

INVESTIGATION OF AMBIENT GRADIENTS USING CNN BASED IMAGE STEGANOGRAPHY

Submitted in partial fulfilment of requirements for the award of
Bachelor of Engineering in
Electronics and Communication Engineering

by

THOTA JYOTHSNA (37130412)

VUMMIDICHETY THEJASWINI (37130447)



**DEPARTMENT OF ELECTRONICS AND COMMUNICATION ENGINEERING
SCHOOL OF ELECTRICAL AND ELECTRONICS**

SATHYABAMA

**INSTITUTE OF SCIENCE AND TECHNOLOGY
(DEEMED TO BE UNIVERSITY)**

**Accredited with Grade "A" by NAAC
JEPPIAAR NAGAR, RAJIV GANDHI SALAI, CHENNAI – 600119**

APRIL – 2021



SATHYABAMA

INSTITUTE OF SCIENCE AND TECHNOLOGY

(DEEMED TO BE UNIVERSITY)

Accredited "A" Grade by NAAC | 12B Status by UGC | Approved by AICTE

www.sathyabama.ac.in

DEPARTMENT OF ELECTRONICS AND COMMUNICATION ENGINEERING

BONAFIDE CERTIFICATE

This is to certify that this Project Report is the bonafide work of THOTA JYOTHSNA (Reg.No.37130412) and VUMMIDICHETY THEJASWINI (Reg.No.37130447) who carried out the project entitled "**INVESTIGATION OF AMBIENT GRADIENTS USING CNN BASED IMAGE STEGANORAPHY**" under our supervision from December 2020 to March 2021.

Internal Guide

Dr.A.SAHAYA ANSELIN NISHA M.E., Ph.D.

Head of the Department

Dr.T.RAVI M.E., Ph.D.

Submitted for Viva voce Examination held on _____

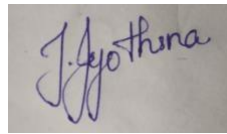
Internal Examiner

External Examiner

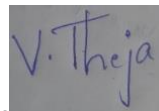
DECLARATION

We, THOTA JYOTHSNA (Reg.No.37130412) and VUMMIDICHETY THEJASWINI (Reg.No.37130447) hereby declare that the Project Report entitled **“INVESTIGATION OF AMBIENT GRADIENTS USING CNN BASED IMAGE STEGANOGRAPHY”** done by us under the guidance of Dr.A.SAHAYA ANSELIN NISHA M.E., Ph.D. is submitted in partial fulfillment of the requirements for the award of Bachelor of Engineering degree in Electronics and Communication Engineering.

1.



2.



DATE:

PLACE:CHENNAI

SIGNATURE OF THE CANDIDATES

ACKNOWLEDGEMENT

We are pleased to acknowledge our sincere thanks to Board of Management of **SATHYABAMA** for their kind encouragement in doing this project and for completing it successfully. We are grateful to them.

We convey our thanks to **Dr.N.M.NANDHITHA M.E., Ph.D.** Dean, School of Electronics and Communication Engineering and **Dr.T.RAVI M.E., Ph.D.** Head of the Department, Department of Electronics and Communication Engineering for providing me necessary support and details at the right time during the progressive reviews.

We would like to express our sincere and deep sense of gratitude to our Project Guide **Dr. A. SAHAYA ANSELIN NISHA M.E., Ph.D.** for her valuable guidance, suggestions and constant encouragement paved way for the successful completion of our project work.

We wish to express our thanks to all Teaching and Non-teaching staff members of the Department of **Electronics and Communication Engineering** who were helpful in many ways for the completion of the project.

We would like to thank our beloved **PARENTS** for their constant love and support, without whom this wouldn't be possible and finally we would like to thank the **ALMIGHTY** for his wonderful blessing.

ABSTRACT

Data connectivity has been facilitated by the development of high-speed laptop networks and Web pages. Unlike analog and digital media, it has a wide range of benefits, including high content, fast editing, repeating loyalty, and prestige. However, this sort of development has doubled the fear that information is being dissimulated, causing data from sender to recipient within the field of information of knowledge of information. Because of this aspect, data protection is the most serious information downside. Steganography is important in the field of data protection. Video and photographs are a common replacement for activity awareness. For an efficient and eminent embedding operation, it is critical to pick suitable pixels in the video frames that are used to store the key details.

We use video Steganography due to the large scale and memory limitations. When concealing information in a carrier disk, we use the least significant bit (LSB) insertion technique. We change the LSB of a video file with the information bits in the least significant bit (LSB) technique to hide information. This project focuses on information concealment by LSB substitution in specific photos of the film and in a specific location of the camera. The core abstract of the framework is to secure data transmission using LSB technology. The project focuses on various steganography approaches. Image steganography is performed on images, and related data is decrypted for message image retrieval. Since image steganography can be done in a variety of ways, it is studied and one of the techniques used to demonstrate it.

The PowerPoint Steganography file's name refers to a new kind of steganography solution.' The sound/custom animated effects in the PowerPoint file are used to hide data. This technique would be enforced by a codebook containing confidential sound details and information about the text that must be transferred.

To improve confidentiality, a non-traditional approach known as steganography is used to communicate data while maintaining anonymity. This is a big problem in every area. "Steganography" is a "material concealment" technique. Stego is a general term for "covering," and "books" is a term for "reading." Steganography is described by the combination of these two terms, which means "secret prose." It's similar to cryptography in that cryptography uses data scrambling techniques, while steganography stores information in a carrier file of some kind.

TABLE OF CONTENTS

Chapter No.	Title	Page No.
	ABSTRACT	V
	LIST OF FIGURE	VII
	LIST OF TABLES	IX
	LIST OF GRAPHS	X
1	INTRODUCTION	1
	1.1 AIM OF THE PROJECT	2
2	LITERATURE SERVEY	3
	2.1 SUMMARY	6
3	INVESTIGATION	
	3.1 EXISTING POINT	6
	3.2 PROPOSED POINT	7
	3.3 BLOCK DIAGRAM	8
	3.4 STEGANOGRAPHY	
	3.5 STEGANOGRAPHY MEDIA	9
	3.6 PRACTICAL APPOACH	10
	3.7 ADVANCE VIDEO CODING	10
	3.8 COLOUR SPACE	10
	3.9 SAMPLING PATERNS	11
	3.10 PREDITION	12
	3.11TRANSFORM	13
	QUANTIZATION	
4	STENOGRAPHY SECURITY	
	4.1 STEGANOGRAPHY CHANNELS	15
	4.2 EMBEDDING METHOD	16
	4.3 +1 EMBEDDING	17
	4.4 F5-ENBEDDING	18
	4.5 PRACTICAL EMBEDDING	18
	4.6 MATRIX EMBEDDING	18
	4.7 NON-SHARED SELECTION	19

CHAPTER NO	TITLE	PAGE NO
	4.8 PROPOSED FEATURES	19
	4.9 PROPOSED DISTORSION METHOD	21
	4.9.1KULLBACK-LCIBLER DRIVERGENCE	21
	4.9.2MAXIMUM MEAN DISCREPANCY	22
	4.10IMPLEMENTATION	22
	4.10.1STEGOSAURUSGUI	23
	4.10.2THE XML BACKEND	24
	4.10.3THE .FV FILE FORMAT	25

CHAPTER NO	TITLE	PAGE NO.
	4.11 NORMALIZATION	26
	4.11.1 RESCALING	26
	4.11.2 STANDARDIZING	26
	4.12 THE VIDEO COLLECTION	26
	4.13H.264.SYNTAX	27
	4.14FEACTURE EXTRACTION	28
5	SOFTWARE DISCRPTION	
	5.1 EXTRACT.SH	28
	5.2 EMBEDDING SIMULATION	31
	5.3 UNTUN.SOFTWARE	33
	5.4 MMD	34
	5.5 EMBEDDING STIMULATION	34
	5.6 MMD ESTIMATION	35
6	EXPERIMENTAL RESULTS	
	6.1 VISUAL DISTORTION	36
	6.2 FUTURE ANALYSIS	36
	6.3 EMBEDDED DETECT	40
	6.4 F5 EMBEDDING	42
	6.5 COEFFICIENT ARE BEST	42
	6.6 LIMITATIONS	44
7	CONCLUSION	45
	7.1 FUTURE WORK	46
	7.2 PERSONAL REPORT	46
8	REFERENCES	47

LIST OF FIGURES

FIGURE .No	TITLE	PAGE. NO
3.1	BLOCK DIAGRAM OF STEGANOGRAPHY	8
3.2	VIDEO CODING FLOW WORK	10
3.3	ORIGINAL FRAME AND RGV COMPONENTS	11
3.4	ORIGINAL FRAME AND RGV	15
3.5	INTRA PREDICTION MODES	12
3.6	EXAMPLE X264 OUTPUT STREAM	13
3.7	LUMA DEFAULT TRANSFORM	13
3.8	LUMA INTRA 16X16 TRANSFORM	14
3.9	CROMA TRANSFORM	14
3.10	ZIG-ZAG SCAN ORDER	15
4.1	STEGANOGRAPIC CHANNEL	16
4.2	LSB EMBEDDING	17
4.3	± 1 EMBEDDING	18
4.4	IMPACT ON COEFFICIENT HISTOGRAM	18
4.5	1 EMBEDDING WITH THRESHOLD 1	18
4.6	EMBEDDING ON COEFFICIENT HISTOGRAM	19
4.7	GUIMENUS	24
4.8	SNAPSHOT OF FUTURE DIALOG	26
4.9	SYNTAX OF H.264BITSTREAM	29
5.1	ELESTRATE THE COMPUTATION OF MMD ESTIMTE	37
6.1	UNMODIFIED FOREMAN SNAPSHOTS	38
6.2	EMBEDDING IN LUMA AND CROMA	39
6.3	EMBEDDING IN LUMA	39
6.4	EMBEDDING IN CROMA	39
6.5	F5 EMBEDDING IN LUMA AND CROMA	40
6.6	F5 EMBEDDING IN LUMA	40
6.7	F5 EMBEDDING IN CROMA	40
6.8	LSB EMBEDDING IN LUMA AND CROMA	41

LIST OF TABLES

TABLE.NO	TITLE	PAGE.NO
4.1	METHOD CODES	24
5.1	UNUSED COEFFICIENT OF EMBEDDING SYSTEM	34
6.1	MMD GROWTH RATES	46

LIST OF GRAPHS

GRAPH NO	TITLE	PAGE NO
6.1	BPNC->MMD FOR 1 EMBEDDING IN AC AND DC USING THRESHOLD 1 (P SCLICES)	42
6.2	BPNC->MMD FOR $\pm$ EMBEDDING IN AC AND DC USING THRESHOLD 1 (B SCLIECS)	43
6.3	BPNC->MMD FOR $\pm$ EMBEDDING IN AC AND DC USING THRESHOLD 1 (P SCLINES)	43

CHAPTER - 1

INTRODUCTION

Now every day is that the uses of devices like pc, mobile and plenty of additional alternative device for communication likewise as for information storage and transmission has will increase. As a result there's increase in no of user's conjointly there's increase in no of unauthorized user's that are attempting to access an information by unfair suggests that. This arises the matter of knowledge security. to unravel this downside an information is keep or transmitted within the encrypted format. This encrypted information is illegible to the un-authorized user. Cryptography may be a science of data security, that secures the information whereas the information is being transmit-ted and keep. each coding and coding method has 2 aspects: the algorithmic rule and also the key use for the coding and coding. However, it's the key used for coding and coding that produces the method of cryptography secure. There square measure 2 styles of cryptological mechanisms: stellate key cryptography within which constant key's use for coding and coding. just in case of uneven key cryptography, 2 completely different keys used for coding and coding. stellate key algorithmic rule is way quicker and easier to implement and requiered less process power as compare to uneven key algorithmic rule.

The advance coding commonplace (AES) specifies a federal IP standards publication (FIPS) approved cryptological algorithmic rule that may be wont to defend electronic information. it had been publish by National Institute of ordinary and Technology (NIST) in 2001 developed by Joan Daemen and Vincent Rijmen, Associate in Nursing algorithmic rule referred to as Rijdael, Priya. Deshmukh is M.E student of faculty member. Ram Meghe Institute of Technology and analysis, Badnera due to the drawbacks within the 3DES like the algorithmic rule is comparatively slow in computer code and it uses 64-bit block size that is a smaller amount for additional security massive block size is needed. thanks to this reason, AES supposed to exchange 3DES. AES has blessings over 3DES like high procedure potency, 128-bit block size, and cryptology resistance is robust against differential truncated differential, linear, interpolation and sq. attacks. the appliance of image process is especially found

in military communication, Forensic, Robotics, intelligent system, etc. during this paper, we tend to enforced the AES algorithmic rule on image with the assistance of MATLAB computer code

The idea behind image-based Steganography is extremely easy. pictures square measure composed of digital information (pixels), that describes what's within the image, sometimes the colours of all the pixels. Since we all know each image is created from pels and each pixel contains 3-values (red, green, blue). each computer memory unit of knowledge is reborn to its 8-bit computer code exploitation ASCII values. currently pixels square measure scan from left to right in an exceedingly cluster of three containing a complete of nine values. the primary 8-values square measure wont to store binary information. the worth is created odd if one happens and although zero happens Steganography is the art of hiding. In this art we hide some information in some other in formation. The fact that there is touch is a fact. There are several different types of carrier files that can be used, but digital images are the most common due to their popularity on the internet. To mask secret features in photos, a variety of steganographic methods are available. Some are complex, and they all have their high and weak points. Various implementations have different steganography technique requirements. For example, although some implementations require complete invisibility, others allow for the concealment of a larger hidden message. The study's aim was to provide a description of the use of steganography sand techniques.

The aim of this project was that enables a person to send data over network by hiding it into an image so that adversary could not intercept the message. Cryptography is used to encrypt secret information before it is embedded in the cover files. see Steganography is the art of concealing data in a channel, as the term suggests.

SCOPE OR PROJECT:

Steganography is the practice of concealing the fact the communication has occurred by enclosing information in other information. The main of this project is to conceal information from any image file. The project's goal is to provide steganographic information tools, such as all types of information files as well as the path to save the image and extrude files. The project's goal is to prevent unauthorized access and enhance message distribution security. To meet the requirements I use a simple and basic steganography process.

This project is designed to hide details from every picture file. The project aims to incorporate steganography hiding information resources which includes some kind of file and image file.

The aim of this project was that enables a person to send data over network by hiding it into an image so that adversary could not intercept the message. Cryptography is used to encrypt secret information before it is embedded in the cover files. see Steganography is the art of concealing data in a channel, as the

CHAPTER:2

Literature survey

1. Hybrid Approach to Text & Image Steganography victimization AES and LSB Technique Author: Vikas M, Yashwanth E, Veeresh, Sanath Krishna S, Narender .M To transmit secret messages or data from one location/source to a different program, we tend to use steganography and cryptography. In cryptography, the secret message's content is normally disorganized, while in Steganography, the secret message is encased in a highly opaque medium. We tend to propose a secure model based on a mixture of Advanced Commonplace cryptography and Most Significant Bit algorithms in this article. In this case, AES is used for cryptography, and the LSB technique is used for steganography. In a canopy, the intended system encrypts an image or text.

M Saritha, Sushravya M, Vishwanath M Kadabadi, The text is encoded using an isobilateral victimization scheme's XOR algorithm and a sequent software algorithm called to hide|to cowl, which uses information from the cover image. It includes the following details: "MATLAB Cryptography Image and Text Steganography: An Introduction" is the title of a paper published by MATLAB. In terms of the amount of security given, it is inadequate. Furthermore, the gadget mentioned is prohibited by calibers and a few forms of images.

Aman Arora, This article describes a scheme for modifying the area unit bits based on the original bit set. If the image has the same component length around a district, the bit variations will create differences in the full pixels, and therefore the correction to the color values will be incredibly easy for human eyes.

Utsav Seth and Shiva Saxena, In comparison to other articles on the topic of AES victimization, this method provides a genuine degree of continuity. When Steganography is used as the final shot, the picture quality is low and there is a lot of noise compared to other documents. As the product of the least significant nibble, this can be chosen over a couple of least significant bits. The true truth that is hidden in the photographs will then be discovered by human eyes.

Kamal deep Joshi and Rajkumar Yadav, 'A new approach to learning about different forms of Steganography imaging techniques and stallion techniques for finding coded

messages in images in the space and reworking domains.' Using XOR's for image steganography: This method presupposes that the sender and receiver settle on a canopy image that will be a present on both ends. This method, however, is difficult to implement since it will be difficult to preserve the same image with the same size, resolution, and other features.

[2]. Otori and Kuriyama pioneered the work of mixing experience writing with pixel-based texture synths. Secret messages to be concealed squarely in a colored dotted pattern and directly on a blank frame. With the aid of a pixel-based algorithmic rule, the remaining pixels mistreat the technique of pixel texture while concealing the existence of spotted patterns. Until applying the data detection function, the display of the stego texture image synthesized is photographed for the purpose of extracting messages.

[4]. Liang et al. developed a patch-based sampling methodology and applied it. By rising the exponential growth and secret touch of potential laptop users over the internet, steganography is becoming more important in the digital industry. It can also be described as the analysis of secret invisible communication, which focuses on the various ways in which the transmitted message is hidden. Material incorporation is obtained in general for copyright purposes in communication such as images, audio, voice, or multimedia, as well as in military communication for authentication and a variety of other purposes. A message is embedded in a canopy image that is used because of the media to insert m. the most hidden touch in the Steganography photographs.

[1]. Since the output is an object, pixel-based algorithms synthesize the component image and use neighborhood abstract comparisons to fix the most associated component during the sample texture. That the already synthesized pixels incorrectly represent each output part. The sum of the patterns determines the potential of the Otori and Kuriyama techniques. Their technique, on the other hand, had a very low message extraction quality.

[3]. Patches are based on patches, and there is no part to synthesize textures; instead, patches are pasted from a supply texture. This Cohen et al. and Xu et al. solution improves the image transparency of artificial pixel textures by preserving the textures inside the patches square. Since patches are plated with a small, overlapping field, the rotation method for adjacent patches' overlapped regions is still possible in the

artificial process. Efros and citizens[17] have provided the gift of a patch-handi crafting technique known as 'picture quilting.' The algorithmic rule searches for a new patch to be synthesized and sewed first.

Message in the image to generate a stego image that could contain a hidden coded message. We tend to propose a special method for covering hidden data in this document, in addition to AES and LSB. The customers' choice of image consistency assessment aided in the determination of key messages square. As a result, the user has complete freedom to choose any size output permitted by the source texture and one patch that meets the pre-defined tolerance to neighbors in the overlapping region. Then, using dynamic programming methods, the minimum error path through the overlapping field is shown. This declares an optimal cap between the candidate chosen and the syntactic patch, resulting in visually plausible patch stitching.

[5]. Ni et al. proposed a reversible image data hiding Algorithm that, after retrieving the secret data, could recover a cover image while preserving the stego image. Histogram shifting is a popular technique among reversible image data hiding approaches because it can track pixel change, eliminating embedding distortion, and it only requires a small position map, decreasing the amount of overhead discovered. general framework is the latest state of the art for concealing reversible image data.

zahraa khudhair taha et al To ensure secure internet transmission, high safety and confidentiality data sharing is required; information security is a major issue in network and Internet data exchange. Our main goal in this paper is to improve the current secure communication mechanism, perhaps by integrating cryptography and steganography. Cryptography and steganography are used to conceal information transmission. It is used in this article to adapt the Advanced Encryption Standards (AES) algorithm to incoherent information (figure contents), after which the contents of the images are hidden by the frame's smallest significant bit The content is key encoded in 128 bits. In fact, the scrambled content is often used for a variety of image arrangement types and sizes. The tests demonstrate that the spread image influences the nature of the stego image; hence, PSNR increases with MSE and decreases with the vast spread image. The findings reveal that the proposed approach is strong and adds protection thresholds for information sharing.

If more data is exchanged over the internet, safer communications are expected. The large number of clients necessitates security, particularly because it is accessible

globally with numerous PC, network, and Internet usages. To protect customer data, a variety of techniques are used. Encryption, the method of converting readable data from state to obvious babble, is one such tactic. During the encryption method, confidential data was encrypted into an encryption form, which proved quite challenging (even unauthorized to recognize or read the information). The second technique is steganography, in which data is concealed within some visual or auditory media, such as a picture, sound, or video. A secure host for data transfer is used in this article for cryptography and steganography. A bit key AES-128 is used to encrypt the message, and the LSB is added to the image to be sent in the encrypted text.

This section discusses the Stable Data Transmission System. [2] Formal paraphrase [2] Formal paraphrase Sharma et al. proposed a scheme that combines cryptography and steganography to improve data security. The message byte is an arandom key XORing created by a pseudo-random generator that is then inserted into the plaintext in a picture format. Sharma and colleagues suggest the BLOWFISH Algorithm for encrypting secret images and the LSB process for hiding the encrypted image inside a frame.

[2]. Vijay and Swati used the Data Encryption Standard (DES) algorithm to encrypt the message and the MD5 algorithm to measure a message digest used to check the authenticity of a message. The picture file is then concealed. Singh and Attri suggested the LSB dual data protection layer, which uses the Advanced Encryption Standard Algorithm to encode and encrypt data. Sine and cosine Inconspicuous. The RGB layer has been converted (cover image).

2.1 SUMMARY:

This project aims to offer an overview of the use and techniques of image steganography. Steganography is also provided in audio files. For a more secure solution, the project encrypts the letter with a hidden key before transferring it to the receiver. In order to receive the original, the receiver decrypts the letter. This project is intended to conceal information in each image file. The project's goal is to develop steganography hiding techniques.

CHAPTER-3

INVESTIGATION

EXISTING SYSYTEM:

The scheme extracts binary bit planes from a basic image and manages bit level output and ambiguity using a pseudo-random image given by the Logistic Map. With the four LSBPs permuting rows and columns with the same pseudo-random series and the plain-image encryption method's statistical features disabled, In this paper to develop a tool that can encrypt/decrypt text using bit plane extraction

3.1 PROPOSED POINT:

The proposed method for hiding the message in the video image is based on video steganography, which retrieves the hidden message from the video using the LSB technique (least significant bit). LSB steganography techniques are widely used in steganographic photographs to investigate the conditions under which viewers can move from stego images to cover images. The first is the letter image carrier, and the second is the stego photo with the hidden message. The other picture is a video of Stego. It is impossible to tell the difference between the original and Stego video clips. The data is concealed in the video file using the LSB algorithm.

The disadvantage of the LSB coding technique is that it requires a low level of computer complexity and a high channel bit rate. This method modifies small bits of the message bits of the carrier files' single pixels. Every pixel has three hidden bits, and each RGB component has one hidden message. To mask three bits of message in each pixel color, we use 24-bit images such as BMP (Bitmap). The human eye cannot tell the difference between 21-bit and 24-bit images.

Steganography, a modern technique, was developed, carried out, and tested. The proposed method hides the coded message by examining the same bits between the secret messages and picture pixel values. The proposed solution provides better security and would protect the message from stego attacks.

3.1 Block diagram:

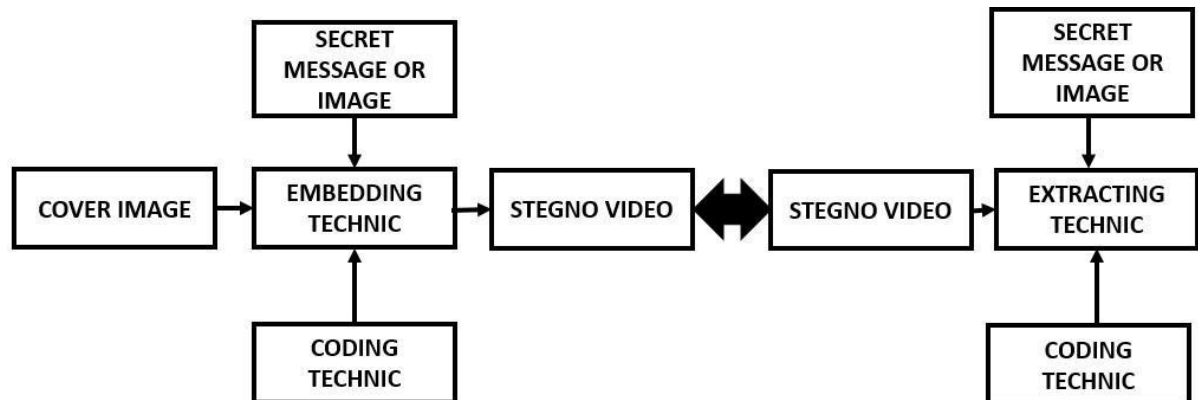


Figure 3.1: Block diagram of steganography

3.2 STEGANOGRAPHY VS CRYPTOGRAPHY AND WATERMARKING

Steganography is the process of hiding data in cover objects, the name comes from the Greek words “steganos” which means “covered” and “graphia” which means “writing” [1]. The aim of steganography is to hide the fact that communication happened at all which is different from the aim of cryptography. If communication happens encrypted, the fact of communication is not hidden.

Water marking uses similar techniques as steganography but with a different intention, a water marked video could contain a hidden key which identifies the special copy of video file. This would allow us to trace copies of the video. However, in watermarking, the absence of knowledge that a watermark is being used is not crucial, more importantly; the watermark shall be robust against attacks like reencoding and video filters. A steganographic system is broken as soon as the fact that communication took place is revealed, no matter if the message s are deciphered

successfully extracted 141 documents containing detailed plans for future attack embedded into porno graphic video found on an suspected al-Qaeda operative who was arrested in May 2011 in Berlin²³. It is important to investigate if video can be used for steganography and how steganography in videos can be detected.

3.3 STEGANOGRAPHIC MEDIA

In this project we try to assist Alice and Bob who are imprisoned and trying to escape.

They can exchange videos but all their communication is controlled by warden Eve-

who observes their communication.

There are three different approach to digital steganography for Alice and Bob to use

[1]: **Steganography by cover.** Selection There is a fixed database of cover objects each assigned with a secret meaning.

Alice needs to pick the correct cover to transmit in order to convey a message.

synthesis Alice creates a cover with special properties to transmit a hidden meaning. This can be colors of clothes in a picture or the length of a text rather than the content.

Modification Given a cover object Alice modifies it slightly so that the change is not at all or at least hardly noticeable. For example, Alice could apply different video filters until the modified video contains the desired message or Alice can change.

The syntax elements inside the video files lightly so that Bob can extract a message from them.

Steganography by covers election is not interesting to Alice and Bob, they would both have to synchronize a large collection of video files and agree on a meaning of each individual video. Cover synthesis is hard to do on video since it demand shoot the same video multiple times. Therefore, Alice and Bob will use cover modification. In this project, we will discuss

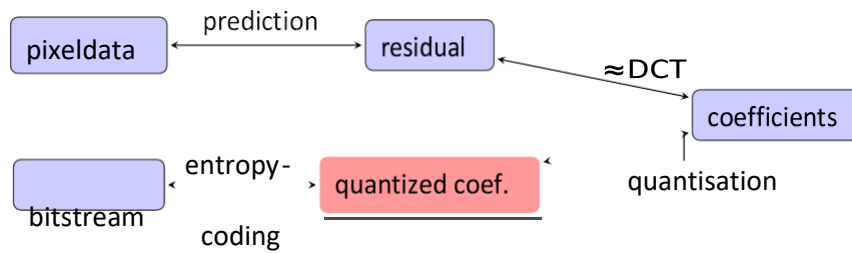


Figure 3.2 Video coding flow work

Figure3.2: Video coding work flow. We modify only the quantized coefficients in the compressed video stream when simulating embedding.

When encoding a frame using h.264 first a prediction is made, this can either be intra- or an inter prediction. Intra prediction uses data from the same frame that already has been encoded and inter prediction uses past and/or future frame(s). The prediction is subtracted from the pixel data, the residual gets transformed using an integer approximation to the Discrete Cosine Transform (DCT) and quantized. The quantized coefficients are then entropy-coded into the final bit stream.

3.4 PRACTICAL APPROACH

In this project we will simulate steganography by modification of quantized coefficients in an h.264 video stream and compare different approaches with respect to their detectability. The embeddings simulation will happen on a set of dvd that are transcoded to h.264 with the x264 encoder and using ffmpeg to read the resulting video bit streams.

A set of features will be proposed to represent a section of video as a high dimensional feature vector. The detectability of different embedding schemes with different parameters will be measured as distortion in these features using the Kullback-Leibler Divergence (KL-D) and Maximum Mean Discrepancy (MMD). The distortion measures are described in detail in section.

3.5 THE H.264 ADVANCED VIDEO CODING(AVC) STANDARD

H.264 is a block-based video codec, the pipeline mentioned in section 2.3 is traversed for each 16x16 pixel macroblock in a video frame. Within each



Figure 3.3: Original frame and RGB components

YCRCB

The human eye is more sensitive to luminance rather than to colour. That's the reason for most video codecs to convert the input video data into a different representation,

YCrCb. The aim is to separate luminance and colour so that colour information can be encoded in lower resolution.

The luminance component can be calculated as follows:

$$Y = k_r R + k_g G + k_b B \quad (1)$$

The constants k_r , k_g , k_b are given by the h.264 specification [3].

Now only the difference from Y gets stored in the colour information:

$$C_r = R - Y$$

$$C_b = B - Y$$

(2)

$$C_g = G - Y$$

C_g can be expressed as a linear sum of the other variables [2], so only Y , C_r and C_b need to be stored. Figure 3 shows the decomposition of the same snapshot as above.



Figure 3.4: Original frame and YCrCb components

3.8 SAMPLING PATTERNS:

There are two chroma channels, thus in 4:2:0 sampling half as many chroma as luma samples are being stored.

3.9 PREDICTION:

A frame is divided into slices and each slice is divided into 16x16 pixel macroblocks. A macroblock can be further partitioned down to block size 4x4.

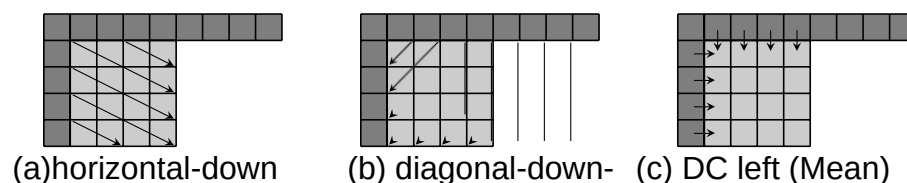


Figure 3.5: Intra prediction modes

Figure 3.5: A few different Intra Prediction Modes. Dark grey Pixels are already decoded, the light grey ones need to be predicted from them.

Figure 3.5: Inter Prediction. Showing I-, P- and B-Macroblocks.

There are three different kinds of frames, I, P and B frames, similarly there are I, P and B macro blocks. An I macro block makes a prediction based on the pixels to its left and top in the same frame (intraprediction), we can regard an I frame as a picture with a compression technique that is similar to JPEG. P macro blocks use a portion of a past frame as a prediction where the so-called motion vector usually has half pixel precision to capture fractional pixel movements. B macroblocks come with two motion vectors that can point in both future and past frames in order to make a more accurate prediction.

The first frame in a video has to be an I frame since there are no possible reference frames present. There are two kinds of I frames in H.264, regular I frames and IDR (Instantaneous Decode Refresh) frames. IDR frames contain only I macro blocks as well and also they don't allow references by future P and B frames behind them. IDR frames partition the video into sections that

can be decoded independently allowing the user to jump to a future position in the video stream without having to decode all frames in between. The first frame is an IDR .

Pixel precision B macro blocks combine both reference frame stoapredictionusing a weighted average or other combination methods. Prediction is a key part of each video codec.

Frames can also be partitioned in to horizontal slices, there are, P and B slices. Slices are separate units that can be decoded independently allowing the decoder to process a frame on parallel hardware. Blu rays for example have to contain at least 4 slices to ensure playability by relatively cheap hardware in blurayplayers.

In order to visualize howan encoder might use the different frame type sand how much they differinsize ,Figure7 shows the output ofx264 when encoding for eman with QP = 20. I frames are red, P frames blue and B framesgreen.

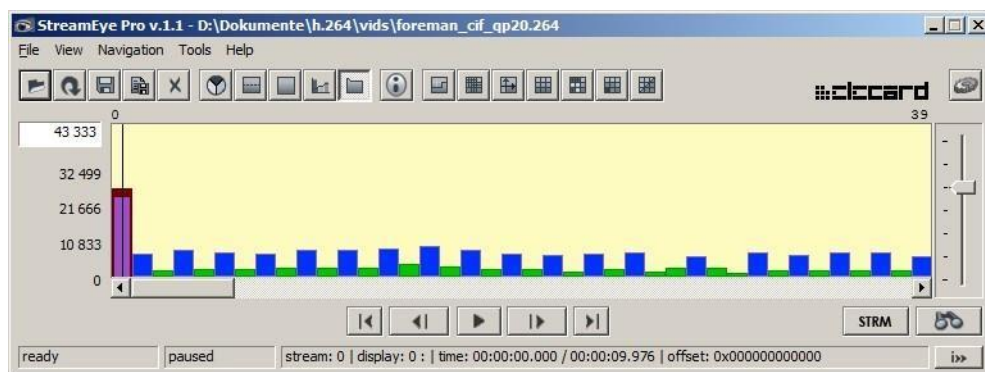


Figure 3.6: Example x264 output stream

3.10 TRANSFORMATION, QUANTIZATION AND ENTROPYCODING

H.264 supports two different luma transforms: the default and Intra 16x16 (AC/DC) transforms which are illustrated in figure8 and figure9 Chroma uses only the AC/DC transform, see figure10.

The Quantisation Parameter (QP) is constant for each 4x4 block and the

actual step sizes is exponential in QP, it doubles as QP increases by 6[2]. On input coefficient x we

Figure 3.7: Luma default Transform. Lower intensity indicates lower magnitude of coefficients. Quantisation is omitted here.

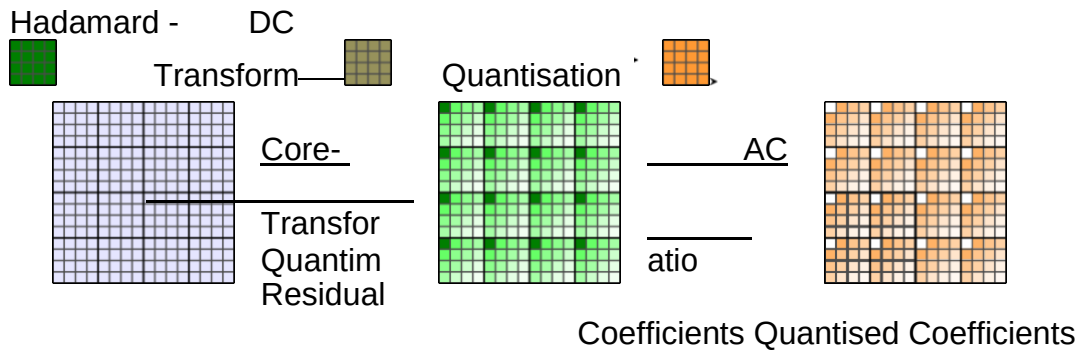


Figure 3.8: Luma intra 16x16 transform

Figure 3.8: Luma Intra 16x16 Transform. Partitions are not allowed here.

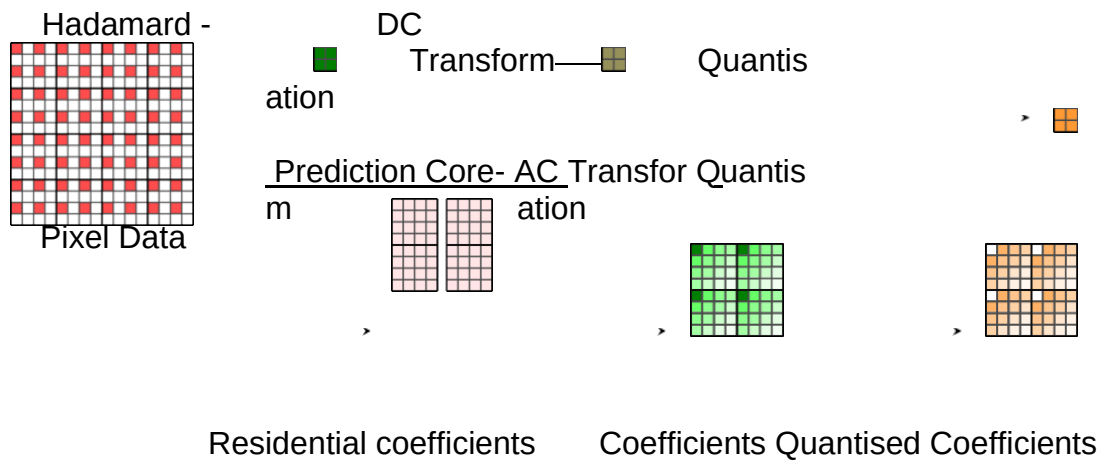


Figure 3.9: Chroma Transform.

Compute quantized coefficient by $y = \text{round}(\frac{x}{s})$. When decoding the video x will

approximated as $x = y \cdot \text{Pixel}$. values are integers, h.264 uses an integer approximation to the DCT as core transform, the Had amard transform is an integer trans for maswell, quantization in h.264 takes integers as input and produces integers as output.

Each video file has a profile defining which encoding techniques need to be

CABAC (Context Adaptive Binary Arithmetic Coding). Entropy coding is a lossless process whereas the quantisation is lossy, the decoded frame will differ from the original one. In CAVLC, the coefficient values are scanned in the zig-zag pattern which is illustrated in figure.

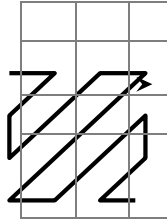


Figure 3.10: Zig-zag scan order

CHAPTER 4

STENOGRAPHY

4.1 STEGANOGRAPHIC CHANNELS:

A steganographic channel (system/stego system) fully describes how two parties called Alice and Bob communicate secretly. It consists basically of the following elements:

- i) cover source
 - ii) embedding and extraction methods
 - iii) source of stego keys (similar to cryptography)
 - iv) source of messages
 - v) data exchange channel, observed by warden Eve
- The part of Figure 12 can be defined formally[1]:

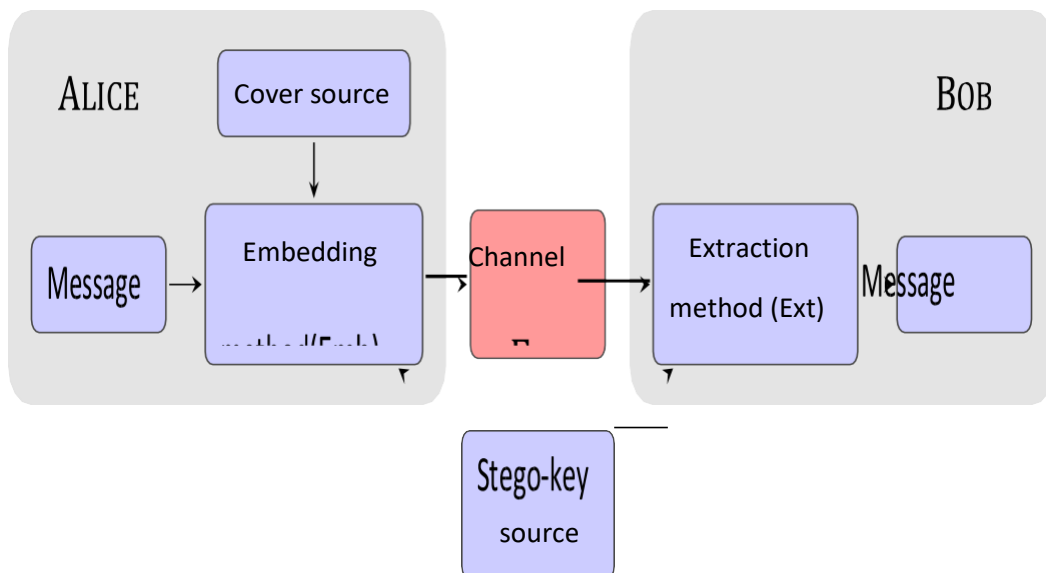


Figure 4.1: Steganographic channel

The exchanged data is controlled by warden Eve who we assume to be a passive observer. Alice and Bob require a stego system that achieves maximum payload within a certain detectability. We will now present different embedding methods and approaches to estimate their performance.

4.2 EMBEDDING METHODS

An obvious thing to do is to just set the LSB to the desired value. We may assume that in half of the cases the LSB already contains the correct value. The following diagram illustrates the hiding procedure:

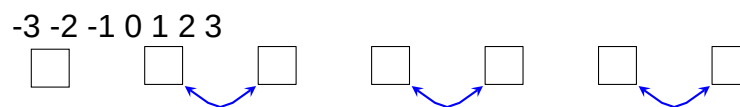


Figure 4.2: LSB Embedding.

Flipping the LSB changes a coefficient to a fixed target and is self-inverse. Therefore the coefficients group into pairs swapping values. If we use LSB embedding on each coefficient an example histogram of coefficient occurrences will change in the following way:

Figure 4.2: Impact of LSB Embedding on the coefficient histogram. Let l and r be two coefficient values that differ only in their LSB. Then half of the coefficients with value l will change into r and vice versa, the histogram will contain the average value in position l and r .

4.2 ± 1 EMBEDDING:

A problem of the LSB embedding is its asymmetry, the coefficients group into unnatural pairs. We can add symmetry by randomly increasing or decreasing a coefficient instead of flipping the LSB, this means that more than one bit may need to be changed.

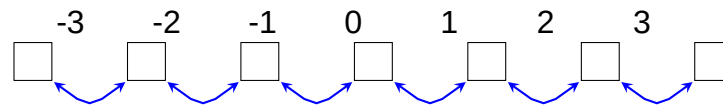


Figure 4.3: ± 1 Embedding.

We can perform 1 embedding on the example histogram above, assuming That the number of increments is exactly the number of decrements for any coefficient value:

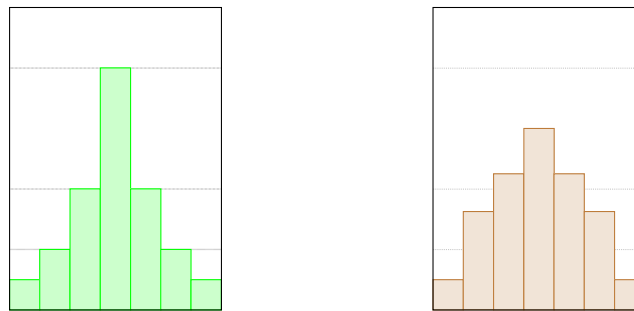


Figure 4.4: IMPACT ON COEFFICIENT HISTOGRAM

Figure 4.4: Impact of 1 Embedding on the coefficient histogram. The stego histogram is symmetric as the input histogram is as well, but the shape has $\pm$ changed. Now it is not two values that average out exactly but the entire histogram smoothens.

If we do not want to change the number of zeros on embedding but still wish to use the value 1 coefficients we have to always increase the absolute value of value 1 and -1 coefficients. This is the hiding scheme with threshold:

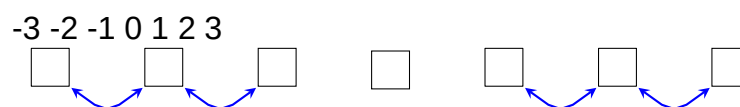


Figure 4.5: 1 Embedding with threshold 1.

Coefficients with values 1 or -1 always
Increment their absolute value.

The outcome on our example histogram is shown in Figure 18.

4.3 PRACTICAL EMBEDDING:

We have seen different methods to properly set the LSB of a particular coefficient in section 4.2 but the question remains which coefficients need to be changed to transmit the hidden message. The naive way is to set a random path through all coefficients in the video file and read their LSBs in sequence. We would expect Alice to change half.

We can use the fact that all coefficients have their LSB set to some value. Given a group of seven coefficients we have a binary vector of seven LSBs and the product $Hv = u$ is well defined. If Alice wants to communicate three dimensional vector x to Bob and $x = u$ she needs to find $u = ux$ where f all operations happen modulo-2 so that u again is a binary vector. Since $x=u$, u is non f zero and therefore it is a column in H , say column number j . Then the LSB in coefficient number j need to be flipped so that $Hv^j = x$.

The coefficients on the way to get the message right. Higher payload can be achieved by using Matrix Embedding.

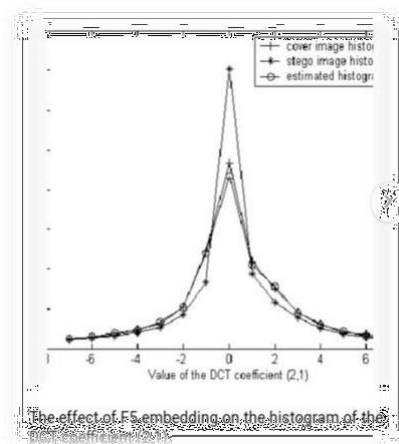


Figure 4.6: Embedding on coefficient histogram

4.4 MATRIX EMBEDDING:

Hamming codes are error correcting block codes based on the parity check matrix H containing all non-zero binary vectors in a fixed dimension[4]. In the case of 3dimensional vectors H is the following matrix

Three dimensional Hamming codes achieve three communicated bits by changing one coefficient in worst case where as the naïve approach requires to change bits on average to transmit 3 bits.

4.5 NON-SHARED SELECTION CHANNEL:

Alice having the clean cover object has strictly more information available than Bob does. For example Bob does not know if a particular coefficient was changed or not. For instance if Alice and Bob agree to use coefficients with absolute value "2" only for embedding to keep a distance to value 0 coefficients Bob can not know if a coefficient with value 1 had value 2 before Alice changed it or if it is an original 1. Thus Alice always has to in crementones.

Alice might even have the original uncompressed video and only want to use coefficients for embedding that are highly affected by quantisation. For example if the step size is 20 and the DCT coefficient is 89, quantized to 4, Alice would want to change it to 90 which is quantized to 5. If the DCT output is 80 Alice will not want to use that coefficient at all. This information is unknown to Bob, Alice and Bob need wet paper codes[1].

A possible solution to the problem is Alice and Bob to share a large binary Matrix D which will map vector v of LSB values from the video to the desired message u . Alice needs to solve $D v = u$ with the restriction that only dry coefficients are changed. This can be achieved using any solving method, Gaussian elimination for example. Bob will simply evaluate the product without knowing which parts of were manipulated by Alice. The matrix D can efficiently be defined using a common r and on generator seed.

4.6 PROPOSED FEATURES:

We assume the outcome of the DCT approximation that is used by h.264 to be an array where coefficients are scanned in zig-zag order.

We have seen in the last section that all of the mentioned hiding methods have a noticeable impact on the coefficients histogram, thus we certainly want to include the histogram in our set of features. Our histograms include zeros but only zeros before the last non-zero coefficient. Histograms alone are not good enough

because Alice can change coefficients that are not containing any hidden information read by Bob simply to restore histograms. Statistical restoration makes additional unnecessary changes in the video which may introduce additional visual distortion or distortion in other sets of features. In most cases statistical restoration reduces the security of a steganographic system[1].

The magnitude of coefficients tends to decrease as we move down the zigzag scan list, therefore we will drop coefficients at the end to reduce dimensionality. Also we want to use individual ranges for each coefficients in case ranges differ, the range can be determined by data visualization in the Stegosaurus GUI program.

Section 5.4 contains details about the chosen ranges.

In addition to histograms we will use co-occurrences as features. An array of 10 coefficients will allow to extract 9 co-occurrence pairs. Restoration of co-occurrence statistics is a much more complicated task and is likely to require a reduction of embedded payload. The DCT coefficients are highly decorrelated but h.264 uses a rough approximation to the DCT. The definition of the DCT is:

$$Y = A \cdot X \cdot A^T$$

The magnitude of coefficients tends to decrease as we move down the zigzag scan list, therefore we will drop coefficients at the end to reduce dimensionality. Also we want to use individual ranges for each coefficients in case ranges differ, the range can be determined by data visualization in the Stegosaurus GUI program.

Section 5.4 contains details about the chosen ranges.

occurrence statistics is a much more complicated task and is likely to require a reduction of embedded payload. The DCT coefficients are highly decorrelated but h.264 uses a rough approximation to the DCT. The definition of the DCT is:

$$Y = A \cdot X \cdot A^T$$

The magnitude of coefficients tends to decrease as we move down the zigzag scan list, therefore we will drop coefficients at the end to reduce dimensionality. Also we want to use individual ranges for each coefficients in case ranges differ, the range can be determined by data visualization in the Stegosaurus GUI program.

Section 5.4 contains details about the chosen ranges.

In addition to histograms we will use co-occurrences as features. An array of 10

coefficients will allow to extract 9 co-occurrence pairs. Restoration of co-occurrence statistics is a much more complicated task and is likely to require reduction of embedded payload. The DCT coefficients are highly decorrelated but h.264 uses a rough approximation to the DCT. The definition of the DCT is:

$$Y = A \cdot X \cdot A^T$$

There are likely to be correlations between coefficients specific to h.264 that will get lost when embedding data.

Each macro block has one luma channel but two chroma channels U and V. We may use the correlation between both chroma channels as features as well. Again we take histograms of 2 dimensional vectors, the first component being a coefficient in the U channel and the second component being the corresponding coefficient in V. We keep separate histograms for each coefficient, that is up to 16 if no coefficient is dropped.

All features mentioned above, that is (i) histograms, (ii) co-occurrences and, (iii) U vs V vectors are concatenated to the final feature vector. One vector is extracted per IDR section in a video. Histograms and co-occurrences of the Cr and Cb components are added up so that we have a luma, achroma DC and a chroma AC part.

P and B slices are treated separately since we are interested in which one of the two is capable of carrying more hidden information. An IDR section may only contain P slices which is why there can be more P vectors than B vectors. B slices rely on P slices as reference points, hence there is a corresponding P feature vector for each B feature vector. There are targeted attacks again LSB embedding using custom feature space tuning the similarity of the histogram with expected histogram after LSB embedding [5], estimating the embedded payload as well [6].

These features can be derived from the coefficients histogram and therefore they do not need to be included in our feature set. These method specific features allow us to check for specific methods using very low dimensional features but we attempt to use high dimensional features in order to detect any embedding method. Our features will be tested on the three different embedding methods mentioned above,

The KL-D is the expected number of extra bits needed to encode easily m bits that

follows distribution P when a code for distribution Q is used. If $P = Q$, $DKL(P \parallel Q) = 0$, the two distributions are identical, therefore they are not distinguishable.

We model a type as a multivariate normal distribution or Gaussian since the central limit theorem states that Gaussians are appropriate to model independent and identically distributed (iid) random variables. The KL-D of two Gaussians $N_0(\mu_0, \Sigma_0)$ and $N_1(\mu_1, \Sigma_1)$ in dimension k is:

This expression makes the KL-D computable, the computationally most intensive part is the inversion of Σ_1 , this can be achieved using the Householder QR-factorisation, details follow in section 5.6.

The MMD takes a more geometric approach and is related to the performance of support vector machines (SVMs). We are particularly interested in small KL-D estimates since these impose a bound on the performance on any detector. Small MMD estimates only tell us that SVMs will not perform well as detectors, but others might do. Both measures will be briefly introduced now.

PROPOSED DISTORTION MEASUREMENTS:

There are two distortion measures that we will use, the Kullback-Leibler Divergence (KL-D) and the Maximum Mean Discrepancy (MMD). The KL-D is an information theoretic concept and therefore any results shown using the KL-D are universally true for all possible detectors. The MMD takes a more geometric approach and is related to the performance of support vector machines (SVMs).

We are particularly interested in small KL-D estimates since these impose a bound on the performance on any detector. Small MMD estimates only tell us that SVMs will not perform well as detectors, but others might do. Both measures will be briefly introduced now.

4.9.1 KULLBACK-LEIBLER DIVERGENCE:

In information theory there is a notion of a type or empirical probability distribution P_x which is the fractional occurrence of each code word [4]. We assume that clean videos are of a certain type and different embedding methods will produce videos of a different type. Information theory also tells us that there is at most a polynomial number of types of sequences of length n while the number of possible sequences

is exponential. There are only few types each containing many sequences.

There is more noise in video than just steganographic embedding, different cameras, light settings or video of liters may

Leave at race in the video and form different types as well, we use similar video sources and identical encodings settings to reduce this problem. Detecting if a video is manipulated is testing whether it follows the clean type P_{clean} or a steganographic type, for example PF5, there are information theoretic bound on the performance of hypothesis test in given different types, the probability of missed detection is approximately [4]:

$$PMD \approx 2^{-DKL(P_{\text{stego}} \| P_{\text{clean}})}$$

We want to maximize this value since we want the warden to miss embedded data.

Therefore we need to minimize the KL-D, which is defined to be [4]:

4.92 MAXIMUM MEAN DISCREPANCY: The MMD of two distributions p and q and function class F is defined to be [7]: $MMD(p, q) = \sup_{f \in F} (E_{x \sim p} f(x) - E_{x \sim q} f(x))$

$F - f \in F$

In case of a finite sample sets X and Y of size N this is equivalent .

We have to make strong assumptions on the class of function to make the MMD F computable, Gaussian kernels have been shown to perform well when used in SVMs[8]. On this class of kernel functions an MMD estimate is given by:

$$MMD(F, X, Y) = \sqrt{k(x, x) + k(y, y) - 2k(x, y)}$$

where $k(x, y) = \exp(-\gamma \|x - y\|_2^2)$

We choose γ to be η^{-2} where η is the median of L_2 distances of all pairs of feature vectors. The features need to be normalized before calculating the MMDs, more on this in section 5.1.

The MMD checks for each pair of feature vectors how well they are separable by applying a Gaussian kernel which is directly related to the performance of an SVM on each pair of input vectors.

To answer these we will simulate embedding with different methods and two parameters, the threshold and a flag indicating if the DC coefficient shall be used

as well. The threshold is a lower limit on the coefficient value that is used for hiding, it is interpreted differently for each embedding method.

We are interested in the detectability of a certain payload that is embedded in a video, before we can interpret the distance measures in a meaningful way we need to measure the impact. In this project we will use bits per non-zero coefficient (bpnc) since the number of non-zero coefficients is a good indicator for the amount of data in a specific part of a video file. For each method we will have an early proportional map from embedding probability to bpnc.

For each embedding scheme we will extract features with embedding probabilities 0.001, 0.002, . . . , 0.009 and plot the bpnc of p on the x-axis and the y-axis. The square root law of steganographic capacity [9] tells us that larger covers require smaller embedding probabilities to ensure steganographic security. This explains the very small embedding probabilities we use in our experiments.

The KL-D has shown to be not computable in practice, see section 5.6 for details.

4.10 IMPLEMENTATION:

There is more than one step involved in the practical portion of this project:

- (I) create a video collection
- (II) exploratory data analysis
- (III) simulate hiding, extract features
- (IV) compare different embedding methods

Extraction is performed using a modification of ffmpeg called ffmpeg-extract. Before we can extract features we need to determine appropriate ranges for the histograms, this is achieved by visually plotting sets of features in the Stegosaurus GUI program. The GUI is also interface to manage feature sets and launch any calculations. First we will describe the video collection used for our experiments. left to right. Row i contains the histogram of the i 'th coefficient. Afterwards we have the co-occurrence features, again for L, U and V. The left coefficient is on the y axis, that is the reason for the stretch in y direction. The range for

The first coefficient is larger than the range for the second one. The remaining dots are UvsV features, one per coefficient with non-zero range.

The open feature sets are managed by the Stego Model. The Stego Model is the

back end of the GUI controlling all information displayed including the progress bar. See the Appendix for class definitions. Figure 25: shows the ranges that we use in our experiments, they have been set by visually exploring mean vectors in the Stego saurus program. We can calculate the dimension of each part of the feature vector: **histograms** We need to sum up the ranges, multiply by two to get positive and negative part

and add 1 for the zeros in each interval: $2 \cdot (15+11+10+3 \cdot 8+2 \cdot 5+3 \cdot 3)+16 \cdot = 168$ for luma, $2 \cdot (8+26+\dots+4)+4=52$ for chroma DC and $2 \cdot (24+3 \cdot 3)+15=49$ for chroma AC, all adding up to 269.

The total dimension is 1882.

Figure 25: shows the ranges that we use in our experiments, they have been set by visually exploring mean vectors in the Stego saurus program. We can calculate the dimension of each part of the feature vector: **histograms** We need to sum up the ranges, multiply by two to get positive and negative part

and add 1 for the zeros in each interval: $2 \cdot (15+11+10+3 \cdot 8+2 \cdot 5+3 \cdot 3)+16 \cdot = 168$ for luma, $2 \cdot (8+26+\dots+4)+4=52$ for chroma DC and $2 \cdot (24+3 \cdot 3)+15=49$ for chroma AC, all adding up to 269.

The total dimension is 1882.

The name is rooted in the etymological connection with steganography, figure 4.7 shows a screenshot. It is based on Qt1 and GPU accelerated by several CUDA9 kernels and the cuBLAS9 (CUDA Basic Linear Algebra Subroutines) library.

On the first level in the Collection tree we have a code for the embedding scheme, codes are listed in table 1.

The blue histograms and multi colour shapes below represent the same information, one uses bar height to indicate a certain value and the other uses opacity. Each pixel in the bottom figure has a corresponding bar and its opacity is the square root of the bar

Height divided by the maximum bar height. The square root gives the picture as mother look. Figure 23 gives a more detailed view.

4.10 STEGOSAURUSGUI:

The name is rooted in the etymological connection with steganography, figure 4.7 shows a screenshot. It is based on Qt1 and GPU accelerated by several CUD9 kernels and the cuBLAS9 (CUDA Basic Linear Algebra Subroutines) library.

On the first level in the Collection tree we have a code for the embedding scheme, codes are listed in table 1.

The blue histograms and multi-colour shapes below represent the same information, one uses bar height to indicate a certain value and the other uses opacity. Each pixel in the bottom figure has a corresponding bar and its opacity is the square root of the bar

Height divided by the maximum bar height. The square root gives the picture as mother look. Figure 23 gives a more detailed view.

	AC+DC			AC		
threshold	1	2	4	1	2	4
± 1	1	2	3	4	5	6
F5	1	1	1	1	1	1
	1	2	3	4	5	6
LSB	21	22	23	24	25	26

Table 4.1: method codes

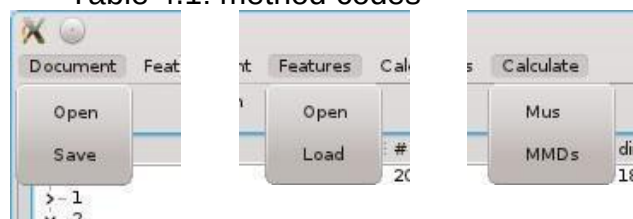


Figure 4.7: Gui menus



Figure 4.7: The first three units represent histogram features: L, U and V read

Initially the feature vector contained multiple QP values, the columns slider inside the Settings dock define how many QP chunk saresh own in each row in the central widgets. The scale option defines a linear scaling parameter on the histograms and the height defines the maximum shown height of the histograms without changing the scaling. Increasing the height makes the program display more information.

4.10.1 THE XMLBACKEND:

The GUI uses an xml document to store all files containing feature vectors that are available. The document is hardcoded to be stegodoc.xml in the current directory, other documents can not be opened. The xml parsing is done using Qt's DOMAPI. We can see that the collection is represented by a tree structure other user which makes the tree-like DOM apia convenient back end. This part of a stegodoc.xml file:

Listing 1: stegodoc.xml

```
< stegosaurus >
<sets >
<featureSetmethod="0"qp_range="1"qp_offset="20"type="P">
<file>/mnt/stego/features/seta/0/p_clean_0_minc_1_qp_20.fv</file>
<file>/mnt/stego/features/setb/0/p_clean_0_minc_1_qp_20.fv</file>
<file>/mnt/stego/features/setc/0/p_clean_0_minc_1_qp_20.fv</file>
<file>/mnt/stego/features/setd/0/p_clean_0_minc_1_qp_20.fv</file>
<file>/mnt/stego/features/sete/0/p_clean_0_minc_1_qp_20.fv</file>
<file>/mnt/stego/features/setf/0/p_clean_0_minc_1_qp_20.fv</file>
<file>/mnt/stego/features/setg/0/p_clean_0_minc_1_qp_20.fv</file>
<file>/mnt/stego/features/seth/0/p_clean_0_minc_1_qp_20.fv</file>
</featureSet>
```

```
<featureSetmethod="1"qp_range="1"qp_offset="20"type="P">
```

There are 120gb of features divided into 7994 files for each qp offset value, the xml document allows to quickly browse load with out having to re-open the m every time. Files are inserted into the system by Features Open which opens a Q File Dialog that can select and open multiple. Fv files. They will be inserted in to the DOM tree automatically and can be stored to disk via Document Save. Duplicates are ignored. Features Load opens a dialog asking which feature sets are to be loaded.



Figure 4.8: snapshot of feature dialog

Figure 4.8: A snapshot of the feature loading dialog. The combo boxes from left to right present minimum method, maximum method, qp offset.

Qp range and type (PorB).The corresponding set of clean features is loaded as well.

4.10.2 THE .FV FILE FORMATE:

Features are stored as unsigned 32 bit integer counts in the feature files. An advantage is that 32 bit integers require only half as much disk space as 64 bit doubles do.

We need a header for Stegos auru store a feature files properly, the are the header elements and their types: **slice_type [char]** Type 0 represent P slices, 1 represents B slices. **method [char]** Method code, as described above. **probability [double]** Embedding probability, only stored if method $f=0$.

Accept [char] Channels used for embedding, again only stored if method = 0. The first bit indicates if luma is being used, the second bit chroma DC and the third bit chroma AC. Therefore we have 1 representing luma only, 6 representing chroma only and 7 representing both.

qp_offset [char] We use 20 as qp offset in our experiments.

qp_range [char] We use single qp, i.e. range 1, in our experiments. Increasing the range not only increases the dimensionality but also then umber of vectors that are extracted since it gets less likely to have a zero count vector in an IDR section.

ranges [unsigned char] The histogram ranges for each individual coefficient.

The dimension is not stored explicitly but all information is given that is required to compute the features dimensionality. Also the number of vectors is not included

4.11 NORMALIZATION:

Prior to any normalization technique we need to scale counts appropriately. One vector of counts might be a multiple of another count vector, we consider such vectors to be identical. We scale counts so that the sum of all components equals the dimension, each component is one on average. This allows us to change the dimension without having to adjust the scaling in the GUI each time. We have implemented two normalization techniques:

4.11.1 Rescaling:

The minimum component, which is typically zero in our case, is mapped to zero, the maximum component is mapped to one. Values in between are linearly interpolated. For component j we get scaled value $c^j = (c - \min) / (\max - \min)$.

4.11.2 Standardizing:

We first find the mean of all raw feature vectors. On an input vector we subtract the mean and multiply each component afterwards so that we have variance one in every component. The mean of standardized vectors is zero.

All parameters, these are minimum/ maximum values, mean vector and soon, that are needed for normalization are found on basis of the clean set only. It is important that we use the same parameters across both clean and stego sets, otherwise we might lose a steganographic impact.

The purpose of normalization is to treat every component of the feature vector equally.

We have very non-uniform data, the snapshot in figure. shows non-normalized data.

Changes in large coefficients that occur very rarely will not have a large impact on the shape of the vector, but they should not be treated with less significance than changes in smaller coefficients.

Listing 2: encode.sh (raw)

```
#!/bin/sh
```

```
#Usage:encode.sh<bitrate><file> output="../../raw/$1/dvd_$2_cavlc_$1.avi"
```

```
ffmpeg-y-pass1-i$2-vcodeclibx264-x264optscabac=0
```

Listing 4.1: encode.sh (denoising)

```
$1 - deinterlace - vf scale = 1024:576 ,hqdn3d ,unsharp
```

Of these 61 videos then 16 were picked so that the number of contained feature vectors was roughly 10 times the dimension to aid stable results, these were the videos containing fewest vectors, we work with a set of 20577 vectors. Extractions were made with bit rates 1000k, 2000k and 3000k for both raw and denoised video but we only had enough time to process one set of videos, the 3000k raw encodings.

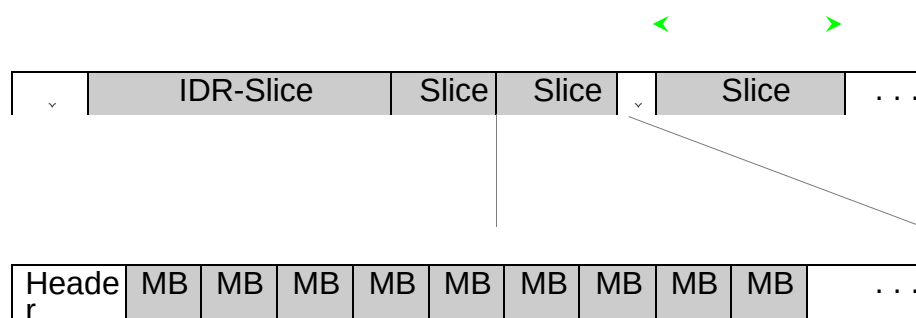
4.12 H.264SYNTAX:

Formally a video bitstream is not stored in a file but transmitted over a Network Abstraction Layer (NAL) which contains different NAL units. A frame is divided into slices, this allows the decoder to decoded part of the Frame on different processors. An IDR (Instantaneous Decoder Refresh) slice signals the decoder to store refresh itself, Motion Vectors cannot point behind a Frame that contains an IDR-Slice.

The details of the parameter of the slice Header are not important here, but we will explain the macro block header elements:

MB Type Defines type (I, P or B) and partition of the macro block.

elements



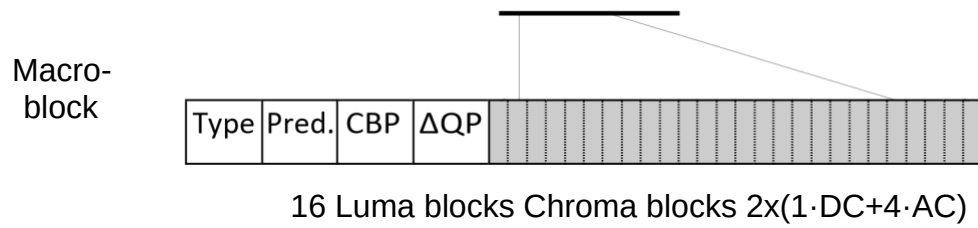


Figure 4.9: Syntax of H.264 Bitstream.

4.13 FEATUREEXTRACTION:

The extraction takes place in the `decoded_residual` method in the `h264_cavlc.c` file in `ffmpeg` library. Extractional ways happens when an h.264 video is decoded, we transcode the input video sin to a computationally cheaper codec (MPEG-1) to launch the extraction. This is the extraction script:

```
#!/bin/sh
# extract.sh<file>
```

CHAPTER 5

SOFTWAREDISCRIPT

5.1 Extract.sh

All parameters are exhausting coded, we'd like to vary code and recompile if we would like to vary embedding strategies or the other parameters. Additionally to the higher than the extraction program outputs video statics aswell, that's the quantity of vectors extracted from every file and also the average bpnc. The various log files will simply be concatenated by victimization bash practically. This can be a part of associate example concentrated log file:

Listing 5: logs_qp_20 (for the primary 2 DVDs victimization methodology

```
[qp_20,C,p=0.001]averagep_bpnc:0.000181641--1549vectors.
```

```
[qp_20,C,p=0.001]averageb_bpnc:0.000262209--1213vectors.
```

```
[qp_20,C,p=0.001]averagep_bpnc:0.000211074--  
1402vectors.
```

```
[qp_20,C,p=0.001]averageb_bpnc:0.000379492--  
1202vectors.
```

```
[qp_20,C,p=0.002]averagep_bpnc:0.000361263--  
1549vectors.
```

```
[qp_20,C,p=0.002]averageb_bpnc:0.000514996--  
1213vectors.
```

```
qp_20,C,p=0.002]averagep_bpnc:0.000361263--  
1549vectors.
```

```
[qp_20,C,p=0.002]averageb_bpnc:0.000514996--  
1213vectors.
```

```
[qp_20,C,p=0.001]averagep_bpnc:0.000211074--  
1402vectors.
```

```
[qp_20,C,p=0.001]averageb_bpnc:0.000379492--
```

1202vectors.

[qp_20,C,p=0.002]averagep_bpnc:0.000488388--
1402vectors.

[qp_20,C,p=0.002]averageb_bpnc:0.000780432--
1202vectors.

[qp_20,C,p=0.001]averageb_bpnc:0.000262209--1213vectors.

[qp_20,C,p=0.001]averagep_bpnc:0.000211074--
1402vectors.

[qp_20,C,p=0.001]averageb_bpnc:0.000379492--
1202vectors.

[qp_20,C,p=0.002]averagep_bpnc:0.000361263--1549vectors.

[qp_20,C,p=0.002]averageb_bpnc:0.000514996--1213vectors.

[qp_20,C,p=0.002]averagep_bpnc:0.000488
388--140202

[qp_20,C,p=0.002]averageb_bpnc:0.000780
4321202vect

[qp_20,C,p=0.003]averagep_bpnc:0.000547
533--1405

[qp_20,C,p=0.003]averageb_bpnc:0.0007
90526--1213

[qp_20,C,p=0.001]averagep_bpnc:0.000211074--
1402vectors.

[qp_20,C,p=0.001]averageb_bpnc:0.000379492--
1202vectors.

[qp_20,C,p=0.004]averagep_bpnc:0.000488388--
1402vectors.

[qp_20,C,p=0.004]averageb_bpnc:0.00078043

2--1202vectors

[qp_20 , LC , p = 0.001] average p_ bpnc: 0 .000931314 --- 1549

vectors. [qp_20 , LC , p = 0.001] average b_ bpnc: 0 .00081836 --
- 1213vectors.

[qp_20 , LC , p = 0.001] average p_ bpnc: 0

.000904551 --- 1402 vectors. [qp_20 , LC , p =

0.001] average b_ bpnc: 0 .000697173 ---
1202vectors.

[qp_20 , LC , p = 0.002] average p_ bpnc: 0

.00199814 --- 1549 vectors. [qp_20 , LC , p = 0.002] average b_ bpnc: 0
.00197204 --- 1213vectors.

[qp_20,C,p=0.001]averagep_bpnc:0.000181

641--1549vectors.

[qp_20,C,p=0.001]averageb_bpnc:0.000262

209--1213vectors.

[qp_20,C,p=0.001]averagep_bpnc:0.000211

074--1402vectors.

[qp_20,C,p=0.001]averageb_bpnc:0.000379

492--1202vectors.

[qp_20,C,p=0.002]averagep_bpnc:0.000361

263--1549vectors.

[qp_20,C,p=0.002]averageb_bpnc:0.000514

996--1213vectors.

[qp_20,C,p=0.002]averagep_bpnc:0.000488

388--1402vectors.

[qp_20,C,p=0.002]averageb_bpnc:0.000780

432--1202vectors.

[qp_20,C,p=0.003]averagep_bpnc:0.000547

533--1549vectors.

[qp_20,C,p=0.003]averageb_bpnc:0.000790

526--1213vectors.

...

[qp_20,C,p=0.009]averagep_bpnc:0.0024

3165--1402vectors.

[qp_20,C,p=0.009]averageb_bpnc:0.0036

6531--1202vectors.

[qp_20 , LC , p = 0.001] average p_ bpnc: 0 .000931314 --- 1549
vectors. [qp_20 , LC , p = 0.001] average b_ bpnc: 0 .00081836 --
- 1213vectors.

[qp_20 , LC , p = 0.001] average p_ bpnc: 0
.000904551 --- 1402 vectors. [qp_20 , LC , p =
0.001] average b_ bpnc: 0 .000697173 ---
1202vectors.

Program:

```
voidsimulate_hiding_plusminus(H264FeatureContext*fc,intblocknum,intthresh){ int i;  
int * coefs = fc -  
> tape; intmin;  
doubler;  
int sl = fc ->slice_type ;
```

//check if we are on correct channel

```

    if(!(fc->accept_blocks&(1<<block num)))
    {
        return ;
        for(i=min;i<num_coefs[blocknum];i++)
        {
            // checkrange
            if(coefs[i]<thresh&&coefs[i]>-thresh)continue; r=(((double)ran
            //workoutthecorrectstartingpointinarray switch(blocknum)
            {
            case 0:
            min=MIN_COEF-1;
            b
            r
            e
            a
            k
            ;
            c
            a
            s
            e
            1
            :
            if(MIN_COEF>1) return;
            //weonlyhavefirstcoefficientshere
            m
            i
            n
            =
            0
            ;
            b
            r

```

```

e
a
k
;
c
a
s
e
2
:
min=max(MI
N_COEF-
2,0); break ;
}

for(i=min;i<num_coefs[blocknum];i++)
{
// checkrange
if(coefs[i]<thresh&&coefs[i]>-thresh)continue;
r=(((double)rand())/((double)RAND_MAX));
voidsimulate_hiding_plusminus(H264FeatureContext*fc,intblocknum,intthresh){ int i;
int * coefs = fc -
> tape; intmin;
doubler;
int sl = fc ->slice_type ;

//checkifweareoncorrectchannel
if(!(fc->accept_blocks&(1<<blocknum)))
{
return ;
}

//workoutthecorrectstarti

```

```

ngpointinarray
switch(blocknum)
{
  case 0:
min=MIN_COEF-1;
b
r
e
a
k
;
c
a
s
e
1
:
if(MIN_COEF>1) return;
//weonlyhavefirstco
efficientshere
min=0;
b
r
e
a
k

.00199814 --- 1549 vectors. [ qp_20 , LC , p = 0.002] average b_ bpnc: 0
.00197204 --- 1213vectors.

...

[qp_20,L,p=0.008]averagep_bpnc:0.0060
3218--1402vectors.

```

[qp_20,L,p=0.008]averageb_bpnc:0.0046
1397--1202vectors.

[qp_20,L,p=0.009]averagep_bpnc:0.0073
7204--1549vectors.

[qp_20,L,p=0.009]averageb_bpnc:0.0065
5301--1213vectors.

[qp_20,L,p=0.009]averagep_bpnc:0.00677
329--1402vectors.

[qp_20,L,p=0.009]averageb_bpnc:0.0055
5186--1202vectors

We used the weighted average of these bpnc values, weighted according to the number of vectors they represent, to construct the bpnc MMD maps. We only computed the weighted average for $p=0.009$ and interpolated the remaining $\rightarrow$ values in `cethebpnc` has shown to be practically linear for the numbers of vectors we worked with. It was not feasible in the time given to compute the weighted average for every probability value

```
for(i=min;i<num_coefs[blocknum];i++){  
  // checkrange  
  if(coefs[i]<thresh&&coefs[i]>-thresh)continue;  
  r=(((double)rand())/((double)RAND_MAX));  
  if ( r < fc -> p_hide) { //statisticsusedforbpnc calculation coefs[ i] -= 1;  
  }  
  statistics used for bpnc calculation
```

5.1 EMBEDDED SYSTEM

The implementations of the different embedding methods only differ the very end where the actual embedding is performed and in the range checking, we will therefore give the for loop only for the sefunctions:

Listing 5.2: F5 embedding simulation

The implementations of the different embedding methods only differ at the very end where the actual embedding is performed and in the range checking, we will therefore

give the for loop only for these functions

The embedding procedure of our proposed image steganography technique is presented in the following lines.

Select initial values for generating two S-boxes SbM and SbN with length m and n respectively, where the size of the cover image is $m \times n$.

Expand the secret image S with 8-bit and size $m \times n$

2 2 to image Sx

coefs[i] -= 1;

}

with 2-bit and size $m \times n$.

Use the generated S-boxes to select the position of pixels to embed the image Sx as stated

UNTUN SOFTWARE:

The threshold is interpreted differently for each embedding method, table 2 shows intervals of coefficients that are not used for embedding.

The H264FeatureContext contains the current feature counts as well as the statistics needed to calculate bpnc values, details are given in the appendix.

Table 5.1: Unused coefficients for every embedding methodology reckoning on threshold. we have a tendency to see that the interval size for LSB embedding is uneven and increasing most quickly.

We 1st ought to estimate the mean μ and also the variance matrix Σ from the extracted feature vectors X. Let N be the amount of vectors, then we have a tendency to have:

$$N\Sigma = 1$$

$$\mu = \frac{1}{N} \sum_{i=1}^N x_i$$

$$i=1$$

$$N\Sigma = \sum_{i=1}^N (x_i - \mu)(x_i - \mu)^T$$

$$\Sigma = \sum_{i=1}^N (x_i)$$

We need to invert Σ to calculate a KL-D estimate, as delineate in section four.5.1. We want to figure as exactly as potential, associate degree approximated resolution isn't satisfactory, thus we have a tendency to determined to use the Housholder QR resolving to search out the inverse. A letter of the alphabet R factgorization of matrix A may be a decomposition of A into the merchandise letter of the alphabet R wherever Q is orthogonoal and R isupper -triangular. A homeowner matrix is of the subsequent form:

$$ww^TH = I - a \text{ pair of } w^T w$$

On input vector v the a part of v in direction of w can amendment sign, the remainder are unchanged. we {are able to} at once see that there are 2 eigenvalues, 1 and 1, one one =1 – having solely was eigenvector. additionally H is symmetrical . If we have a tendency to use w_0 specified w we have a tendency to get associate degree orthonormal rather than simply associate degree orthogonal matrix H , since: $H \cdot H^T = H \cdot H = (I - 2w_0w^T)(I - 2w_0w^T) = I - 4w_0w^T + 4w_0w^T = I$

Starting with matrix Σ within the opening move we would like to remodel the primary column vector to a multiple of the primary unit vector employing a homeowner transformation. The corresponding

We will perpetually need the sign of element txt amendment for additional stable results. If v was terribly about to the x axis we have a tendency to might encounter issues with numerical exactness.

We next apply H_2 to the complete matrix and shortly. The diagonal components have to be compelled to be off from the matrix and square measure keep in a very separate array. this fashion QR resolving may be computed in situ. The determinant of a homeowner matrix is one since the determinant is that the product of chemist values. We have a fair variety of dimensions, thus the determinant of Σ is that the product of R 's diagonal entries that we've got keep in a very separate array. Itturns out that particularly towards the top the entries get terribly tiny, resulting in a much zero product, Σ isn't invertible, thus we have a tendency to cannot calculate the KL- D this fashion. this is for all standardisation

techniques we've got enforced.

If we have a tendency to reduced the feature set by removing all options that a true ways that or nearly perpetually zero and linearly dependent options this approach may work. There is an entirely totally different approach to estimate the KL-D of 2 sets of points primarily based on k nearest neighbours (kNN)[7] however these square measure shown to be ever. MMD: GPU provide massively parallel computing units that achieve more FLOPS compared to CPU singular, but peak performance on GPU is only achieved under certain conditions. Threads on the GPU are organized in to three dimensional blocks and blocks are organized in a three dimensional grid. A number of threads are executed simultaneously, but they all have to perform the same operations at any given time. If there is a branch, for example if the first half of the threads satisfies an if condition and the second half does not, the first half will be executed simultaneously and afterwards these conditions as well [10]. For best performance we want to minimize branching. The performance of GPU world memory depends on the manner it's accessed, reading consecutive blocks is quick however performance decreases to a tiny low fraction if random locations in GPU world are accessed [11]. There square measure completely different levels of memory during a GPU, there's the worldwide memory that is that the largest and slowest, then there's shared memory that is shared between threads within the same block and native memory accessible by individual threads solely. We could use cuBLAS to accelerate the kernel computation, but we do not achieve best performance that way. If we were to implement the vector sum on the GPU we would launch one thread per element for the first half of the vector and compute the sum with another element in the second half. This can be repeated until there is only one element left which is the result [10]. We see that this implementation does not achieve

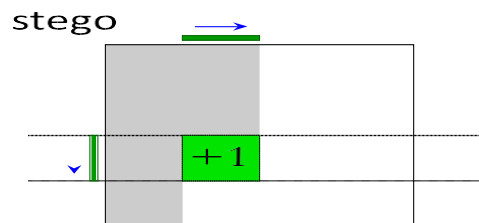
5.2 EMBEDDING STIMULATION:

Peak performance since then number of busy threads decreases overtime. we tend to assume that Cu BLAS uses an analogous implementation. The dimension of the feature vector is comparatively little, every CUD kernel launch comes with associated overhead, decreases cuBLAS.

Instead we would like to launch a thread for every pair of vectors (x, y) and let it figure $\|x - y\|_2$ by running squares of $x_i - y_i$ consecutive as we tend to would do on the

central processing unit. we've to partition the feature vector into partitions since there's a timeout on kernel executions. This computation is that the opening move in each the estimation of γ moreover because the calculation of the MMD itself. For the MMD calculation we'd like to multiply with $-\gamma$ and calculate the exponential once wards. Since GPU international memory is incredibly restricted we'd like to figure block-wise and launch the kernel multiple times so as to hide any vector is calculable on the clean set solely, $\|x - y\|^2 = \|y - x\|^2$, we tend to solely ought to look into half the pairsthere. Instead we would like to launch a thread for every try of vectors (x, y) and let it figure $\|x - y\|^2$ by rundown sqaures of $x_i - y_i$ consecutive as we'd do on the central processing unit. For the MMD calculation we'd like to multiply with $-\gamma$ and calculate the exponential once wards. Since GPU international memory is incredibly restricted we'd like to figure block-wise and launch the kernel multiple times so as to hide any vector is calculable on the clean set solely, $\|x - y\|^2 = \|y - x\|^2$, we tend to solely ought to look into half the pairs there. MMD ESTIMATION:

We need to cover three combinations, clean x clean, clean x stego and stego x stego. We compute the differences of the three directly to increase numerical stability, if we Were to sum up all results for clean x clean and the other two cases separately and find the difference afterwards we may lose precision. Doubles are



represented by main and exponent an close accuracy in absolute terms as the absolute value of a number increases [12].

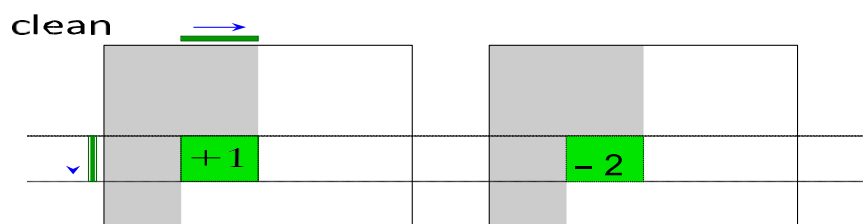


Figure 5.1: illustrates the computation of the MMD estimate.

Figure 5.1: Our MMD implementation. The green parts are buffers on the GPU where each dark green unit represents an array of features vector while the light green squares are two dimensional arrays of doubles. We can imagine this picture to have depth D where D is the feature dimension and the light green parts scan through the vectors towards or a way from us. Both right and down moving dark green arrays are always in the same position.

CHAPTER - 6

RESULTS

6.1 VISUALDISTORTION:

For visible effects we need to maximize impact, set the threshold and minimum coefficient as well as the hiding probability to one. Now we present a series of snap shots of the 20 Foreman encoding done by x 264. Frame number 252 is an IDR frame, thus we see a discontinuity there. All types of slices and macro blocks are used for embedding here, when extracting features we consider P and B slices only. The main focus of this project is on feature analysis and not on the examination of visual distortion.

6.2 FEATUREANALYSIS:

We have a collection of options for every (i)embedding methodology, (ii)embedding likelihood, (iii) slice sort, (iv) channel, (v) threshold worth and (vi) use of AC solely or AC and DC coefficients. every purpose within the graphs below represents the MMD of 1 such feature set



(a) frame1 (b) frame51 (c) frame101 (d) frame151



(e) frame201 (f) frame251 (g) frame252 (h) frame301

Figure 6.1: Unmodified Foreman snapshots





(e) frame201 (f) frame251 (g) frame252 (h) frame301

Figure 6.2.: ± 1 embedding in Luma and Chroma

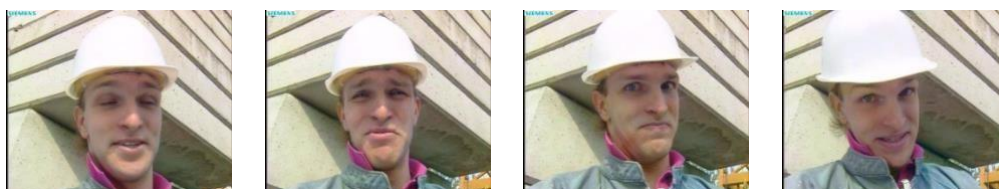


(a) frame1 (b) frame51 (c) frame101 (d) frame151



(e) frame201 (f) frame251 (g) frame252 (h) frame301

Figure 6.3: ± 1 embedding in Luma

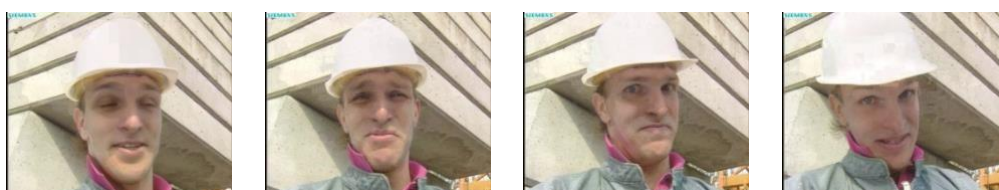


(a) frame1 (b) frame51 (c) frame101 (d) frame151



(e) frame201 (f) frame251 (g) frame252 (h) frame301

Figure 6.4: ± 1 embedding in Chroma

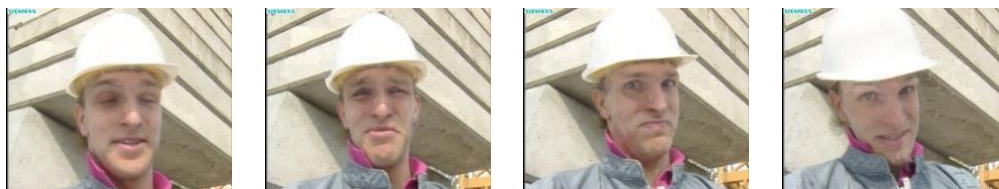


(a) frame1 (b) frame51 (c) frame101 (d) frame151



(e) frame201 (f)frame251 (g)frame252 (h) frame301

Figure 6.5: F5 embedding in Luma and Chroma



(a) frame1 (b)frame51 (c)frame101 (d) frame151



(e) frame201 (f)frame251 (g)frame252 (h)frame253

Figure 6.6: F5 embedding in Luma

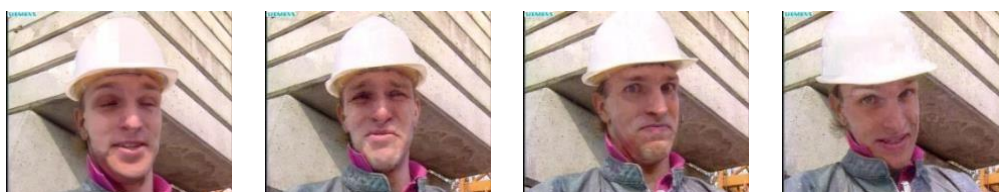


(a) frame1 (b)frame51 (c)frame101 (d) frame151



(e) frame201 (f)frame251 (g)frame252 (h) frame301

Figure 6.7: F5 embedding in Chroma

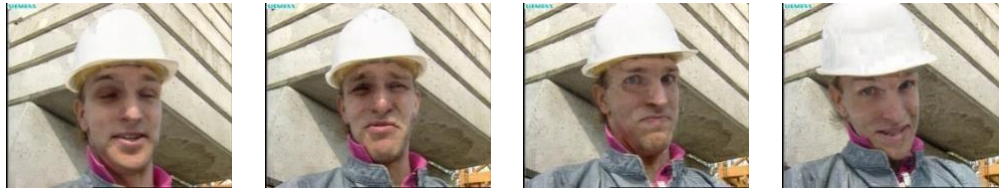


(a) frame1 (b)frame51 (c)frame101 (d) frame151



(e) frame201 (f)frame251 (g)frame252 (h) frame301

Figure 6.8: LSB embedding in Luma and Chroma

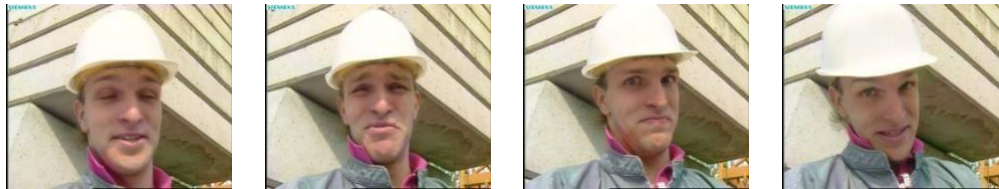


(a) frame1 (b)frame51 (c)frame101 (d) frame151



(e) frame201 (f)frame251 (g)frame252 (h) frame301

Figure 6.9: LSB embedding in Luma



(a) frame1 (b)frame51 (c)frame101 (d) frame151



(e)frame201 (f)frame251 (g)frame252 (h) frame301 Figure

6.9: LSB embedding in chroma

Figure 6.9: LSB embedding in Chroma and the corresponding set of clean features. every feature set contains a corresponding average bpnc (bits per non zero coefficient) price, we tend to investigate the MMD in term soft bpnc to answer our

questions on embedding. Figure thirty-nine shows the MMD as a perform of bpnc.

where the embedding probability ranges from 0.1 to 1. Each point is the result of an MMD calculation on two sets of 20577 vectors with dimension 1882, extracted from 16DVDs.

Figure 6.1: bpnc MMD for 1 embedding in AC and DC using threshold 1 (P slices). The embedding probability ranges from 0.1 to 1.

The MMD is locally linear near zero[13], therefore it can be accurately represented by a function of type $y=ax$ for a small embedding·Probability We present the parameter a of the least squares approximation for all graphs we computed in table 3 and we will give full graph son a few example:answer our questions about embedding. The embed in probability in the following experiments ranges from 0.001 to 0.009, in uniform steps of size 0.001.

6.3 EMBEDDING DETECT:

We will look at the detectability three different channel embeddings (luma only, both chroma channels and luma and chroma) 1 embedding with threshold 1 since it uses all non zero coefficients for embedding. These are the corresponding MMD values in terms of bpnc for P slices:

We can see that the MMD estimate is very stable and that chroma is least suited for embedding. The UvsV features that are available for chroma only increase detectability in chroma. But we can also see that a combination of luma and chroma is slightly better than using luma alone.

figure 6.3:shows the same graph for B slices. We can see that the value sare less table and that using luma alone achieves lowest detectability here from the

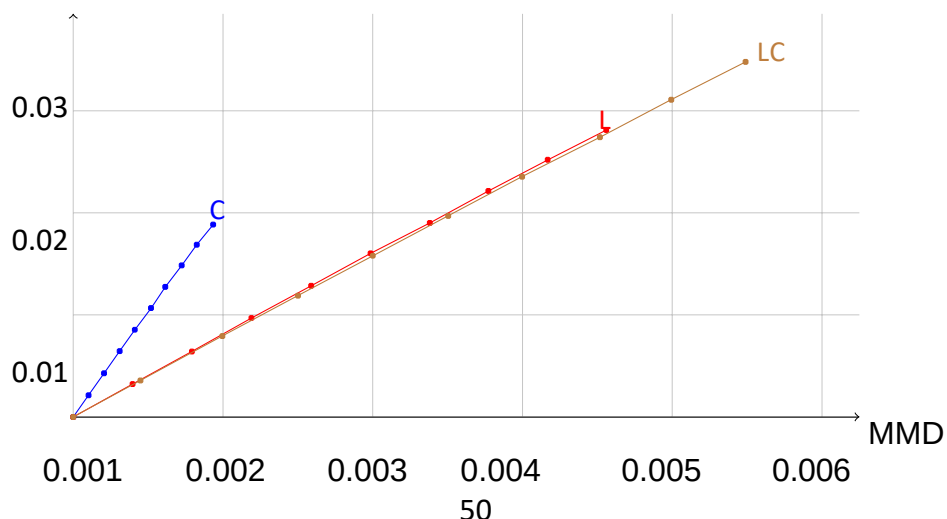


Figure 6.2: bpnc $\rightarrow$ MMD for ± 1 embedding in AC and DC using threshold 1 (P slices).

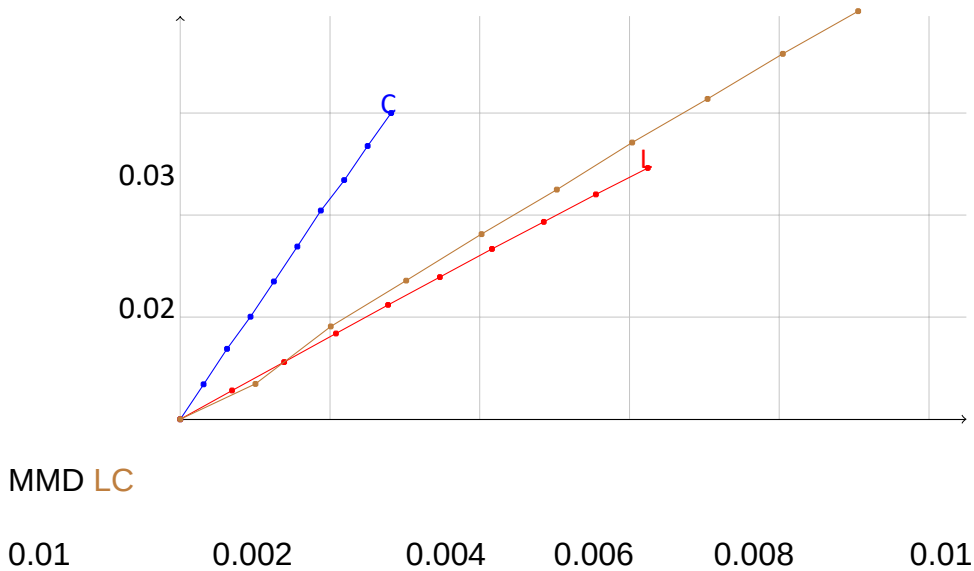


Figure 6.3: bpnc $\rightarrow$ MMD for ± 1 embedding in AC and DC using threshold 1 (B slices).

It is more detectable than in luma alone in this case. If we were to use al owner probability on chroma embedding we could possibly beat luma alone embedding here as we

6.4 F5 EMBEDDING:

We can only compare1 embedding with threshold2 and F5 embedding with threshold directly since they use the same coefficients for embedding, see table for preciserange definitions. These are the corresponding graphs:

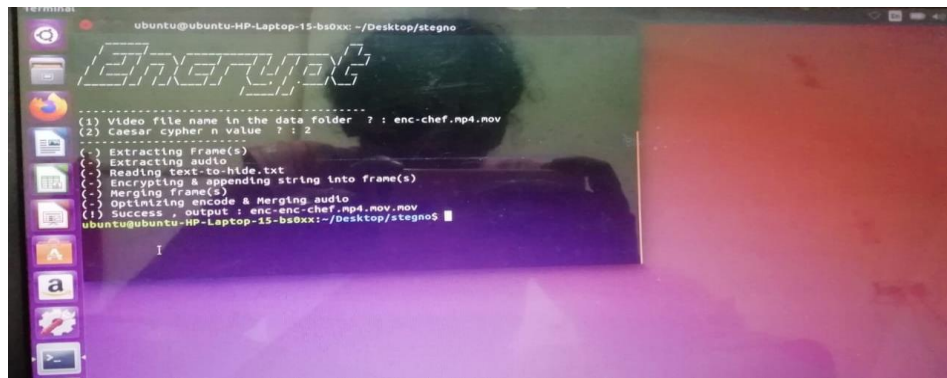


Figure no:6.10 Encrypt of messages

MMD:

1. The different threshold values are needed for both methods to embed in the same coefficients.

2.F5 is better performing, chroma embeddings using F5 embedding are less detectable than luma embeddings using 1embedding.

We cannot compare the LSB embedding as directly since it is a symmetrical and never uses the same coefficients for embedding as the other methods do. But we can infer from table 3 that it does not achieve lower growth rates than F5 embedding for thresholds 1 and 2. With threshold 4 the LSB embedding does not yield meaningful results in chroma since there is hardly any embedding at all. F5 is the least detectable among the embedding methods tested. Comparing values at the same threshold LSB embedding out-performs embedding.

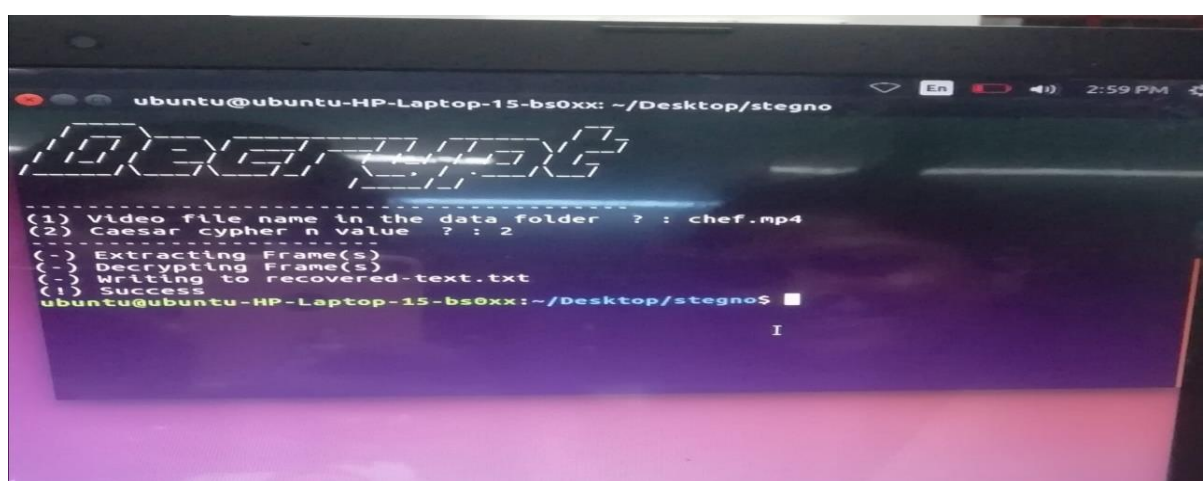


Figure no 6.11 decrypt result

COFFIENT AND BEST:

We can see in table 3 that a cross all embedding methods we get minimum detectability at threshold 1 and when using both AC and DC coefficients. This tells us that we want to hide in coefficients that occur frequently.

MMD growth rates. The threshold is taken otherwise for every methodology, seetable a pair of for precise ranges. a awfully little variety of Na NMM Destimates was unnoticed within the computation of the smallest amount squares approximation. we tend to get NaN result if the add below the root of the MMD estimate is negative.

We observe that for threshold 1 and embedding in the DC coefficient as well P and B slices show similar behaviour, but as the threshold increases or the DC coefficient not use detect ability in B slices increases faster than it does in P slices. We can deduce that coefficients in Bslices are generally smallerthan coefficients in Pslices and higher concentrated linearly places in the scanorder. Since small larval uesand earlier places are preferable for hiding both slice types are equally well suited for embedding The threshold is taken otherwise for every methodology, seetable a pair of for precise ranges. a awfully little variety of Na NMM Destimates was unnoticed within the computation of the smallest amount

			P			B		
		t	L	C	LC	L	C	LC
±1	AC AC+DC	1		10.230			10.729	
		2	3.988 6.582	24.795	3.903 6.652	3.979 7.046	37.170	4.450 8.736
		4	11.122	47.643	11.185	15.596	40.352	14.109
		1	7.282 11.560	11.747	7.090 11.315	8.805 15.016	15.977	8.631 13.759
		2		28.002			22.831	
		4	19.661	85.349	19.482	25.311	89.874	22.514
F5	AC AC+DC	1		5.463			4.250	
		2	1.475 2.970	10.809	1.460 2.885	1.417 3.044	10.428	1.446 3.243
		4	5.708	24.750	5.505	8.405	22.682	9.033
		1	2.655 5.164	7.295	2.526 4.937	3.077 5.447	6.649	2.975 5.572
		2		12.762			14.178	
		4	8.939	15.455	8.872	10.426	59.427	10.974

LSB	ACAC+DC	1		6.125			8.679	
		2	3.529	20.56	3.416	3.388	21.52	3.643
		4	5.087	2	7.522	5	7.353	
			7.514	29.00	4.952	16.53	39.19	17.57
		1	6.162	8	7.242	6	1	4
			8.681	12.08	5.841	7.635	14.90	7.385
		2		8	8.483	12.88	6	10.45
				21.61		5	21.63	4
		4		1			9	
			9.679	0.614	9.061	43.41	0.000	36.66
						1		8

Table 6.1: MMD growth rates.

Table 6.1: MMD growth rates. The threshold is taken otherwise for every methodology, seetable a pair of for precise ranges. a awfully little variety of Na NMM Destimates was unnoticed within the computation of the smallest amount squares approximation. we tend to get NaN result if the add below the root of the MMD estimate is negative.

We observe that for threshold 1 and embedding in the DC coefficient as well P and B slices show similar behaviour, but as the threshold increases or the DC coefficient not use detect ability in B slices increases faster than it does in P slices. We can deduce that coefficients in Bslices are generally smallerthan coefficients in Pslices and higher concentrated linearly places in the scanorder. Since small larval uesand earlier places are preferable for hiding both slice types are equally well suited for embedding.

6.5LIMITATIONS:

Our video collection contains very uniform data, the same encoder was used togenerate all videos in it with identical parameters, the resolution is constant at 1024x576 (the standard DVD resolution) and DVDs area unit a top quality supply. of these conditions don't apply to real videos that area unit transmitted on-line, the noise would possibly build the detection of steganographic embedding tougher. We use the CAVL Centropy secret writing to avoid interference once ever-changing the quantal coefficients. In CABAC the modification of the values would possibly already cause interference, CABAC maintains a probabilistic model of the

coefficients to supply codes that are unit additional economical than CAVLC codes.

We look at the quantal DCT coefficients solely however different parts within the macro block are often used yet, as an example the motion vectors or ΔQP .

We simulate the embedding within the compressed videostream, instead we tend to might engraft whereas secret writing the video. this might avoid interference with any entropy code and conjointly minimize visual impacts in steganographic distortion is employed for predictions created by the encoder.

MMD growth rates. The threshold is taken otherwise for every methodology, see table 4 for precise ranges. a awfully little variety of Na NMM Destimates was unnoticed within the computation of the smallest amount squares approximation. we tend to get NaN result if the add below the root of the MMD estimate is negative.

We observe that for threshold 1 and embedding in the DC coefficient as well P and B slices show similar behaviour, but as the threshold increases or the DC coefficient not use detect ability in B slices increases faster than it does in P slices. We can deduce that coefficients in B slices are generally smaller than coefficients in P slices and higher concentrated linearly places in the scanorder. Since small larval ues and earlier places are preferable for hiding both slice types are equally well suited for embedding

CHAPTER-7

CONCLUSION

We have implemented several embedding mechanisms for h.264 video files, propose data set of features useful to detect if a video is modified and evaluated a benchmark on a large video collection started with ffmpeg version 0.8("Love") as code base to extract the proposed features and used a GUI program named Stegosaurus for exploratory data visualization and analysis. Stegosaurus manages feature vectors located in different files and can apply different normalization techniques to the input data which is an integer vector of occurrence counts. All Calculations are launched graphically and are GPU accelerated. Ffmpeg was used to create the video collection, taking x264 to transcode a set of DVDs.

Ø P and B slices were treated separately to which of both is better suited for embedding. We have used two different methods for benchmarking.

(ii) MMD that is said to the performance of associate degree SVM on the 2 computer file sets. The KL-D estimate was not computable in practice but the MMD estimate has shown to very stable and linear as expected[7]. We have implemented three different embedding methods, the (i) 1, (ii) F5 and LSB embeddings. For each method we have extracted features with different lower thresholds on the coefficient values as well as with using DC and AC coefficients or AC coefficients only for embedding. Also each extra action was done on chroma only, luma only or both. Across all embedding methods we have observed similar behaviour when varying the parameters or channels, we make the following conclusions:

1. The F5 method is the least detectable among the set tested. The LSB embedding is slightly less detectable than the ± 1 embedding.
2. Embedding in the chroma channel is most detectable. Using luma alone or luma and chroma leads to similar detectability where generally P slices are better suited for luma and chroma embedding while luma only embeddings are less detectable in B slices. Including the DC coefficient decreases detectability, especially in B slices since the essential part of their data is in DC coefficients. P and B slices

behave similarly allow threshold and if both AC and DC coefficients are used for embedding.

To our information no one has used the KL-D or MMD as detectability measures in video steganography before, most papers work on the idea of visual distortion. we've projected, enforced and tested a replacement feature set that uses each luma and color property data. In most papers concerning video or image steganography solely the luma channel is being employed, our experiments have however n that embedding in just luma doesn't succeed minimum discover ability in our options. we've used h.264 for our check s which savery common video for mat atthet time of writing, it's used on bluray disc swidely spread on the net, it's employed by YouTube10 and alternative video sharing websites. Our project uses terribly similar information, all video shared contagion movies, trans coded with identical encoder and same parameters. Real information retrieved on-line can contain loads additional noise introduced by completely different encoders, quality settings, cameras, video filters and shortly. We use the CAVLC entropy coder which is generally out-performed by CABAC, therefore most online media and bluraysuse CABAC. We use CAVLC to avoid interference with the CABAC entropy coder.

7.2 FUTUREWORK:

There is plenty of scope for future add the implementation, the practicality of the user interface is very restricted, we tend to used alternative computer code yet to judge the experiments, particularly a computer program to reckon the proper bpnc values for every extraction. we might wish Stegosaurusto calculate bpncs mechanically and let the user flick thru the results of previous computations. this is able to conjointly need amo restructured backend, like data} base storing all necessary information concerning the feature files or presumably the options directly which might build a lot of largest spark off eature stractable.

It would be fascinating to examine if associate actual detector confirms the gap meaurements we've got found. This detector will bean SV Mora easier classifier like associate averaged perceptron.

7.3 PERSONALREPORT:

we had ne'er detected of steganography before we have a tendency to the project description, neither had we have a tendency to ever worked within the field of

video writing before. However, the idea of concealment knowledge in video measured large in order that we started to scan books and on-line media concerning video compression and the H.264 codec. We have a tendency to try to implement 4:2:2 color areas up to 26411 get accustomed to the algorithms details further because the code layout, H.264 was the most important program we have a tendency to have changed at that time. It had been a new experience to set long term goal over months for a project that may take over one and a half years. It had been fascinating to examine scientific theory and machine learning, particularly as we have a tendency to have a group of application. The experiments disbursed during this project are on a distinct scale than experiments we've done before. Entombed soft computation demand and knowledge load in each input and output. We use two machines for the experiments, one having a strong CPU (AMD 8 core) for the feature extraction and the other one having a strong GPU (Nvidia GeForce GTX 59012) large amount of disk space (9TB RAID) for the MMD calculations. Both were running constantly for dozens of hours doing calculations or copying data. Planning the hardware required and setting up calculations on this scale was new to us. Now we look forward to write a scientific

CHAPTER-8

REFERENCES

[1].Jessica Fridrich. Steganography in Digital Media. Cambridge University Press,2010.

ISBN:978-0-521-19019-0.

.Richardson The H.264 Advanced Video Compression Standard. John Wiley a Sons Ltd, 2010. ISBN:978-0-470-51692-8.

Telecommunication and Standardization Sector of ITU.H.264.2009.

[3].Thomas M. Cover and Joy A. Thomas. Elements of Information Theory. 2nd ed. JohnWileyandSonsLtd,2006.ISBN:978-0-471-24195-9.

4[].Andreas Westfeld and Andreas Pfitzmann. 'Attacks on Steganographic Systems'. In: Information Hiding. Ed. by Andreas P fitz mann. Vol. 1768. Lecture

[5].Notes in Computer Science.10.1007/10719724_5. Springer Berlin/He idel berg, 2000,pp.6176.ISBN:978-3-540-67182-4.

URL:http://dx.doi.org/10.1007/10719724_5.

[6.]JessicaFridrichetal.Quantativesteganalysisofdigitalimages:estimatingthesecret message length.2003

[7.]T. Pevný and J. Fridrich. 'Benchmarking for Steganography'. In: Information Hid- ing, 10th International Workshop. Ed. by K. Solanki. Lecture Notes in Computer Science. Santa Barbara, CA: Springer-Verlag, New York, 2008.

[8.]TomášPevný.'KernelMethods inSteganalysis'.PhDthesis.BinghamtonUniver- sity, SUNY,2008.

⁹Which has been accomplished by the developers at the time we finish this report.

¹⁰<http://www.geforce.co.uk/hardware/desktop-gpus/geforce-gtx-590>

[11].AndrewD.Keretal.'TheSquareRootLawofSteganographicCapacity'.In:10thAC M Workshop on Multimedia and Security. 2008, pp.107–116.

[12.]JasonSanders and Edward Kandrot. CUDA by Example. Morgan KaufmannPub- lishers,2011.

[13.]RobFarber. CUDA Application Design Development. Morgan Kaufmann Publish- ers,2011.

[14.]DaviB Kirk and Wen mei W. Hwu. Programming Massively Parallel Processors. Morgan Kaufmann Publition.

IMAGE STEGANOGRAPHY

ORIGINALITY REPORT

6% SIMILARITY INDEX	5% INTERNET SOURCES	4% PUBLICATIONS	2% STUDENT PAPERS
-------------------------------	-------------------------------	---------------------------	-----------------------------

PRIMARY SOURCES

1	www.coursehero.com Internet Source	2%
2	www.cs.ox.ac.uk Internet Source	2%
3	Ahmed A. Abd EL-Latif, Bassem Abd-El-Atty, Salvador E. Venegas-Andraca. "A novel image steganography technique based on quantum substitution boxes", Optics & Laser Technology, 2019 Publication	1%
4	mafiadoc.com Internet Source	<1%
5	www.ijrte.org Internet Source	<1%
6	Submitted to Engineers Australia Student Paper	<1%
7	Submitted to Woodcroft College Student Paper	<1%
8	ijerd.com Internet Source	
		<1%
9	vibdoc.com Internet Source	<1%
10	Submitted to Multimedia University Student Paper	<1%
11	infoscience.epfl.ch Internet Source	<1%
12	hdl.handle.net Internet Source	<1%
13	www.slideshare.net Internet Source	<1%