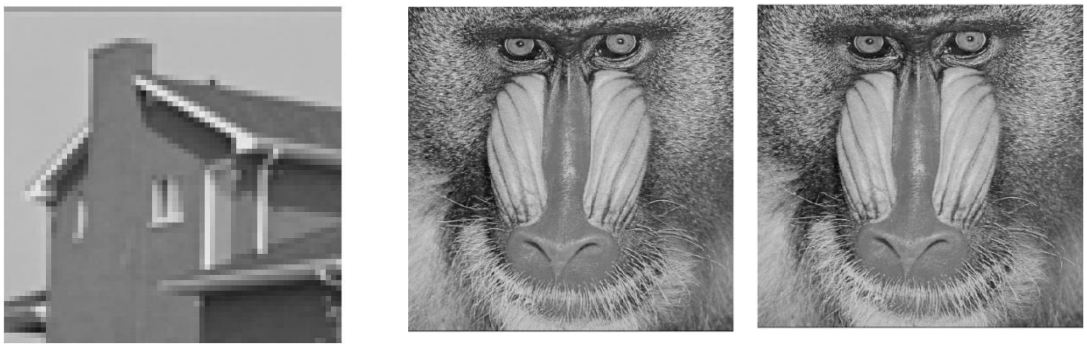
There are two kinds of information hiding techniques namely steganography and watermarking. Both these techniques work in a similar manner of invisibly hiding of secret information inside some naive media. When secret message is hidden inside the cover media, it becomes stego-media in case of steganography. In case of watermarking, it is watermarked-media. However, these techniques differ from each other in terms of objectives and carrying capacity [6]. Robustness is the main objective of watermarking technique [57]. Embedded watermark may be visible [63] or invisible [111]. But in case of steganography, the main objective is imperceptibility of secret message inside stego-media. Hidden secret message bits should not be visible to naked eyes of intruders. When the size of secret message is large, then steganography is preferred. For small

payloads such as a company banner, watermarking is done. Watermarking is used to protect copyright of media [60] whereas steganography is used to transfer secret information from one source to another. Table 1.1 shows the differences between these two techniques.

Table 1.1. Comparison between watermarking and steganography techniques

Criteria	**Watermarking**	**Steganography**
Objective	Robustness	Imperceptibility
Size of secret message	Very small; generally, a company logo	Secret message can be one-quarter of the size of cover media
Visibility of hidden data	May be visible or invisible.	Invisible

Fig. 1.3 shows the imperceptibility characteristic of steganography a secret image of house is shown in Fig. 1.3(a). Fig. 1.3 (b) shows the baboon cover image, which is used to hide house image. Fig. 1.3 (c) shows the stego-image, that is obtained by hiding secret house image. Least Significant Bit (LSB) steganography technique [4] is used as the steganography technique for hiding house image in cover image of baboon. It can be seen from Fig. 1.3(b) and Fig. 1.3(c) that there is no visible difference between cover image and stego-image.



(a) (b) (c)

Fig. 1.3. (a) Secret House image, (b) cover baboon image, and(c) stego-image obtained from LSB steganography

An intruder cannot make out from the stego-image that some important information is being carried away inside it. Therefore, the imperceptibility is a very crucial feature for any steganography algorithm.

1.2 Steganography

Hiding a secret message inside an innocuous file (called cover media) in such a manner that the hidden message remains unperceivable to the eyes of intruder is Steganography. The cover media is considered as a file. The secret message is taken as the stream of binary bits. Stego-media is obtained after embedding of secret bits inside cover media. Table 1.2 shows the comparison between cryptography and steganography [5].

Table 1.2. Comparison between cryptography and steganography techniques

Criteria	Cryptography	Steganography
Aim	It converts secret message into non-understandable form.	It hides secret message inside a harmless media so that it is invisible to human eye.
Outcome	Outcome of cryptography is a cipher.	Outcome of steganography is stego-media.
Modification of Secret Image	Cipher can be modified by the intruders.	Secret message hidden inside the harmless media cannot be modified by intruders.
Application	Cryptography works only on secret message.	Steganography works on secret data as well as a cover media.
Structure of secret message	Cryptography alters the structure of secret message	Steganography does not change the structure of secret message.

A typical steganography system has two phases such as embedding and extraction phases. Embedding phase takes place at the side of the sender where the secret message is hidden inside the cover media using some algorithm. Extraction phase takes place at the receiver side where the secret message is extracted from the stego-media. There are four kinds of steganography. These are text steganography, audio steganography, video steganography, and image steganography. Type of cover media used for hiding secret

message determines the type of steganography. If the cover media used is a text file, then it is known as text steganography. In case of audio steganography, the cover media is an audio file, while video steganography uses a video file as cover media. In case of image steganography, the cover media is an image.

The focus of this thesis is image steganography. Image steganography techniques outnumber the other three. A video file can be considered as moving frames of images, so if a secret data is concealed inside an image, then it can be embedded inside the video file also. Thus, image steganography paves the way for video steganography. Audio and text steganography techniques have not received much attention compared to image steganography because of the larger carrier required in the former two when same amount of payload is to be hidden in all three [6]

The mathematical formulation of steganography system can be described as follows. Assuming C to be cover media and M is secret message that needs to be embedded. $\dot{C}$ is the stego-media obtained after embedding secret message M inside the cover media using a key K. This takes place at the sender side. The embedding (Em) process can be described as:

$$Em: C\emptyset\ K\emptyset\ M = \dot{C} \tag{1.1}$$

where $\emptyset$ is an operation applied between C, K and M applied in order to embed the secret bits invisibly.

Similarly, the extraction phase (Ex) gives out the secret message M at the receiver side. If the cover media is also obtained in its exact form, then the steganography is called reversible. If there is no need of cover media for extraction of secret message, then it is termed as blind steganography.

$$Ex\left(\dot{C}, K\right) = M \tag{1.2}$$

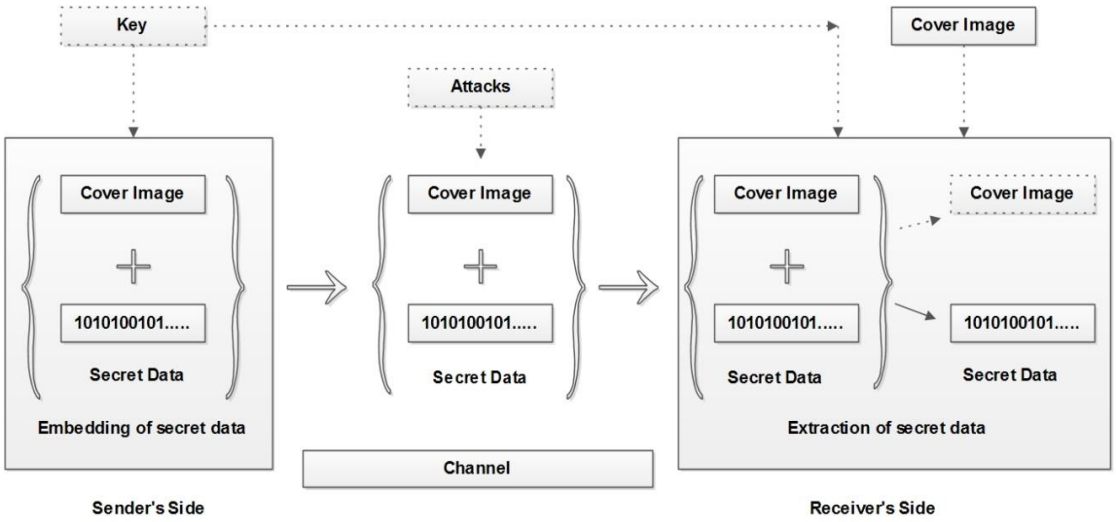


Fig. 1.4. Typical steganography system

Fig. 1.4 illustrates a steganography system. In this figure, superfluous elements are shown with dotted arrows. The cover media in this figure is an image; hence, this is a typical image steganography system. Key may or may not be used at the sender side for embedding of secret bits inside the cover media. When the stego-media is sent over the communication channel then it may be subjected to attacks. These attacks can hamper the contents of secret message inside the stego-media. Thus, the steganography system should be able to withstand attacks. When the stego-image reaches receiver, it may have been attacked by intruders present over communication channel. Hence, the pre-processing of stego-image may be required at the receiver side. Afterwards, the extraction of secret message from stego-media is done. If key is used at the sender side for embedding of data, then it is sent to the receiver side. If there is no need of cover image in the extraction process, then the extraction is termed as blind extraction. Steganography algorithm is termed as reversible if the cover image is obtained in its original form after extraction of secret bits from stego-image [9].

1.2.1 Characteristics of steganography approaches

A steganography approach should have the following characteristics in order to efficiently hide the secret bits inside the cover media invisibly. If a steganography system exhibits these characteristics then it is an efficient steganography system.

1. Imperceptibility

   Steganography system should be able to hide the secret message inside the cover media in such a way that the human eye cannot make out its presence. This is a highly

desirable characteristic for a steganography system. This characteristic is also referred to as perceptual transparency sometimes. The difference between stego-media and cover-media should not be made out by Human Visual System (HVS).

2. Robustness

Stego-media should be robust enough to safeguard the hidden message even stego-media is affected from scaling, transformations, and rotation attacks. The stego-image may be inspected by eavesdroppers and intruders during the communication. The steganography system should make sure that hidden message contents are not hampered even intruders or attackers perform any operations on the stego-image.

3. Embedding Capacity

Stego-image should be able to carry a secret message which is approximately quarter of its size. It is also termed as payload capacity and measured in terms of bits per pixel in case of image steganography. A trade-off between embedding capacity and imperceptibility has to be maintained. Increasing the embedding capacity may decrease the imperceptibility of the hidden message inside the stego-message, thereby revealing the hidden message. Thus, a balance between these two components is required.

4. Security

The stego-image must be secure enough to withstand attacks on it. Robustness and security are two different characteristics. Robustness of a steganography system is the ability of the stego-image to resist attacks. Security of the system is determined by its ability to withstand attacks by steganalysis. Steganalysis is the science in which stego-media is studied and checked to identify the existence of hidden message inside it [3].

Quality of steganography approach is determined by imperceptibility, robustness, and embedding capacity. Fig. 1.5 depicts the characteristics of steganography and watermarking techniques. When more capacity is desired, then steganography system may not be secure enough and it may fail to withstand attacks on stego-media. In case of digital watermarking, embedding capacity is low but the embedded watermark is robust. Ability of a stego-media to withstand attacks determines its level of security.

If a secure steganographic system is developed which resists the attacks, then the embedding capacity and robustness characteristics are compromised.

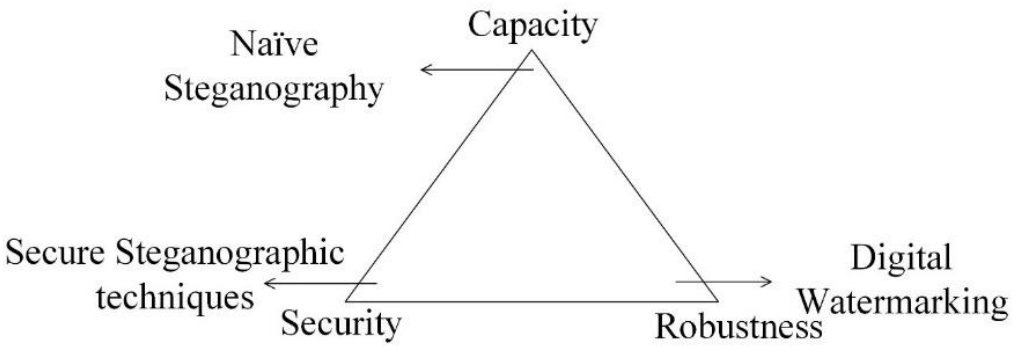


Fig. 1.5. Characteristics of information hiding techniques

This triangle examines the inter-relationships of the three characteristics and thus is called the magic triangle.

1.3 Image Steganography

In image steganography, the cover media is an image file. Secret message is a stream of binary bits and output of image steganography is stego-image.

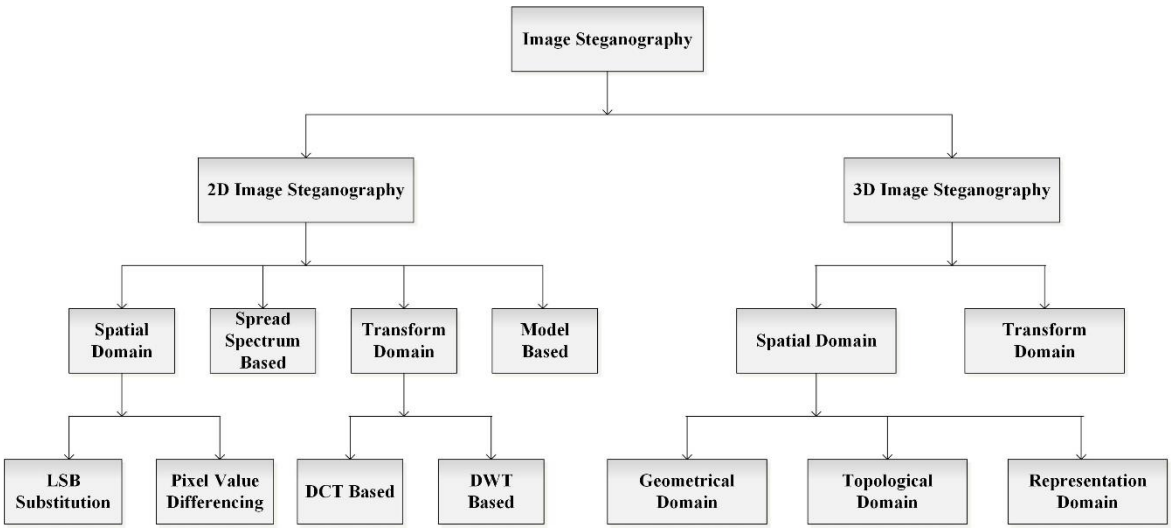


Fig. 1.6. Image steganography types [1]

On the basis of cover image file, Image Steganography can be further classified into two types. These are 2D Image Steganography and 3D Image Steganography. If the cover image file is a 2D image, then it is called 2D image steganography. In case of 3D image model being used then it is called 3D image steganography. Fig.1.6 shows

two types of Image Steganography. 2D image steganography approaches are developed in domain of Spread Spectrum, Spatial domain, Transform (or frequency) domain [19], and Model-based. 3D image steganography approaches are developed in spatial domain and transform domain.

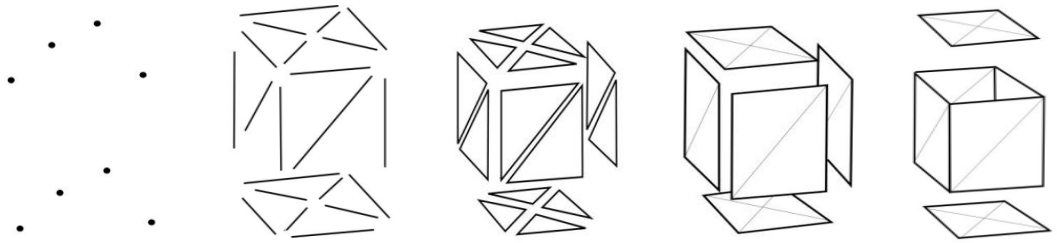
2D and 3D image steganography techniques are different from each other in various terms. In 2D image steganography, a 2-dimensional matrix containing pixel intensities is used as cover file. But in case of 3D image steganography, a 3-dimensional image model (which may be a mesh model, point cloud or NURBS surface, etc) is used as cover medium. In case of former, pixel intensities are modified to hide the secret information. In 3D image steganography, vertices are modified in order to hide secret message inside it. In case of 3D steganography, embedding capacity is measured as bits per vertex and for 2D steganography, embedding capacity is measured in terms of bits per pixel.

A single 3D stego-model is able to carry larger payload than a single 2D stego-image. This is because, a 3D image model is large and has more data-points (vertices in this case) to hide secret bits than 2D image. The former one has large embedding capacity than the latter. However, bandwidth required for sending a 3D image model is much higher. Thus, 2D image steganography is preferable for small embedding capacity. When the size of secret message is large enough to adjust the transmission costs, then 3D steganography is preferred. The focus of this thesis is 3D image steganography.

1.3.1 3D image models

3D image steganography system requires a 3D image cover model and/or a key. 3D image model can be represented as a NURBS surface, point cloud model, polygon mesh model, etc. Among these different representations, polygon mesh model is preferred. This is because polygon mesh model can be transferred from one place to another at the highest rate in comparison to all other representations. Thus, the present work uses polygon mesh model representation.

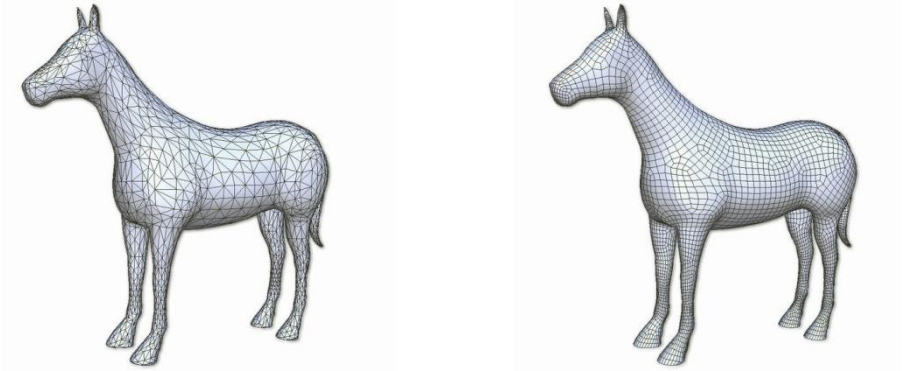
A 3D image model is composed of vertices, faces, and edges. A single point in a 3D mesh is called a vertex and is represented by three coordinates (x, y, and z) in Cartesian system. Two such vertices are joined together by an edge. A face is formed by enclosing a set of vertices. Fig. 1.7 shows 3D image model.



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

Fig. 1.7. 3D Mesh Model (a)vertices, (b) edges, (c)faces, (d)polygon, and (e)surface

A mesh model having only triangle faces is known as a triangle mesh; while a mesh containing only quadrilateral faces is called a quadrilateral mesh.



(a) (b)

Fig. 1.8. 3D mesh model of horse(a) triangle mesh (b)quadrilateral mesh

Fig. 1.8 depicts how a 3D horse can be represented using a triangle mesh or a quadrilateral mesh. Type of mesh affects the shading effect of 3D model. A 3D image model is composed of vertices in 3D space. All the connected vertices of the mesh are visited for hiding secret bits inside the 3D mesh model. Thus, for traversing vertices of the mesh, a mesh traversing algorithm should also be designed along with steganography technique. Mesh traversing algorithm should give same and unique traversal order of vertices over a particular mesh. It should give same mesh traversal order every time it is run. Also, it should give a single mesh traversal order on a particular mesh. For instance, in the mesh shown below in Fig. 1.9, if Breadth First Search (BFS) algorithm is applied, there can be many traversal orders. Taking vertex 1 as the source vertex, traversal orders can be $1 \rightarrow 2 \rightarrow 3 \rightarrow 4$, $1 \rightarrow 3 \rightarrow 2 \rightarrow 4$ and $1 \rightarrow 4 \rightarrow 3 \rightarrow 2$. Since mesh is more of an undirected graph, thus there can be more than one traversal order of vertices.

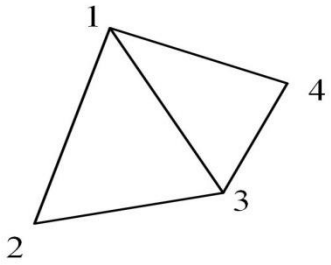


Fig. 1.9. A simple mesh

A mesh traversal algorithm that fails to give a unique mesh traversal order for a particular mesh fails to withstand vertex reordering attack. Thus, need for devising a mesh traversal algorithm arises, which gives a unique traversal order of vertices of mesh irrespective of the number of times it is run over the mesh

1.3.2 Types of 3D image steganography

3D image steganography is broadly classified into two classes namely -Spatial and Transform domain. In spatial domain, vertices of 3D cover model are directly modified while in transform domain, 3D model is first taken to transform domain and then modifications are done on it. A typical 3D image model has a large number of vertices. So, in order to move 3D image model in and out of transform domain is time-consuming and has high space complexity. Thus, spatial domain is preferred over transform domain. 3D image steganography approaches based in spatial domain are further classified into three categories. These are geometrical domain-based steganography, topological domain-based steganography, and representation domain-based steganography.

3D image steganography approaches based on geometrical domain utilise the geometrical aspects of 3D image cover model to hide the secret message bits. Embedding of secret message bits in geometrical aspects of 3D model fails to withstand changes in geometry of 3D model. This implies any change in geometry of 3D stego-model can destroy the secret message inside it. Thus, embedding processes should be resistant against the geometrical transformations. Ohbuchi et al. [18] listed some of geometrical aspects of 3D image model that can be taken up for hiding of secret message bits.

Topological domain-based steganography algorithms modify the connectivity or topology information of 3D mesh model. Connectivity between different vertices is

changed in such a way that any observable change is not made in the cover model. As topology of a mesh remains unaffected by changes in its geometry, geometrical transformations will not have any effect on 3D stego-model. Thus, these approaches are fail-proof from geometrical transformations. However, connectivity information in a 3D mesh model is less as compared to its geometrical aspects. Thus, embedding capacity of topological domain-based steganography approaches is less.

Representation domain-based 3D image steganography approaches modify the redundancy present in a mesh model. These approaches are infallible to the geometrical transformations as they remain intact in case of any change in geometry. Redundant information is added to mesh and then this redundant information is modified in order to hide secret message bits.

1.4 Attacks on 3D image steganography

Attacks on 3D image steganography can be broadly classified into two categories such as distortion and distortion less attacks. An attacker of 3D stego-model may not have knowledge of 3D steganography. Few attacks on the 3D image steganography are depicted in Fig. 1.10.

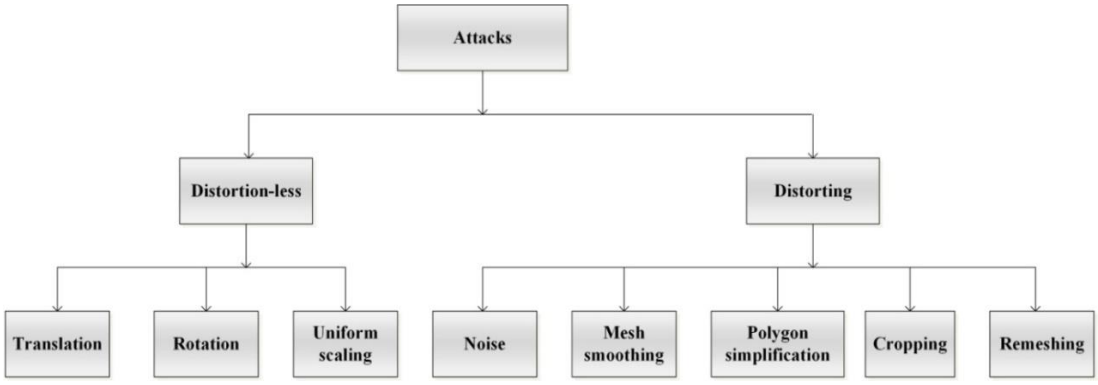


Fig. 1.10. Attacks on 3D stego-model

Distortion-less attacks are those attacks which do not tamper mesh model but may destroy hidden secret bits inside the mesh model. Rotation, translation, uniform scaling and vertex reordering are examples of distortion less attacks. Vertex reordering attack changes the vertex reference indices. It does not hamper the mesh, but changing the reference indices will result in incorrect extraction of secret message bits at the receiver side [16].

Rotation, uniform scaling, and translation are other examples of distortion-less attacks. These attacks do not modify mesh connectivity. Since steganography systems based on topological domain are designed by changing the topology or connectivity of the mesh. These systems are not affected by RST attacks. As they can modify the vertices' locations, geometrical domain-based steganography algorithm can fail to withstand these attacks. The resistance of algorithms designed geometrical domain should be checked against distortion-less attacks. As far as representation domain-based algorithms are concerned, they make use of the redundant information of mesh. If the redundant information is about the vertices' locations, then they are unable to extract correct secret message. Another type of attack is distortion attack. Distortion attacks change stego-mesh along with secret message bits. They change mesh and while modifying mesh vertices and their connectivity, secret message bits embedded may be changed. Some well – known distorting attacks are Noise, mesh simplification, and cropping, Algorithms based on geometrical, topological, and representation domain fail to withstand these types of attacks. The extracted secret message from attacked stego-model may contain incorrect information.

1.5 Steganalysis

Steganalysis is the science of generating algorithms that detects the existence of secret data inside stego-model [3]. It may or may not be able to steal the secret message hidden inside it.

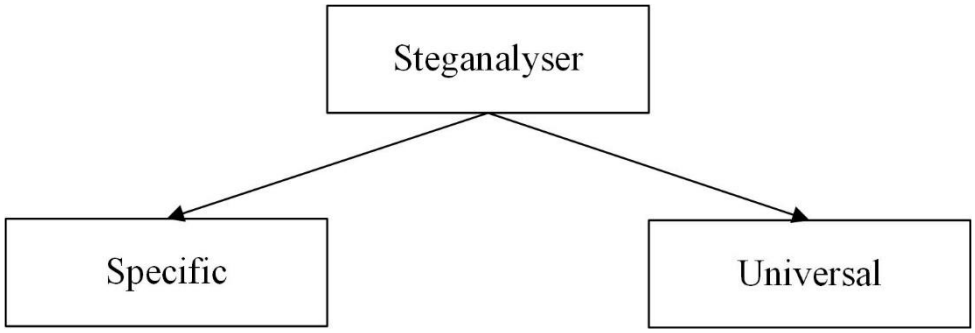


Fig. 1.11. Types of 3D steganalyser

There are two kinds of steganalysis techniques named-specific and universal [21] as shown in Fig. 1.11. Specific steganalyser is designed to attack stego-model embedded with specific steganography algorithm. On the other hand, universal steganalyser algorithms are developed for detection of hidden secret message inside cover model

embedded using any steganography algorithm. 3D image steganalyser algorithms are developed by observing the statistical changes that might have occurred because of embedding of secret message bits. Although, secret message bits are invisible to human eyes, the natural statistics of 3D image model are disturbed because of embedding [3, 22].

1.6 3D Image Steganography applications

Steganography has various applications in different fields as enlisted below. It is a medium of covert communication between two parties sitting at different corners of the world. Since presence of secret bits is invisible to the naked eye, there lies a possibility that the steganography algorithms are being used but have not seen the daylight yet. Some of the applications where steganography algorithms are being used have been put up below.

1. Military and defense organizations

Steganography has been used by terrorist organizations for communicating secret information among their various units [23-25]. A few years ago, a US special agent from FBI filed complaint against some alleged Russian agents that they have been using steganography for hiding encrypted messages [112]. News of using 2D cover images for steganography by defense and criminal organizations has surfaced time to time [23-25, 112]. It might be a possibility that 3D image steganography has also been used for covert communications but the news has not broken out yet [15]. Thus, development of steganography algorithms using 3D image models can be helpful in efficient working of defense organizations.

2. Medical area

Another application of steganography is in medical area. Steganography algorithms can be used for hiding the patient history and other such useful information inside the reports prepared on 3D model of human organs [113]. It should be noted that the embedding done in this case should be reversible in nature so that it does not alter the patient's report.

3. Monitoring copyrighted material on internet

Availability of various 3D computer graphics software such as Blender, Maya, Mudbox, etc. [115] has made the task of designing of 3D models easy and simplified. As a result, need to protect these 3D models against their copyrighted use arises. Steganography plays an important role in this case as it secretly carries the owner's name and other related information inside the 3D model and inhibits its illegitimate use. It should be noted that the steganography algorithm used for hiding this information is robust against different attacks. In other words, attackers or duplicate copy makers should not be able to remove the information from the original work however hard they may try.

1.7 Research Motivation

3D steganography algorithms have not received much attention due to some inherent problems in this area as pointed out in [116]. Also, low embedding capacity of 3D image models thrusts the researchers away from 3D steganography. Robustness of 3D stego-models is low and they are susceptible to fall down in case of attacks.

Apart from this, rotation, scaling and translation transformations over a 3D image model can be carried out by any intruder or attacker. These transformations do not change 3D model. Thus, if a 3D model is attacked by affine transformations, it does not appear to be attacked at all at the receiver side. Receiver may extract incorrect secret message unknowingly.

Secret message is embedded as such inside 3D model. This is only one layer of data security. In place of this, the secret message should first be converted to some other form and then embedded inside the 3D image model.

Distortion to 3D image model and its embedding capacity are both in direct relationship with one another. Increasing one factor also increases the other. However, more capacity and low distortion is desired for an ideal steganography algorithm. A trade-off between capacity and distortion has to be maintained. Increasing embedding capacity is bound to enhance distortion in 3D mesh model. Low embedding capacity will cause less distortion. Sampling of data points of 3D image model is a very difficult task. Thus, embedding effects on 3D mesh model should be reversible. Cover model should be obtained in its exact form after extraction of secret bits at the receiver side. Hence, same 3D image model may be reused again.

These factors motivate a research work on 3D image steganography system which could increase embedding capacity without causing much distortion to stego-model. The proposed steganography system should also withstand attacks and should be reversible. Secret message bits should be first encrypted and/or scrambled, as the case may be and then embedded inside the 3D model.

1.8 Thesis Contributions

The main contributions of present thesis are as follows:

- Lorenz-Rossler hyper chaotic system is used in this work in such a way that the randomness in plain image is increased. Inter channel operations in three color channels of the RGB image even further enhance the randomness.
- Image encryption algorithm has also been proposed in this thesis. The proposed algorithm is based on DNA cryptosystem.
- Mesh traversal algorithm has been proposed in this thesis. This algorithm traverses mesh vertices and gives unique and same traversal order every time it is run over a particular mesh.
- Adaptive steganography algorithm is proposed in this thesis. Adaptive steganography implies that in the noisy surfaces of the mesh model a greater number of secret bits are hidden than the smooth surfaces of mesh model. A noisy surface and smooth surface are distinguished from each other on the basis of number of vertices present on the surface.
- The proposed steganography system embeds secret message bits adaptively inside the 3D mesh model. Thus, the distortion caused to the cover model is as low as possible.
- Logistic chaotic map is also used in the steganography algorithm for the first time in 3D image steganography. Use of chaotic map imparts randomness in embedding of secret bits inside the 3D mesh model. This makes the steganography algorithm robust to withstand the attacks by steganalysts.
- The proposed steganography system has blind extraction at the receiver side. This implies that the cover model is not required while extraction of secret message from stego-model.

The proposed steganography system is reversible in nature. This means that the stego-model gives back the original cover model after extraction of secret bits from it. Thus, the effects of steganography are not permanent on the 3D mesh model.

1.9 Thesis Organisation

Introduction to the thesis has been done in Chapter 1, rest of the structure of thesis is explained as follows:

In chapter 2, literature survey of various steganography techniques is presented in detail. Various techniques based on geometrical, topological and representation domain are explained and compared with one another. The ability of these techniques to withstand attacks has also been discussed.

In chapter 3, technique to encrypt secret 2D image is proposed. This chapter discusses in detail the proposed technique. Also, this chapter lists down various performance metrics to measure the performance of the encryption algorithm. The image encryption algorithm proposed in this work is for encryption of colour images.

In chapter 4, the main objective of this thesis is covered. In this chapter, a 3D image steganography algorithm is presented. In order to enable the 3D image steganography algorithm to withstand vertex reordering attack, a 3D mesh traversal algorithm is also proposed which gives a unique mesh traversal order for a particular 3D mesh. Also, PCA is performed on the 3D mesh in order to detect if the 3D mesh model has been attacked by rotation, scaling and translation.

In chapter 5, the ability of 3D stego-model to resist rotation, scaling and translation attacks is checked. Secret bits hidden inside the 3D stego-model should not be hampered when the 3D stego-model is attacked. Perceptual distortions on the surface of 3D mesh model is also looked out using Meshlab tool [55]

In chapter 6, conclusion of the thesis is presented along with directions for future work in this research area. Reversible data hiding in 3D mesh models is an emerging field.

Chapter 2
Literature Review

This chapter studies the various 3D image steganography techniques proposed in the literature. 3D image steganography techniques are very less in literature than those of 2D image steganography techniques.

Algorithms designed in spatial domain are more in number as compared to those in transform domain. In spatial domain algorithms, modifications are done directly on the vertices of cover mesh. Embedding done in spatial domain may fail to withstand distortion-less attacks. In other words, if the stego-model is rotated, scaled and/or translated; vertices on which embedding is done may get affected. So, incorrect secret information might be extracted at the receiver side. In case of transform domain, mesh model is first taken to transform domain and then modifications are done. These algorithms withstand distortion-less attacks. So, if the stego-model is rotated, scaled or translated then there is no effect on the embedded secret message bits. Nevertheless, algorithms in transform domain are less in number as compared to the algorithms in spatial domain. This is because of the fact that taking a 3D model in and out of transform domain has more time and space complexity [34].

2.1 Transform domain-based steganography

Aspert et al. [26] proposed steganography algorithm based on watermarking algorithm in [27]. Cover model was first transformed to a different space which could only partially resist rotation attack. In this algorithm, textual secret message is converted to its ASCII form and then hide inside the *normal* vector of the vertices. These *normal* vectors are not normal vectors of the vertices as per the definitions of geometry. They redefined *normal* vectors in their work as per the algorithm. RMSE (Root Mean Square Error) between the cover model and stego-model was calculated for cover model in order to find out distortions caused to cover model because of embedding.

Maret et al. [28] proposed a new approach for designing a steganography algorithm which could resist rotation, scaling and translation attacks. Based on algorithm proposed in [26], a new similarity-transform invariant space was created and the cover model was taken to it. Embedding algorithm has three stages. In the first stage, the cover model was taken to similarity invariant space. The outcome of this step was cover model sampled over a unit sphere. Embedding of secret message bits was done inside the cover model in the second stage. In third stage, other modifications were done on the cover model in order to embed secret message bits. Steganography system has a blind extraction method and results of distortion and embedding capacity were compared with [26].

2.2 Spatial domain-based steganography

Based on the embedding location of secret message bits, spatial domain-based steganography techniques are classified into three main categories:

1. Geometrical domain-based steganography
2. Topological domain-based steganography
3. Representation domain-based steganography

These steganography techniques are compared in terms of

- blind extraction method,
- reversible nature and
- ability to withstand attacks, etc.

A steganography system is called reversible if after the extraction of secret message bits from 3D stego-model, 3D cover model is received back accurately. The effect of embedding secret bits on 3D image model should not be permanent. The original cover model should be received in its original form at the receiver side.

Another desirable feature in 3D image steganography is a blind extraction method. If the cover media is not required for extraction of secret message bits, then the extraction method is termed as blind. 3D image model is large because of large data points. Hence, bandwidth required to send one 3D image model from one place to another over communication channel is huge. If secret message bits are extracted from 3D stego-model without 3D cover model, then it uses less bandwidth. Thus, it is a desirable characteristic of 3D image steganography algorithms.

Attacks on a 3D image model are of two types namely, distortion-less and distorting. These attacks do not require any knowledge of 3D mesh and can be carried out by intruders of communication channels. Also, distortion-less attacks do not cause any change in 3D mesh and are able to modify the mesh completely without knowledge of sender and receiver. But they may affect the embedded secret message bits inside 3D image model. Thus, the embedded secret message bits can be easily modified by causing distortion-less attacks over 3D mesh model without coming into notice of the sender and receiver.

2.2.1 Geometrical domain-based steganography techniques

In geometrical domain-based steganography approach, geometrical primitives such as vertices, edges, strips of triangles, etc. of 3D cover object in order to hide the secret message [3]. Geometrical transformations are able to destroy embedded message bits. Thus, Ohbuchi et al. [18] listed geometrical primitives that are invariant to one or more such transformations. A few of them are listed in Table 2.1.

Table 2.1. Geometrical primitives for embedding

Geometrical Primitive	**Affected by**		
	Rotation	**Scaling**	**Translation**
Location of vertex	√	√	√
Distance between two vertices	×	√	×
Area of a polygon	×	√	×
Volume enclosed by a polygon	×	√	×
Ratio of area of two polygons	×	×	×
Ratio of volumes enclosed by two polygons	×	×	×

In Table 2.1 a tick mark indicates that the geometrical primitive can withstand the corresponding attack. On the other hand, a cross mark indicates that the embedding fails when attacked by transformation/s. This is a very short list of embedding primitives that can be used for embedding in geometrical domain. There may be other embedding primitives that can be used for embedding of secret bits in this domain. Fig. 2.1 shows

the development of 3D steganography approaches based in geometrical domain over the past years.

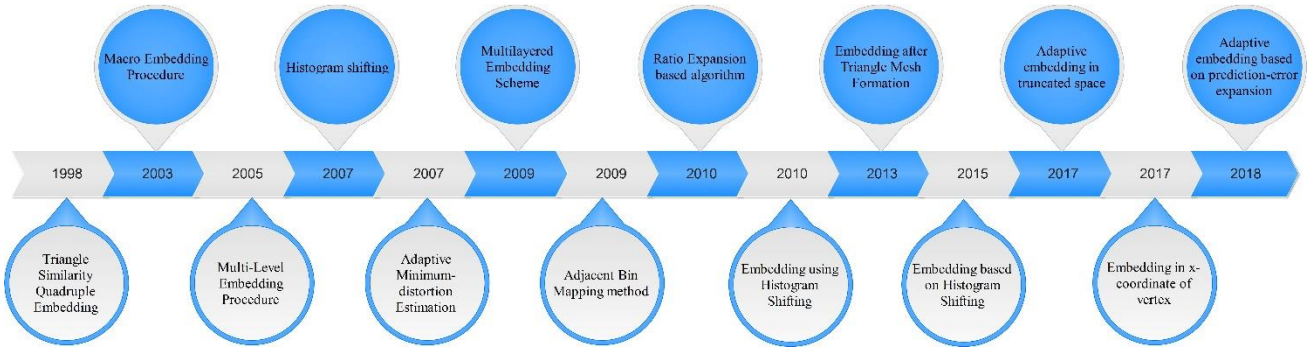


Fig. 2.1. Timeline showing geometrical domain-based steganography techniques

Ohbuchi et al. [18] proposed five novel algorithms, namely

1. Triangle Similarity Quadruple Embedding
2. Tetrahedral Volume Ratio Embedding.
3. Triangle Strip peeling symbol-sequence Embedding
4. Polygon stencil pattern-embedding
5. Mesh density pattern embedding

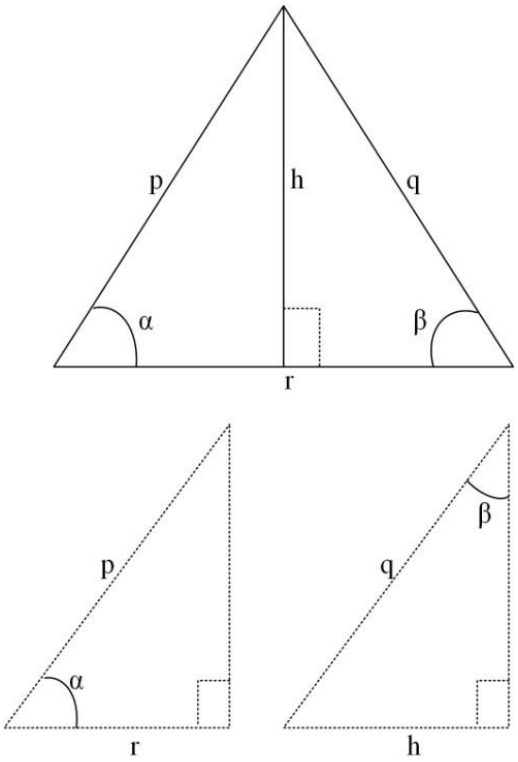


Fig. 2.2. Similar triangles [18]

Triangle Similarity Quadruple Embedding algorithm is based on the concept of similar triangles. As shown in Fig. 2.2, two right-angled triangles are proved to be similar if the dimensionless quantity pairs {α, β} or $\{q/p, h/r\}$ are equal. For embedding secret bits, modifications are done in these dimensionless quantities so that embedding withstands geometrical attacks/ distortion-less attacks. The extraction procedure of this embedding algorithm was blind.

The other algorithm in this research work, namely Tetrahedral volume ratio embedding steganography technique was based on modifying ratio of volumes of two tetrahedrons. In this algorithm, vertices of cover mesh were slightly modified in order to modify the volume enclosed by the tetrahedron. In addition to withstand geometrical attacks, stego-mesh was able to resist topological changes to the mesh, e.g. remeshing. In place of finding an initial vertex, the algorithm works on finding an initial edge in the cover mesh which has highest tetrahedron volume. Blind extraction method of this steganography system was based on trial-and-error method of finding the initial edge. Since an edge is composed of two vertices, for traversing mesh from the initial edge there are two equally likely options as shown in Fig. 2.3. This ambiguity was also resolved using trail-and-error method.

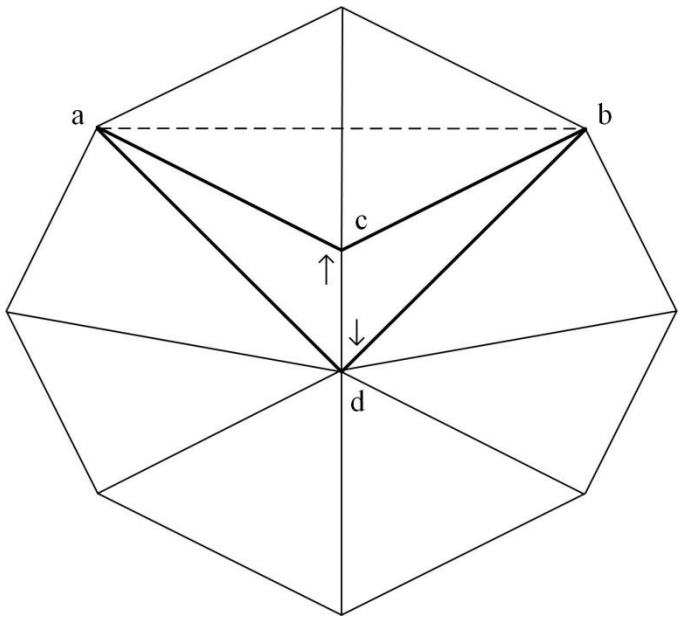


Fig. 2.3. Two equally likely options for mesh traversal [18]

Cayre et al. [29] proposed a Macro Embedding Procedure (MEP) based on the concept of TSPS algorithm. Considering AB as the entry edge in Fig. 2.4, orthogonal projection

of triangle summit C on edge AB defines the state of this triangle. There are two states such as S_0 and S_1. The present state of C is indicated by P(C). Thus, there are two cases, for k=0,1:

- $P(C) \in S_k$, no modification is needed
- $P(C) \notin S_k$, C needs to be shifted to new C' so that $P(C') \in S_k$

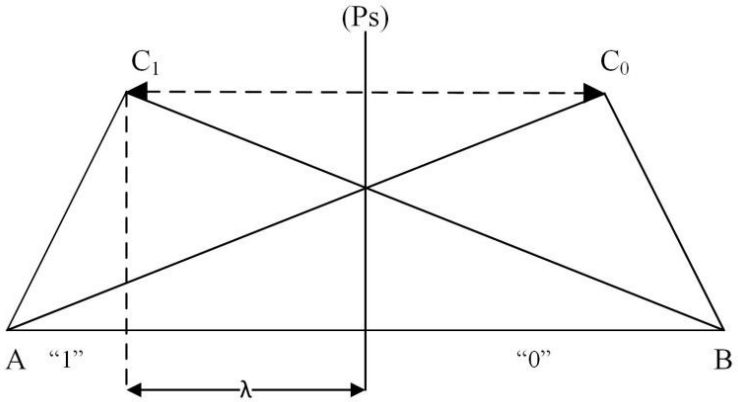


Fig. 2.4. Change of state for embedding secret bit 0 and 1

An erasing key recorded if any modification was done in order to embed the secret bit. Using erasing key steganography embedding effects were removed from the stego-mesh. Also, the proposed steganography algorithm withstands distortion less attacks. Difference in this scheme and TSPS algorithm is that in this algorithm, the triangle sequence generated need not be peeled off the mesh. This algorithm could withstand rotation, scaling and translation attacks but could not withstand the mesh simplification or remeshing attacks. A major drawback of this algorithm was its huge time complexity. If large 3D image models are used as cover model, then time taken up for embedding is too high.

Cheng et al. proposed three different steganography approaches. In [12], sequence list was obtained as a prerequisite of the embedding algorithm. In order to obtain the sequence list, the authors proposed three strategies, namely- triangle neighbour table, hierarchical k-dimension tree and advanced jump strategy. In triangle neighbour table, information about neighbours of a vertex was stored which was discovered from hierarchical k-dimension tree structure. The advanced jump strategy was based on MEP idea of Cayre et al. [29] and was called as Multi-level Embedding Procedure (MLEP). In MLEP, there are three levels, namely Sliding, Extending and Rotation level.

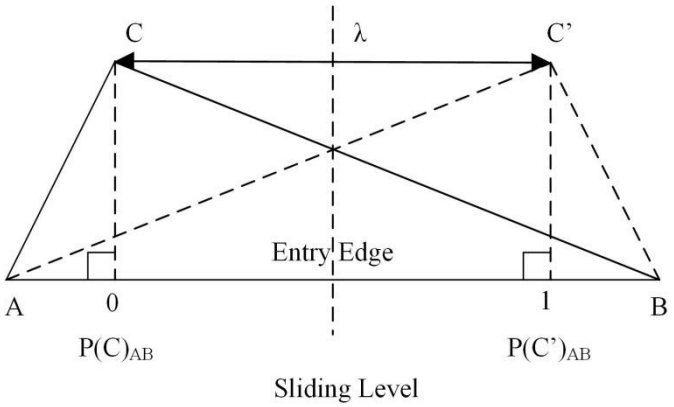


Fig. 2.5. Sliding level

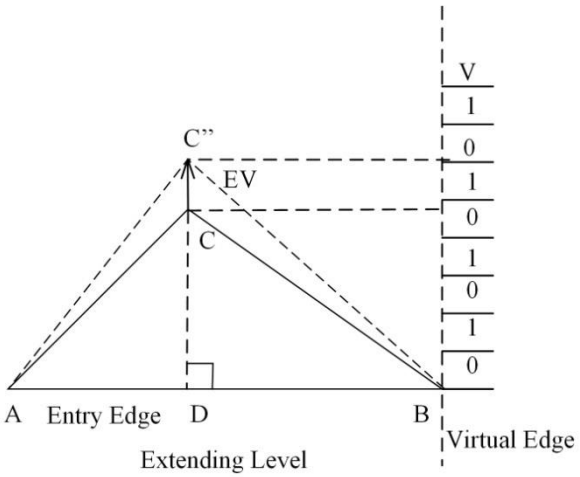


Fig. 2.6. Extending level

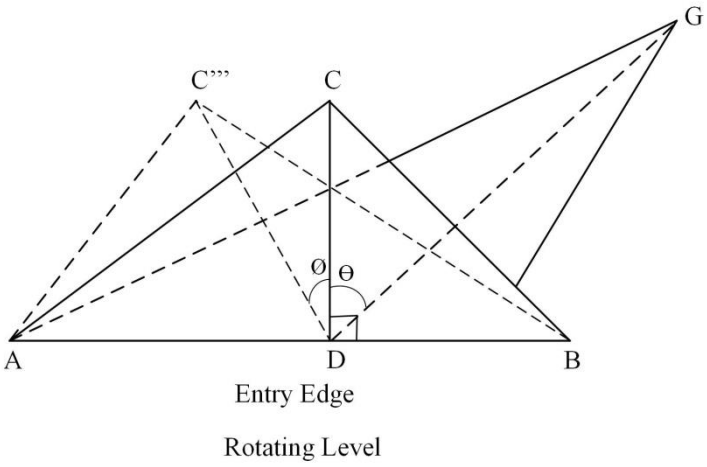


Fig. 2.7. Rotating level

In order to make the embedding in sliding level, the projection of triangle summit C has to be shifted as per the secret bit to be hidden in vertex C of the triangle. Interval AB is

divided into M_0 and M_1 as shown in Fig. 2.5. If $P(C)|_{AB} \in M_i$, for $i = 0,1$, then no modification is needed, otherwise C vertex has to be slided to C' so that $P(C')|_{AB} \in M_i$.

For extending level, embedding of message bits is done on the basis Distance Threshold Ratio (DTR) derived from Distance Threshold (DT). As shown in Fig. 2.6, VB is the virtual edge. Virtual edge is divided into M_0 and M_1intervals. For i=0,1, two cases occur in extending level, if the$\left(\frac{CD}{DT}\right)$ is multiple of 2, then it falls in M_0 interval. Otherwise, it falls in M_1interval. Adjustment in CD is done in order to embed secret bits by adding or subtracting EV in it, and thus shifting C to C''.

For rotating level, gravity centre G of each triangle is taken which makes θ with the orthogonal projection of triangle summit C. An angle $\emptyset$ is defined such that the ratio of θ and $\emptyset$ modulus 2 is divisible by 0. For hiding secret bits 0 and 1, two cases can be defined in order to keep P(C) in interval M_i for i = 0,1. Rotating level is illustrated in Fig. 2.7. These levels were defined so that the embedding is reversible and affine-invariant. A limitation of this algorithm was machine precision errors that may occur when small 3D image models are used as cover models.

Cheng et al. [14] proposed an adaptive steganography technique for 3D meshes. The embedding algorithm made up of two tables such as Vertex Body Table (VBT) and Polygon Neighbour Table (PNT). In VBT, the information about polygons sharing vertices is stored and in PNT information about neighbouring polygons sharing an edge is saved. Adaptive embedding was done for the first time in the history of 3D image steganography. This is done by embedding more secret bits in noisy surface than the smooth surface. The authors used the sliding, extending and rotating levels proposed in [12], and proposed Minimal Distortion Distance Estimation (MDDE). The proposed method has a blind extraction and resists the affine transformations.

Chao et al. [15] proposed Multi-layered Embedding Scheme for high capacity steganography system. Principal Component Analysis (PCA) projects all vertices of mesh onto the principal axes. Two extreme ends of the first principal axes are chosen as the end vertices. On the second principal axes, vertex on the furthest extreme end is taken as the third end vertex. The proposed approach was based on Quantization Index Modulation (QIM) [30, 31]. Single layer embedding was later extended to multi-layer embedding. The proposed embedding algorithm cannot withstand affine transformations

and vertex reordering attacks. The proposed algorithm cannot use perfect smooth 3D image models as the cover model.

Wu et al. [32] proposed steganography algorithm based on LSB+ algorithm in 2D image steganography [33]. In LSB hiding method of steganography, the least significant bit of pixels of cover image are changed in order to hide secret message bits. But as a result of this, the histogram of the cover image is affected. To preserve the histogram of cover image, LSB+ data hiding technique is proposed. LSB+ algorithm is based on Adjacent Bin Mapping (ABM) method. A typical histogram bin is shown in Fig. 2.8.

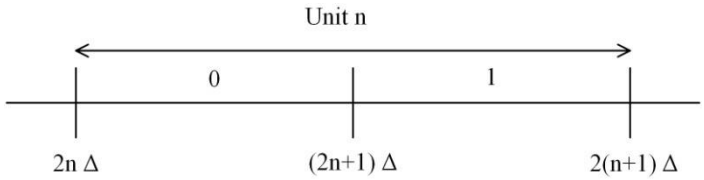


Fig. 2.8. Unit n in a histogram

Two adjacent bins of histograms are combined to form a unit. In each unit the number of 0's and 1's are preserved while embedding of secret bits. Applying same concept to 3D image steganography, three histograms are made for mesh vertices of mesh each for three coordinate axes. Extraction method of the system was not blind.

Chuang et al. [34] and Jhou et al. [35] used histogram shifting idea of Ni et al. [36] in reversible data hiding in 2D images. In histogram shifting method of reversible data hiding in 2D images, histogram of pixel values is drawn.

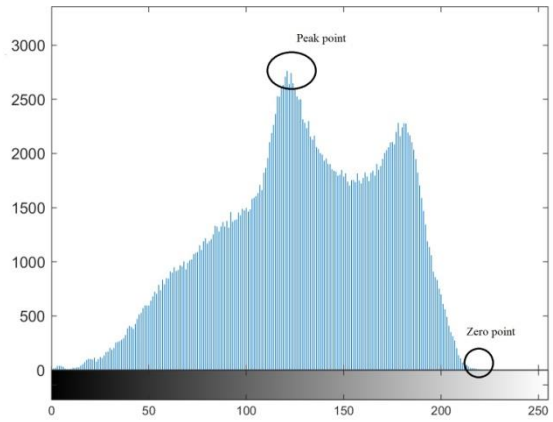


Fig. 2.9. Histogram of a 2D image showing peak point and zero point

In histogram, the intensity value for which number of pixels is maximum, is known as peak point (PP). Intensity value having minimum number of pixels is known as zero point (ZP). Peak point and zero point in the histogram are discovered as shown in Fig. 2.9.

In reversible data hiding, the cover media should be recovered after extraction of secret bits from stego-media at the receiver end. In this method, secret bits are embedded inside the frequency of the peak point as shown in Fig. 2.10.

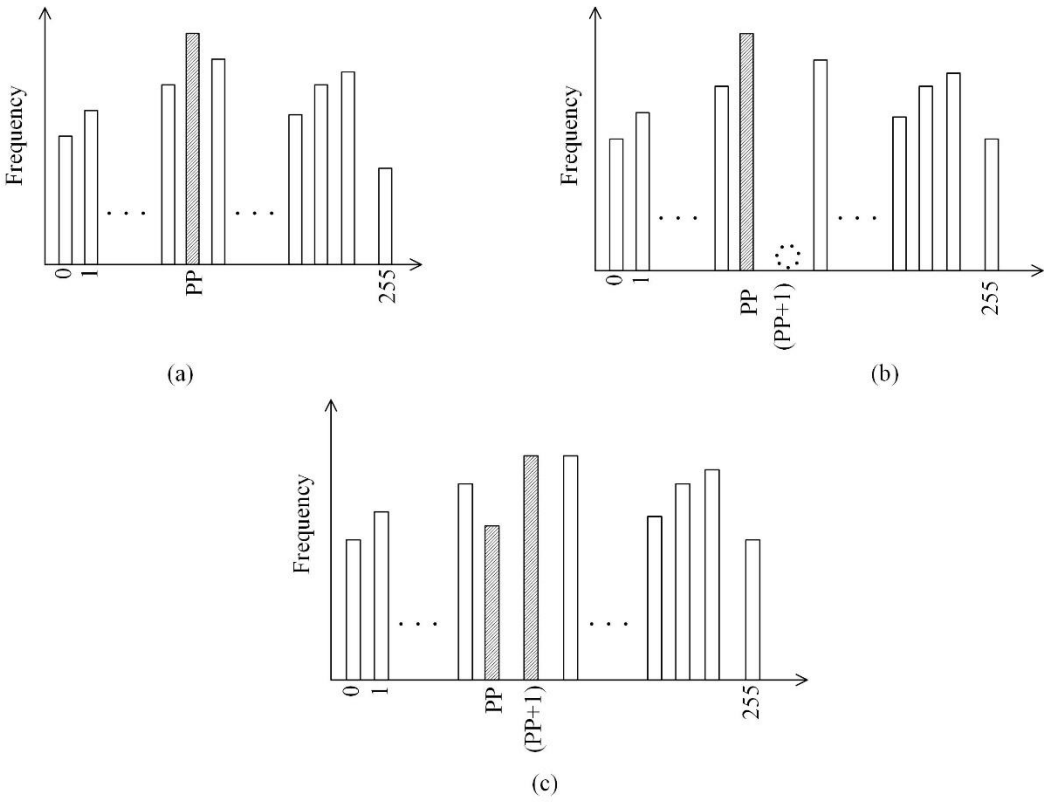


Fig. 2.10. Steps in histogram shifting(a) before embedding; (b) shifting histogram;(c) after embedding

1. Shift the entire histogram from intensity value of (PP+1) to ZP by 1. This is done by increasing the pixel intensity values of all the pixels in the range of [PP+1, ZP] by 1. This results in histogram being shifted from [PP+1, ZP] to [PP+2, ZP+1]. It creates a hole at intensity value (PP+1).
2. Now the image is scanned and as soon as a pixel with pixel value PP is found, secret bit from bit stream is taken. If the secret bit is 0 then no change is done. Otherwise, the pixel value of the pixel is increased by 1. Likewise, secret bits of

bit stream are embedded in the cover image.

Thus, secret bits are embedded in the cover image while retaining the original statistical properties of cover image. When the secret bits are extracted from stego-image at the receiver side then the original cover image is received.

In case of 3D image steganography, histogram is drawn for the distances of points from the gravity centre. After normalizing the distances with the largest distance and then multiplying with magnification factor, histogram is drawn. For instance, taking normalization factor 100, the histogram can be drawn for values 0 to 100. The histogram is made for the integral values in the range [0 100]. The fractional part obtained is also saved for each vertex so that the recovery of cover model can be done at the receiver side.

From the histogram, PP and ZP are obtained. Histogram is then modified as done above in 2D image steganography. This is done be increasing the distance of all vertices by 1 which have distances in the range [PP+1, ZP]. After obtaining histogram, bits of secret message are embedded one by one in the vertices. This is done by increasing the distance of vertices having distance equal to PP by 1 in case of '1' to be hidden. If the secret bit to be embedded is '0', then no change is done to the vertices.

After the extraction of secret bits is done from the stego-model, cover model can be obtained in its original form by constructing the histogram and obtaining PP and ZP. The entire histogram is then shifted to obtain the original histogram. From original histogram, distances are returned to their original locations.

Jhou et al.'s proposed algorithm [35] could not withstand affine transformations. In order to solve this problem, Chuang et al. [34] used Helmert transformations [37] and registration of stego-model.

Another difference between Jhou et al. [35] and Chuang et al. [34] scheme is that of embedding of secret bits in the cover model. In Jhou et al.'s proposed method [35], the secret bits are embedded directly into the vertices of mesh model. In Chuang et al.'s proposed algorithm, secret bits are embedded by modifying the normalized distances of model centre and vertices of the mesh model.

Another reversible data hiding technique in 3D image steganography has been proposed by Ji et al. [38]. This technique is based on difference expansion technique of 2D image steganography proposed by Tian et al. [39] Difference expansion technique in 2D image steganography is explained taking an example.

Assuming a pair of pixels valued at 201 and 207 is there in cover image. Secret bit to be hidden is 1. Then, average and difference of the two intensity values is calculated. Average and difference come out to be 204 and 6 respectively. Difference is then converted from decimal to binary, i.e. $(6)_{10} = (110)_2$. Then, secret bit is appended to the difference and new difference is obtained. In this case, the new difference is $(1101)_2 = (13)_{10}$. New pixel values can be obtained from the modified difference values which are 198 and 211 in this case. Thus, the difference between the pixels is *expanded* from 6 to 13.

At receiver side, the difference between the modified pixel values is calculated as 13(=211-198). Then this difference is converted to its binary representation and LSB (Least Significant Bit) is extracted from it. Thus, secret bit is obtained and original pixel values can be restored with original difference and average values.

In 3D image steganography, Ji et al. [38] proposed ratio expansion technique for reversible data hiding based on difference expansion technique. For a particular mesh vertex, its distance with the gravity centre is calculated. Then, its distance with all its neighbouring vertices is calculated. Ratio of distance between a vertex and its neighbouring vertex and distance between it and the gravity centre is obtained. In order to hide secret bits this ratio is modified. The proposed algorithm was shown to resist rotation, translation and uniform scaling. Distortions caused due to embedding of secret bits are also low. However, the embedding capacity was also very less.

Anitha et al. [40] proposed a new algorithm for embedding secret bits inside the 3D model by modifying LSB (Least Significant Bit) of vertices. For embedding of secret bits, an initial triangle is constructed as shown in Fig. 2.11.

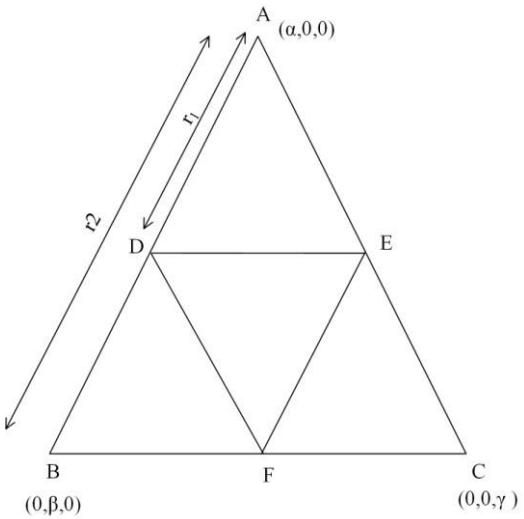


Fig. 2.11. Triangle subdivision

This triangle is constructed by taking maximum values of vertices on the three coordinate axes. In ΔABC, α is the maximum value of vertices on x-axis, β is the maximum value of vertices on y-axis and γ is the maximum value of vertices on z-axis. These points are discovered and a triangle is formed. This triangle is formed only for the purpose of embedding. For obtaining D, E and F points on ΔABC, decomposition ratio and length of edges AB (r_2 in this case), BC and AC are used as shown in equation below.

$$r_1 = decomposition\_ratio \times r_2 \tag{2.1}$$

Smaller triangles are constructed in a similar fashion until a threshold is reached. After triangle subdivision process is complete, embedding of secret bits is done at these newly formed vertices. Two or three bits are embedded at each vertex depending upon the stego-key used in the proposed steganography system. The blind extraction method required stego-key for extracting secret binary bits. The proposed steganography algorithm resists rotation, scaling and cropping attacks. However, the proposed system cannot withstand vertex reordering attack.

Tsai et al. [41] proposed an improvement over the reversible data hiding scheme using histograms proposed by Chuang et al. [34] and Jhou et al. [35]. In this method, the distances between neighbouring vertices was taken for constructing histogram in place of distance between vertex and centre. In this steganography system, mesh traversal algorithm was also proposed in order to determine a unique referencing order for every 3D mesh. The proposed mesh traversal algorithm was based on BFS (Breadth First

Search) algorithm. The ambiguity in visiting the vertices was resolved by measurement of included angle between the ray joining the vertex and gravity centre and its normal. Vertices having smallest included angle would be visited first. Distortions on 3D mesh model increased when embedding capacity was increased. For a huge 3D mesh model, more than one vertex can have same included angle value. Thus, the mesh traversal algorithm could give more than one traversal orders for a given 3D mesh. Also, the proposed algorithm withstands affine transformations because of PCA (Principal Components Analysis) performed over the stego mesh.

Li et al. [42] proposed a steganography algorithm which embeds secret bits in 3D model with adjustable distortion. The authors constructed a truncated space first and then embedding of secret bits is done by making use of the proposed shifting strategy and a secret key. The shifting strategy used for embedding made use of the secret key constructed. The embedding is done on the vertices represented in truncated space. The distortions caused to the 3D mesh model because of embedding were less. The performance of the proposed system was measured in terms of PSNR (Peak Signal to Noise Ratio) and compared with some of the steganography systems. Although the embedding capacity achieved is high, the proposed steganography algorithm could weakly resist the vertex reordering attack. Also, it fails to withstand noise and smoothing attack on the stego-model. As PCA (Principal Components Analysis) is used in this system, it fails to work on uniform 3D mesh models.

Anish et al. [43] proposed 3D image steganography based on PCD file format of 3D mesh models. In this steganography system, data is hidden by manipulating x-coordinate of cover 3D image models. Secret text is considered as a stream of ASCII characters and hidden inside the fractional part of x-coordinates. Performance of the proposed system was not measured in terms of embedding capacity. However, distortions caused to stego model were measured in terms of SNR (Signal to Noise Ratio).

Prediction Error Expansion strategy proposed by Thodi et al. [45] for 2D image steganography is another reversible data hiding approach used in 3D image steganography. In Prediction Error Expansion scheme, intensity of a pixel is first predicted based on its context. Different algorithms are used for predicting intensity of a pixel based on the context. One of the approaches for finding the predicted intensity is mentioned below.

c1	c2
c3	p

Fig. 2.12. Context of pixel p

In [47], predicted intensity $\tilde{p}$ can be predicted by following equation.

$$\tilde{p} = \begin{cases} \max(c2, c3), & if\ c1 \leq \min(c2, c3) \\ \min(c2, c3), & if c1 \geq \max(c1, c3) \\ c2 + c3 - c1, & otherwise \end{cases} \quad (2.2)$$

Where p, c1, c2 and c3 are pixel values in its context as shown in Fig. 2.12 above. From $\tilde{p}$, $\hat{p}$ (predicted pixel value) can be found using equation written below.

$$\hat{p} = 2 \times \lfloor \tilde{p}\ /2 \rfloor \quad (2.3)$$

After discovering the predicted intensity value $\hat{p}$, prediction error is calculated.

$$e = p - \hat{p} \quad (2.4)$$

Secret bit is embedded is done in the prediction error e as:

$$e_s = 2e + b \quad (2.5)$$

where e_s is the changed prediction error with binary bit b embedded in it.

Thus, the changed pixel value can be obtained as:

$$p_s = \hat{p} + e_s \quad (2.6)$$

where $p_s$is the changed pixel value with binary bit b embedded in it.

At the extraction side, embedded binary bit is extracted from LSB of $p_s$and the original pixel value is obtained.

$$p = \hat{p} + \lfloor e_s/2 \rfloor \quad (2.7)$$

Based on this, Zhang et al. [44] proposed reversible data hiding scheme in 3D image steganography. Prediction of value is based on a novel prediction algorithm proposed in this research work. Adaptive embedding of secret bits was done in order to reduce

distortions. In case of large amount of secret data to be hidden, multiple layers of embedding were done to hide the secret bits. In order to make the steganography effects reversible on 3D cover model, some auxiliary information was sent along with 3D stego-model. This auxiliary information composed of a Location Map (LM), an Ending Marker (EM) and the value of threshold(T). Extraction of the secret bits from stego-model was blind. SNR (Signal to Noise Ratio) was calculated to detect the distortions to the cover model. Table 2.2 mentions geometrical domain-based steganography approaches over the time.

Table 2.2. Comparison of various approaches in Geometrical domain

Year	**Authors**	**Algorithm/ Technique**	**Reversible**	**Blind**	**Withstands geometrical transformations**
1998	Ohbuchi et al. [18]	Triangle Similarity Quadruple Embedding, Tetrahedral Volume Ratio Embedding	No	Yes	Yes
2003	Cayre et al. [29]	Macro Embedding Procedure	Yes	Yes	Yes
2005	Cheng et al. [12]	Multi-Level Embedding Procedure	Yes	Yes	Yes
2007	Jhou et al.[35]	Histogram shifting	Yes	Yes	No
2007	Cheng et al. [14]	Adaptive Minimum-distortion Estimation	No	Yes	Yes
2009	Chao et al. [15]	Multi-layered Embedding Scheme	No	Yes	No

2009	Wu et al. [33]	Adjacent Bin Mapping method	No	Yes	No
2010	Ji et al. [38]	Ratio Expansion based algorithm	Yes	Yes	Yes
2010	Chuang et al. [34]	Embedding using Histogram Shifting	Yes	Yes	Yes
2013	Anitha et al. [40]	Embedding after Triangle Mesh Formation	No	Yes	Yes
2015	Tsai et al. [41]	Embedding based on Histogram Shifting	Yes	Yes	Yes
2017	Li et al. [42]	Adaptive embedding in truncated space	No	Yes	Yes
2017	Desai et al. [43]	Embedding in x-coordinate of vertex	No	Yes	No
2018	Zhang et al. [44]	Adaptive embedding based on prediction-error expansion scheme	Yes	Yes	No

2.2.2 Topological domain-based steganography

In topological domain-based 3D image steganography, connectivity of vertices or topology of mesh model is slightly modified in order to embed secret binary bits [22]. As connectivity information in 3D model is less than the geometrical primitives that can be used up for embedding secret bits. Algorithms in topological domain-based steganography are less in number as compared to geometrical domain-based steganography. These algorithms can withstand geometrical transformations because the connectivity information remains intact when geometrical transformations are done to the mesh. However, these algorithms are vulnerable to mesh simplification, vertex

reordering and other topological modification transformations. Fig. 2.13 shows the development of topological domain steganography approaches over the past years.

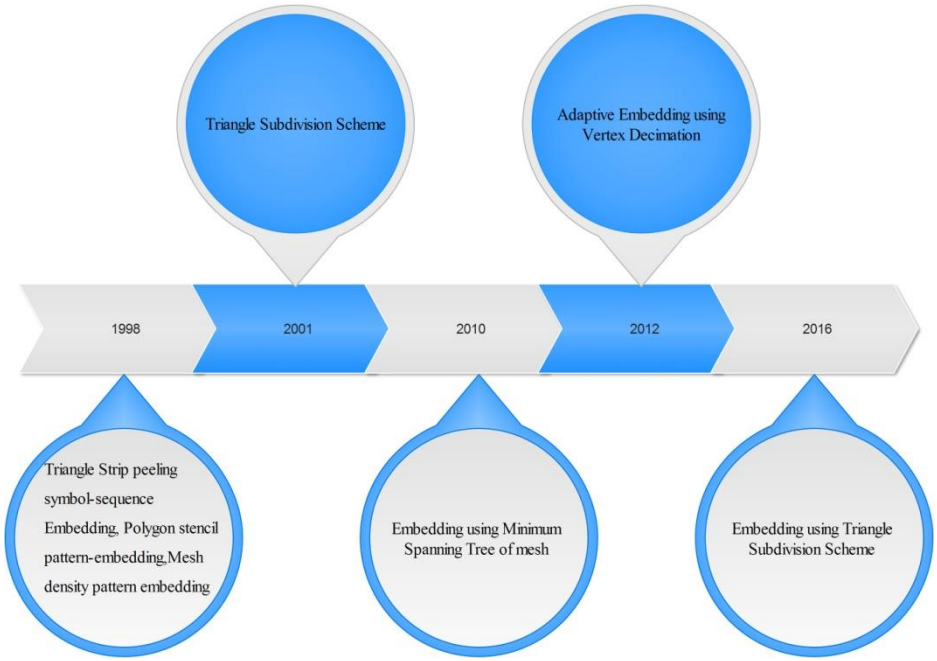


Fig. 2.13. Timeline showing topological domain-based steganography techniques

Ohbuchi et al. [18] enlisted some of the topological primitives that can be used for embedding of secret data. They proposed three algorithms for embedding of secret data, such as:

1. Triangle Strip peeling symbol-sequence Embedding
2. Polygon stencil pattern-embedding
3. Mesh density pattern embedding

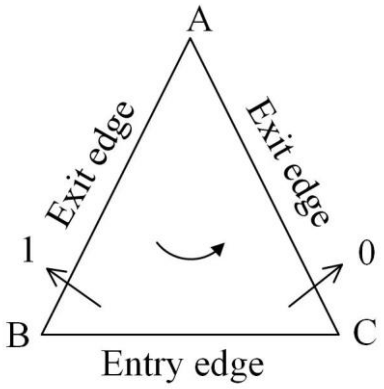


Fig. 2.14. Entry edge and exit-edges in a triangle

Triangle Strip Peeling Symbol-sequence-embedding (TSPS) embeds secret bits by making use of adjacent triangles in a mesh. By traversing over triangles of the mesh, secret bits are embedded. As shown in Fig. 2.14, a triangle can be thought of as one entry edge and two exit edges.

In this triangle, edge BC is the entry edge and there are two exit edges, AB and AC. If after entering into the triangle from edge BC, exit edge AC is chosen (which is present in its clockwise direction), then secret bit embedded is 0. Likewise, if after BC, edge AB is visited (present in the anti-clockwise direction), then secret bit embedded is 1. In a similar manner, all the secret bits are embedded by traversing triangles and forming a triangle strip as shown in Fig. 2.15.

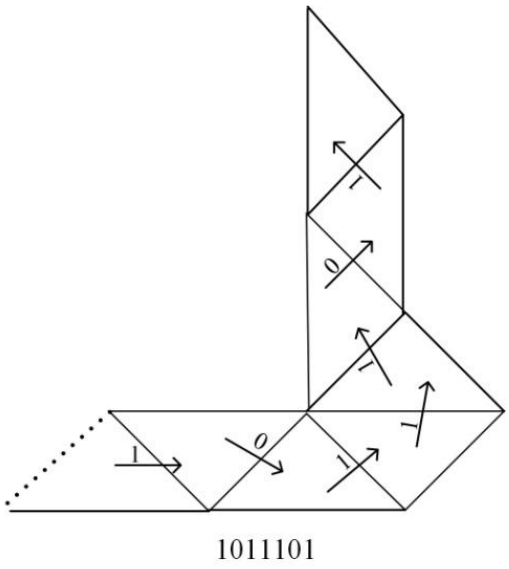


Fig. 2.15. TSPS algorithm for embedding ‘1011101’

This strip is then peeled off from the mesh except the starting initial edge. The termination condition is the open end of the triangle strip. This steganography system worked on blind extraction method. Low space efficiency and inability to resist polygon simplification attack are some of the disadvantages of this algorithm.

Second algorithm proposed was Polygon stencil pattern-embedding. This algorithm is similar to TSPS proposed above. Difference in TSPS algorithm and Polygon stencil pattern-embedding algorithm is that a pattern can be embedded in the latter while in case of former, binary bits can be embedded. Polygon stencil pattern-embedding algorithm is used for watermarking purposes.

Third algorithm proposed was Mesh density pattern embedding. The embedding algorithm was used for watermarking symbols onto the mesh. It cannot withstand polygon simplification attack.

Mao et al. [47] argued that the ratio of two-line segments on a line remains unchanged even when the line is subjected to geometrical transformations as shown in Fig. 2.16.

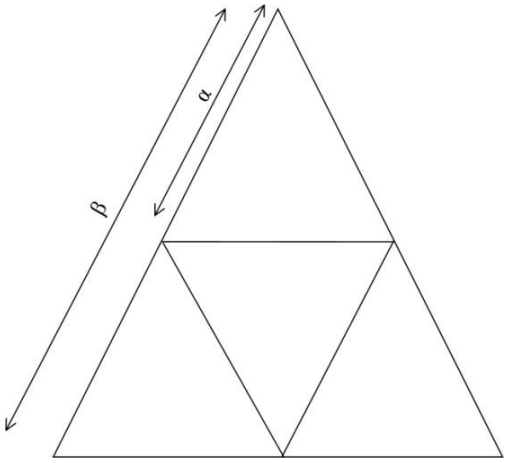


Fig. 2.16. Affine invariance of ratio of line segments in mesh

The triangle subdivision process is carried out on the triangle mesh. New vertices are created with the help of pseudorandom numbers generator. The ratio in which an edge of a triangle is subdivided to create a new triangle is determined by the secret data to be embedded. For embedding of secret bits, those triangles are searched in mesh present on a same straight line. The shared vertex was then shifted slightly from its position to hide the secret bits. The blind extraction of secret data requires same seed as that used in the embedding process for pseudorandom number generation. The algorithm withstands affine transformations and mesh simplification attacks. Major drawbacks in this approach are listed below [48].

1. Addition of new vertices to the mesh increases the size of 3D mesh model.
2. Imperceptibility of embedded bits is less. Thus, distortions in 3D mesh are visible to human eye.
3. As per the adjacency property of 3D mesh, each edge present inside 3D mesh should have only two sides. In case of edges present on the outside of mesh, there is only one side of the edge. In this work, 3D stego mesh did not maintain this adjacency property.

Puech et al. [48] used Minimum Spanning Tree (MST) in the proposed topological based steganography algorithm. The algorithm consists of three steps such as

1. construction of MST in the mesh;
2. analysing embedding areas in MST; and finally
3. embedding of secret data bits by joining the common edge or uncommon edge in between two triangles.

For construction of MST of the cover 3D mesh model, starting point in the mesh is taken from the PCA (Principal Components Analysis). PCA determines the three directions in which the cover 3D mesh model should be placed. MST is taken as a prerequisite for embedding of secret bits because it is unique for a particular given mesh. The weights in a MST are taken as Euclidean distances between the vertices in the mesh. After construction of MST, selection of triangles is done as shown in Fig. 2.17.

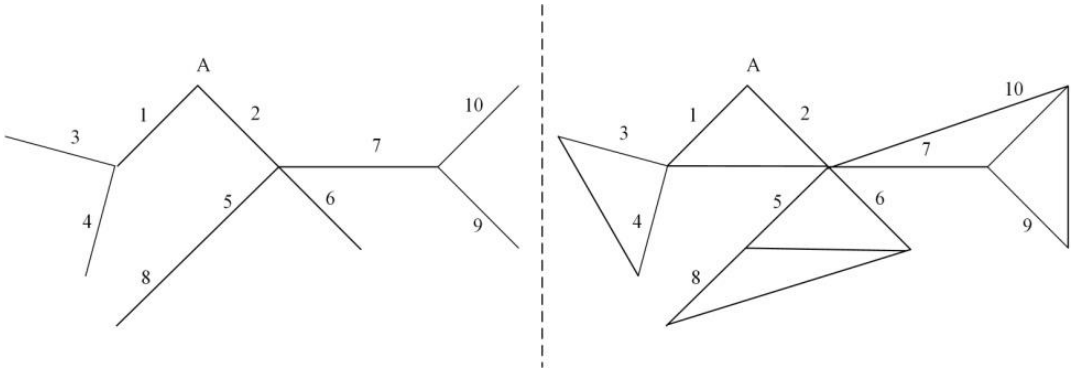


Fig. 2.17. Selection of vertices involved in making MST

After selection of triangles, synchronization of quadruples is done in order to hide secret bits. For embedding of secret bits, connectivity of triangles is modified. Since modifying the topology of the mesh could result in errors causing the surface to become disorderly. The proposed method is lossless as no new edges are formed while embedding is carried out. The extraction of secret data bits from stego-model does not need the cover model. Some constraints such as co planarity constraint, convexity constraint and overlapping constraint restrict the embedding of secret bits into 3D mesh model. Thus, the embedding capacity of this method is very low and is only 0.128 bit per vertex for some 3D objects. However, this method withstands rotation, scaling and translation attacks.

Tsai [16] proposed an adaptive embedding algorithm for 3D image Steganography that is based on vertex decimation by Schroeder [49]. Information from vertex decimation is

used for embedding secret data bits. First vertex decimation step is done in which vertices, edges and faces are removed so that the resulting mesh is rendered at a high speed. For this selection of vertices is done using three different orders such as Input-First (IF), Minimum-Neighbours-First (MinNF) and Maximum-Neighbours-First (MaxNF). The embedding is done by taking 3D cover model to PCA coordinate system. As shown in Fig. 2.18, V^E is the embedding vertex that is slightly shifted in order to embed secret bits.

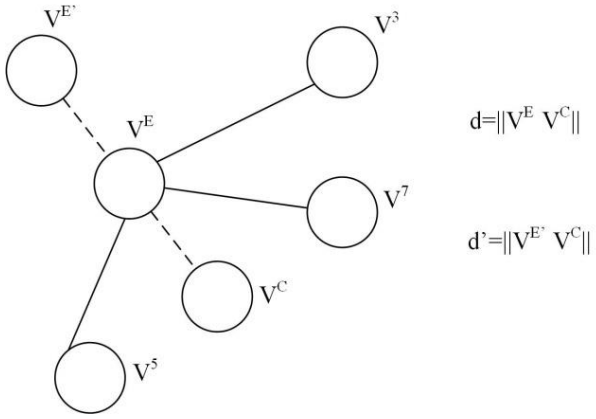


Fig. 2.18. Slight modification in embedding vertex with respect to gravity centre

The embedding capacity of the proposed approach was proportional to distance between embedding vertex and the centre of 3D mesh model. This makes 3D stego-model resistant to the rotation, scaling and translation transformations. The blind extraction algorithm has high embedding capacity and less distortion.

Tsai [50] proposed steganography scheme in 3D image mesh models by modifying the topology of the cover model, using recursive triangle subdivision process. Subdivision of triangles is done. New vertices were added to 3D mesh model during triangle subdivision process. As a result, file-size of 3D mesh model increased. Stopping condition for subdivision process is the preciseness of 3D mesh model. It is robust against the vertex reordering attack. The blind extraction of secret message from the stego-model fails when the stego-model is attacked by noise. However, the proposed approach has high embedding capacity.

Table 2.3 compares various topological domain-based 3D image steganography approaches developed over the past few years. As can be observed from the table, all the

topological domain based algorithms resist geometrical transformations. However, these algorithms are not reversible except [50]. This implies that the changes done to 3D cover models for embedding of secret bits cannot be reversed.

Table 2.3. Comparison of various approaches in topological domain

Year	**Authors**	**Algorithm/ Technique**	**Reversible**	**Blind**	**Withstands geometrical transformations**
1998	Ohbuchi et al.[18]	Triangle Strip peeling symbol-sequence Embedding, Polygon stencil pattern-embedding, Mesh density pattern embedding	No	Yes	Yes
2001	Mao et al. [47]	Triangle Subdivision Scheme	No	Yes	Yes
2010	Puech et al. [48]	Embedding using Minimum Spanning Tree of mesh	No	Yes	Yes
2012	Tsai [16]	Adaptive Embedding using Vertex Decimation	No	Yes	Yes
2016	Tsai [50]	Embedding using Triangle Subdivision Scheme	Yes	Yes	Yes

2.2.3 Representation domain-based steganography

Representation based steganography algorithms uses redundant information in representation of mesh model [22]. The redundant information in mesh is added and modified later so that the secret message can be hidden in the 3D model. However, the payload capacity of representation-based methods is less than above two mentioned above. This is the reason of a smaller number of contributions made in this area. As can

be observed from Fig. 2.19 showing the timeline of representation domain-based approaches, only two approaches have been proposed in this domain so far.

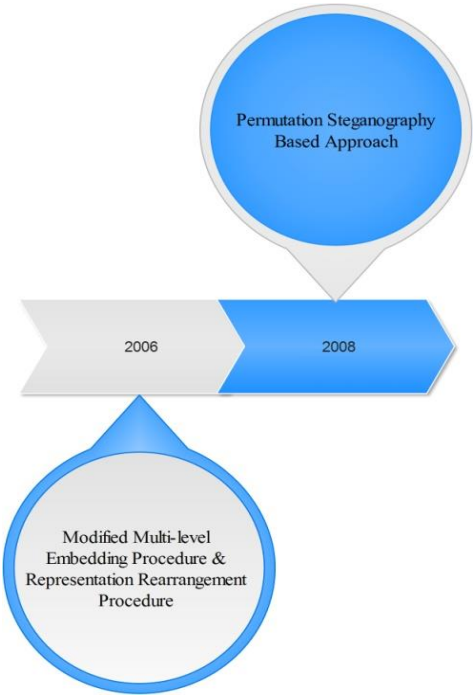


Fig. 2.19. Timeline showing representation domain-based steganography techniques

Cheng et al [13]. used representation domain for embedding of secret bits for the first time in 3D Image Steganography. They used representation domain along with the proposed geometrical domain-based steganography algorithm. Thus, the embedding capacity of the total system (geometrical-domain based algorithm and representation-domain based algorithm) is high. 3D mesh model compression algorithms are based on the fact that changing the order of vertices and polygons arbitrarily does not affect 3D image model [51,52]. Using this idea, vertices and polygons are rearranged in the mesh model. Then, one bit per vertex and one bit per polygon embedded. The blind extraction of secret data requires just secret key (which is an added layer of security). As the embedding is done in the representation domain, it withstands geometrical transformations.

Bogomjakov et al. [53] proposed a time efficient algorithm using indexed representation of the mesh for hiding secret data. The proposed algorithm is based on permutation steganography. In permutation steganography, elements are ordered in any arrangement.

The difference in the arrangement with a particular reference order is the hidden secret message. The order in which faces and vertices are stored is rearranged. This rearrangement is done according to the secret data to be embedded. As it requires a reference order, the extraction is not blind. This algorithm is robust against affine transformations. It has a fairly good embedding capacity. Table 2.4 summarizes the two 3D image steganography algorithms proposed in representation domain.

Table 2.4. Comparison of various approaches in representation domain

Year	**Authors**	**Algorithm/ Technique**	**Reversible**	**Blind**	**Withstands geometrical transformations**
2006	Cheng et al. [13]	Modified Multi-level Embedding Procedure & Representation Rearrangement Procedure	No	Yes	Yes
2008	Bogomjakov et al. [53]	Permutation Steganography Based Approach	Yes	Yes	Yes

2.3 Research gaps

Based on the literature survey done above, the following research gaps are identified.

1. Lack of steganography algorithm having high embedding capacity and less distortion

3D image steganography algorithms have low embedding capacity and do not fully utilize the carrying capacity of 3D image model. If embedding capacity is increased, then distortions caused are inevitable. Thus, there is a need to design a 3D image

steganography algorithm such that the embedding capacity of 3D mesh models is fully utilised without causing much distortion to the 3D mesh model.

2. Robustness against affine transformations attack

Affine transformations can be caused to 3D mesh model without having knowledge of 3D mesh model. These transformations can destroy the hidden secret bits inside the 3D mesh model. Thus, robustness of 3D mesh models against these transformations should be ensured. Hidden secret bits should not be destroyed even if the stego-model is attacked by rotation, scaling and translation.

3. Scrambling of secret data before embedding

Secret data is embedded as such in the 3D mesh model. Steganalysis may reveal the pattern of hiding secret bits inside 3D mesh model. Thus, the secret message may reach an undesired destination. Thus, secret data should be scrambled first and then hidden inside 3D mesh model.

Thus, there is a need to develop 3D image steganography algorithm which utilizes embedding capacity without causing much distortion to the 3D mesh model. Also, the proposed 3D image steganography should be able to withstand rotation, scaling, translation and vertex reordering attacks. The secret message bits should be scrambled before embedding it into 3D mesh model. Embedded secret bits in the proposed 3D image steganography algorithm should not cause any visible distortion to the 3D surfaces.

2.4 Objectives

To address the aforementioned research gaps, the following objectives were identified for the research work:

1. To study and analyse 3D image model steganography.
2. To develop a proficient scrambling technique for secret data.
3. To develop novel 3D image steganography techniques for achieving the trade-off between distortion and capacity.
4. To validate novel 3D image steganography techniques on various types of attacks.

2.5 Summary

In this chapter, a detailed literature survey of the steganography approaches proposed in literature has been done. Steganography approaches proposed in spatial domain and transform domain are discussed in detail in the chapter. Spatial domain-based techniques can be further classified into three types - geometrical domain, topological domain and representation domain. Various algorithms proposed in literature in the above-mentioned domains are explained in detail in this chapter. Ability of steganography approaches to withstand attacks is also discussed in this chapter. As geometrical primitives of 3D mesh models for embedding are more in number than that of topological primitives, more steganography approaches are based in geometrical domain. Representation domain based steganography algorithms lags behind both of them in number because of even less embedding primitives available in this domain.

Chapter 3
Image Encryption

This chapter discusses image encryption algorithm proposed for encryption of color images using DNA cryptosystem [65] and Lorenz-Rossler hyper-chaotic system [88]. The proposed image encryption algorithm enhances information entropy in plain images by using chaotic sequences obtained from the hyper-chaotic system. Inter-channel operations between red, green and blue channels increase information entropy further more. A unique encrypted image is generated for different plain colour images. Using two chaotic sequences increase key space; thus, providing resistance against brute-force attacks.

3.1 Introduction

Image encryption is the process of conversion of a meaningful image into some random arrangement of pixels such that its intelligible property is destroyed. Image encryption process transforms a plain image into a jumbled collection of pixels (encrypted image). Image encryption process has two properties- permutation (also called confusion) and substitution (also called diffusion). In permutation, pixels in plain image are interchanged in such a way that the confused image and plain image do not look similar at all. In substitution, the pixel intensities are changed using some function which enhances randomness in plain image.

Encryption of image cannot be done using traditional block ciphers such as Data Encryption Standard (DES), Advanced Encryption Standard (AES), and International Data Encryption Algorithm (IDEA). These ciphers cannot be used for encryption of image because of its storage format [66]. Hence, different image encryption algorithms are devised for encrypting plain images.

Use of chaotic functions in image encryption process by Matthews [106] in 1989 encouraged its use in the research works of image encryption. Properties of chaotic system such as sensitivity to initial conditions, pseudo-random behaviour, complex

structures, etc approve their usefulness in image encryption. In case of permutation, the pixel values of image are interchanged with the help of chaotic systems in a pseudo-random way. In case of substitution, pixel values of image are enhanced in pseudo-random fashion in order to enhance randomness in image.

In this chapter, image encryption algorithm uses Lorenz-Rossler hyper-chaotic system [88]. DNA cryptosystem rules are applied to encode the pixel values of RGB planes.

3.2 DNA cryptosystem

A DNA structure comprises of four nucleotide bases- Adenine (A), Thymine (T), Cytosine(C) and Guanine (G). These four nucleotides follow complementary relationship with one another as per Watson-Crick rule [74]. This implies that if A is given value '00', then T should be assigned '11'. Similarly, for C and G bases, values are assigned in complimentary fashion. A pair of bits is assigned to one nucleotide base. Since RGB planes in color image are 8-bit channels, there are four nucleotides for a particular pixel value. This implies that conversion of a decimal valued pixel will have a four nucleotide DNA sequence as its equivalent. Only $8(= 2! \times 2! \times 2!)$ combinations out of $24(= 4!)$ possible combinations are allowed. These encoding rules are mentioned in Table 3.1.

Table 3.1. Encoding rules for DNA sequences [65]

Rule No.	**DNA Nucleotides**			
	A	**T**	**G**	**C**
1	00	11	01	10
2	00	11	10	01
3	11	00	01	10
4	11	00	10	01
5	10	01	11	00
6	01	10	11	00
7	10	01	00	11
8	01	10	00	11

Applying these rules, pixel values are converted to DNA sequences. For example, pixel value $(45)_{10}$ is converted into binary format first, which is$(00101101)_2$.

Using rule no.1, this pixel is converted to DNA sequence – ACTG. Similarly, all the pixels are converted to DNA sequences by applying these rules mentioned in Table 3.1. Encoding of pixel values into DNA sequences is done taking into account the location of the particular pixel. Operations such as XOR, addition and subtraction can be applied on DNA sequences [75]. Results obtained by applying operations on these two nucleotides are mentioned. By assigning values from a particular rule of Table 3.1 to the four nucleotides, these results can be obtained. For instance, if rule no. 6 is applied, then A-01, T-10, G-11 and C-00 values will be assigned. Now, C ($\oplus$) T, i.e. 00($\oplus$) 10 is done. This gives 10 as a result, which is T. In a similar way all the entries in Table 3.2 are generated. For addition and subtraction operations also, entries are generated in a similar order and Tables 3.3 and 3.4 are made.

Table 3.2. XOR operation on DNA nucleotides

$\oplus$	**A**	**T**	**C**	**G**
A	C	G	A	T
T	T	A	T	A
C	A	T	C	G
G	T	A	G	C

Table 3.3. Addition operation on DNA nucleotides

+	**A**	**T**	**C**	**G**
A	T	G	A	C
T	G	C	T	A
C	A	T	C	G
G	C	A	G	T

Table 3.4. Subtraction operation on DNA nucleotides

-	**A**	**T**	**C**	**G**
A	C	G	A	T
T	A	C	T	G
C	G	T	C	A
G	T	A	G	C

Rule number to be applied for encoding of DNA sequences depends upon the location of pixel in image matrix. It is given by the following equation.

$$Rule\_number \;=\; (i+j) mod\ 8 \;+1 \tag{3.1}$$

where i and j are row and column index of a pixel respectively.

Using this equation, different DNA sequences can be generated for same pixel value at different locations, for example, suppose $(18)_{10}$ pixel value is present at $(2,5)$ and $(14{,}10)$ locations. Binary equivalent of $(18)_{10}$ is $(00010010)_2$. For pixel value at $(2,5)$, rule no. 8 is applied and DNA sequence comes out to be GAGT. For pixel value at $(14{,}10)$ location, rule no. 1 will be applied and the DNA sequence is AGAC. Thus, two different DNA sequences are generated for same pixel intensity value. It enhances the randomness in image.

All the pixel values in RGB planes are thus converted to DNA sequence representations. This is done by first converting the decimal values pixels to their binary equivalents. Then, 0's are padded in front in order to make the binary equivalents in 8-bit formats. Afterwards, rules are applied according to the location of pixels in RGB planes. This changes dimensions of decimal-valued RGB matrices. Assuming that the dimension of RGB matrices is 128×128, conversion to DNA sequence, it becomes 128×512. For each decimal valued pixel, a four-length DNA sequence is generated. This increases the columns from 128 to 512(=128×4). Number of rows remains intact.

3.2.1 Hamming distance

Hamming distance in a pair of sequences is measured as the least number of substitutions that needs to be in order to make the two sequences exactly similar. Hamming distance values can be obtained for two DNA sequences [77]. For example, hamming distances for two DNA strands- 'ATTGCT' and 'GCTATT' needs to be calculated. The process is shown in Fig. 3.1.

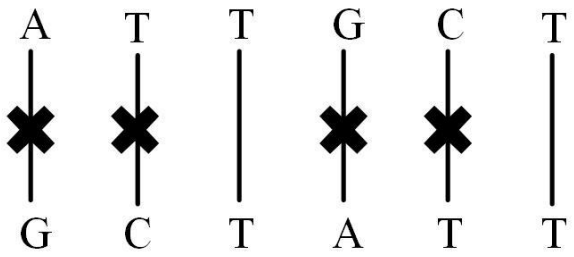


Fig. 3.1. DNA strands mismatches

It can be observed from these two DNA sequences, that both sequences are of length 6 whose hamming distance has to be evaluated. Substituting A and G on first place, T and C on second place, G and A on fourth place and C and T on fifth place gives similar sequences. Thus, total four substitutions are required in this case. Hence, hamming distances for this pair of sequences is four.

In a similar manner, hamming distance values can be obtained for DNA sequences generated from Red, Green and Blue channels of plain image. Three pairs can be formed for hamming distance calculation. These pairs are- Red & Green (R&G), Green & Blue (G&B), and Blue & Red (B&R). The difference between DNA sequences of two channels is obtained by reading the DNA sequences pixel-by-pixel and comparing the values. The numbers of mismatches are counted and added to give the hamming distance value of two channels. The pseudo-code for finding hamming distance between a pair of strings is given below.

Algorithm 3.1: Calculate Hamming Distance

Input: String str_one, str_two

Result: hamming_count

1. hamming_count = 0 //both str_one and str_two are assumed to have equal length
2. position=1
3. **while** position <= str_one.length
4. **if** str_one[position]! = str_two[position] //compare genes at *position*
5. hamming_count = hamming_count + 1
6. **end-if**
7. position = position + 1
8. **end-while**

Here, position denotes the position of pointer in string that is being compared at a time. As length of two strings being compared is taken to be same. So, for every mismatch in the two strings, variable hamming_count is increased by one. Total number of mismatches in the two strings is found which gives hamming distance between them.

Hamming distance between R and G is denoted byh_{rg}. Likewise, hamming distance between G and B is denoted by notation- h_{gb} and that between B and R is denoted as h_{br} . These values are modified which can be used in the image encryption process. These values are divided by a suitable power of ten so that these values become less than one.

$$h_{rg} = h_{rg} \div 10^{m}, h_{gb} = h_{gb} \div 10^{n}, h_{br} = h_{br} \div 10^{p} \quad (3.2)$$

where m, n and p are suitable powers of ten for h_{rg}, h_{gb} and h_{br}respectively.

This results in generation of unique chaotic sequences for each plain image. A slight change in initial conditions results in a completely different chaotic sequence [65]. This is because of the fact that the initial conditions are modified as per the pixel values of red, green and blue channels. As pixel values of these three channels are utilized in forming the initial conditions, a unique mask image is generated for each plain image. This helps the encryption algorithm to generate a unique mask image for each plain image. Thus, the attackers cannot identify the pattern of conversion of plain image into encrypted image.

3.3 Chaotic maps

Chaotic maps are widely used in the field of image encryption. It enhances randomness in plain image. They have various characteristics that are utilized for image encryption:

1. Sensitivity to initial conditions

Initial conditions of the system change the chaotic sequences completely. Thus, the initial conditions can be modified to generate a completely different chaotic sequence. Hence, chaotic systems can be used to generate different mask images for different plain images.

2. Pseudo-random behaviour

Chaotic systems show pseudo- random behaviour. This implies that they have random properties but they have extremely large time periods. Therefore, they are seemingly random in behaviour. So, they can be used to increase the randomness in plain image.

3. Mixing

Chaotic systems exhibit this characteristic in such a way that the system spreads itself over the full phase space. This is an intrinsic feature of chaotic system which can be utilized for spreading the pixel values of plain image in a full phase space.

4. Ergodicity

Ergodicity of a chaotic system ensures trajectory of the chaotic system essentially restricts itself to a spatial object called attractor. Density of such points is time invariant which is a desirable property of the chaotic systems in image encryption.

Based on the above-mentioned properties, chaotic systems are a favourable choice for the image encryption algorithm designers. Logistic map has been used for enhancing the randomness in image encryption techniques [64, 69, 83, 91-93], it is used alone or along with other chaotic system in confusion or diffusion phase of image encryption. Zhang et al. [64] used logistic map on DNA sequences of Red, Green and Blue channels of color image. Pareek et al. [91] used two logistic chaotic maps to increase randomness in grayscale image. Based on the outcome of logistic map, encryption rule to be used for encrypting a particular pixel was determined. Li et al. [93] proposed an image encryption algorithm that uses three tables. The two tables are used to store the coordinate of X and Y direction and third table is used for swapping. These tables are constructed through logistic map.

Some image encryption algorithm use Chen's hyper chaotic system for diffusion process of image encryption [66, 86, 87, 94]. Chen's hyper-chaotic system gives four chaotic sequences which can be used to enhance randomness. In [61], Henon map has been used as the chaotic system for scrambling of pixels in a grayscale image. Arnold's cat map [56] is also used for scrambling of pixel values in gray images. In [95], Rossler system is used for introducing randomness in the plain image. Spatiotemporal chaotic systems have been used for image encryption in [67, 96, 97]. DNA cryptosystems for encrypting a plain image using DNA computing rules have emerged [64-73]. Image encryption

problem has been formulated as optimization problem. Genetic Algorithms can be applied for solving the image encryption problem using entropy as an objective function [65, 68, 70]. Optimization algorithms can also be used to generate encrypted images with high entropy values [103, 114]. DNA cryptosystems can be combined with other systems for encryption of plain images such as S-box [71], Trellis Algorithm [72], and CBC encryption mode [73].

The literature survey entails that the use of chaotic systems is a good choice to encrypt the image. Chaotic systems enhance the randomness in image when used in permutation and/or substitution processes of image encryption. There are several benefits of using a hyper-chaotic system over a chaotic system. A hyper-chaotic system is the chaotic system which has two or more than two positive Lyapunov exponents. Thus, a hyper-chaotic system gives more randomness in the system. The proposed image encryption uses the combination of two hyper-chaotic systems namely Lorenz chaotic system and Rossler chaotic system. For a particular set of values of the control parameters, the system goes into the state of hyper-chaos [88]. The combined chaotic system has six control parameters formed by combining three control parameters each from Lorenz chaotic system and Rossler chaotic system. When two chaotic systems are combined, the key space increases. Thus, resistance of the chaotic system towards brute force attacks increases.

3.3.1 Lorenz-Rossler hyper-chaotic system

Lorenz-Rossler chaotic system is obtained from combining both Lorenz chaotic system [99] and Rossler chaotic system [100]. The combined system is in a hyper-chaotic state when a particular set of values is assigned to control parameters [88]. Lorenz-Rossler hyper chaotic system is obtained by simply adding equations from Lorenz chaotic system and Rossler chaotic system. Resulting hyper chaotic system has six control parameters (three from each) and integration step is taken a small value. The mathematical representation of Lorenz-Rossler chaotic system is given below.

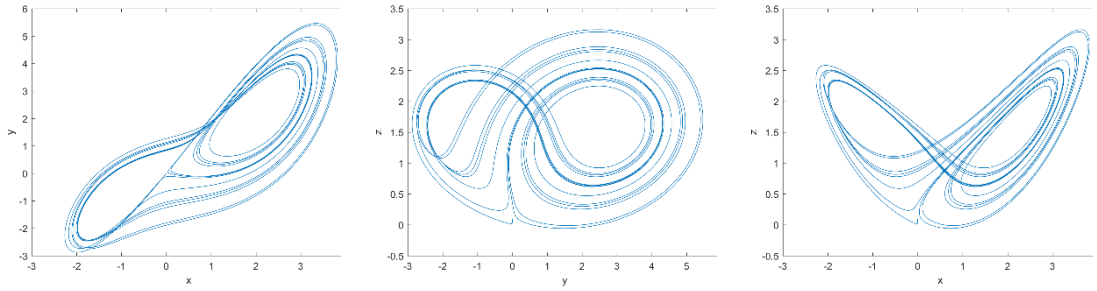
$$\hat{x} = (\delta - 1)y - \delta x - z;$$

$$\hat{y} = (r + 1)x - (1 - a)y - 20xz;$$

$$\hat{z} = 5xy - \beta z + b + xz - cx \tag{3.3}$$

where $\hat{x}$, $\hat{y}$, and $\hat{z}$ are integrals of x, y, and z. $\delta, r, \beta,\ a, b$, and c are control parameters.

Investigations done in [88] reveal that the system exhibits chaotic behaviour when the control parameters are set as $a = 9,\ b = 0,\ c = 8,\ \delta = 20,\ \ \beta = 8.5$, and $r = 20$. Chaotic attractors of the system are shown in Fig. 3.2.



(a) (b) (c)

Fig. 3.2. Chaotic attractors of Lorenz-Rossler chaotic system in (a) xy-plane, (b)yz-plane and (c) xz-plane

Lorenz-Rossler chaotic system gives three chaotic sequences- x,y,and z. Initial values of chaotic sequences are modified according to the hamming distance values so that the chaotic sequences generated are unique for a particular plain image.

3.4 Encryption algorithm

Image encryption algorithm is based on Lorenz-Rossler chaotic system and DNA cryptography rules. For same pixel values in image, different DNA sequences can be generated depending on the location of pixel. It enhances randomness in image and destroys the correlation properties of the adjoining pixels.

The steps of encryption algorithm are described below.

Step 1. *Separation of color components*

Read plain image (*P*) and decompose the color image into its three different color channels such as *R*, *G*, and *B*.

$$P \rightarrow \{R, G, B\}$$

Step 2. *Conversion to DNA strands representation*

The pixel values (in decimal) of *R*, *G*, and *B* are translated into their corresponding binary representation. According to the location of pixel, the

binary values are converted into DNA strands $\left(R_{DS},\ G_{DS},\ B_{DS}\right)$ using encoding rule mentioned in Table 3.1.

$$\begin{aligned} &DNA\_Encode\left(Binary\left(R\right)\right)\rightarrow R_{DS}\\ &DNA\_Encode\left(Binary\left(G\right)\right)\rightarrow G_{DS}\\ &DNA\_Encode\left(Binary\left(B\right)\right)\rightarrow B_{DS} \end{aligned} \tag{3.4}$$

Step 3. *Generation of Chaotic Sequences*

Generate chaotic sequences x, y, and z using procedure as mentioned in Section 3.3.1. The mask images such as DX, DY, and DZ are generated from chaotic sequences.

$$DX(i,j)=X_k,\quad DY(i,j)=Y_k,\quad DZ(i,j)=Z_k \tag{3.5}$$

where i and j are computed as follows:

$$\begin{aligned} &i=\left\lfloor \left(k-1\right)/row\_size \right\rfloor+1\\ &j=k-\left(\left(i-1\right)\times col\_size\right) \end{aligned} \tag{3.6}$$

Here,

$$\begin{aligned} &X_k=\left(\left(\left\lfloor x_k\times 10^{14}\right\rfloor \%n\right)+1\right)\%256\\ &Y_k=\left(\left(\left\lfloor y_k\times 10^{14}\right\rfloor \%n\right)+1\right)\%256 \qquad \forall x_k\in x,\ y_k\in y\ \ and\ \ z_k\in z\\ &Z_k=\left(\left(\left\lfloor z_k\times 10^{14}\right\rfloor \%n\right)+1\right)\%256 \end{aligned} \tag{3.7}$$

Step 4. *Conversion of Chaotic Sequences into DNA Strands*

Convert the decimal values of DX, DY and DZ into DNA strands using encoding rule mentioned in Table 3.1.

$$\begin{aligned} &DNA\_Encode\left(Binary\left(DX\right)\right)\rightarrow DX_{DS}\\ &DNA\_Encode\left(Binary\left(DY\right)\right)\rightarrow DY_{DS}\\ &DNA\_Encode\left(Binary\left(DZ\right)\right)\rightarrow DZ_{DS} \end{aligned} \tag{3.8}$$

This step produces arrays $\left(DX_{DS},\ DY_{DS},\ DZ_{DS}\right)$. Size of each array is 256×1024.

Step 5. *Apply XOR and addition operation on DNA sequences*

First, XOR operation is applied on DNA strands obtained from Steps2 and 4.

$$R_{DS} \oplus DX_{DS} \rightarrow R'_{DS}, G_{DS} \oplus DY_{DS} \rightarrow G'_{DS}, \quad B_{DS} \oplus DZ_{DS} \rightarrow B'_{DS} \tag{3.9}$$

where $\oplus$ represents XOR operation. This step is explained in Fig. 3.3

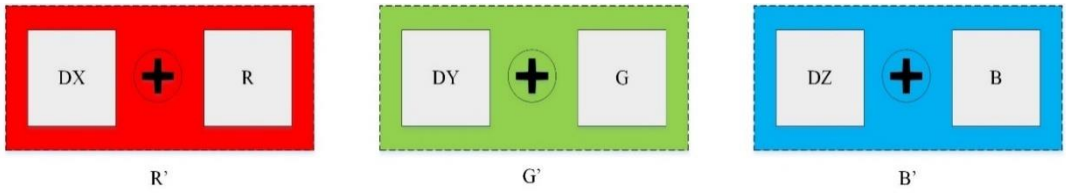


Fig. 3.3. XOR operation

Thereafter, the addition operation is applied on R'_{DS}, G'_{DS}, and B'_{DS} to produce new DNA strands (R''_{DS} G''_{DS} and B''_{DS}) as explained in Fig. 3.4.

$$G'_{DS} + R'_{DS} \rightarrow G''_{DS}, \quad B'_{DS} + G''_{DS} \rightarrow B''_{DS}, \quad R'_{DS} + B''_{DS} \rightarrow R''_{DS} \tag{3.10}$$

where '+' represents addition operation between DNA strands. The XOR and addition operation are mentioned in Tables 3.2 and 3.3 respectively.

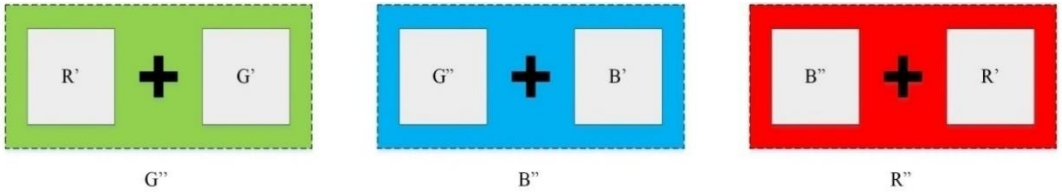


Fig. 3.4. Addition operation

Step 6. *Generation of Encrypted Image*

Convert R''_{DS}, G''_{DS}, and B''_{DS} into its decimal representations and combine them to produce the encrypted image (*E*).

$$Combine\left(Decimal\left(R''_{DS}, G''_{DS}, B''_{DS}\right)\right) \rightarrow E \tag{3.11}$$

Encrypted image is received in the last step. This encryption process is and illustrated in Fig. 3.5 below.

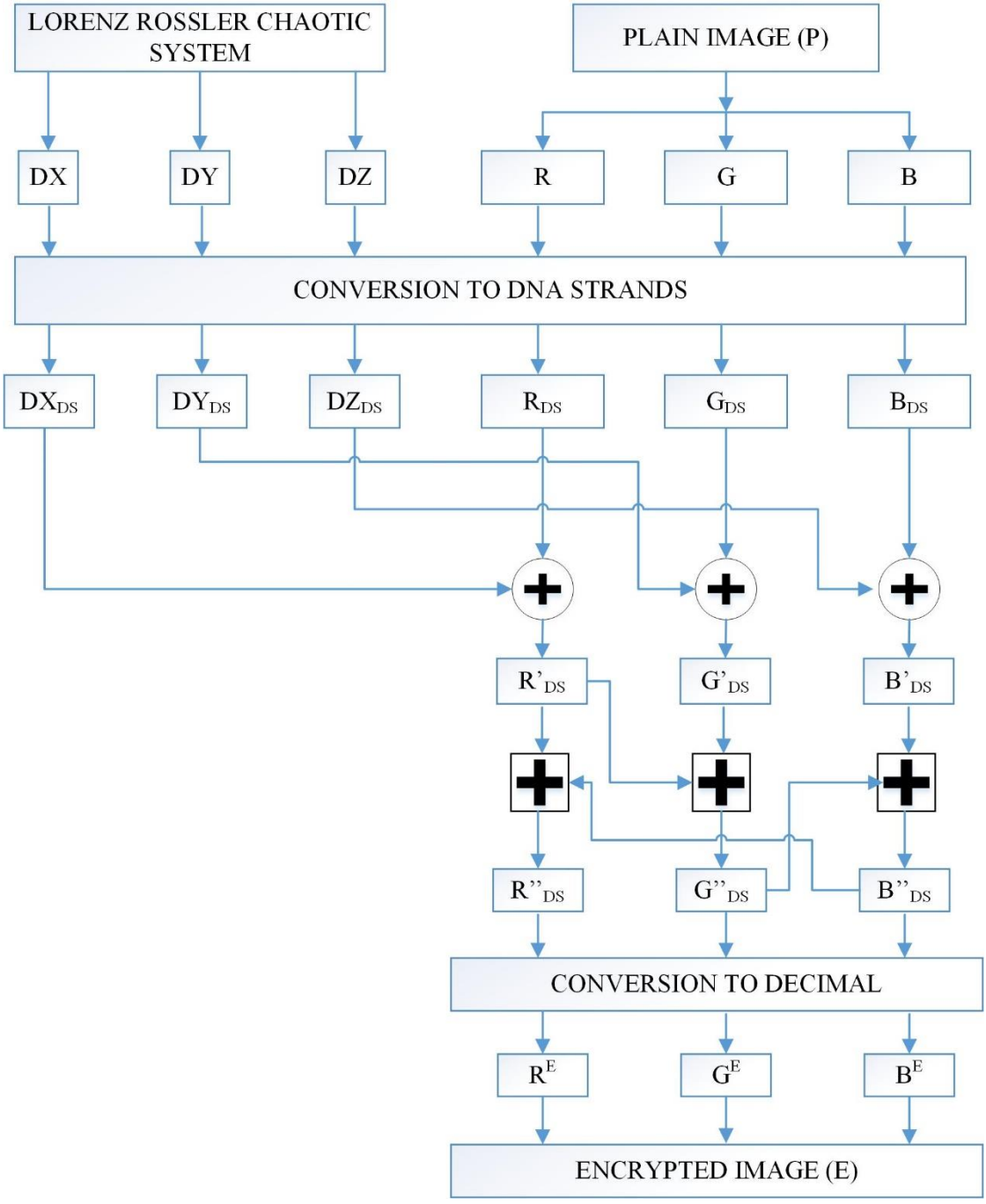


Fig. 3.5. Proposed image encryption procedure

3.5 Decryption algorithm

The decryption of an image is the inverse process of encryption. The addition operation is replaced with subtraction. Using encryption keys; $\widehat{x_1}$, $\widehat{y_1}$, $\widehat{z_1}$, δ, β, r, a, b and c; chaotic sequences are generated. With same initial conditions, chaotic sequences can be regenerated as they are pseudorandom in nature. Thus, chaotic sequences generated

in decryption phase are exactly similar to the chaotic sequences generated in encryption phase.

The step by step decryption procedure of the proposed approach is given below:

Step 1. *Separation of color components from encrypted image*

Read encrypted image (*E*) and decompose it into three planes *R, G,* and *B*.

$$E \rightarrow \left\{R^E, G^E, B^E\right\}$$

Step 2. *Conversion of encrypted image into DNA strands representation*

The pixel values of R^E, G^E, and B^E are converted into their corresponding binary representation. Thereafter, the binary values are converted into DNA strands $\left(R_{DS}^E,\ G_{DS}^E,\ B_{DS}^E\right)$ according to the location of pixels.

$$\begin{aligned} &DNA\_Encode\left(Binary\left(R^E\right)\right) \rightarrow R_{DS}^E \\ &DNA\_Encode\left(Binary\left(G^E\right)\right) \rightarrow G_{DS}^E \\ &DNA\_Encode\left(Binary\left(B^E\right)\right) \rightarrow B_{DS}^E \end{aligned} \tag{3.12}$$

Step 3. *Generation of Chaotic Sequences*

Generate chaotic sequences through same parameter as used in encryption process and produce mask images DX, DY, and DZ from these chaotic sequences. The chaotic sequence generation procedure is same as mentioned in Step 3 of encryption procedure.

Step 4. *Conversion of Chaotic Sequences into DNA Strands*

Convert the decimal values of DX, DY, and DZ into DNA strands using encoding rule mentioned in Table 3.1.

$$\begin{aligned} &DNA\_Encode\left(Binary\left(DX\right)\right) \rightarrow DX_{DS} \\ &DNA\_Encode\left(Binary\left(DY\right)\right) \rightarrow DY_{DS} \\ &DNA\_Encode\left(Binary\left(DZ\right)\right) \rightarrow DZ_{DS} \end{aligned} \tag{3.13}$$

This step produces DNA strands $\left(DX_{DS},\ DY_{DS},\ DZ_{DS}\right)$ of chaotic sequence generated from decryption process.

Step 5. *Apply Subtraction and XOR operations on DNA sequences*

Subtraction operation is applied on DNA strands obtained from Steps 2 and 4.

$$B_{DS}^{E} - G_{DS}^{E} \to B_{DS}^{E'}, \quad R_{DS}^{E} - B_{DS}^{E'} \to R_{DS}^{E'}, \quad G_{DS}^{E} - R_{DS}^{E'} \to G_{DS}^{E'} \tag{3.14}$$

where '-' represents subtraction operation. This step is shown in Fig. 3.6.

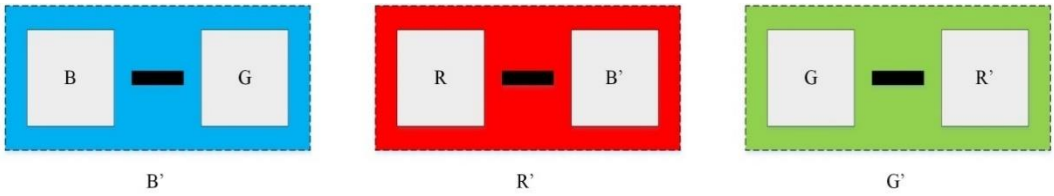


Fig. 3.6. Subtraction operation

Thereafter, XOR operation is applied on $R_{DS}^{E'}$, $G_{DS}^{E'}$, and $\boldsymbol{B}_{DS}^{E'}$ to produce new DNA strands ($\boldsymbol{R}_{DS}^{E''}$, $G_{DS}^{E''}$, and $\boldsymbol{B}_{DS}^{E''}$).

$$R_{DS}^{E'} \oplus DX_{DS}^{E} \to R_{DS}^{E''}, \quad G_{DS}^{E'} \oplus DY_{DS}^{E} \to G_{DS}^{E''}, \quad B_{DS}^{E'} \oplus DZ_{DS}^{E} \to B_{DS}^{E''} \tag{3.15}$$

The XOR and subtraction operation are mentioned in Tables 3.2 and 3.4 respectively.

Step 6. *Generation of Decrypted Image*

Convert the DNA strands obtained from Step 5 into their binary representation. Thereafter, the binary values are transformed into decimal notation (R', G', and B').

$$\begin{aligned} &Decimal\left(Binary\left(R_{DS}^{E''}\right)\right) \to R' \\ &Decimal\left(Binary\left(G_{DS}^{E''}\right)\right) \to G' \\ &Decimal\left(Binary\left(B_{DS}^{E''}\right)\right) \to B' \end{aligned} \tag{3.16}$$

These components are combined back to produce color image (P').

$$Combine\left(R', G', B'\right) \to P'$$

Fig. 3.7 shows the proposed image decryption procedure in brief detail.

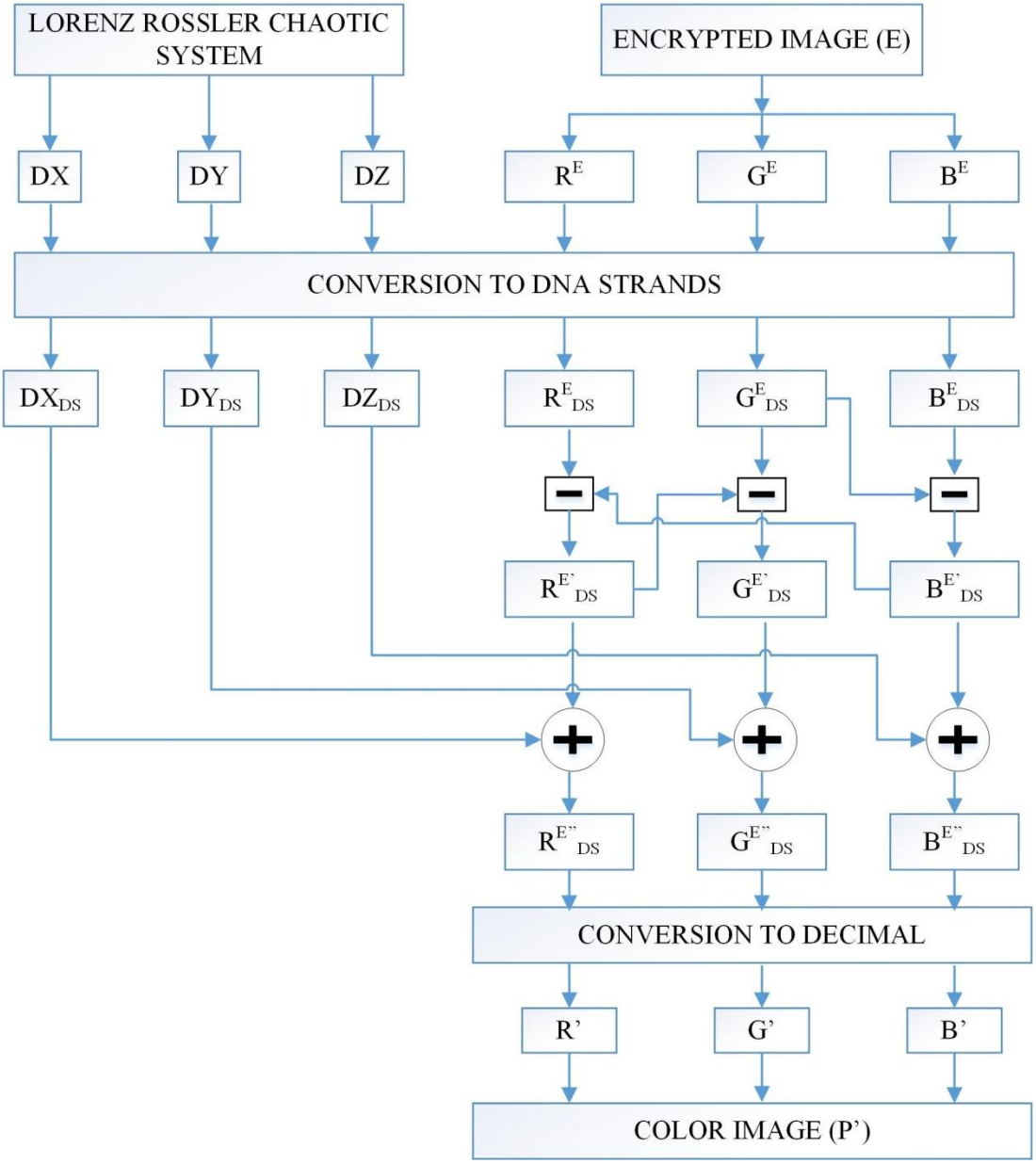


Fig. 3.7. Generation of decrypted image

3.6 Simulation results and discussion

The proposed image encryption algorithm was tested on i7-7500U processor with 8GB RAM. Test images from SIPI database [98] were taken and were resized to 256×256. Encrypted images from the proposed image encryption are shown in Fig. 3.8 below. It can be observed from the figure that the encrypted images reveal no information regarding their respective plain images. Encrypted images are decrypted and then compared with the plain images. It can be observed that the plain and decrypted images are exactly similar to each other.

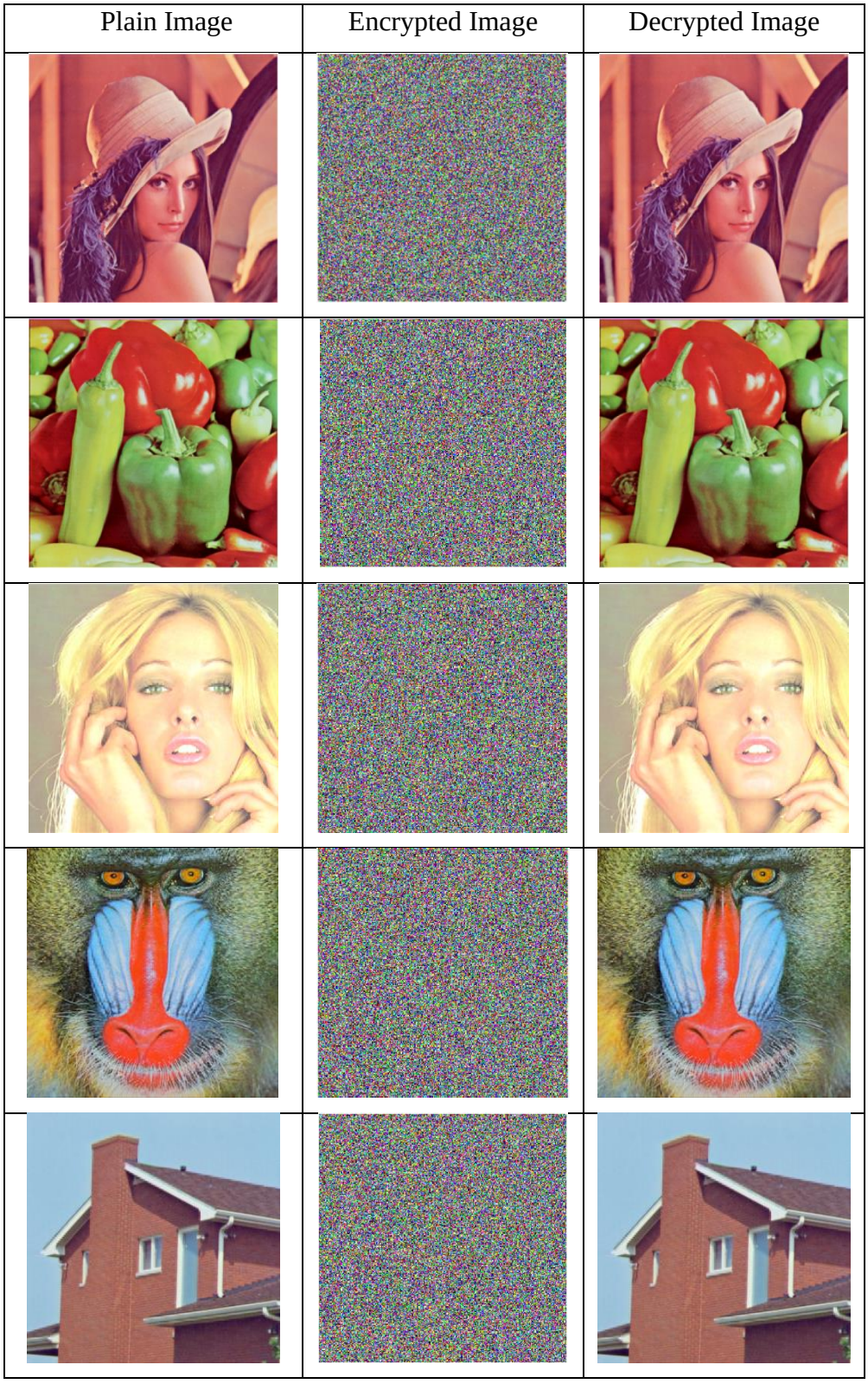


Fig. 3.8. Encrypted and decrypted images of the respective plain images

A good image encryption algorithm should be able to resist key exhaustive attacks, statistical attacks and differential attacks. Performance of the image encryption is

measured from its ability to resist these attacks. Different parameters indicate strength of the algorithm to withstand these attacks as mentioned below.

Information entropy of image indicates the degree of randomness in image. Higher the value of entropy of image, more is the disturbance in the image. Correlation coefficient indicates whether adjacent pixels in an image carry related information. High value of correlation coefficient implies high dependence of the adjacent pixels on each other. This indicates that the image is carrying some meaningful information and is not random. Thus, high entropy and low correlation coefficient value of encrypted image is desired.

Also, there are some attacks which can be carried out by the attackers. Encrypted image should be able to resist these attacks. Performance of the image encryption algorithm is measured using the performance metrics mentioned below.

3.6.1 Resistance towards attacks

An encrypted image should be able to resist various attacks carried out by the attackers on it. Response of encrypted image is analysed in the preceding subsections when differential attacks, statistical attack and exhaustive attacks are carried out.

3.6.1.1 Differential attack analysis

Differential attacks reveal the relationship between pixels of plain image and encrypted image. This is done by changing a few pixels in the plain image and encrypting the changed image. Difference between the two encrypted images is observed in order to obtain relationship between plain image and encrypted image [65]. In other words, the image encryption algorithm should be able to give a unique encrypted image for each plain image. Even a slightest change in image should result in a completely different encrypted image. Ability to resist differential attack analysis is measured using NPCR (Number of Pixel Change Rate) and UACI (Unified Average Changing Intensity) values [65]. These two parameters are calculated from two encrypted images. These two parameters are calculated by eq. 3.17-3.19 given below. Plain image of Lena is changed in one pixel and this changed image is used for calculating NPCR and UACI. The two encrypted images T1 and T2 are obtained by encrypting Lena image and changed Lena image respectively. Since the initial parameters x1, y1 and z1 are obtained by adding the hamming distances of red, green and blue channels of plain image and chaotic systems

are sensitive to initial parameters; every plain image has a unique encrypted image. This can be observed by high NPCR and low UACI values in Table 3.5.

$$C(i,j) = \begin{cases} 0\ , if\ T1(i,j) == T2(i,j); \\ 1\ , if\ T1(i,j)\ != T2(i,j); \end{cases} \tag{3.17}$$

$$NPCR = \frac{\sum_{i=1}^{M}\sum_{j=1}^{N} C(i,j)}{M \times N} \times 100 \tag{3.18}$$

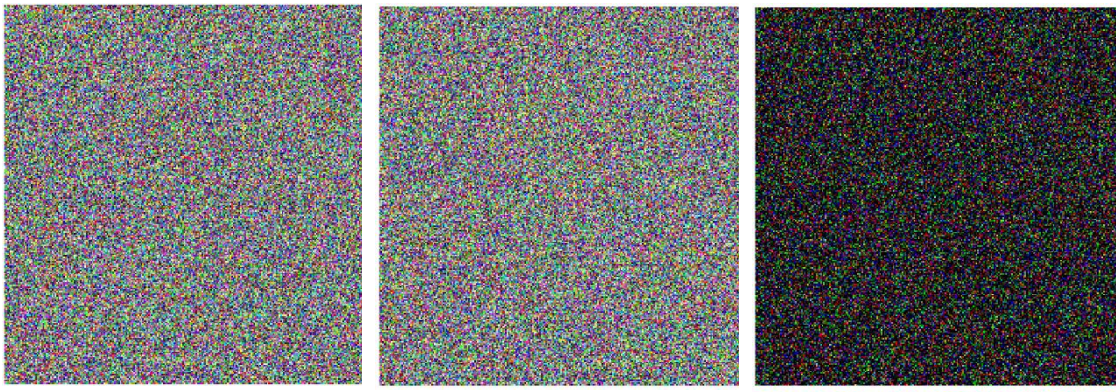
$$UACI = \frac{\sum_{i=1}^{M}\sum_{j=1}^{N} |T1(i,j) - T2(i,j)|}{255 \times M \times N} \times 100 \tag{3.19}$$

where M and N are height and width of the image respectively.

Fig. 3.9 shows the encrypted images T1 and T2 used for calculation of NPCR and UACI. In order to get a better understanding of difference between T1 and T2, difference between corresponding pixels of T1 and T2 has been obtained and shown in the extreme right.

Table 3.5. NPCR and UACI values

	Red	**Green**	**Blue**
NPCR	99.623%	99.606%	99.652%
UACI	33.245%	33.362%	33.521%



(a) (b) (c)

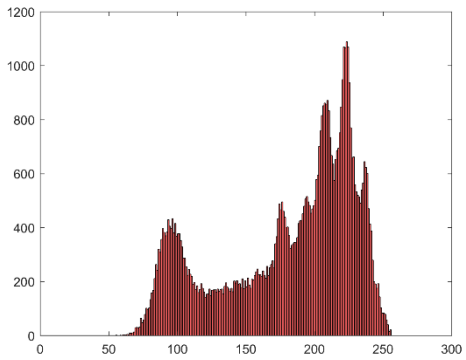
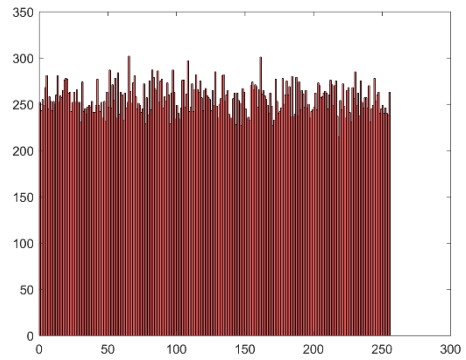
Fig. 3.9. (a) Encrypted image of Lena, (b) encrypted image of Lena with one-pixel change and (c) difference between (a) and (b)

3.6.1.2 Statistical attack analysis

Statistical properties of a plain image should be completely destroyed during the encryption process. This means that from the statistical properties of the encrypted image, any meaningful information is not revealed to the attacker. In statistical analysis of the encrypted image, there are two criteria of evaluating the image encryption algorithm as written below.

3.6.1.2.1 Histogram analysis

Histogram of an image is plotted against the frequency of pixels having a particular intensity value. Ideally, a true random image should have flat histogram bars having same height. In a true random image for any pixel intensity value, number of pixels is constant. Thus, same heights of histogram bars are obtained for it. A plain meaningful image is having unequal height of histogram bars which implies some information is there in the image. The image encryption algorithm should be able to convert the plain image into an encrypted image which gives a histogram that has uniform histogram bars. When histogram analysis of the encrypted image is carried out by an attacker of the encrypted image, it does not cause suspicion in the minds of the attacker that it is not a true random image. Histogram analysis was carried out on the three channels of both plain and encrypted images as shown below in Fig. 3.10 and Fig. 3.11. This is shown for Lena and Peppers images and their respective encrypted images.

Plain *Lena* Image	**Encrypted *Lena* Image**
	

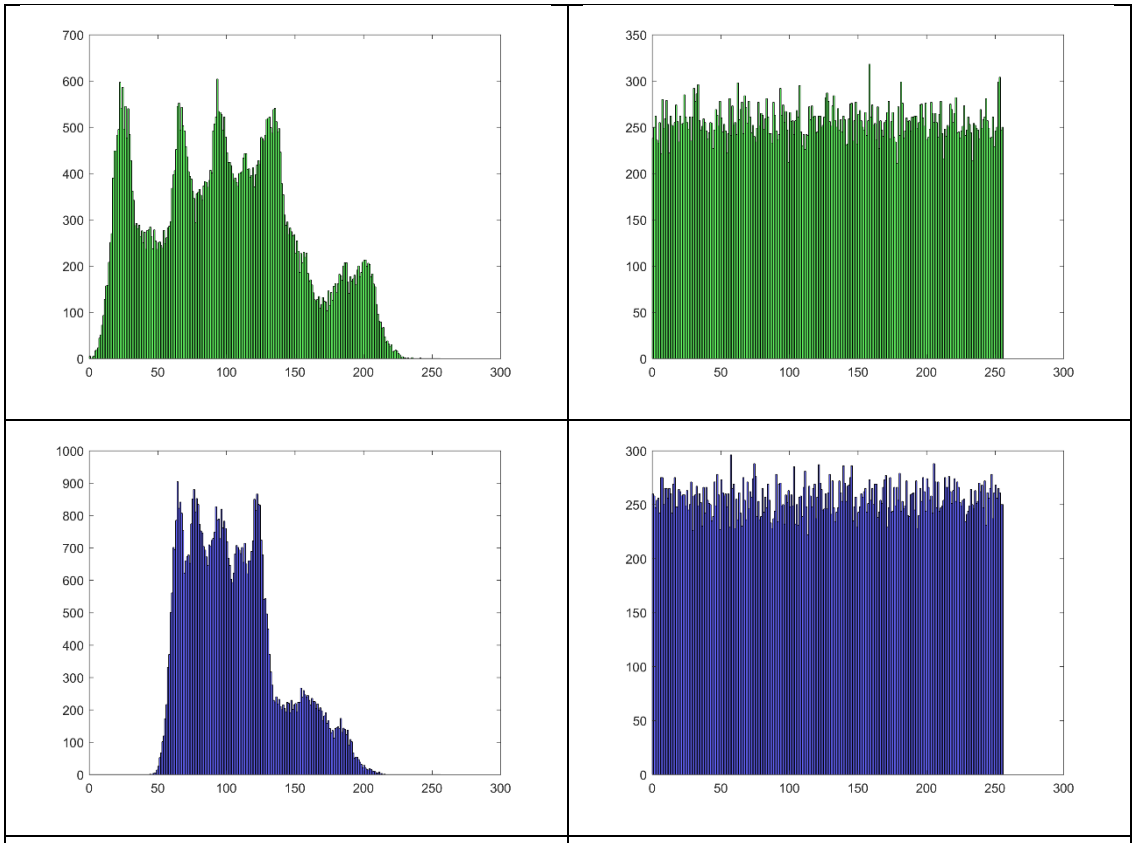
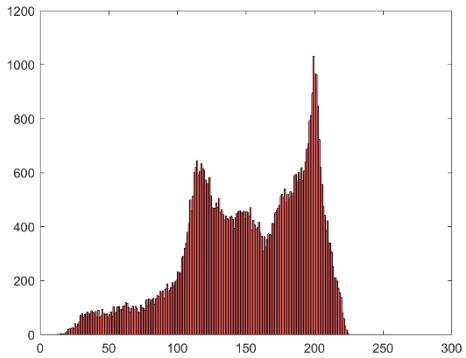
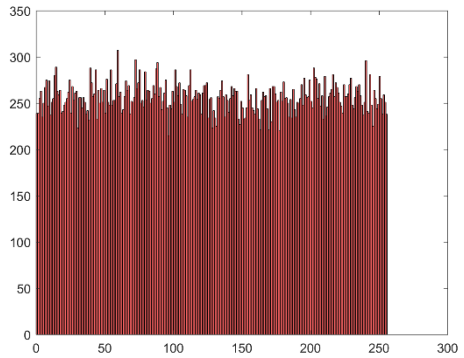


Fig. 3.10. Histogram of RGB planes of (a) plain *Lena* image and (b) encrypted *Lena* image.

Plain *Peppers* Image	**Encrypted *Peppers* Image**
	

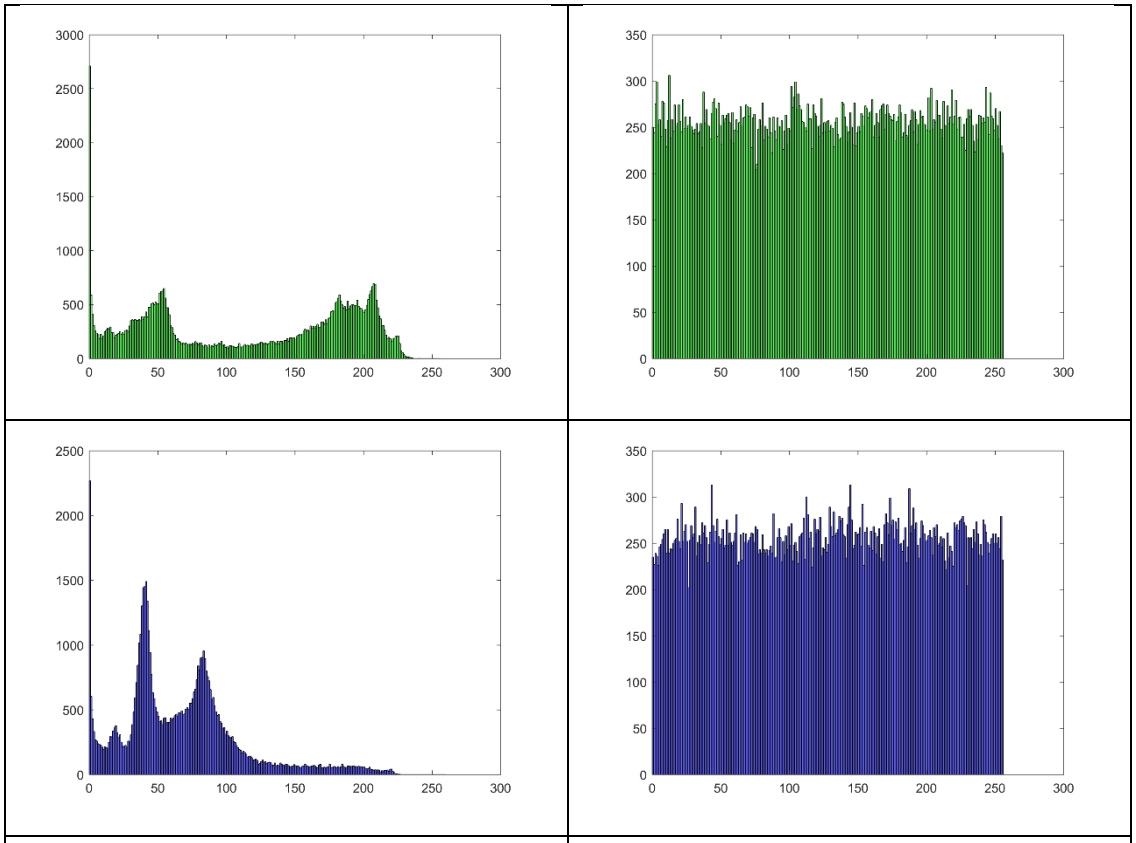


Fig. 3.11. Histogram of RGB planes of (a) plain *Peppers* image and (b) encrypted *Peppers* image

As can be observed, histogram plots obtained for the RGB planes of encrypted images are flat and thus seem to be histogram plots of a noisy image. Image encryption algorithm is thus able to withstand histogram analysis by statistical attackers.

3.6.1.2.2 Correlation analysis

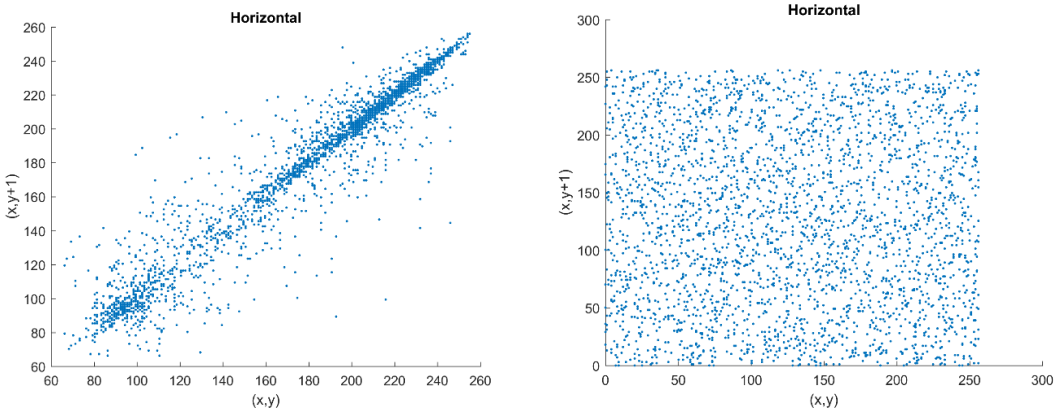
Another statistical attack is correlation analysis. Correlation between two pixels of a plain image is very high, leading to a high value of correlation coefficient of plain image. The correlation coefficient (CC) is calculated for two adjacent pixels in horizontal, vertical and diagonal directions. It can be defined as: [101]

$$CC = \frac{N\sum_{j=1}^{N}\left(x_j \times y_j\right) - \sum_{j=1}^{N} x_j \times \sum_{j=1}^{N} y_j}{\sqrt{\left(N\sum_{j=1}^{N} x_j^2 - \left(\sum_{j=1}^{N} x_j\right)^2\right)\left(N\sum_{j=1}^{N} y_j^2 - \left(\sum_{j=1}^{N} y_j\right)^2\right)}} \tag{3.20}$$

where x and y are pixel values of two adjacent pixels. N represents the total number of pixels.

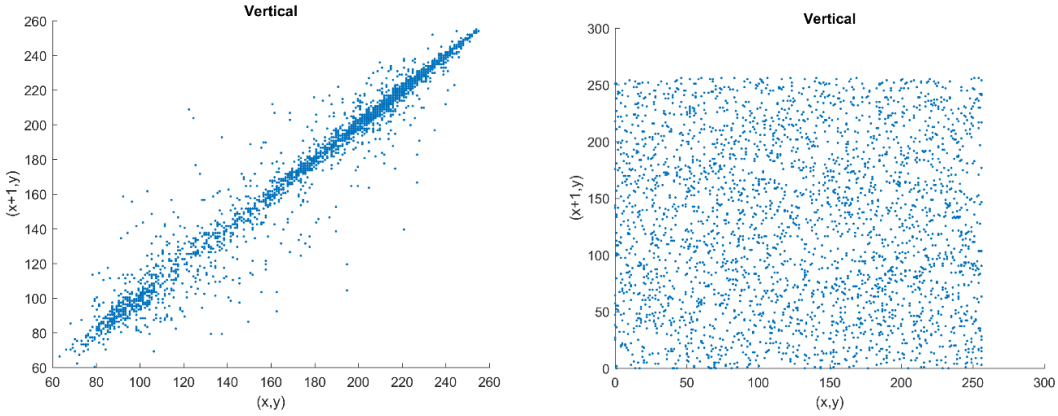
The correlation between two adjacent pixels of plain images and encrypted images has been computed. The value of correlation between two adjacent pixels in plain image is high as each pixel is highly correlated with its adjacent pixels. However, the value of correlation for an encrypted image should be as small as possible. The low value of correlation coefficient for adjacent pixels in horizontal, vertical and diagonal directions indicate that the adjacent pixels are uncorrelated.

The correlation values are computed for 3000 pairs of two adjacent pixels in all three directions-horizontal, vertical and diagonal from plain image and encrypted image. Figures 3.12, 3.13, and 3.14 show the correlation distribution of two horizontal adjacent pixels, two vertical adjacent pixels, and two diagonal adjacent pixels of red plane of plain image with its encrypted image, respectively. The correlation coefficient between two adjacent pixels for original and encrypted image are depicted in Table 3.6. This table reveals that the proposed image encryption system guarantees decreased correlation coefficient values amongst two neighbouring pixels. Therefore, these results prove that the proposed encryption system is able to increase randomness among pixels.



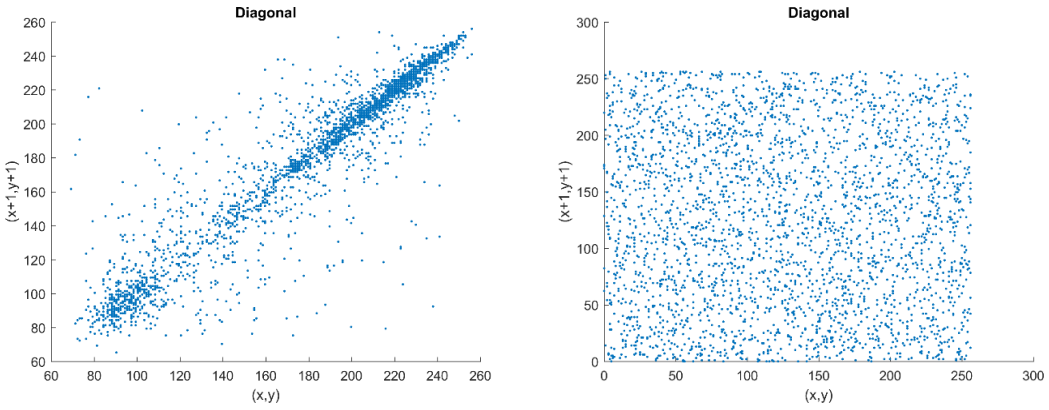
(a) (b)

Fig. 3.12. Correlation analysis of two horizontally adjacent pixels of red plane (a) plain *Lena* image and (b) encrypted *Lena* image.



(a) (b)

Fig. 3.13. Correlation analysis of two vertically adjacent pixels of red plane (a) plain *Lena* image and (b) encrypted *Lena* image.



(a) (b)

Fig. 3.14. Correlation analysis of two diagonally adjacent pixels of red plane (a) Plain *Lena* Image and (b) Encrypted *Lena* Image.

Table 3.6. Correlation coefficient of two adjacent pixels of original image and encrypted image

Images	**Plane**	**Correlation Coefficient**	**Direction of adjacent pixels**		
			Horizontal	**Vertical**	**Diagonal**
Lena	Red	Original	0.9572	0.9789	0.9339
		Encrypted	-0.0001	0. 0026	-0.0053
	Green	Original	0.9432	0.9714	0.9193
		Encrypted	-0.0011	0.0009	0.0026

	Blue	Original	0.9284	0.9559	0.9007
		Encrypted	-0.0010	-0.0030	-0.0051
Peppers	Red	Original	0.9646	0.9680	0.9369
		Encrypted	-0.0016	0.0023	0.0004
	Green	Original	0.9698	0.9750	0.9466
		Encrypted	-0.0043	0.0053	-0.0008
	Blue	Original	0.9570	0.9636	0.9263
		Encrypted	0.0013	0.0005	0.0008
Girl	Red	Original	0.9779	0.9294	0.9129
		Encrypted	0.0026	-0.0019	-0.0051
	Green	Original	0.9748	0.9106	0.8941
		Encrypted	-0.0054	-0.0078	-0.0123
	Blue	Original	0.9726	0.9130	0.8958
		Encrypted	-0.0046	0.0033	-0.0042
Baboon	Red	Original	0.9474	0.9277	0.9034
		Encrypted	-0.0017	-0.0007	0.0015
	Green	Original	0.8728	0.8380	0.7925
		Encrypted	0.0028	0.0039	0.0015
	Blue	Original	0.9216	0.9139	0.8763
		Encrypted	0.0041	0.0061	0.0025
House	Red	Original	0.9671	0.9353	0.9126
		Encrypted	0.0024	0.0010	0.0047
	Green	Original	0.9805	0.9474	0.9320
		Encrypted	-0.0014	0.0035	-0.0071
	Blue	Original	0.9820	0.9749	0.9625
		Encrypted	0.0026	-0.0002	0.0055

The performance of proposed technique is compared with eight well-known image encryption techniques on *Lena* image. Table 3.7 shows the correlation coefficient between the plain image and encrypted image. The results reveal that the proposed approach provides small correlation coefficient as compared to other techniques.

Therefore, the proposed approach generates better encryption performance than the others.

Table 3.7. Performance comparison in terms of horizontal correlation coefficients for encrypted *Lena*

Component of image ⟶	**Red**	**Green**	**Blue**
Proposed algorithm	-0. 00009	-0.0011	-0.0010
Niyat et al. [86]	0.0757	0.0744	0.0687
Zhang et al. [79]	-0.0065	0.0009	-0.0008
Wang et al. [80]	-0.0108	-0.0181	-0.0061
Murillo et al. [81]	0.0135	-0.0835	-0.0170
Mishra et al. [82]	0.0219	-0.0046	-0.0211
Kumar et al. [83]	0.0181	-0.0067	0.0154
Kumar et al. [84]	0.0035	-0.0097	0.01857
Zhang et al. [66]	0.0017	0.0016	0.0013

3.6.1.3 Exhaustive attack analysis

In this attack analysis, key space and key sensitivity of the proposed image encryption algorithm is checked. Ability to resist brute-force attack is evaluated in this attack.

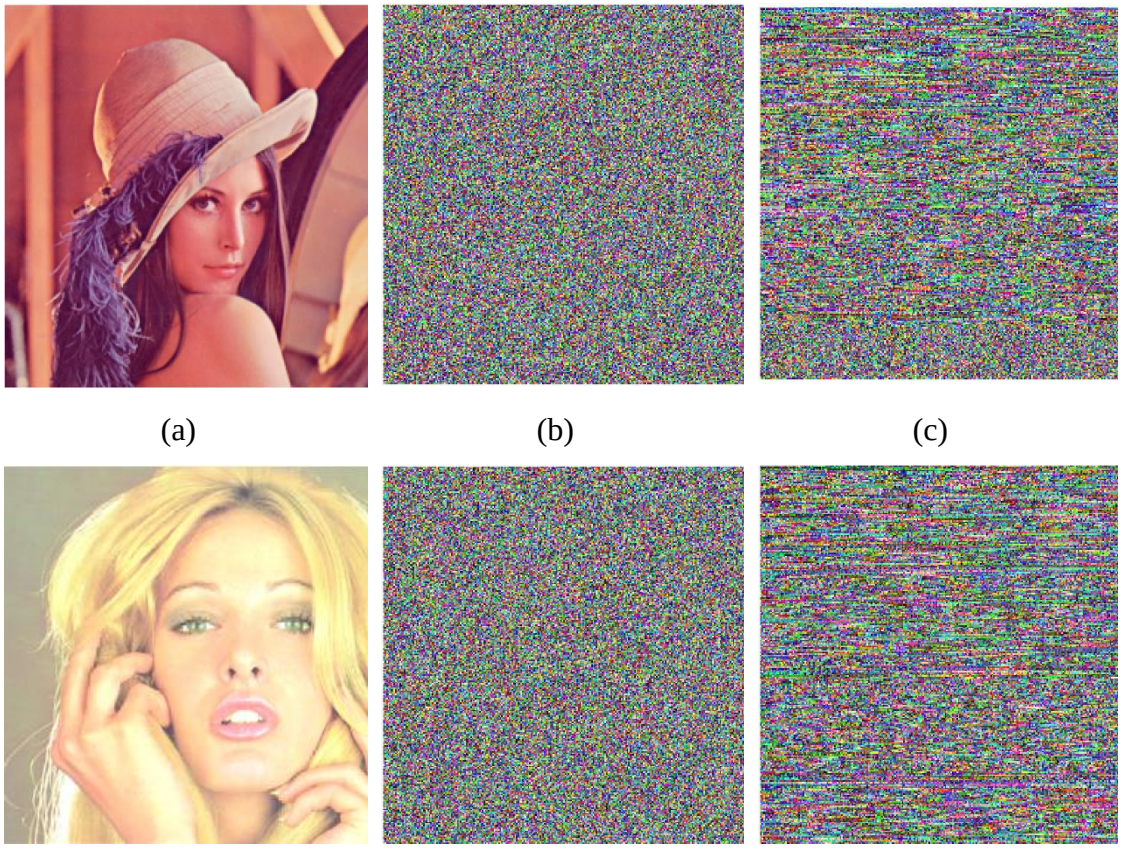
3.6.1.3.1 Key space analysis

Key space should be as large as possible, thus resisting the brute-force attack on encryption keys by attackers. The proposed approach uses three floating point values; x1, y1 and z1 as secret keys. As per IEEE 754-2008[90] floating point standards, double precision of 64-bit takes 15 decimal digits; hence the key space is 10^{45}. The proposed approach uses the set of (δ, β, r, a, b and c) as keys; hence leading to a large key space.

3.6.1.3.2 Key sensitivity analysis

Key sensitivity of encryption keys used in the proposed approach can be observed by using a slightly changed key in decryption than that used in encryption. If there are more than one key in the process, any one of them can be modified. A slight modification is done in the fractional part of key value. Then, modified key is used for decryption of

encrypted image. All the other parameters are kept same in encryption and decryption process. Fig. 3.15 shows that the decryption process fails to yield the original plain image when a slightly changed key. In case of Lena, if slightly modified key, i.e. $0.001 + 10^{-14}$ is used in place of 0.001 for decrypting encrypted Lena image, then the decryption algorithm does not give back the original image Same procedure was repeated for tiffany image and it was observed that the decryption image could not give back the original tiffany image when modified key $0.001 + 10^{-14}$ is used. Thus, even a slight change at the 14^{th} decimal place in encryption keys fails to give back the original plain image; indicating highly sensitive encryption keys.



(a) (b) (c)

(d) (e) (f)

Fig. 3.15. Key sensitivity analysis; (a), (d) plain images of Lena and tiffany; (b), (e) respective encrypted images; (c), (f) respective decrypted images using slightly changed key

3.6.2 Information entropy analysis

Information entropy is used to measure the degree of disturbance in the given image. Entropy (H) can be defined as follows [76]:

$$H(v)=\sum_i p(v_i)log_2\frac{1}{p(v_i)} \tag{3.21}$$

where $p(v_i)$ represents the probability of occurrence of pixel value v_i of given image.

A true random image system would produce 2^8 pixel values with an equal probability, i.e., $v=\{v_1,v_2,.....,v_{2^8}\}$. Thus, the entropy of a true random image will be eight $(i.e.,\ H(v)=8)$ [76]. The value of entropy should be as close to 8 as possible for an ideal encrypted image.

Table 3.8 depict the values of entropy for plain images and encrypted images. A look at the table reveals that the entropy of the encrypted image is almost equal to 8.

Table 3.8. Entropy analysis

Image	**Original**			**Encrypted**		
	Red	**Green**	**Blue**	**Red**	**Green**	**Blue**
Lena	6.6099	7.0245	6.9171	7.9974	7.9969	7.9979
Peppers	7.3009	7.5570	7.0929	7.9974	7.9973	7.9969
Girl	5.7150	5.3738	5.7117	7.9973	7.9976	7.9973
Baboon	7.6058	7.3581	7.6665	7.9970	7.9972	7.9973
House	6.4311	6.5389	6.2320	7.9974	7.9970	7.9967

Comparison of entropy calculated on encrypted image of Lena for the proposed work and previous works is done in Table 3.9. Entropy values of encrypted RGB image of Lena of previous works is compared with that of proposed work. As can be observed from the table that the proposed image encryption algorithm outperforms these recent image encryption approaches in terms of information entropy.

Table 3.9. Information entropy compared with other approaches

Entropy	**Red**	**Green**	**Blue**
Proposed work	7. 9974	7. 9969	7.9979
Zhang et al. [64]	7.9894	7.9884	7.9866
Zhang et al. [66]	7.9971	7.9969	7.9962
Niyat et al. [86]	7.9973	7.9968	7.9976

3.6.3 Avalanche Effect

A small change in the encryption key or plain image should entirely change in the encrypted image. This effect is known as an avalanche effect. The well-known performance measure namely Mean Square Error (MSE) is used to estimate the avalanche effect of cryptosystem [102]. It is computed between two cipher images generated from slightly different encryption keys. Assuming C_1 and C_2 be two cipher images whose encryption keys are slightly different. The mathematical formulation of MSE is given below [102]:

$$MSE = \frac{1}{M \times N} \sum_{i=1}^{M} \sum_{j=1}^{N} \left(C_1(i,j) - C_2(i,j) \right)^2 \tag{3.22}$$

Where M and N are the width and height of images, respectively.

For the proposed encryption system, MSE is computed between cipher image of *Lena* using original encryption key and cipher image obtained through modified encryption key as mentioned in Table 3.10. Cipher Image 1 was obtained by modifying only key x at twelfth decimal place, while key y and key z were intact.

Table 3.10. MSE between cipher images of *Lena* generated from slightly different encryption keys

Images	***X***	***Y***	***Z***	**MSE**
Cipher Image 1	$0.001 + (10^{-12})$	0.001	0.1	10953
Cipher Image 2	0.001	$0.001 + (10^{-12})$	0.1	10921
Cipher Image 3	0.001	0.001	$0.1 + (10^{-12})$	10932

Table 3.11. MSE between cipher images of *Peppers* generated from slightly different encryption keys

Images	***X***	***Y***	***Z***	**MSE**
Cipher Image 1	$0.001 + (10^{-12})$	0.001	0.1	10897
Cipher Image 2	0.001	$0.001 + (10^{-12})$	0.1	10954
Cipher Image 3	0.001	0.001	$0.1 + (10^{-12})$	10860

Similarly, Cipher Image 2 and Cipher Image 3 were generated by modifying only key y and key z respectively. Table 3.11 depicts MSE computed for *Peppers* image by slightly altering the encryption keys in a similar manner. It can be observed from large values of MSE in both tables that the proposed image encryption algorithm exhibits avalanche effect.

3.7 Summary

A novel image encryption algorithm is proposed in this chapter. The proposed algorithm is based on Lorenz-Rossler hyper chaotic system. It encrypts a plain color image. The three color channels are encoded into DNA strands using DNA cryptosystem. Chaotic sequences obtained from the hyper-chaotic system are also encoded to DNA strands. In the proposed approach, its novelty lies in usage of Lorenz-Rossler hyper-chaotic system on colour images for encryption. Other advantage of the proposed approach is application of inter-channel operations among red, green and blue channels. Three types of attacks are carried out on the encrypted image and analysis of the results obtained reveal that the proposed encryption algorithm fairs well when being attacked by attackers. Proposed image encryption algorithm is compared with some previous works in terms of information entropy and correlation coefficients to evaluate its performance. It is evident from comparisons done that it outperforms the previous works. Information entropy analysis and avalanche effect on the encrypted image validate randomness and uniqueness nature respectively of the encrypted image.

Chapter 4
3D Image Steganography

4.1 Introduction

A 3D Image Steganography approach takes a 3D image model as cover media. Mesh model representation of the 3D image model is used. Secret message is considered as a stream of binary bits. 3D image mesh model has large number of data points which can be used for embedding of secret bits. A novel reversible data-hiding algorithm is proposed in this chapter which uses 3D mesh model as cover file. It has blind extraction method. Embedding of secret bits is adaptive to surface of 3D mesh model. The proposed approach is the first 3D image steganography approach to use a chaotic system. It may inspire research-works of this area to use different chaotic maps in their approaches such as sine map, tent map, baker map, and/or their combinations.

4.2 Proposed 3D steganography system

The proposed image steganography system is reversible in nature. This implies that the effects of steganography on the 3D image cover model are not permanent. At the receiver side, the extraction of secret message is obtained. Along with it, the original 3D image model is also obtained in its exact form in case of reversible steganography. The extraction method is blind in nature, i.e., extraction method does not require the cover model. Fig. 4.1 illustrates general framework of the proposed approach.

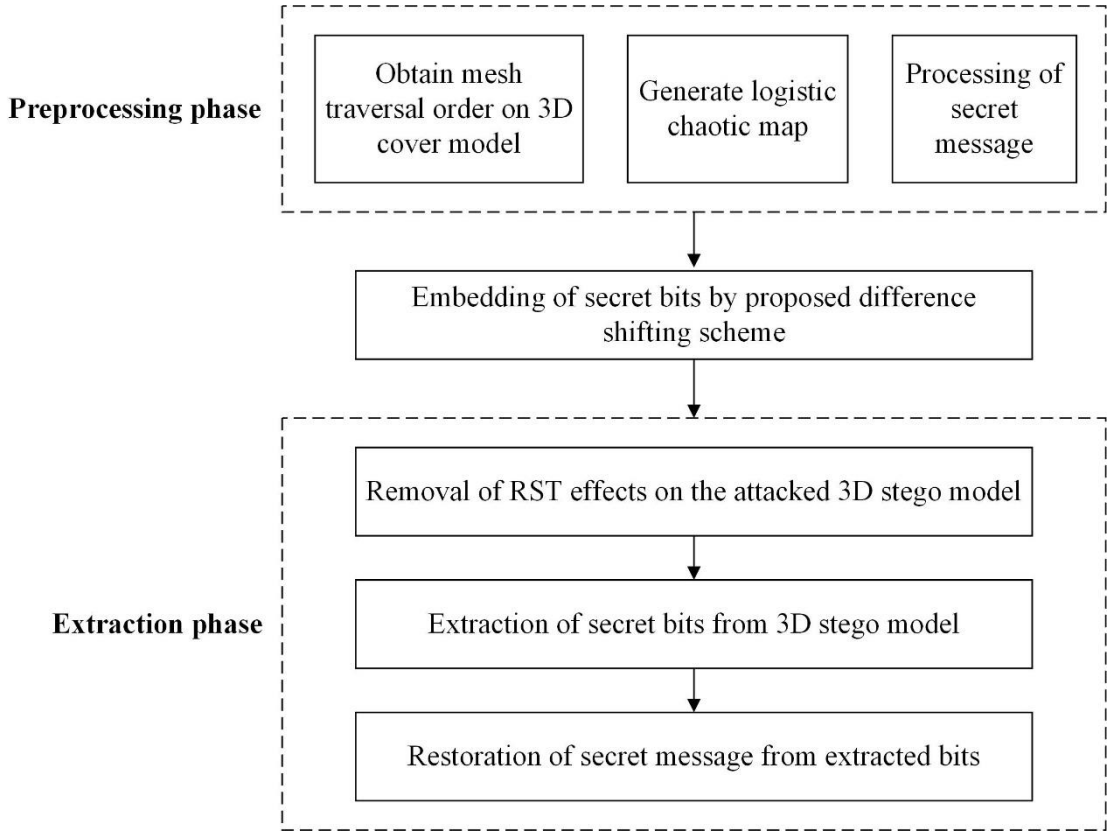


Fig. 4.1. Framework of the proposed steganography system

There are mainly three phases of the proposed steganography approach. These are preprocessing, embedding process, and extraction process. In the preprocessing phase, secret message is also processed and turned in an appropriate form so that it can be embedded inside 3D cover model. Mesh traversal order is obtained during the preprocessing stage. Vertices of the mesh are referred in order they appear in the mesh traversal order. After embedding of secret message inside 3D mesh, 3D stego model is sent from sender to receiver side. At the receiver side the stego model is checked for RST attacks and rectified for errors that have crept because of RST attacks. After extracting the secret bits from stego model, secret message is restored back from the bits. The detail description of the proposed 3D image steganography system is given in preceding sections.

4.2.1 Preprocessing phase

During both embedding and extraction process in 3D image steganography system, preprocessing phase is carried out. In this phase, 3D image model to be used as cover model is preprocessed. A 3D image model is composed of vertices having three

coordinates. A face is formed from a closed set of vertices. Thus, a typical 3D image model has two arrays such as Vertices and Faces [34].

In Vertices array, location of a vertex is specified in three coordinates in Cartesian coordinate system. In Faces array, indices of vertices are mentioned which form a particular face. For example, Vertices array shows the location of vertex with index 1031 as $(2.5, 4.1, 3.2)$ and Faces array represents that face with index 10 comprises of vertices indices as $(1031, 450, 601)$. Thus, utilizing information from these two arrays, a 3D image model is constructed by 3D rendering software, e.g. Meshlab [55].

Faces array lists the vertices used to form a particular face. From Faces array, connection of a vertex to other vertices in mesh can be found out. If an edge exists between two vertices, then those two vertices are neighbours to each other. Thus, neighbours of a vertex can be found out from Faces array. Identifying vertices is one of the first preprocessing task of the proposed algorithm. From a given starting vertex, *find_neighbors* method finds neighbours of the given vertex. Neighbour table is constructed for all the vertices of the mesh.

Algorithm 4.1: Construct Neighbour Table

Input: Faces and Vertices arrays

Result: Neighbour table

1. **for** face_index = 1 to all faces in Faces array
2. v = Faces(face_index,1)
3. neighbour_vertices = find_neighbours(v)
4. add vertices not present already in row v of neighbour table
5. repeat steps 2-4 for Faces(face_index,2) and Faces(face_index,3)
6. **end for**

Algorithm 4.2: Find Neighbours

Input: Vertex v of Vertices array for which neighbours are to be discovered

Result: neighbour_vertices of v

1. **for** i = 1 to all faces in Faces array
2. **if** v is present in any column of row i of Faces array

3. add vertices of the other two columns to neighbour_vertices
4. remove duplicate vertices if any
5. **end if**
6. **end for**

From faces array, neighbour table is constructed. In case of some 3D mesh models, some vertices may not have any connection with other vertices in the mesh. Thus, such unconnected vertices are also identified in the procedure.

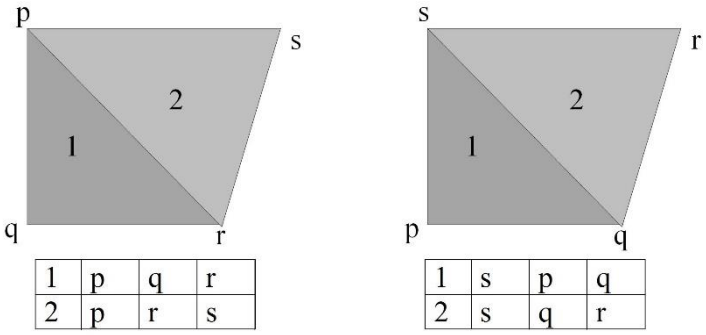
From the neighbour table, smooth and noisy surface over the mesh model can be identified. In the proposed steganography system, a surface is considered to be noisy if it has more vertices on it. It implies that if a vertex in mesh model has more neighbours than a particular threshold value then it is termed a noisy surface. On the other hand, if a vertex has less than the threshold neighbours, then it is called a smooth surface. Threshold value for 3D mesh models is different for small and big 3D mesh models. Threshold value for small 3D mesh models is smaller than that for large 3D mesh models.

After constructing the neighbour table, the next step is to obtain a mesh traversal order in 3D cover model. A unique mesh traversal order for each 3D mesh model has to be obtained. This is done in order to make the steganography robust against vertex reordering attack. In vertex reordering attack, the reference indices of vertices is changed, keeping the topology of the mesh intact. As topology of mesh is not changed and the location of vertices also remains unmodified, vertex reordering attack is a distortion less attack. But if the reference indices of the mesh are used for embedding of secret bits then the extraction process would fail to correct secret message from the attacked stego-model.

4.2.1.1 Mesh traversal algorithm

The mesh traversal algorithm is based on the shortest distance between neighbour vertices. It will generate the same referencing order every time over a particular mesh. It gives only one traversing sequence without any ambiguity. This is a necessary condition for mesh traversal algorithm to meet. Hiding bits inside mesh vertices should not alter the referencing order of vertices. It should remain exactly same during both embedding and extraction phases.

Mesh traversal algorithm makes the stego-mesh resistant against vertex reordering attack. Vertex reordering of the mesh changes the reference index of vertices. Fig. 4.2 shows an example of vertex reordering attack. In left mesh, Face 1 consists of vertices *p, q,* and *r* while Face 2 consists of vertices *p, r,* and *s*. In right mesh, Face 1 has vertices *p, q,* and *s*; and Face 2 has vertices *q, r,* and *s*. No change has been done in these two meshes except the reference indices of vertices of mesh. These meshes are vertex reordering attacks for each other. Thus, if these reference indices are taken for hiding bits, then the extraction procedure would fail.



1	p	q	r
2	p	r	s

1	s	p	q
2	s	q	r

Fig. 4.2. Vertex reordering attack

Geometry and topology of the mesh remains intact but the reference indices are changed. Since only the vertex indices are changed; mesh looks similar as the previous one after attack. However, the retrieved information from stego mesh may be incorrect as the reference indices are different during embedding and extraction phases. Thus, it is a desirable characteristic of steganography algorithm to follow a mesh traversal algorithm while referring to vertices instead of indices.

The proposed mesh traversal algorithm is based on breadth first search. A source vertex is enqueued and all its neighbours are discovered. Neighbours are enqueued in the queue in increasing order of distance from source vertex. The source vertex is a vertex nearest to the gravity centre of 3D mesh model. All neighbours of it are dequeued one by one. Afterwards, neighbours of dequeued vertex are discovered and enqueued into queue. This process goes on until all the vertices have been discovered and visited.

Even after vertex reordering attack, the proposed mesh traversal algorithm will visit the vertices in the same order. It is observed from Fig. 4.2, if the mesh traversal order is p, q, r, and s for mesh on left hand side, then it would be s, p, q, and r for the mesh on right

hand side. Therefore, the traversal-order of vertices of mesh remains same irrespective of their reference indices.

The proposed traversal algorithm gives same traversal order even if 3D model has been rotated, scaled or transformed. In case of rotation of 3D mesh model, distances between vertices would remain unaffected and same mesh traversal order would be generated. In case of uniform scaling, all the vertices would be scaled and hence no change in the traversal order would occur. In case of translation, 3D model in some other location would not alter the distances between the vertices and thus the traversal order remains intact. Thus, the traversal order is indifferent towards any changes in 3D model. It shows resistance against vertex reordering attacks. Algorithm 1 describes the proposed mesh traversal algorithm.

Algorithm 4.3: Mesh traversal algorithm

Input: 3D Mesh and source vertex

Result: Traversal order

1. Mark all vertices of Mesh as unvisited
2. Enqueue in Queue, Q the source vertex
3. Mark source vertex as visited and put in traversal order
4. **while** (Q is not empty)
5. v ← Q.dequeue( ) //Removing vertex whose neighbours will be visited
6. Find all neighbours of v
7. Calculate distances of all neighbours from v.
8. **for** all neighbours w in ascending order of their distances from v
9. **if** w is not visited
10. Q.enqueue(w)
11. mark w as visited and put in traversal order.
12. **end_if**
13. **end_for**
14. **end_while**

Fig. 4.3 shows visiting queue for each step. Vertices being visited are shown in bold. Vertex 2 is nearer to vertex 1 than vertex 3. So, vertex 2 has been enqueued prior to vertex 3 in visiting queue. Likewise, it is done for all the vertices in the mesh.

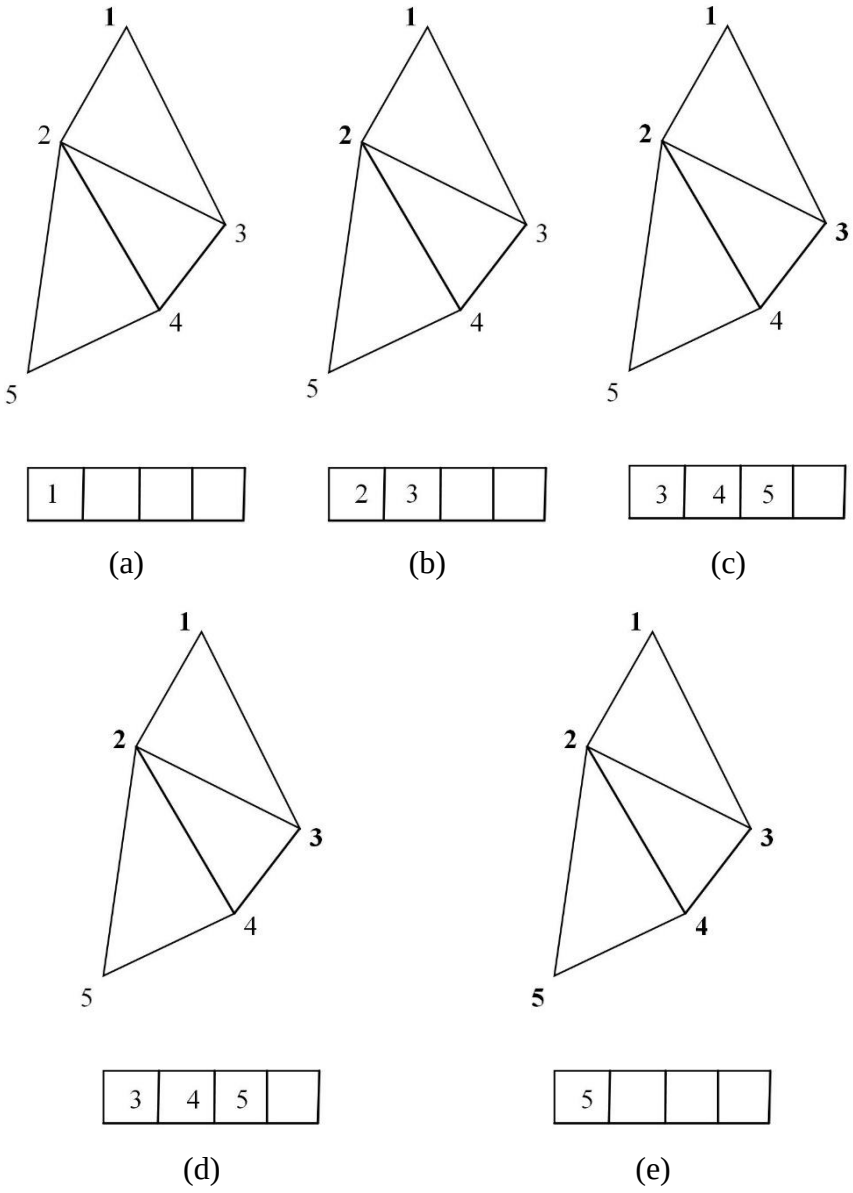


Fig. 4.3. Steps of mesh traversal algorithm from (a)-(e) along with queue

The proposed mesh traversal algorithm gives the same unique mesh traversal order, irrespective of the times it is run over the 3D mesh model. It is based on the distances between neighbouring vertices. So, if the distances between the vertices are not altered when the mesh model is attacked, same mesh traversal order is generated by the algorithm.

When 3D mesh model is attacked by rotation and translation, the distances between vertices is not altered at all. Thus, the mesh traversal algorithm gives same mesh referencing indices order when the mesh model has been rotated or translated. However, when the 3D mesh model is scaled using a uniform scaling factor, the distances between the vertices is changed. Uniform scaling of the 3D mesh model causes the scaling of distance by a factor of s, s being the uniform scaling factor. A comparison between the distances of neighboring vertices is done and even after scaling of 3D mesh model, the chronological order still remains same.

4.2.1.2 Generation of logistic chaotic map

During preprocessing, a chaotic sequence is generated using chaotic logistic map as mentioned in Eq. 4.1

$$a_{i+1} = \mu a_i(1 - a_i) \tag{4.1}$$

where a_i is the chaotic sequence and μ is control parameter. For $3.569945 < \mu \leq 4$, the system goes into chaotic state. Fig. 4.4 shows the bifurcation diagram of logistic map. As per my best knowledge, chaotic map has not been used in 3D steganography system. Chaotic map is used in the proposed steganography system in order to randomly hide data in one of the three coordinates of a 3D vertex, i.e., *x*, *y*, and *z*.

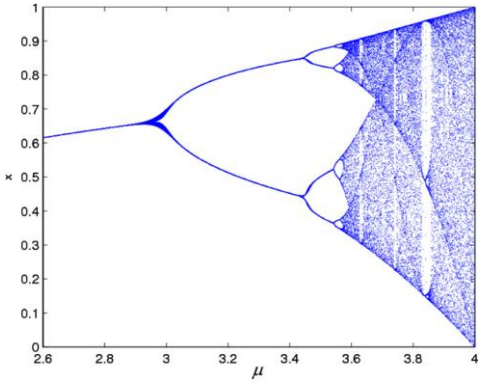


Fig. 4.4. Logistic map bifurcation diagram [54]

Out of three coordinates of a vertex, x, y, and z; only one coordinate value is used for hiding secret bit. Values from logistic map are multiplied by 3 and ceil operation is applied to find out in which coordinate (x, y or z), secret bit is hidden.

Using a chaotic map in embedding has three advantages. First, secret bits are hidden randomly in three coordinates. Embedding of secret bits is done in a random manner. This makes the embedding pattern difficult to deduce for intruders. Second, secret bits are not hidden only in one-coordinate axis. Anish et al. [43] hid secret bits in x-coordinates only of the mesh vertices. Hiding in only one coordinate may cause more distortion to the cover model. Thus, embedding of secret bits is done in all three coordinates. Distribution of secret bit data to all three coordinates causes much less distortion. Third, chaotic systems can be regenerated given the exactly same initial conditions [65]. Exactly same chaotic sequences can be regenerated at the receiver end during extraction of secret bits that results in the correct extraction of secret bits.

4.2.1.3 Processing of secret message

The secret message is hidden inside the 3D mesh model. The secret message is first converted to an encrypted form. This encrypted form is then converted into a bit stream and afterwards it is hidden inside 3D mesh model. In this preprocessing step, the secret message which is in the form of image is encrypted. Image encryption of the secret image is done as mentioned in Chapter 3. Then, the decimal valued pixels of encrypted image are converted to their binary equivalents. Next, the binary equivalents are joined together to form a binary bit stream. This binary bit stream is taken and bits are hidden one-by-one in 3D mesh model.

4.2.2 Embedding process

All vertices of the mesh model are labelled as - *fresh*, *dirty* and *embeddable*. All the unused vertices are labelled as *fresh*. From a set of *fresh* vertices, an *embeddable* vertex (on which embedding can be done) is chosen and after embedding it becomes *dirty*. It implies that it cannot be used anymore as shown in Fig. 4.5. In this figure, a white vertex shows an unused vertex, while a greyish-white implies that it can be used for embedding and a grey vertex shows that it cannot be used for embedding.

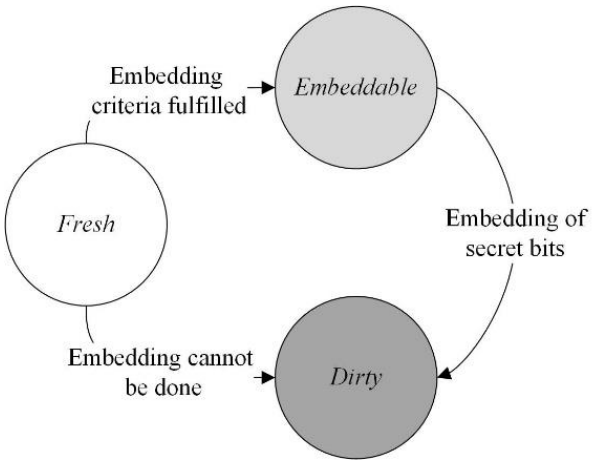


Fig. 4.5. Labels of mesh vertices

As soon as a vertex is discovered using the above-mentioned mesh traversal algorithm, it is labelled as *fresh.* Then this *fresh* vertex is taken and checked if its neighbours are more than the threshold value. In case it satisfies the embedding criteria, this vertex is marked as *embeddable*. If the *fresh* vertex does not satisfy embedding criteria then it is marked as *dirty*, which implies it won't be looked up again as it is already *dirty.* After the embedding of secret bits is done, *embeddable* vertex and its neighbour used up for embedding of secret bits are marked as *dirty.*

Number of neighbours associated with this *fresh* vertex is counted. The proposed steganography scheme is adaptive in nature. It distinguishes between smooth and noisy surfaces for embedding of secret bits. On a smooth surface, there will be less vertices or points than on a noisy surface. Embedding of secret bits is done inside the vertices present on noisy surface while the vertices of a smooth surface are kept undisturbed. This results in very less distortion caused by embedding of secret bits in 3D model. Since the number of vertices on a surface remains invariant to distortion less attacks, the proposed steganography algorithm withstands these attacks. Number of neighbours is a threshold parameter for embedding of secret bits. Once an *embeddable* vertex is found, its first *fresh* neighbour is taken and checked if this neighbour is *embeddable* too. If these two are *embeddable*, then they form the vertex pair on which embedding is done.

The proposed embedding algorithm is loosely based on difference between the coordinates of vertices. This method is based on the difference expansion by Tian [39]. Let the vertex pairs in which b-bit is to be embedded is (x_1, y_1, z_1) and (x_2, y_2, z_2). From logistic map values, it is decided in which of coordinates of vertex pair, the secret bit will

be hidden. Assuming that the secret bit is to be hidden inside x-coordinate of vertex pair. Average (l) and difference (h) between x_1 and x_2 can be computed as follows.

$$l = (x_1 + x_2)/2 \tag{4.2}$$

$$h = x_1 - x_2 \tag{4.3}$$

Note that this difference is the absolute difference and hence it always a positive number.

Afterwards, a suitable value of ten is searched in order to hide secret bit in x_1 and x_2. This suitable power of ten is obtained by making use of x_1 and x_2. These values are multiplied by 10 repeatedly until the non-decimal parts of x_1 and x_2 are unequal.

For example, let's assume that $x_1 = 0.024$ and $x_2 = 0.028$. The difference between these points is 0.004. Now, values of x_1 and $x_2$are multiplied by 10. The non-decimal parts of x_1 and x_2 are both equal to 0. So, they are multiplied with 10 again and 2.4 and 2.8 is received in which still non-decimal part is equal. One more multiplication by 10 produces 4.0 and 8.0 in which non-decimal parts are unequal. Thus, the multiplication with 10^3 gives the desired result. From this particular case, it is observed the suitable power of ten is 3. Note that the actual coordinate values are not altered while performing this step.

The next step of proposed approach is secret bit hiding step. The mathematical formulation is given below.

$$\acute{h} = \frac{h}{10^n} + \frac{\mathrm{b}}{10^n} \tag{4.4}$$

where $\acute{h}$ is the modified difference, b is decimal conversion of secret bit to be hidden and n is the suitable power of ten found above. Eq. (4.4) is the difference shifting step in proposed 3D image steganography. Now that modified difference value has been obtained, coordinate values are modified with it.

If $x_1 > x_2$then $\acute{x_1} = l + \acute{h}/2$; $\acute{x_2} = l - \acute{h}/2$

Else $\acute{x_1} = l - \acute{h}/2$; $\acute{x_2} = l + \acute{h}/2$ (4.5)

where $\acute{x_1}$ and $\acute{x_2}$ are modified x-coordinate values of vertex pair.

Proceeding with the above-mentioned example of $x_1 = 0.024$ and $x_2 = 0.028$, average $l = 0.026$, $h = 0.004$ and assuming $b = 1$ the modified distance is calculated as $\acute{h} =$ 0.001004. Using Eq. (4.5), the values of $\acute{x_1}$ and $\acute{x_2}$ are calculated as 0.025498

and 0.026502 respectively. Fig. 4.6 illustrates the example in detail. It may be noted that no change is done in y and z coordinate value of vertex pair as the embedding is done only in x-coordinate of vertex pair.

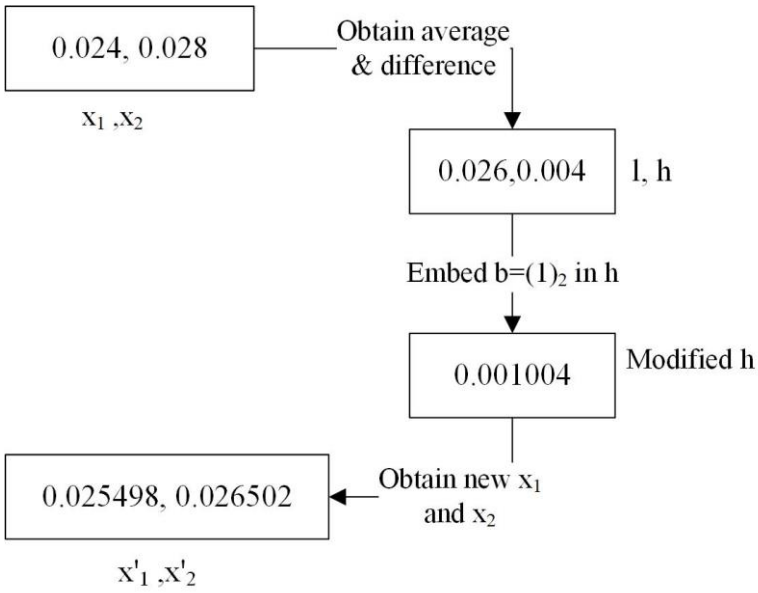


Fig. 4.6. Embedding based on difference shifting

As can be observed, modified values are very close to the original values; thus, resulting in very less distortion. Similarly, when embedding is done in y and z coordinates of vertex pair then only those coordinate values are modified and other values are kept intact. This reduces distortion caused by embedding in all three coordinates as done in some previous reversible data hiding techniques for 3D models [34, 41].

In the proposed approach, vertex pair is taken up for embedding secret bits. Since a vertex has many neighbours, it is important to mark the embedded vertex as used in order to differentiate it from the unused vertices. After the modification of coordinate values, vertices involved in forming this vertex pair are marked as *dirty* so that no more modification is done on them. Thus, while selection of vertex pair for embedding, these *dirty* vertices are not selected.

Embedding capacity can be increased by a small change in embedding process proposed above. In eq. (4.4), b is the decimal value of the secret bit to be hidden. In place of taking 1 secret bit, 2 or 3 bits may be taken simultaneously and its decimal equivalent is then evaluated in order to hide secret information.

For instance, let the secret bit stream be '1011001001…', instead of taking '1' for hiding inside 3D model, '10' or '101' can be taken from bit stream. Thus, in place of hiding 1(= $(1)_2$) inside vertex coordinate, 2(= $(10)_2$) or 5(= $(101)_2$) will be hidden.

An interesting point to observe here is that for embedding, more than 3 bits cannot be taken at once from bit stream. This is because decimal equivalent of more than 3 bits may result in a 2-digit decimal equivalent and adding it to difference value will modify difference in such a way that the original difference cannot be retrieved.

For example, in the above-mentioned case, if $(1011)_2$ is taken, then $(11)_{10}$ is to be hidden. Let's take original difference to be 0.5, shifting the difference would make it 0.05, Now addition of 0.05 and 0.11; makes new difference as 0.16. Extraction process would wrongly interpret 1 from modified difference as the secret information and set difference as 0.6 in place of 0.5! Thus, hiding more than 3 bits at a time inside a vertex pair destroys the reversible characteristic of proposed steganography scheme. The flowchart of proposed approach is shown in Fig. 4.7.

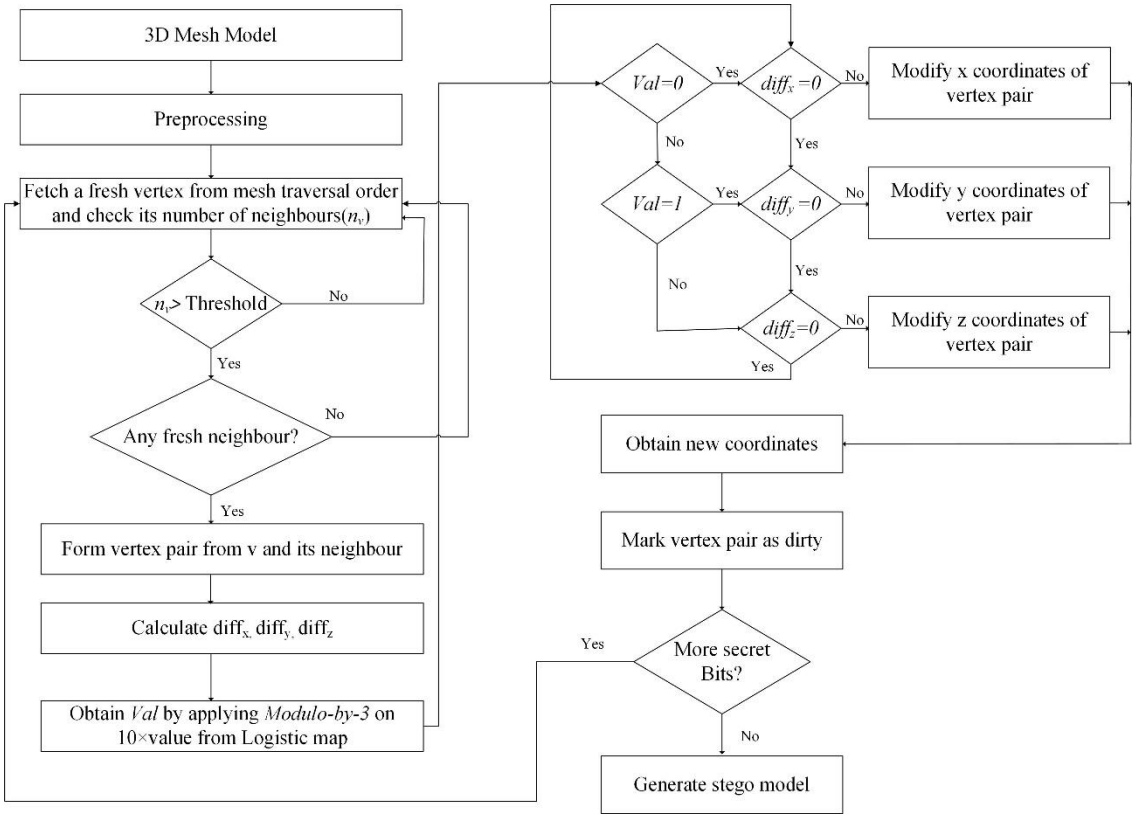


Fig. 4.7. Proposed Steganography system

4.2.3 Extraction process

In this phase, stego-model is processed at the receiver side. It consists of three steps. First, correction of stego-model is done if attacked by RST attacks. Second, extraction of secret bits from the rectified stego-model, followed by restoring of secret bits to form the hidden secret image in the last step. After execution of last step, the secret message is revealed to receiver through restored secret image.

4.2.3.1 Response of stego-model to RST attacks

The stego-model may have been subject to rotation, uniform scaling, and transformation on its way to the receiver. These attacks are distortion less attacks; i.e. no distortion is done in these attacks. But these attacks are potent enough of destroying the secret bits embedded inside the mesh vertices. The proposed steganography algorithm withstands these attacks. Rotation causes rotation of mesh points around some point in space. In the proposed embedding process, angles between vertices are not modified. Also, the rotation of mesh around some point would not alter distances among points, though the absolute difference between points will be altered. In case of translation of mesh to some other location, the distances would not be changed. The absolute differences would also remain same. Scaling would cause the distances among these points may increase or decrease. Average and difference needed for embedding may also get affected by scaling operation. Thus, the secret information hidden may be incorrectly extracted from attacked stego-model subjected to rotation or scaling. To save the secret data from being hampered by scaling of mesh, a simple solution is to undo the impact of scaling on the mesh. This is done by using 3D Helmert transformations for solving RST registration [37].

$$\hat{P}_i = T(t_x, t_y, t_z) + SR(\omega, \varphi, \kappa)P_i \tag{4.6}$$

$$\begin{bmatrix} \hat{x} \\ \hat{y} \\ \hat{z} \end{bmatrix} = \begin{bmatrix} t_x \\ t_y \\ t_z \end{bmatrix} + SR(\omega, \varphi, \kappa) \begin{bmatrix} x \\ y \\ z \end{bmatrix} \tag{4.7}$$

where $\hat{P}_i$is point from attacked stego model, P_i is corresponding point in stego mesh, $T(t_x, t_y, t_z)$ is translation vector, S is the scaling factor and

$$R(\omega, \varphi, \kappa) = \begin{bmatrix} cos\varphi cos\kappa & sin\omega sin\varphi cos\kappa - cos\omega sin\kappa & cos\omega sin\varphi cos\kappa + sin\omega sin\kappa \\ cos\varphi sin\kappa & sin\omega sin\varphi sin\kappa + sin\omega sin\kappa & cos\omega sin\varphi sin\kappa - sin\omega cos\kappa \\ -sin\varphi & sin\omega cos\varphi & cos\omega cos\varphi \end{bmatrix} \tag{4.8}$$

i.e. $R(\omega, \varphi, \kappa)$is rotation vector for rotating x, y and z axes using angles ω, φ and κ respectively. For solving these equations, 3 matched point pairs are required. Hence, these 3 matched point pairs are recorded in original stego model for matching with those in attacked mesh. Scaling increases or decreases distances of all points, so relative distance is not affected and hence the mesh traversing order remains intact. Thus, the points are compared to one another using the indices from mesh traversal algorithm. After solving the unknowns, the effect of RST on stego model can be undone giving back the stego model. This mesh model will be equal to the stego model sent from the sender side to receiver. Hence, it is given to the next stage of extraction of secret bits from it.

4.2.3.2 Extraction of secret bits

In extraction procedure, the same steps are followed as mentioned in Fig. 4.6. However, modifications are done in these steps to reverse the effect of hiding secret bits. That is to say, division is replaced by multiplication and addition by subtraction. Logistic chaotic map is also used here at the receiver side with values of a_0 and μ from sender side. Values obtained from a_i chaotic sequence are multiplied by 3 and then ceil is taken. This operation gives an integer in range [0, 2] which reveals the coordinate where secret bit/s is hidden. The embedding procedure is followed next, but the operations are performed in reverse to obtain the original cover model back.

For instance, let the vertex pair to be processed is $(\acute{x}_1, \acute{y}_1, \acute{z}_1)$ and $(\acute{x}_2, \acute{y}_2, \acute{z}_2)$. From the logistic map, it is revealed that bit is hidden in x-coordinate. So only $\acute{x}_1$ and $\acute{x}_2$ are manipulated. The suitable power of ten, i.e., *n* is obtained again from $\acute{x}_1$ and $\acute{x}_2$.

Obtaining average, $l = (\acute{x}_1 + \acute{x}_2)/2$ (4.9)

$$\acute{h} = \acute{x}_1 - \acute{x}_2 \quad (4.10)$$

$$h = \acute{h} \times 10^n - \lfloor \acute{h} \times 10^n \rfloor \quad (4.11)$$

$$b = \lfloor \acute{h} \times 10^n \rfloor \quad (4.12)$$

Now average and difference values are obtained, original x_1 and x_2 are obtained using eq. (4.5).

In such a way, secret bits from all points are retrieved and combined to get the secret information being hidden inside 3D stego model. The proposed steganography scheme

has blind extraction method as it does not need the original cover model for extraction of secret bits. Also, it is reversible as it gives back the original cover model.

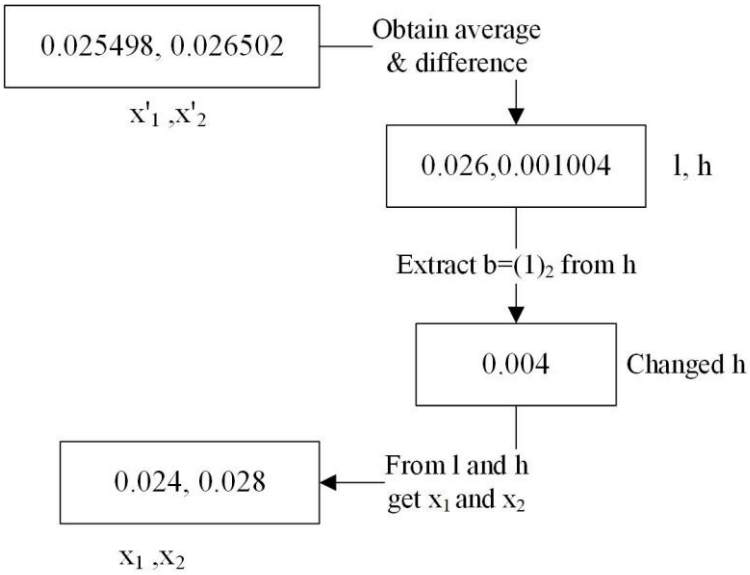


Fig. 4.8. Extraction process

As illustrated in Fig. 4.8, taking the modified values of above-mentioned example, i.e. $\acute{x_1} = 0.025498$ and $\acute{x_2} = 0.026502$. Average comes out to be0.026 and the difference comes out to be 0.001004. The original difference can be restored back using eq. (4.10) as 0.004 and the secret bit is revealed to be 1. Using the original difference and average of x_1 and x_2, they are given as $x_1 = 0.024$ and $x_2 = 0.028$. So, the original values of x_1 and x_2 are restored back after extraction of secret bits from the 3D stego-model. Thus, the entire 3D mesh model is retrieved back at the receiver side after extraction of secret bits from 3D stego-model. This is reversibility feature of the proposed 3D steganography algorithm that the entire 3D mesh model is received back at the receiver side.

The extraction process is able to give back the hidden secret bits without taking help of the cover model. Thus, the extraction process is blind in nature. Hence, the proposed steganography algorithm exhibits the desired characteristics- reversibility and blind extraction method.

4.2.3.3 Restoration of secret message

From extraction process, secret bits are obtained. These secret bits however need to be restored to its original form so that hidden message is conveyed to the receiver. From

binary representations, decimal valued pixels are obtained and then combined to form the secret image. The obtained secret image is encrypted image. Thus, the secret message is conveyed when the encrypted image is decrypted. Secret information is then revealed to the receiver from the decrypted image. This is the last step in proposed steganography algorithm.

4.3 Experimental results and discussions

The performance of the proposed steganography algorithm was tested on Intel i7-7500U processor running on Windows 10 operating system.

4.3.1 Experimentation set up

The performance of 3D steganography is tested on different 3D mesh models obtained from Stanford graphics laboratory Computer graphics (2018) such as dragon, bunny, and drill. Table 4.1 depicts the mesh models to be used as cover models and their number of vertices and faces.

Table 4.1. Description of 3D Mesh models

Cover Model	**Number of vertices**	**Number of faces**
Bunny	34834	69451
Dragon	100250	202520
Drill	1961	3855
Armadillo	172974	345944
Buddha	543652	1087716

For embedding and extraction logistic chaotic map was generated with values $\mu = 3.7893$ and $a_i = 0.3333$. Stego-model was rotated, scaled, and translated using MeshLab (2018). The attacked stego-model is then corrected and extraction of secret bits was done.

4.3.2 Performance metrics

In order to demonstrate the performance of the proposed steganography approach, two performance metrics namely, capacity and distortion are used. Distortion should be as low as possible while capacity of the proposed approach should not be compromised. A

trade-off between these parameters is always a daunting task for the algorithm designers. The proposed approach has been evaluated in terms of capacity and distortion.

4.3.2.1 Embedding capacity

Embedding capacity of a 3D steganography algorithm is the amount of secret data bits that can be hidden inside a vertex of 3D cover model. The proposed approach uses a single coordinate value of a vertex in order to hide 1 secret bit. Thus, the embedding capacity is 1 bit per vertex pair.

Table 4.2 shows the embedding capacity of mesh models using the proposed approach. When 1-bit of secret information is hidden at a time, less than half of the total number vertices carry the secret information. On the other hand, hiding 2 or 3 bits at a time increases the carrying capacity of the mesh models. It approximately doubles or triples the capacity, which is understandable.

Table 4.2. Embedding capacity of the proposed approach

Cover Model	**Threshold value**	**Size of payload attached (in bits)**		
		Attaching 1-bit at a time	**Attaching 2-bits at a time**	**Attaching 3-bits at a time**
Bunny	4	17237	34474	51711
Dragon	5	41215	82430	123645
Drill	3	963	1926	2889
Armadillo	5	42643	85286	127929
Buddha	6	63429	126858	190287

Altering threshold values has a great impact on the carrying capacity of the mesh model. This can be seen in Fig. 4.9. These graphs are obtained by varying threshold values while embedding of secret bits. It is evident from these graphs that in small mesh models like drill, a lower threshold value seems to be producing better embedding results than the higher ones. This is reverse in case of larger mesh models like dragon. Other 3D mesh models such as Armadillo, Buddha and Bunny have been used as cover models in the proposed approach. The results obtained are shown in Fig. 4.9.

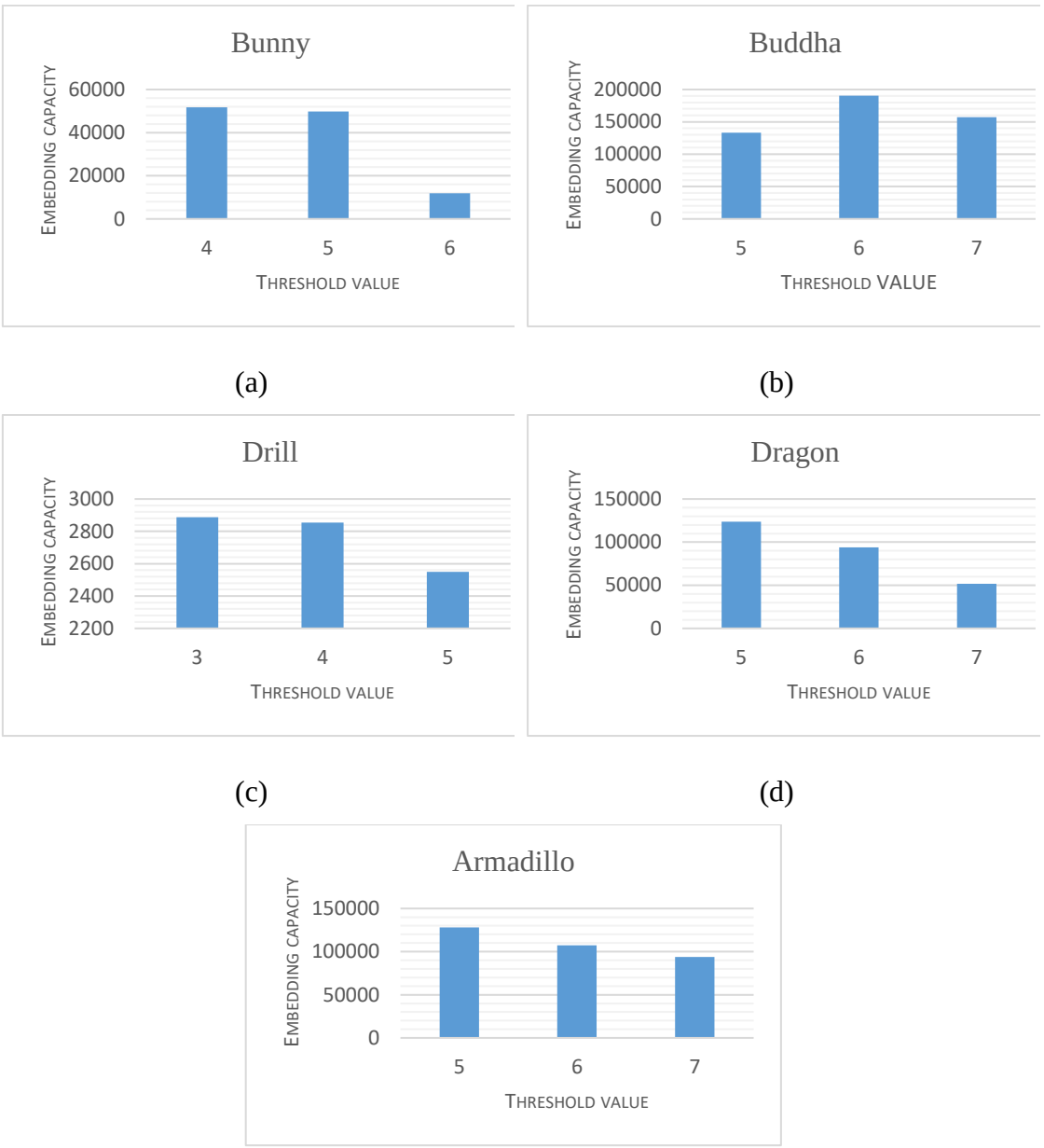


Fig. 4.9. Embedding capacity of (a) Bunny, (b) Buddha, (c) Drill, (d) Dragon and (e) Armadillo cover models for different threshold values

4.3.2.2 Normalised Hausdorff Distances (NHD)

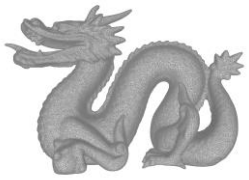
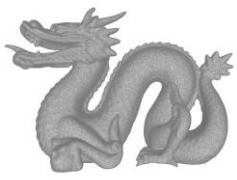
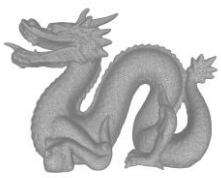
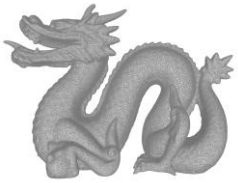
Normalised Hausdorff distance (NHD) is used to measure the similarity between stego model and cover model. It is very sensitive to any changes in the two models being compared. NHD is obtained by dividing Hausdorff Distance by diagonal length of the bounding box of mesh model. Values near 10^{-4} indicate visually acceptable distortion

[40]. From Table 4.3, it is evident that the values are acceptable in all the three cases. However, distortion on mesh may increases when the capacity goes high.

Table 4.3. NHD values after hiding heavy payloads inside 3D meshes

Cover Model	**Threshold value**	**Normalised Hausdorff distances (in 10^{-5})**		
		Attaching 1-bit at a time	**Attaching 2-bits at a time**	**Attaching 3-bits at a time**
Bunny	4	0.60	0.75	0.98
Dragon	5	0.44	0.51	0.73
Drill	3	0.08	0.16	0.33
Armadillo	5	0.51	0.58	0.71
Buddha	6	0.48	0.52	0.67

Table 4.4. Original and stego mesh models after hiding 1-3 bits of secret data

Original cover model	**Stego model after hiding 1-bit**	**Stego model after hiding 2-bits**	**Stego model after hiding 3-bits**
			
			

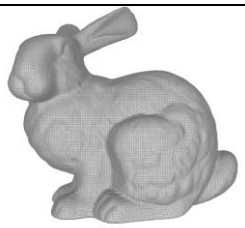
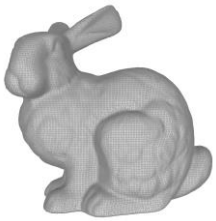
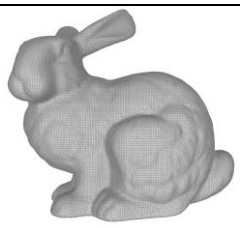
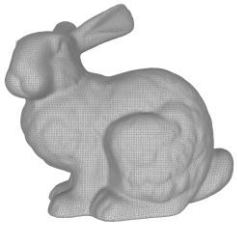
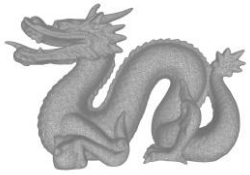
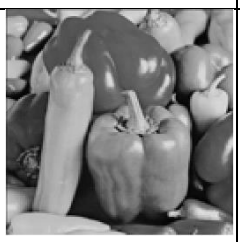
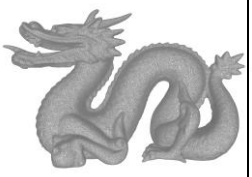
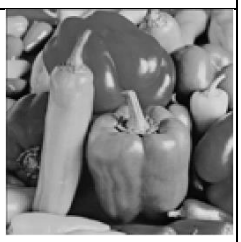
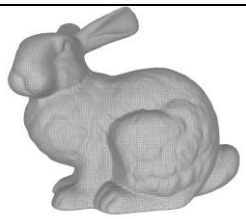
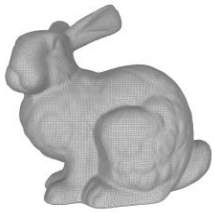
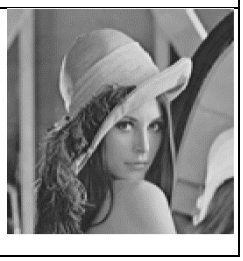
			

Table 4.4 shows 3D mesh models as cover and stego models. It is observed from it that perceptual distortions on the surface of 3D mesh model are minimal or extremely low. Thus, the proposed steganography system embeds secret data invisibly inside 3D mesh model. Wireframe model of the meshes has been included to judge the similarity and/or dissimilarity between stego and cover mesh model. After extraction of secret bits from stego model, the original cover model is retrieved along with correct secret bits. Embedding capacity can be increased but at the cost of increasing distortion. Performance evaluation has been done on both small (e.g. drill) and large (e.g. dragon) mesh models.

Table 4.5. Secret and extracted images by hiding 2 bits at-a-time in cover model

Model Name	**Original cover model**	**Secret image**	**Stego model**	**Extracted image**
Dragon				
Bunny				

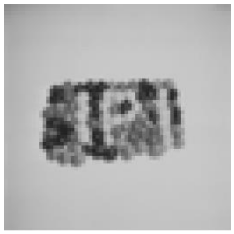
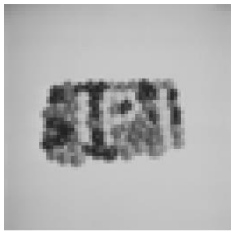
Drill				

Table 4.5 shows secret images hidden inside respective 3D cover models along with the extracted images. As can be noted, secret and extracted images are absolutely similar to each other. The extraction process successfully extracts the hidden secret image from stego-model without any loss of information bits. In Dragon and Bunny cover model, both Lena and Peppers of size 128×128 are hidden respectively. In Drill cover model, marbles image is hidden which is of size 64×64. Since Dragon and Bunny are large 3D mesh models, the dimension of secret image is large. Drill is a small cover model; thus, the secret image is of low dimension. Decimal-valued pixels of these images are first converted to their binary forms and then hidden inside 3D cover models.

4.4 Summary

In this chapter, the 3D steganography algorithm is proposed. The proposed system has three main phases. The preprocessing phase processes 3D mesh model as cover media and the secret image which is to be hidden. A mesh traversal algorithm is proposed in this chapter in order to safeguard 3D stego model against vertex reordering attack. The proposed mesh traversal algorithm is based on breadth first search. Novelty of the proposed mesh traversal approach lies in visiting the nearest neighbour first. Other advantage of the mesh traversal algorithm is that same mesh traversal order is generated even if the mesh has been rotated, scaled and translated. Proposed steganography approach introduces a novel difference shifting scheme. It is reversible in nature and has

blind extraction method. Another advantage of the proposed approach is that logistic chaotic map is used for the first time to hide secret bits in a random fashion inside the 3D mesh model. The results reveal that the proposed steganography algorithm hides the secret bits in 3D mesh models in such a way that the perceptual distortions are minimal. Secret image is extracted in its exact form from the 3D stego model. By changing threshold values, trade-off between distortion and embedding capacity is achieved.

Chapter 5

Robustness assessment of proposed approach

5.1 Introduction

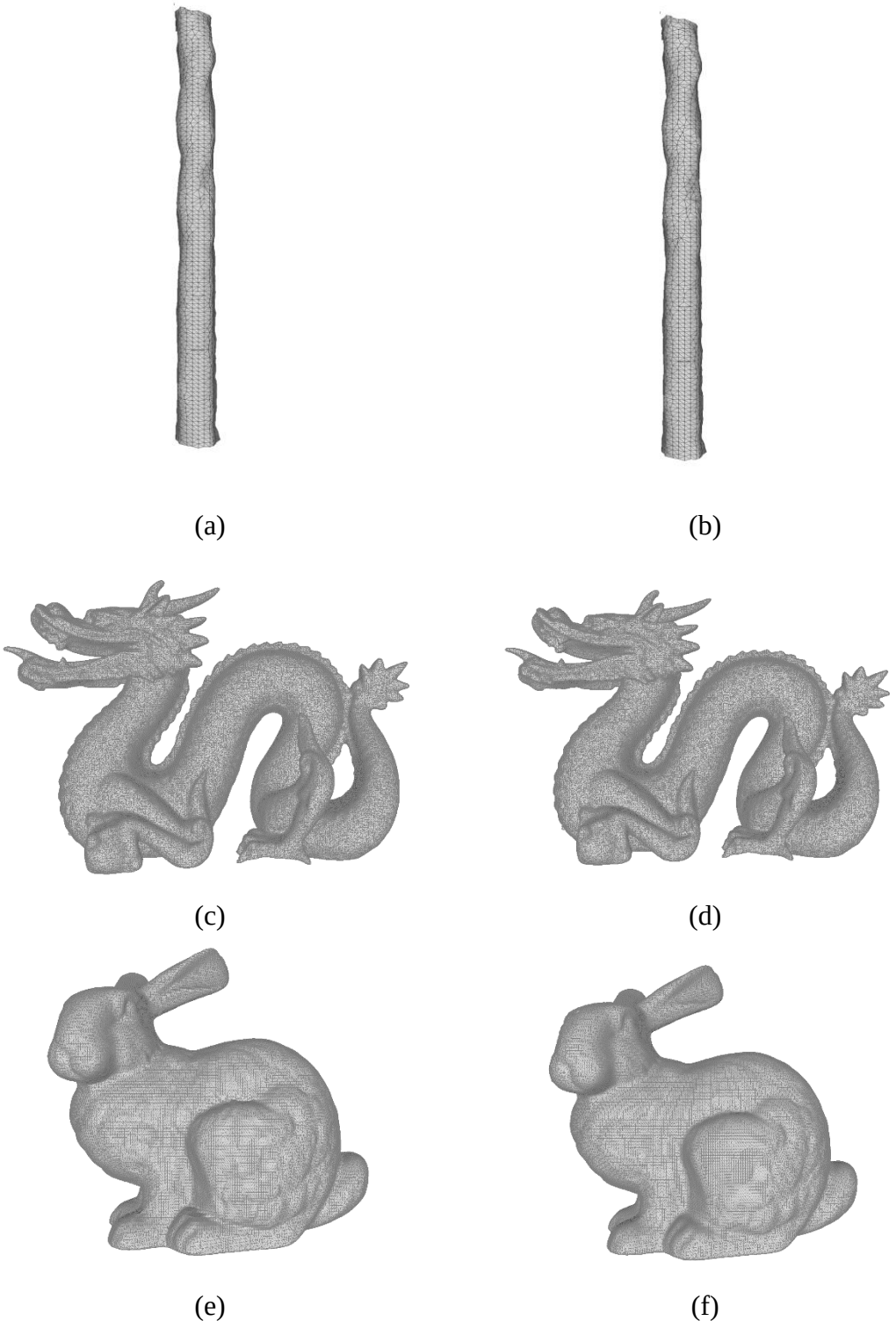
A steganography approach is termed as robust if it is able to yield correct secret message at the receiver side [6]. Robustness and security features are linked with each other, but are different to one another. Robustness of a steganography approach is assured if it is able to safeguard the secret message bits inside the stego media when the stego-media is subjected to attacks. Security, on the other hand, is checked by the ability of the steganography approach to withstand analysis by steganalysts. Attacks can be distortion less or distorting to 3D mesh model. In this chapter, distortion less attacks are discussed in detail and carried out on 3D stego mesh model. Ability of proposed steganography to withstand attacks is examined and its effects on 3D stego model are studied in detail in this chapter. Robustness of the proposed 3D steganography system is thereby evaluated.

5.2 Perceptual Transparency

The stego media should be similar to the cover media perceptually. Secret bits embedded inside the stego-media should not be visible to the naked eyes of intruders. Thus, the perceptual transparency of the embedded secret bits inside the stego-media plays a vital role in 3D image steganography. In the proposed work, the perceptual transparency of the secret bits is ensured.

A 3D mesh model represents a surface. Distortions on the surface of 3D mesh model are revealed when 3D mesh models are printed by 3D printers. Hence, in 3D image steganography imperceptibility is important. The distortions incurred on the 3D mesh model can be judged by the unevenness caused on the surface of the stego-media which might have occurred because of embedding secret bits. The proposed steganography

algorithm ensures the embedded secret bits remain unperceivable to intruders. This can be seen by comparing the cover media and stego media visually.



(a) (b)

(c) (d)

(e) (f)

Fig. 5.1. 3D Drill, Dragon and Bunny as cover model in (a), (c) and (e) and stego model (b), (d) and (f) respectively.

It can be observed from Fig. 5.1 that difference between stego model and cover model b naked eye cannot be made. The proposed steganography system is able to invisibly hide the secret bits inside the 3D mesh model. Therefore, the proposed steganography system withstands visual attacks by intruders over stego model.

5.3 Robustness against attacks

Ability of resisting the attacks defines the robustness of the stego-model. The attacker of 3D stego-model may or may not have any knowledge of stego-model. There are two types of attacks-distortion-less and distorting attacks. Table 5.1 lists attacks on 3D mesh model. It briefly explains the impact of the attack on the mesh model.

Table 5.1. Attacks on 3D mesh model

Attack	Type	Impact on mesh model
Translation	Distortion less	No impact
Rotation	Distortion less	No impact
Uniform scaling	Distortion less	No impact
Vertex reordering	Distortion less	No impact
Noise	Distorting	Random vertices are modified
Mesh smoothing	Distorting	Mesh surface is smoothened
Polygonal simplification	Distorting	Vertices are reduced
Cropping	Distorting	One or more parts of mesh are removed
Remeshing	Distorting	Topology of mesh is changed

5.3.1 Distortion less attacks and its consequences

Distortion less attacks are those attacks which do not change 3D mesh model because its geometry and topology remains intact. However, these attacks are capable of destroying the hidden secret bits in the cover model. In following sub-sections, these attacks are discussed in detail. The impact of these attacks on the proposed mesh traversal algorithm and 3D steganography algorithm are also discussed.

5.3.1.1 Translation

Translation of 3D mesh model causes the entire 3D mesh model to be shifted to some other location in geometry. This drags the entire 3D mesh model from its present location by a certain amount of distance in a particular direction. After translation of 3D mesh model is done, new coordinates can be obtained using Eq. 5.1.

$$\hat{P} = T.P \tag{5.1}$$

where P is coordinate matrix, $\hat{P}$ is changed coordinate matrix and T is translation matrix.

$$T = \begin{bmatrix} 1 & 0 & 0 & v_x \\ 0 & 1 & 0 & v_y \\ 0 & 0 & 1 & v_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{5.2}$$

Fig. 5.2 illustrates the new coordinates that can be obtained by adding the distance the points have moved in certain direction (or axis).

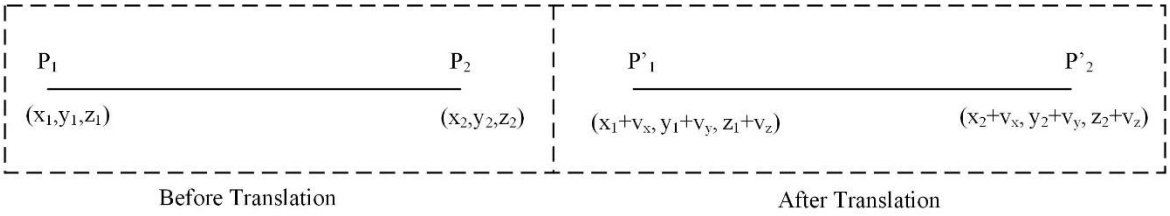
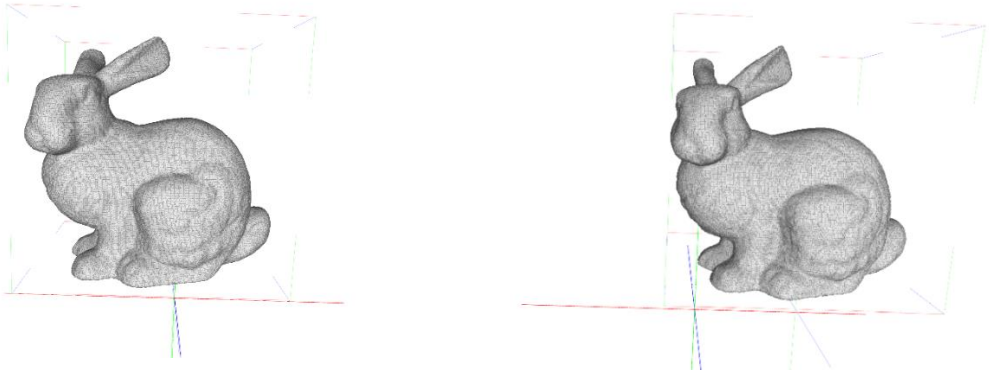


Fig. 5.2. Translation of mesh vertices

Translation of 3D mesh model does not cause any effect on the proposed mesh traversal algorithm. The proposed 3D image steganography algorithm also remains unaffected because it does not alter the distance and difference values.



Before Translation After Translation

Fig. 5.3. Effect of translation

From Fig. 5.3, it can be observed that even after translating stego model of bunny along x-axis, there is no difference between these two visually. No perceptual distortion on the surface of bunny as a result of embedding can be seen.

5.3.1.2 Rotation

Rotation of 3D mesh model can be done along x-axis, y-axis and z-axis. Rotation of points along any axis does not change the distance between points. However, it alters the absolute difference between the points because of rotation angle in it. As shown in Fig. 5.4, rotation of points along any axis causes the coordinates value to be affected by it as a result.

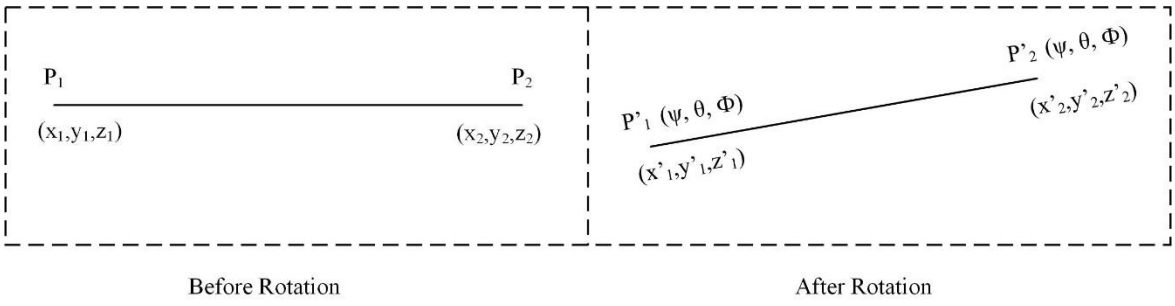


Fig. 5.4. Rotation of mesh vertices

When the points are rotated along z-axis, the new points can be obtained by the rotation matrix mentioned below.

$$R_z(\psi) = \begin{bmatrix} \cos(\psi) & \sin(\psi) & 0 \\ -\sin(\psi) & \cos(\psi) & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{5.3}$$

where ψ is the angle of rotation.

The new coordinates can be obtained by Eq. (5.4) if the rotation is done along the z-axis.

$$\hat{P} = R_z(\psi).P \tag{5.4}$$

where P is the coordinate matrix, $\hat{P}$ is the changed coordinate matrix and $R_z(\psi)$ is the rotation transformation matrix mentioned in Eq. (5.3).

Similarly, rotations along x and y axis can be determined using the following rotation matrices.

$$R_y(\theta) = \begin{bmatrix} \cos(\theta) & 0 & -\sin(\theta) \\ 0 & 1 & 0 \\ \sin(\theta) & 0 & \cos(\theta) \end{bmatrix} \quad (5.5)$$

$$R_x(\Phi) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos(\Phi) & \sin(\Phi) \\ 0 & -\sin(\Phi) & \cos(\Phi) \end{bmatrix} \quad (5.6)$$

where θ and Φ are angle of rotation in y and x axis respectively.

If rotation is carried out along x axis and then about y-axis(changed) and then about z-axis (changed changed)., the rotation transformation matrix becomes

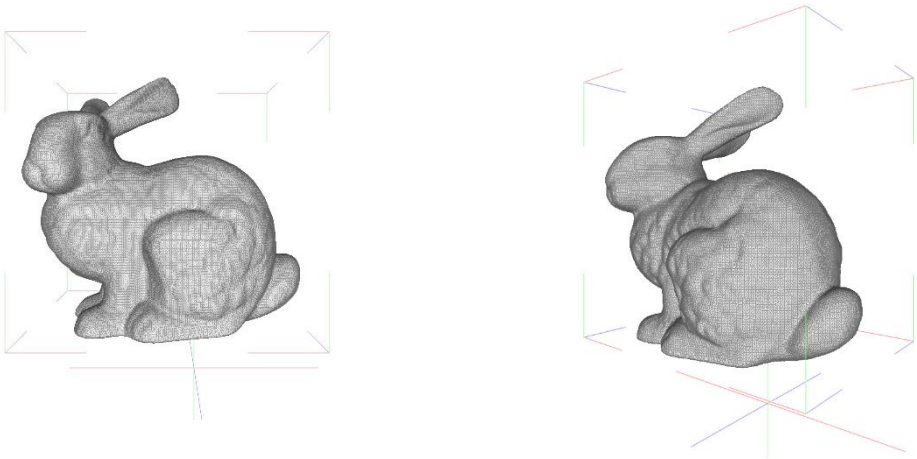
$$R = R_z(\psi).R_y(\theta).R_x(\Phi)$$

$$= \begin{bmatrix} \cos\theta\cos\psi & \sin\Phi\sin\theta\cos\psi + \cos\Phi\sin\psi & -\cos\Phi\sin\theta\cos\psi + \sin\Phi\sin\psi \\ -\cos\theta\sin\psi & -\sin\Phi\sin\theta\sin\psi + \cos\Phi\cos\psi & \cos\Phi\sin\theta\sin\psi + \sin\Phi\cos\psi \\ \sin\theta & -\sin\Phi\cos\theta & \cos\Phi\cos\theta \end{bmatrix}$$

(5.7)

New coordinate values can be obtained for points which have been rotated along x, y and z-axis. Distance between points does not change in case of 3D rotations but the absolute difference values change. As distance do not change, the mesh traversal algorithm will still give the same unique mesh traversal order even when the 3D mesh model is rotated. This is because the proposed mesh traversal algorithm works on the relative distances of one vertex to others. Even when the 3D mesh model is rotated along any axis, the relative distances remain same. Hence, same mesh traversal order is generated for a particular mesh model.

However, as the proposed 3D steganography algorithm is based on the absolute difference between the coordinate values, rotation effect on the points of 3D stego-model has to be reversed.



Before Rotation After Rotation

Fig. 5.5. Effect of rotation

It can be observed from Fig. 5.5, after rotation of stego model of bunny along the x-axis, any perceptual distortion cannot be seen on its surface.

5.3.1.3 Scaling

Scaling operation results in enlargement or shrinking of 3D mesh model by a given magnitude called scale factor. Scaling may be equal in all three directions (or axis) or it may be unequal. The former is called uniform or isotropic scaling in which scaling has same magnitude in all three directions is carried out. The other type of scaling is anisotropic or non-uniform scaling in which the scaling factor is unequal as shown in Fig. 5.6.

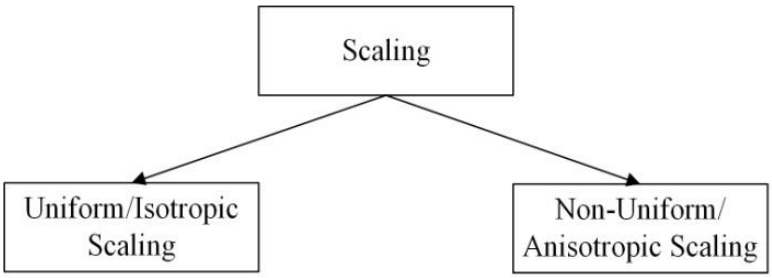


Fig. 5.6. Types of scaling

Scaling operation enhances or diminishes the object. The equation written below determines the new coordinates when scaling operation is carried out.

$$\hat{P} = S.P \tag{5.8}$$

where P is the coordinate matrix, $\hat{P}$ is the changed coordinate matrix and S is the scaling matrix.

$$S = \begin{bmatrix} s_x & 0 & 0 & 0 \\ 0 & s_y & 0 & 0 \\ 0 & 0 & s_z & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{5.9}$$

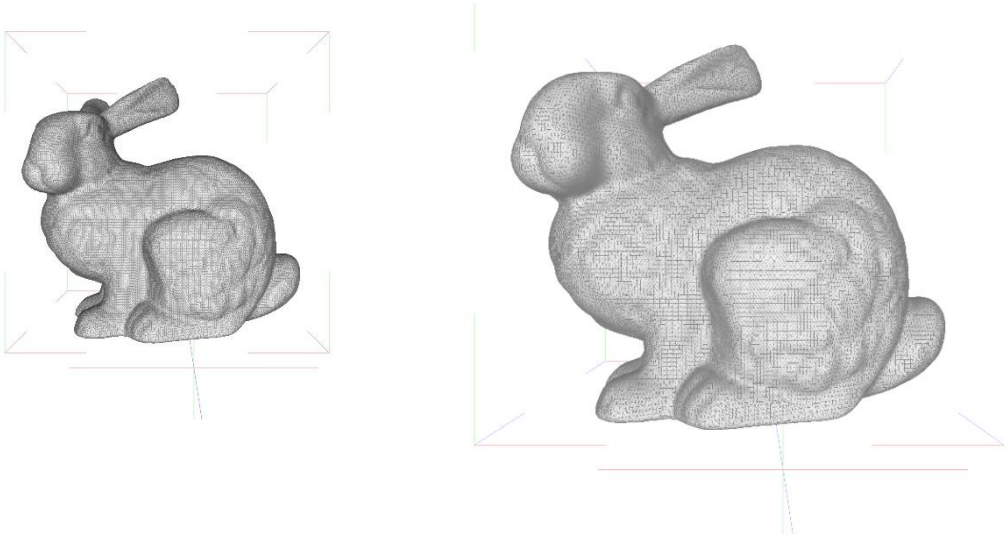
where s_x , s_y and s_z is the scaling factor in x-axis, y-axis and z-axis respectively.

In case of anisotropic scaling, $s_x \neq s_y \neq s_z$ whereas for uniform or isotropic scaling, $s_x = s_y = s_z = s$. In this work, impact of isotropic or uniform scaling is considered.

Fig. 5.7. Scaling of mesh vertices

Scaling on the vertices of the mesh model causes the distance between the vertices to increase or decrease by a factor of s, where s is the scaling factor for uniform scaling. As shown in Fig. 5.7, the new coordinates have been obtained by multiplying old coordinates by s. Area and volume of the 3D object also hence increases by a factor of s^2 and s^3 respectively. Any distortion on the surface of stego model is not seen even when the stego model is scaled as shown in Fig. 5.8.

In case of uniform scaling, the distances are altered in all three directions. Absolute difference between points is also altered. However, the proposed mesh traversal algorithms work on the relative distances between the points and not on the absolute distance values. If all the distances are increased or decreased by a certain amount, then the maximum and minimum distances remain same. Thus, the proposed mesh traversal algorithm remains indifferent towards changes in 3D mesh model.



Before scaling After scaling

Fig. 5.8. Effect of scaling

But the absolute difference in case of isotropic scaling increases or decreases by a factor of s. As shown in Fig. 5.7, before scaling absolute difference between P_1 and P_2 is $(x_2 - x_1)$. After scaling, the new difference becomes $(sx_2 - sx_1)$, which is $s(x_2 - x_1)$, i.e. s times the original difference value between the points. Thus, effect of scaling on the mesh vertices has to be reversed in order to obtain the correct secret bits from the 3D stego-model.

Since the effect of rotation and scaling has to be reversed, in the proposed steganography the 3D stego model is preprocessed at the receiver side. After rotation, scaling and translation of 3D mesh model, new coordinates can be obtained from the following equation.

$$\widehat{\mathrm{P}}_i = T(t_x, t_y, t_z) + SR(\omega, \varphi, \kappa)P_i \tag{5.10}$$

$$\begin{bmatrix} \hat{x} \\ \hat{y} \\ \hat{z} \end{bmatrix} = \begin{bmatrix} t_x \\ t_y \\ t_z \end{bmatrix} + SR(\psi, \theta, \Phi) \begin{bmatrix} x \\ y \\ z \end{bmatrix} \tag{5.11}$$

where $\widehat{\mathrm{P}}_i$ is point from attacked stego model, P_i is corresponding point in stego mesh, $T(t_x, t_y, t_z)$ is translation vector, S is the scaling factor and $R(\psi, \theta, \Phi)$ is the rotation factor written in eq.(5). In order to save the secret data being hampered, a simple solution is to undo the impact of rotation and scaling on the mesh. This is done by using 3D Helmert transformations for solving RST registration [37]. For solving these equations,

3 matched point pairs are required. Hence, these 3 matched point pairs are recorded in original stego model for matching with those in attacked mesh. Scaling increases or decreases distances of all points with each other, so it would not affect the mesh traversing order. Thus, the points are compared to one another using the indices from mesh traversal algorithm. After solving the unknowns, the effect of RST on stego model can be undone giving back the stego model. This mesh model will be equal to the stego model sent from the sender side to receiver.

Another interesting point is 3D stego model may or may not have been attacked by RST transformations. Thus, a difference between the attacked stego model and non-attacked stego model has to be made. This is made by checking the gravity centre of the stego model. Before sending the stego model to the receiver side, the gravity centre of the stego model is recorded. Next, the gravity centre of the stego model is calculated at the receiver side. If both the gravity centre are equal, then the stego model has not been attacked. But RST transformations would result in changing the gravity centre of the stego model as shown in Fig. 5.9. Hence the difference between the attacked stego-model and non-attacked stgeo model can be known.

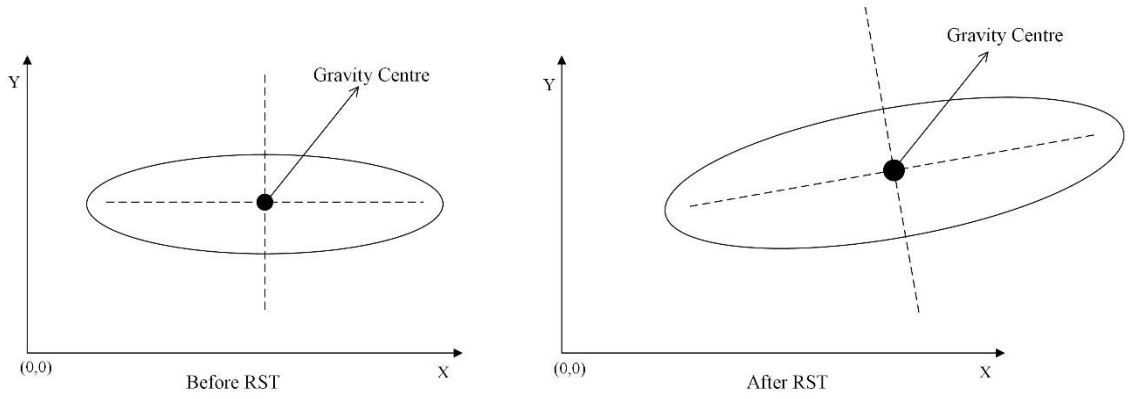


Fig. 5.9. Changed gravity centre after rotation, scaling and translation

Also, the gravity centre of the stego model will be different than that of cover model. This is because of the fact that the embedding of secret bits inside the mesh model would result in shifting of gravity centre from its original location. Thus, the gravity centre of the stego model is recorded in place of cover model.

5.3.1.4 Vertex Reordering

Vertex reordering attack is a distortion-less attack and does not change 3D mesh model.

Reference indices of vertices of 3D mesh model are changed in this attack. As shown in Fig. 5.10, vertex indices of the mesh vertices are changed and the mesh model remains intact.

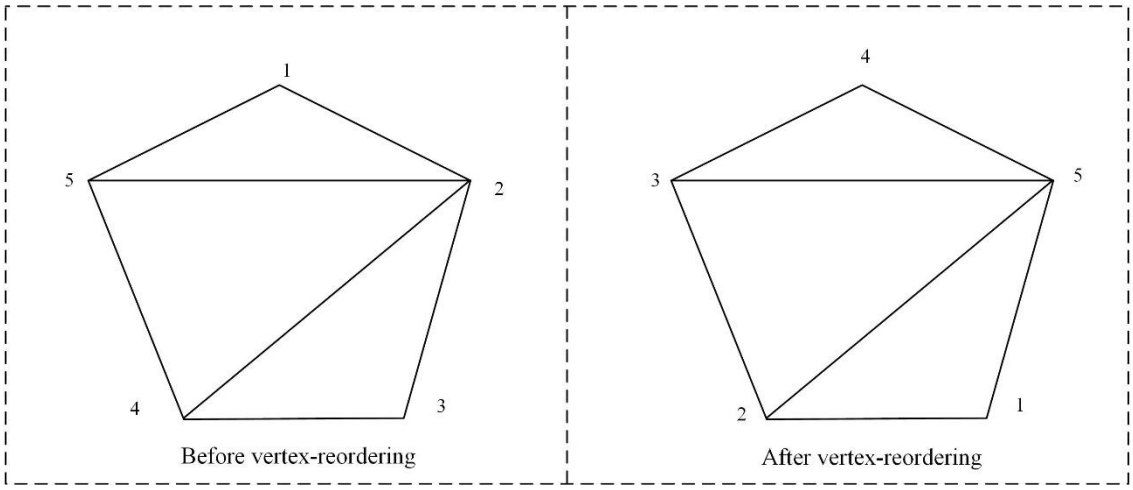


Fig. 5.10. Vertex reordering

There is no change in the mesh after the attack. However, if these reference indices are used for embedding of secret bits, then the attacked 3D stego model may not give the correct secret message at the receiver side. This is because extraction of secret bits from changed reference indices will not result in correct extraction of secret bits.

The proposed mesh traversal algorithm works on Breadth First Search and distances of points with one another in the mesh. It does not take up the reference indices of the vertices for traversing over mesh vertices. It generates a vertex visiting order for a particular 3D mesh model. Thus, the proposed mesh traversal algorithm withstands this attack.

The proposed 3D steganography algorithm uses the vertex visiting order generated by the mesh traversal algorithm in place of the reference indices of the vertices. Thus, the proposed 3D steganography algorithm withstands vertex reordering attack.

Table 5.2 discusses the ability of proposed mesh traversal algorithm and 3D steganography algorithm to resist these attacks. A tick mark (√) indicate that the algorithm resists the attack.

Table 5.2. Resistance towards distortion-less attacks

	Rotation	**Scaling**	**Translation**	**Vertex reordering**
Mesh traversal algorithm	√	√	√	√
Steganography algorithm	√ (after correction)	√ (after correction)	√	√

5.3.2 Distorting attacks and its consequences

Distorting attacks are those attacks which hamper the 3D mesh model. These attacks change the geometry or connectivity or both and destroy 3D mesh model along with the hidden secret message. These attacks include noise, mesh smoothing, geometry or topology compression, remeshing, cropping and polygonal simplification.

Table 5.1 illustrates that distorting attacks are able to change the geometry and/or topology of the mesh model. As the proposed mesh traversal algorithm works on changing the topology(connectivity) of the mesh vertices, any change in topology can affect the mesh traversal order generation mechanism. Geometry of the mesh model is modified in the proposed 3D steganography algorithm. Some distorting attacks may cause the removal of mesh vertices. Secret bit may get deleted along with the deletion of mesh vertices. So, any change in geometry of the mesh model is bound to harm the hidden secret information inside the 3D mesh model.

Hence, distorting attacks can destroy the hidden secret information. Thus, the proposed 3D steganography algorithm cannot withstand these distorting attacks.

5.4 Summary

Robustness analysis of the proposed steganography approach is done in this chapter. In this thesis, mesh traversal algorithm and steganography system are proposed. Both of them are evaluated in terms of their ability to withstand rotation, scaling, translation and vertex reordering attacks. Advantage of conducting robustness analysis of the proposed mesh traversal algorithm and 3D steganography algorithm is to verify if 3D stego-model can withstand distortion less attacks by intruders. It can be concluded that both of them

are able to resist these attacks. Mesh traversal algorithm will generate the same unique mesh traversal order for an attacked mesh model. Secret message bits are extracted correctly from an attacked stego model. This implies that the contents of 3D stego model are not hampered when it is attacked by rotation, scaling, translation and vertex reordering attacks. Distorting attacks such as cropping, mesh simplification, etc. are however, potent enough to destroy the hidden secret bits.

Chapter 6
Conclusion and future scope

6.1 Conclusion

In today's world, Internet has crept in our lives and information of all sorts is available at a few clicks. Hence protection and safeguarding of secret and confidential information from reaching wrong hands has become crucial. Information hiding is a type of information security that ensures that the information does not go to undesired destination. In this type of information security, the secret information is hidden inside a naïve-looking cover media in such a way that it is invisible to human eyes. In this thesis work, cover media is a 3D mesh model, thus the proposed steganography system is 3D image steganography system. The contributions of this thesis are mentioned below:

- A detailed literature survey of 3D image steganography is presented in this thesis. Various 3D image steganography techniques are discussed in detail. Reversible data hiding approaches in 3D image steganography are examined. These reversible data hiding approaches are based on reversible data hiding approaches in 2D image steganography. A 3D image mesh model is more difficult to sample as compared to 2D images. So, effects of embedding should not be permanent on 3D image model. Reversibility feature in data hiding approach assures that the 3D cover mesh model can be obtained in its exact form after extraction of secret bits. A 3D image steganography algorithm is of two types - spatial domain and transform domain. Spatial domain-based steganography algorithm modifies the vertices directly for embedding of secret bits. In case of transform domain, the mesh model is first taken to the transform domain and then modifications are done in order to hide secret bits. Afterwards, the transformed mesh model is brought back to the spatial domain. Algorithms designed in both domains are also discussed in detail. Types of attacks on 3D stego model are also discussed in detail.

- A novel image encryption algorithm is proposed for color image encryption based on Lorenz-Rossler chaotic system and cross-channel operations. The proposed image encryption algorithm reduces the correlation of pixels among each other. It enhances the information entropy of image. DNA cryptography rules are applied in order to increase randomness in plain image. XOR and addition operations between DNA sequences of Red, Green and Blue channel of plain color image are carried out to obtain encrypted image. Simulation of the encryption algorithm was done on several test images from USC-SIPI database [98]. Performance evaluation of the encrypted image was done by checking its correlation coefficient and information entropy values. Statistical analysis, key space and analysis, differential analysis and avalanche effect of the encrypted image was done in order to check the performance of the encrypted image. The results reveal that the proposed image encryption algorithm increases randomness in plain image by approximately 40%. Correlation coefficient of pixel with adjacent pixels reduced to the multiples of 10^{-5} Also, a unique and highly random encrypted image is generated for a given plain image.
- A novel mesh traversal algorithm is also proposed. The proposed mesh traversal algorithm is based on Breadth First Search (BFS). Neighbour nearest to the vertex is first visited, followed by other nearest neighbours. Mesh traversal algorithm gives a unique mesh traversal order for a particular 3D mesh model. Even if the mesh model is attacked by rotation, scaling and translation transformations, the mesh traversal algorithm gives the same mesh traversal order over the attacked mesh model. The order in which the vertices are visited becomes the referencing order for mesh vertices. Using mesh traversal order for referencing vertices ensures that steganography algorithm withstands vertex reordering attack. Vertex reordering attack can cause incorrect extraction of secret message bits.
- A reversible data hiding approach for 3D mesh models is also proposed. It has a blind extraction method and after the extraction of secret bits from the stego model, the cover model is received in its exact form. A novel difference shifting scheme is proposed for embedding of secret bits inside 3D mesh model. This scheme is loosely based on difference expansion scheme by Tian et al. [39] for reversible data hiding in 2D images. Logistic chaotic map is used for the first time in this scheme for hiding of secret bits in mesh vertices. Difference shifting

scheme changes the vertex coordinates by a very small amount and so the distortion caused is minimum. Proposed steganography approach is adaptive in nature. Adaptive nature implies that more secret bits are embedded on noisy surfaces and less or no secret bits are embedded at the smooth surfaces. As a result, distortion caused to the mesh model is less. Difference between noisy and smooth surface is made by counting the number of neighbours of a vertex on the surface. neighbours of a vertex on noisy surface are more than those on smooth surface.

- Perceptually, the hidden secret bits are not visible on the surface of 3D stego model. Response of 3D stego model when it is attacked by rotation, scaling and translation transformations is evaluated in this thesis. When stego model is attacked by these transformations, same mesh traversal order is generated for a particular mesh model. Embedding of secret bits does not get affected by these transformations. But in order to extract correct secret message, these transformations need to be reversed. Afterwards, extraction process is carried out and correct secret message is extracted. Impact of vertex reordering attack on the 3D stego model is determined.

6.2 Future Scope

The present work is in the direction of reversible data hiding for 3D image models. Some directions for future work in this field are laid down below:

- For encryption of color image Lorenz-Rossler hyper-chaotic system has been used. Using one more chaotic system in encryption algorithm will increase the key space and hence enhance security of the encryption algorithm.
- In the proposed image encryption algorithm, only substitution of pixels in three channels of plain image is done. In addition to substitution, permutation of pixels in three channels can be done in order to further increase information entropy in image.
- Mesh traversal algorithm is proposed using BFS (Breadth First Search) algorithm. Neighbour nearest to the vertex is visited first, followed by second nearest and so on. In place of taking distance between mesh vertices, angle made between edge and normal to the surface can be taken. Impact of rotation, scaling

and translation transformations should be checked when a new geometrical primitive is included in 3D Image Steganography system.

- Reversible data hiding approach on 3D mesh models is proposed in this thesis based on reversible data hiding approach in 2D image steganography. Other reversible data hiding approaches in 2D image steganography [108] can also form the basis of data hiding approaches in 3D Image Steganography.
- In the proposed 3D image steganography, logistic chaotic map is used for hiding secret bits in a random fashion so that steganalysts are not able to discover the pattern of hiding the secret bits. In place of using logistic chaotic map, other chaotic functions having less time and space complexity can be thought of as other options.
- In this scheme, neighbours table is constructed using the faces and vertices array. Construction of neighbours table takes up a lot of space and time complexity in case the 3D mesh models is large, e.g. Armadillo, Dragon, etc. This approach can be optimised for time-complexity. Space complexity comes into picture when large 3D mesh models are used as cover models. Hence it can also be optimised so that large 3D mesh models are preprocessed in less time and space complexity.

List of Publications

Papers Published:

1. Girdhar, A. and Kumar, V., 2017. Comprehensive survey of 3D image steganography techniques. *IET Image Processing*, *12*(1), pp.1-10. [**I.F. 1.401**]
2. Girdhar, A. and Kumar, V., 2018. A RGB image encryption technique using Lorenz and Rossler chaotic system on DNA sequences. *Multimedia Tools and Applications*, pp.1-23. [**I.F. 1.541**]
3. Girdhar, A. and Kumar, V., 2019. A reversible and affine invariant 3D data hiding technique based on difference shifting and logistic map. *Journal of Ambient Intelligence and Humanized Computing*, pp.1-15. [**I.F. 1.423**]

Paper Communicated:

1. Girdhar, A. and Kumar, V.,2019. 2D logistic map and Lorenz-Rossler chaotic system based RGB image encryption approach. *The Visual Computer* (**Under Review**).

Bibliography

[1] Cheddad, A., Condell, J., Curran, K. and Mc Kevitt, P., 2010. Digital image steganography: Survey and analysis of current methods. *Signal processing*, *90*(3), pp.727-752.

[2] Ye, G., 2010. Image scrambling encryption algorithm of pixel bit based on chaos map. *Pattern Recognition Letters*, *31*(5), pp.347-354.

[3] Amsden, N.D., Chen, L. and Yuan, X., 2014, July. Transmitting hidden information using steganography via Facebook. In *Fifth International Conference on Computing, Communications and Networking Technologies (ICCCNT)*, pp. 1-7. IEEE.

[4] Chan, C.K. and Cheng, L.M., 2004. Hiding data in images by simple LSB substitution. *Pattern recognition*, *37*(3), pp.469-474.

[5] Zaidan, B.B., Zaidan, A.A., Al-Frajat, A.K. and Jalab, H.A., 2010. On the differences between hiding information and cryptography techniques: An overview. *Journal of Applied Sciences (Faisalabad)*, *10*(15), pp.1650-1655.

[6] Morkel, T., Eloff, J.H.P. and Olivier, M.S., 2005, June. An overview of image steganography, information and computer security architecture (ICSA) research group. In *Proceedings of the Fifth Annual Information Security South Africa Conference (ISSA2005), Sandton, South Africa*.

[7] Fridrich, J., Goljan, M. and Du, R., 2001, April. Invertible authentication watermark for JPEG images. In *Proceedings International Conference on Information Technology: Coding and Computing*, pp. 223-227. IEEE.

[8] Xuan, G., Zhu, J., Chen, J., Shi, Y.Q., Ni, Z. and Su, W., 2002. Distortionless data hiding based on integer wavelet transform. *Electronics Letters*, *38*(25), pp.1646-1648.

[9] Celik, M.U., Sharma, G., Tekalp, A.M. and Saber, E., 2002. Reversible data hiding. In *Proceedings. International Conference on Image Processing*, (Vol. 2, pp.157-160). IEEE.

[10] Li, X. and Wang, J., 2007. A steganographic method based upon JPEG and particle swarm optimization algorithm. *Information Sciences*, *177*(15), pp.3099-3109.

[11] Sun, S., 2016. A novel edge based image steganography with 2k correction and Huffman encoding. *Information Processing Letters*, *116*(2), pp.93-99.

[12] Wang, C.M. and Cheng, Y.M., 2005, September. An efficient information hiding algorithm for polygon models. In *Computer Graphics Forum* (Vol. 24, No. 3, pp. 591-600). Amsterdam: North Holland, 1982-.

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

[14] Cheng, Y.M. and Wang, C.M., 2007. An adaptive steganographic algorithm for 3D polygonal meshes. *The Visual Computer*, *23*(9-11), pp.721-732.

[15] 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), pp.274-284.

[16] Tsai, Y.Y., 2014. An adaptive steganographic algorithm for 3D polygonal models using vertex decimation. *Multimedia Tools and Applications*, *69*(3), pp.859-876.

[17] Computer Graphics (2014). The Stanford 3D Scanning Repository. [online]. Available from: http://graphics.stanford.edu/data/3Dscanrep/.

[18] Ohbuchi, R., Masuda, H. and Aono, M., 1998. Watermarking three-dimensional polygonal models through geometric and topological modifications. *IEEE Journal on selected areas in communications*, *16*(4), pp.551-560.

[19] Kumar, V. and Kumar, D., 2018. A modified DWT-based image steganography technique. *Multimedia Tools and Applications*, *77*(11), pp.13279-13308.

[20] Yang, Y., Pintus, R., Rushmeier, H. and Ivrissimtzis, I., 2017. A 3D steganalytic algorithm and steganalysis-resistant watermarking. *IEEE transactions on visualization and computer graphics*, *23*(2), pp.1002-1013.

[21] Goljan, M., 2002, January. Practical Steganalysis—State of the Art. In *proceedings SPIE Photonics West, Electronic imaging 2002, Security and Watermarking of Multimedia Contents*, 4675, pp. 1-13.

[22] Yang, Y., 2013. *Information analysis for steganography and steganalysis in 3D polygonal meshes* (Doctoral dissertation, Durham University).

[23] Daily Telegraph (2012). Terror plot horror hidden in porn film, Adelaide now. [Online]. Available: http://www.adelaidenow.com.au/news/world/terror-plot-horror-hidden-in-porn-film/news-story/49886a10ddeccdfb341cf72c48bbf5d5.

[24] Kolata, G. (2001). Veiled messages of terror may lurk in Cyberspace, in Science, The New York Time. [Online]. Available: http://www.nytimes.com/2001/10/30/science/veiled-messages-of-terror-may-lurk-in-cyberspace.html.

[25] McCullagh, D. (2001) Bin Laden: Steganography master? WIRED. [Online]. Available: http://archive.wired.com/politics/law/news/2001/02/41658?currentPage=all.

[26] Aspert, N., Drelie, E., Maret, Y. and Ebrahimi, T., 2002, November. Steganography for three-dimensional polygonal meshes. In *Applications of Digital Image Processing XXV* (Vol. 4790, pp. 211-220). International Society for Optics and Photonics.

[27] Wagner, M.G., 2000, April. Robust watermarking of polygonal meshes. In *Proceedings Geometric Modeling and Processing 2000. Theory and Applications* (pp. 201-208). IEEE.

[28] Maret, Y. and Ebrahimi, T., 2004, September. Data hiding on 3D polygonal meshes. In *Proceedings of the 2004 workshop on Multimedia and security* (pp. 68-74). ACM.

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

[30] 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), pp.1423-1443.

[31] Pérez-González, F. and Balado, F., 2002, September. Quantized projection data hiding. In *Proceedings. International Conference on Image Processing* (Vol. 2, pp. 889-892). IEEE.

[32] Wu, H.T. and Dugelay, J.L., 2009. Steganography in 3D geometries and images by adjacent bin mapping. *EURASIP Journal on Information Security*, *2009*(1), pp.1-10.

[33] Wu, H.T., Dugelay, J.L. and Cheung, Y.M., 2008, May. A data mapping method for steganography and its application to images. In *International Workshop on Information Hiding* (pp. 236-250). Springer, Berlin, Heidelberg.

[34] Chuang, C.H., Cheng, C.W. and Yen, Z.Y., 2010. Reversible data hiding with affine invariance for 3D models. In *IET International Conference on Frontier Computing. Theory, Technologies and Applications*, Taichung, pp. 77-81.

[35] Jhou, C.Y., Pan, J.S. and Chou, D., 2007, November. Reversible data hiding base on histogram shift for 3D vertex. In *Third International Conference on Intelligent Information Hiding and Multimedia Signal Processing (IIH-MSP 2007)* (Vol. 1, pp. 365-370). IEEE.

[36] Ni, Z., Shi, Y.Q., Ansari, N. and Su, W., 2003, May. Reversible data hiding. In *Proceedings of the 2003 International Symposium on Circuits and Systems, 2003. ISCAS'03.* (Vol. 2, pp. 912-915). IEEE.

[37] Gruen, A. and Akca, D., 2005. Least squares 3D surface and curve matching. *ISPRS Journal of Photogrammetry and Remote Sensing*, *59*(3), pp.151-174.

[38] Ji, H., Yang, X., Zhang, C. and Gao, X., 2010, October. A new reversible watermarking of 3D models based on ratio expansion. In *2010 3rd International Congress on Image and Signal Processing* (Vol. 8, pp. 3899-3903). IEEE.

[39] Tian, J., 2003. Reversible data embedding using a difference expansion. *IEEE transactions on circuits and systems for video technology*, *13*(8), pp. 890-896.

[40] Thiyagarajan, P., Natarajan, V., Aghila, G., Venkatesan, V.P. and Anitha, R., 2013. Pattern based 3D image Steganography. *3D Research*, *4*(1), pp. 1-8.

[41] Huang, Y.H. and Tsai, Y.Y., 2015. A reversible data hiding scheme for 3D polygonal models based on histogram shifting with high embedding capacity. *3D Research*, *6*(2), p.20.

[42] Li, N., Hu, J., Sun, R., Wang, S. and Luo, Z., 2017. A high-capacity 3D steganography algorithm with adjustable distortion. *IEEE Access*, *5*, pp.24457-24466.

[43] Anish, K., Arpita, N., Nikhil, H., Sumant, K., Bhagya, S. and Desai, S.D., 2017. Intelligence system security based on 3-D image. In *Proceedings of the 5th International Conference on Frontiers in Intelligent Computing: Theory and Applications* (pp. 159-167). Springer, Singapore.

[44] Zhang, Q., Song, X., Wen, T. and Fu, C., 2018. Reversible data hiding for 3D mesh models with hybrid prediction and multilayer strategy. *Multimedia Tools and Applications*, pp.1-17.

[45] Thodi, D.M. and Rodríguez, J.J., 2007. Expansion embedding techniques for reversible watermarking. *IEEE transactions on image processing*, *16*(3), pp.721-730.

[46] Weinberger, M.J., Seroussi, G. and Sapiro, G., 1996, March. LOCO-I: A low complexity, context-based, lossless image compression algorithm. In *Proceedings of Data Compression Conference-DCC'96* (pp. 140-149). IEEE.

[47] Mao, X., Shiba, M. and Imamiya, A., 2001, August. Watermarking 3D geometric models through triangle subdivision. In *Security and Watermarking of Multimedia Contents III* (Vol. 4314, pp. 253-261). International Society for Optics and Photonics.

[48] 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), pp.400-412.

[49] Schroeder, W.J., Zarge, J.A. and Lorensen, W.E., 1992, July. Decimation of triangle meshes. In *Siggraph* (Vol. 92, No. 26, pp. 65-70).

[50] Tsai, Y.Y., 2016. A distortion-Free data hiding scheme for triangular meshes based on recursive subdivision. *Advances in Multimedia*, pp. 1-10.

[51] Arkin, E.M., Held, M., Mitchell, J.S. and Skiena, S.S., 1996. Hamiltonian triangulations for fast rendering. *The Visual Computer*, *12*(9), pp.429-444.

[52] Chow, M.M., 1997. *Optimized geometry compression for real-time rendering* (pp. 347-354). IEEE.

[53] Bogomjakov, A., Gotsman, C. and Isenburg, M., 2008, April. Distortion free steganography for polygonal meshes. In *Computer graphics forum* (Vol. 27, No. 2, pp. 637-642). Oxford, UK: Blackwell Publishing Ltd.

[54] Andrecut, M., 1998. Logistic map as a random number generator. *International Journal of Modern Physics B*, *12*(09), pp.921-930.

[55] Cignoni, P., Rocchini, C. and Scopigno, R., 1998, June. Metro: measuring error on simplified surfaces. In *Computer Graphics Forum* (Vol. 17, No. 2, pp. 167-174). Oxford, UK and Boston, USA: Blackwell Publishers. For Meshlab.

[56] Arnol'd, V.I. and Avez, A., 1968. Ergodic problems of classical mechanics. *Mathematical Physics Monograph Series*. New York:W.A. Benjamin.50(7-9) pp.506.

[57] Mukherjee, D.P., Maitra, S. and Acton, S.T., 2004. Spatial domain digital watermarking of multimedia objects for buyer authentication. *IEEE Transactions on multimedia*, *6*(1), pp.1-15.

[58] Qi, D., Zou, J. and Han, X., 2000. A new class of scrambling transformation and its application in the image information covering. *Science in China Series E: Technological Sciences*, *43*(3), pp.304-312.

[59] Kumar, V. and Kumar, D., 2017. Performance evaluation of modified color image steganography using discrete wavelet transform. *Journal of Intelligent Systems*, pp.1-10.

[60] Hassanien, A.E., 2006. A copyright protection using watermarking algorithm. *Informatica*, *17*(2), pp.187-198.

[61] Ping, P., Mao, Y., Lv, X., Xu, F. and Xu, G., 2015, August. An image scrambling algorithm using discrete Henon map. In *2015 IEEE International Conference on Information and Automation* (pp. 429-432). IEEE.

[62] Jolfaei, A., Wu, X.W. and Muthukkumarasamy, V., 2016. On the security of permutation-only image encryption schemes. *IEEE transactions on information forensics and security*, *11*(2), pp.235-246.

[63] Verma, A. and Tapaswi, S., 2009, May. A novel reversible visible watermarking technique for images using Noise Sensitive Region Based Watermark Embedding (NSRBWE) approach. In *IEEE EUROCON 2009* (pp. 1374-1377). IEEE.

[64] Liu, L., Zhang, Q. and Wei, X., 2012. A RGB image encryption algorithm based on DNA encoding and chaos map. *Computers & Electrical Engineering*, *38*(5), pp.1240-1248.

[65] Enayatifar, R., Abdullah, A.H. and Isnin, I.F., 2014. Chaos-based image encryption using a hybrid genetic algorithm and a DNA sequence. *Optics and Lasers in Engineering*, *56*, pp.83-93.

[66] Wei, X., Guo, L., Zhang, Q., Zhang, J. and Lian, S., 2012. A novel color image encryption algorithm based on DNA sequence operation and hyper-chaotic system. *Journal of Systems and Software*, *85*(2), pp.290-299.

[67] Liu, H. and Wang, X., 2012. Image encryption using DNA complementary rule and chaotic maps. *Applied Soft Computing*, *12*(5), pp.1457-1466.

[68] Saranya, M.R., Mohan, A.K. and Anusudha, K., 2015, February. Algorithm for enhanced image security using DNA and genetic algorithm. In *2015 IEEE International Conference on Signal Processing, Informatics, Communication and Energy Systems (SPICES)* (pp. 1-5). IEEE.

[69] Gupta, S. and Jain, A., 2015, March. Efficient image encryption algorithm using DNA approach. In *2015 2nd International Conference on Computing for Sustainable Global Development (INDIACom)* (pp. 726-731). IEEE.

[70] Saranya, M.R., Mohan, A.K. and Anusudha, K., 2014, November. A composite image cipher using DNA sequence and genetic algorithm. In *2014 International Conference on Contemporary Computing and Informatics (IC3I)* (pp. 1022-1026). IEEE.

[71] Kadhim, F.A., Majeed, G.H.A. and Ali, R.S., 2016, May. Proposal new s-box depending on DNA computing and mathematical operations. In *2016 Al-Sadeq International Conference on Multidisciplinary in IT and Communication Science and Applications (AIC-MITCSA)* (pp. 1-6). IEEE.

[72] Srividhya, N. and Vino, T., 2016, March. Genome based highly secured image using DNA cryptography and trellis algorithm. In *2016 International Conference on Wireless Communications, Signal Processing and Networking (WiSPNET)* (pp. 1658-1662). IEEE.

[73] Gupta, R. and Jain, A., 2014. A new image encryption algorithm based on DNA approach. *International journal of computer applications*, *85*(18), pp. 27-31.

[74] Watson, J.D. and Crick, F.H., 1953. Molecular structure of nucleic acids. *Nature*, *171*(4356), pp.737-738.

[75] Mills Jr, A.P., Yurke, B. and Platzman, P.M., 1999. Article for analog vector algebra computation. *Biosystems*, *52*(1-3), pp.175-180.

[76] Gao, T., Chen, Z., Yuan, Z. and Chen, G., 2006. A hyperchaos generated from Chen's system. *International Journal of Modern Physics C*, *17*(04), pp.471-478.

[77] Gaborit, P. and King, O.D., 2005. Linear constructions for DNA codes. *Theoretical Computer Science*, *334*(1-3), pp.99-113.

[78] Wu, Y., Noonan, J.P. and Agaian, S., 2011. NPCR and UACI randomness tests for image encryption. *Cyber journals: multidisciplinary journals in science and*

technology, Journal of Selected Areas in Telecommunications (JSAT), *1*(2), pp.31-38.

[79] Zhang, Q. and Wei, X., 2013. RGB color image encryption method based on Lorenz chaotic system and DNA computation. *IETE Technical Review*, *30*(5), pp.404-409.

[80] Wang, X., Teng, L. and Qin, X., 2012. A novel colour image encryption algorithm based on chaos. *Signal Processing*, *92*(4), pp.1101-1108.

[81] Murillo-Escobar, M.A., Cruz-Hernández, C., Abundiz-Pérez, F., López-Gutiérrez, R.M. and Del Campo, O.A., 2015. A RGB image encryption algorithm based on total plain image characteristics and chaos. *Signal Processing*, *109*, pp.119-131.

[82] Mishra, D.C., Sharma, R.K., Kumar, M. and Kumar, K., 2014. Security of color image data designed by public-key cryptosystem associated with 2D-DWT. *Fractals*, *22*(04), p.1450011.

[83] Kumar, M., Powduri, P. and Reddy, A., 2015. An RGB image encryption using diffusion process associated with chaotic map. *Journal of Information Security and Applications*, *21*, pp.20-30.

[84] Kumar, M., Iqbal, A. and Kumar, P., 2016. A new RGB image encryption algorithm based on DNA encoding and elliptic curve Diffie–Hellman cryptography. *Signal Processing*, *125*, pp.187-202.

[85] Tiwari, A.K., Rajpoot, A., Shukla, K.K. and Karthikeyan, S., 2015. A Robust Method for Image Steganography based on Chaos Theory. *International Journal of Computer Applications*, *113*(4), pp. 35-41.

[86] Niyat, A.Y., HeiHei, R. and Vafaei Jahan, M., 2015. A RGB image encryption algorithm based on DNA sequence operation and hyper-chaotic system. In *International Congress on Technology, Communication and Knowledge (ICTCK)*.

[87] Niu, Y., Zhang, X. and Han, F., 2017. Image encryption algorithm based on hyperchaotic maps and nucleotide sequences database. *Computational intelligence and neuroscience*, *2017*, pp. 1-9.

[88] Alsafasfeh, Q.H. and Al-Arni, M.S., 2011. A new chaotic behavior from Lorenz and Rossler systems and its electronic circuit implementation. *Circuits and Systems*, *2*(02), p.101.

[89] Ni, Z., Kang, X. and Wang, L., 2016, August. A novel image encryption algorithm based on bit-level improved Arnold transform and hyper chaotic map. In *2016 IEEE International Conference on Signal and Image Processing (ICSIP)* (pp. 156-160). IEEE.

[90] IS Committee, 2008. 754–2008 ieee standard for floating-point arithmetic. *IEEE Computer Society Std, 2008*.

[91] Pareek, N.K., Patidar, V. and Sud, K.K., 2006. Image encryption using chaotic logistic map. *Image and vision computing*, *24*(9), pp.926-934.

[92] Zhou, Y., Bao, L. and Chen, C.P., 2014. A new 1D chaotic system for image encryption. *Signal processing*, *97*, pp.172-182.

[93] Xu, L., Gou, X., Li, Z. and Li, J., 2017. A novel chaotic image encryption algorithm using block scrambling and dynamic index based diffusion. *Optics and Lasers in Engineering*, *91*, pp.41-52.

[94] Guan, Z.H., Huang, F. and Guan, W., 2005. Chaos-based image encryption algorithm. *Physics Letters A*, *346*(1-3), pp.153-157.

[95] Cao, Y.Y. and Fu, C., 2008, October. An image encryption scheme based on high dimension chaos system. In *2008 International Conference on Intelligent Computation Technology and Automation (ICICTA)* (Vol. 2, pp. 104-108). IEEE.

[96] Xiang, T., Wong, K.W. and Liao, X., 2007. Selective image encryption using a spatiotemporal chaotic system. *Chaos: An Interdisciplinary Journal of Nonlinear Science*, *17*(2), p.023115.

[97] Wang, Y., Wong, K.W., Liao, X. and Chen, G., 2011. A new chaos-based fast image encryption algorithm. *Applied soft computing*, *11*(1), pp.514-522.

[98] SIPI Image Database – Misc, *Sipi.usc.edu*, 2017. [Online]. Available: http://sipi.usc.edu/ database/ database.php?volume=misc.

[99] Lorenz, E.N., 1963. Deterministic nonperiodic flow. *Journal of the atmospheric sciences*, *20*(2), pp.130-141.

[100] Rössler, O.E., 1976. An equation for continuous chaos. *Physics Letters A*, *57*(5), pp.397-398.

[101] Zhang, Y. and Xiao, D., 2014. An image encryption scheme based on rotation matrix bit-level permutation and block diffusion. *Communications in Nonlinear Science and Numerical Simulation*, *19*(1), pp.74-82.

[102] Bashir, Z., Rashid, T. and Zafar, S., 2016. Hyperchaotic dynamical system based image encryption scheme with time-varying delays. *Pacific Science Review A: Natural Science and Engineering*, *18*(3), pp.254-260.

[103] Kaelo, P. and Ali, M.M., 2006. A numerical study of some modified differential evolution algorithms. *European journal of operational research*, *169*(3), pp.1176-1184.

[104] Chai, X., Zheng, X., Gan, Z., Han, D. and Chen, Y., 2018. An image encryption algorithm based on chaotic system and compressive sensing. *Signal Processing*, *148*, pp.124-144.

[105] Lan, R., He, J., Wang, S., Gu, T. and Luo, X., 2018. Integrated chaotic systems for image encryption. *Signal Processing*, *147*, pp.133-145.

[106] Matthews, R., 1989. On the derivation of a "chaotic" encryption algorithm. *Cryptologia*,*13*(1), pp.29-42.

[107] Safi, H.W. and Maghari, A.Y., 2017, October. Image encryption using double chaotic logistic map. In *2017 International Conference on Promising Electronic Technologies (ICPET)*(pp. 66-70). IEEE.

[108] Shi, Y.Q., Li, X., Zhang, X., Wu, H.T. and Ma, B., 2016. Reversible data hiding: advances in the past two decades. *IEEE Access*, *4*, pp.3210-3237.

[109] Sun, J., Liao, X., Chen, X. and Guo, S., 2017. Privacy-aware image encryption based on logistic map and data hiding. *International Journal of Bifurcation and Chaos*, *27*(05), p.1750073.

[110] Wang, X., Zhao, Y., Zhang, H. and Guo, K., 2016. A novel color image encryption scheme using alternate chaotic mapping structure. *Optics and Lasers in Engineering*, *82*, pp.79-86.

[111] Soliman, M.M., Hassanien, A.E. and Onsi, H.M., 2016. An adaptive watermarking approach based on weighted quantum particle swarm optimization. *Neural Computing and Applications*, *27*(2), pp.469-481.

[112] "Special Agent Ricci against alleged Russian agents", 2010. [Online]. Available:https://www.justice.gov/sites/default/files/opa/legacy/2010/06/28/062810complaint2.pdf. [Accessed: 10- Jul- 2017].

[113] "3D human lung model to shed light on respiratory diseases", *www.financialexpress.com*, 2017. [Online]. Available:

http://www.financialexpress.com/lifestyle/science/3d-human-lung-model-to-shed-light-on-respiratory-diseases/667123/]. [Accessed: 10- Jul- 2017].

[114] Chin-Wei, B. and Rajeswari, M., 2010. Multi objective optimization approaches in image segmentation–the directions and challenges. *Int. J. Advance. Soft Comput. Appl*, *2*(1), pp.40-64.

[115] "10 Types of 3D Graphics Software Worth Knowing, Animation Career Review", animation carieerview.com, 2017. [Online] Available: http://www.animationcareer-review.com/ articles/10-types-3d-graphics-software-worth-knowing.

[116] Luo, M., 2006. *Robust and blind 3D watermarking* (Doctoral dissertation, University of York).