Digital Video Forensics:

Detecting MPEG-2 Video Tampering through Motion Errors

Ho Hee-Meng

Technical Report

RHUL–MA–2013– 5

01 May 2013

Royal Holloway
University of London

Information Security Group
Royal Holloway, University of London
Egham, Surrey TW20 0EX,
United Kingdom

www.ma.rhul.ac.uk/tech

Name: Hee-Meng, Ho

Digital Video Forensics:

Detecting MPEG-2 Video Tampering through Motion Errors

Supervisor: Dr Stephen Wolthusen

Submitted as part of the requirements for the award of the MSc in Information Security at Royal Holloway, University of London.

I declare that this assignment is all my own work and that I have acknowledged all quotations from the published or unpublished works of other people. I declare that I have also read the statements on plagiarism in Section 1 of the Regulations Governing Examination and Assessment Offences and in accordance with it I submit this project report as my own work.

Signature: Date:

Contents

Acknowledgements

First and foremost, I would like to thank my supervisor, Dr. Stephen Wolthusen for his readiness in guiding me in this project. I sincerely appreciate his time, effort and above all patience in supervising me. I would also like to thank my ex-colleague and friend, Lim Eyung for helping me out and of course not forgetting my dearest friend, Dr. Khoo Swee-Chern for his extremely generous time and energy in imparting his invaluable knowledge without which I would be struggling even more.

List of Abbreviations & Acronyms

AVC	Advanced Video Coding
CCTV	Closed Circuit Television
CD	Compact Disc
DCT	Discrete Cosine Transform
DFT	Discrete Fourier Transform
FFT	Fast Fourier Transform
fps	Frames per second
GOP	Group of Pictures
HEVC	High Efficiency Video Coding
IEC	International Electro-technical Commission
ISO	International Standard Organisation
JPEG	Joint Picture Experts Group
MPEG	Moving Picture Experts Group
VCD	Video Compact Disc

Executive Summary

This report looks at the study of MPEG-2 video forensics. Specifically, it looks at whether the tampering of an MPEG-2 video can be detected by a popular research methodology if an attacker was to edit the video using one of the (less user-friendly) ways to tamper an MPEG-2 video.

This report is divided into 8 chapters. Chapter 1 (Introduction) looks at the motivation on the study of digital video forensics. Chapter 2 (Background) will introduce important key concepts in understanding images and videos, especially how videos are encoded according to the MPEG-2 standard. In addition, Chapter 2 (Background) will also cover a brief description of the relevant MPEG standards. Chapter 3 (Related Works) will cover the general trends and challenges in digital image and video forensics, followed by a more in-depth discussion on the recent research approaches in MPEG-2 video forensics.

Having covered the relevant background and research in digital video forensics in Chapter 2 (Background) and Chapter 3 (Related Works) respectively, Chapter 4 (Problem Statement) will discuss the aims and objectives of this report as well as to state the research question. Chapter 5 (Design of Experiments) will then discuss how the experiments will be setup and designed in order to validate the research question asked in Chapter 4 (Problem Statement).

The results of the experiments will be available in Chapter 6 (Results of Experiments) and will be analysed in Chapter 7 (Discussion). Finally, Chapter 8 (Conclusion & Summary) will summarise the findings of the experiments and conclude whether the research question in this report has been answered satisfactorily or not, as well as providing a short overview on other possible detection and counter-forensics techniques to the research question in this report. Finally, it ends with a short discussion of future areas of research and improvement.

Chapter 1: Introduction

The widespread availability of the Internet, coupled with the easy availability of video and image capturing devices, such as low-price cameras, digital camcorders and CCTVs and cheaper editing software, meant that it is getting easier for ordinary people with nefarious aims to have access to digital doctoring tools in order to modify images and videos [AR11]. This in turn means that the images and videos that are seen in mass media such as television, popular Internet websites such as YouTube, may have been tampered and the adage “a picture speaks a thousand words” while still holding true – may now have a hidden and subverted meaning, i.e., their authenticity can no longer always be taken for granted [WW06].

Yet, at the same time, organisations and countries continue to rely heavily on the authenticity of images and videos. For example, an increasing number of governments and organisations are employing the use of CCTVs because these devices are not only getting cheaper but also are being heavily relied on as useful (and perhaps even necessary) tools in surveillance. A report by Urbaneye Project published in June 2002 estimated that there were possibly 500,000 CCTVs in London [MM02]. The Urbaneye Project was co-ordinated by the Centre of Technology and Society at the Technical University of Berlin and was supported by the European Commission as part of the Key Action "Improving the Socio-Economic Knowledge Base" within the 5th Framework Programme. Other countries too, such as European countries like France, Italy, etc. [LH04] and even Asian countries such as Singapore [SB12] are also using CCTVs to prevent and detect violence and theft.

Given that images and videos from cameras, digital camcorders and CCTVs can serve as very powerful “evidences” in both legal courts and public opinion, it is important to therefore ask whether the images and videos produced by these devices are truly authentic and has not been tampered with. In addition, given the easy availability of digital doctoring tools, many accused suspects in an incriminating video would often claim that the videos showing themselves are being tampered. For example, in March 2011, an opposition leader in Malaysia, was accused of being involved in a sex video [KO12]. Obviously, for political reasons, if the video was found to be genuine, the politician’s career would be at stake whereas if the video was found to be tampered, then the current ruling party in the Malaysian government would be accused of a political conspiracy to destroy the opposition. Therefore, it is important for both parties to be assured of the authenticity of the video. This suggests that while videos are widely popular and are often quoted as “evidences”, the authenticity of a video cannot be automatically assumed and in some cases (such as mentioned in the case of the Malaysian politician), its authenticity may not be so easily verified.

To a layman, the study of digital video forensics could conjure 2 possible interpretation of video “forensics”. One is the study of the subjects in the video to help in forensics investigation. For example, a criminal suspect was seen in the video that recorded a scene where a crime was committed. The appearance of the suspect in the video forms part of a forensics chain of evidence. The other possible interpretation of video “forensics” is the determination of whether a video is authentic or not , i.e., has it been tampered and if so, how and where? In this report, digital video forensics research aims to uncover and analyse the underlying facts about a video and seeks to detect any form of tampering that could have been applied to a video [AR11].

A video at its core is simply a collection of moving images. As such, it should not be surprising that some of the research on digital video forensics is an extension of the research from digital image forensics. Considering that digital images were much more readily produced and available historically compared to digital videos, the research in digital image forensics is considerably more mature than digital video forensics. For example, the research by [WW09] on the detection of video tampering through double quantisation was built on the research by [AP04, JH06] on the detection of doctored JPEG images.

As highlighted by [MS12], in order to ensure that videos can be stored and transmitted in an efficient manner, virtually all digital video undergoes compression during storage or transmission. Today, the MPEG video compression standards are regarded as the most popular video compression format. Currently, the MPEG-2 video standard is considered to be the most popular video compression standard among the various MPEG video standards such as MPEG-1, MPEG-2 and MPEG-4.

Due to the release of low-cost MPEG-4 hardware encoders, the MPEG-4 video standard is expected to gain more popularity in time to come. At the same time, the MPEG committee is working to develop a newer video compression standard, namely, the HEVC (High Efficiency Video Coding) standard, which is also known as H.265 or MPEG-H Part 2 [WK12b]. However, this report will only focus on the detection of tampered MPEG-2 videos.

In this chapter, this report introduces the motivation for studying digital video forensics. The following chapter, Chapter 2 (Background) will cover the necessary key concepts that are needed in order to understand MPEG-2 video forensics.

Chapter 2: Background

As explained in the previous chapter, in this chapter, the report will introduce certain key concepts in understanding images and videos and how videos are encoded according to the MPEG-2 standard. In addition, this chapter will also cover a brief description of the relevant MPEG standards.

The MPEG-2 standard makes use of the JPEG image compression standard in encoding some of its frames. As such, the study of MPEG-2 usually begin with the study of the JPEG compression standard since some of the research in MPEG-2 video forensics focuses on the properties of the JPEG compression and how it is affected when a video is tampered.

2.1 JPEG compression standard

According to [JM99], the JPEG compression standard (referred to as “JPEG”) is the most commonly used format for storing images. The standard is fairly complex since instead of defining just an image file format, it actually defines a number of related image compression techniques. The power of JPEG comes in that it is able to achieve the most aggressive compression of any bitmap format in common use. All the JPEG compression standards in general are lossy (refer to [Chapter 1 of JM99] to understand the difference between lossy and lossless compression).

The original JPEG standard defined 4 compression standards, namely hierarchical, progressive, sequential and lossless (refer to [Chapter 4 of JM99] for more information on these standards). In general, most JPEG images are encoded using the plain vanilla sequential compression mode.

As highlighted in [JM99], the JPEG encoding typically consists of the following steps:

- Sampling – in this phase, the pixel data is converted from the RGB to the YCbCr colourspace and down sampling is performed. The RGB (Red-Green-Blue) and YCbCr are popular colour models that are used to represent colours in images (refer to [Chapter 1 of JM99] for more information on colour models).

- DCT (Discrete Cosine Transform) – in this phase, the JPEG images are compressed into 8 by 8 pixels blocks called data units. The DCT converts the data unit values into a sum of cosine functions. The DCT is the heart of the JPEG compression. Using the DCT coefficients in a 2-dimension (horizontal and vertical), an image can be easily described succinctly. The higher-order DCT coefficients do not contribute much in describing the image and as such, can be discarded while allowing the decoded JPEG image to resemble very closely (to the human eye) to the original image. As such, compression can be achieved (refer to [Chapter 7 of JM99] for more information on the DCT).

- Quantisation – in this phase, the DCT coefficients that are not essential for recreating a close approximation of the original image is removed. This phase is mainly responsible for making JPEG a lossy compression standard.

- Huffman Coding – in this phase, the quantised DCT coefficients are encoded where runs of zero values (due to the quantisation phase) can be eliminated.

Figure 2-1 below shows an overview of the JPEG encoding:

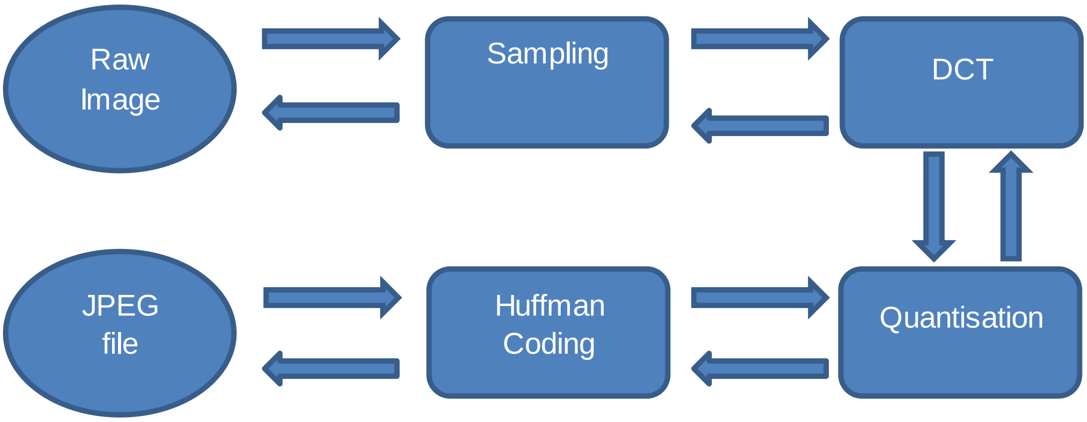


Figure 2-1

The understanding of the JPEG compression in this section are needed in order to understand some of the research presented in Chapter 3 (Related Works), as the tampering of video may also change the properties of the JPEG images in an MPEG-2 video. However, before going into how MPEG-2 videos are encoded, the next section will first provide a brief overview on the evolution of the MPEG video compression standards as well as to describe other key parts of the MPEG standards such as MPEG-4, MPEG-H Part 2 (HEVC), etc.

2.2 Overview of the MPEG video compression standards

According to [LS04], the MPEG video compression standards were set by the Motion Picture Experts Group (MPEG), a committee that comes under the joint control of the International Standard Organisation (ISO) and International Electro-technical Commission (IEC). The IEC handles the international standardisation for electrical and electronic technologies while the ISO handles everything else. The MPEG first had their meeting in 1988 and has since published several MPEG standards that were widely adopted, such as the MPEG-1, MPEG-2, MPEG-4 video compression standards and the MP3 audio compression standard.

According to [JW01], the MPEG standards were novel in that it is not the encoder that was standardised but instead the standards define how the decoder should interpret the encoded MPEG bit streams. A compliant decoder must be able to correctly interpret every allowable bit stream whereas an encoder which produces a restricted subset of the possible codes can still be compliant.

[JW01] highlighted that the first compression standard by the MPEG was the MPEG-1 video compression standard which was basically designed to allow moving pictures and sound to be encoded into the bit rate of an audio Compact Disc (CD) – which resulted in the Video-CD (VCD). According to [LS04], the MPEG-1 essentially uses techniques such as DCT, coefficient quantisation and variable length encoding as well as techniques such as motion compensation for temporal compression. Many of these techniques were taken from the JPEG compression standard (see previous section) and in turn, many of the subsequent MPEG standards use techniques that are described in MPEG-1.

The MPEG-2 video compression standard added video interlacing capability (refer to [Chapter 1 of JW01] for more information on interlacing) as well as expanding the range of picture sizes and bit rates in order to accommodate higher quality videos. Since the MPEG-2 is an extension of MPEG-1, an MPEG-2 compliant decoder can easily read MPEG-1 videos. The MPEG-2 was so successful in handling high quality video compression that the MPEG-3 standard which was originally developed for high-definition television was soon discontinued after it was discovered that MPEG-2, at a high data rate, was able to accommodate high-definition television [LS04]. A more thorough description of the MPEG-2 standard will be covered in the next section, Section 2.3 (MPEG-2 encoding).

MPEG-4 introduces a number of new encoding tools in addition to the techniques used in MPEG-1 and MPEG-2, such as object coding, mesh coding, still picture coding and face and body animation. One of the most important concepts introduced in MPEG-4 is the concept of objects. In MPEG-1 and MPEG-2, a scene is encoded in its entirety. However, in MPEG-4, different parts of the scene can be coded and transmitted separately as video and audio objects. In other words, it is possible to selectively code a foreground object separately from the scene's background. An example quoted in [LS04] is a football match where the game was processed to separate the ball from the rest of the scene and the background (the scene without a ball) was transmitted as a "teaser" to attract a pay-per-view audience. More information on the coding tools of MPEG-4 is available in [JW01]

The MPEG-4 Part 10 which is also known as MPEG-4 AVC (Advanced Video Coding) was subsequently adopted as the H.264 video standard and is best known for its adoption as one of the supported video codes for the Blu-ray discs. Currently, it is also widely used by streaming Internet websites such as YouTube, Vimeo, etc., and is generally considered as the next replacement of the MPEG-2 video standard. However, the hardware H.264 decoders are only starting to become cheaper so at the moment MPEG-2 video standards is still considered as the most popular video compression standard although it is expected that in time to come, the H.264 standards will eventually become dominant.

According to [LS04], due to lack of agreements between the MPEG committee members in the naming notation, the subsequent MPEG standard to be developed after MPEG-4 was named as MPEG-7. However, the MPEG-7 standard is not about compression but rather about the metadata of videos. MPEG-7 is formally known as "Multimedia Content Descriptor Interface" and is concerned about how to add metadata to a video.

In addition to MPEG-7, another new standard which is not related to video compression, known as MPEG-21 was also developed. MPEG-21 seeks to create a framework or structure to manage the digital assets for video. Its mission statement is "to enable transparent and augmented use of multimedia resources across a wide range of networks and devices" [LS04]. There are also a number of subsequent MPEG standards on multimedia technologies, such as MPEG-A which looks at multimedia application format, etc. However, these standards are not relevant to digital video forensics and as such, will not be covered in this report. More information on these standards is available at [WK12c].

The next generation video compression standard after H.264 is called HEVC (High Efficiency Coding) and is also known as H.265 or MPEG-H Part 2. According to [WK12b], the HEVC aims to substantially improve the coding efficiency such as reducing the bit rate requirements while maintaining comparable image quality. In addition, HEVC replaces the macroblocks structure used in previous MPEG compression with a flexible scheme based on coding units, variable size structures which sub-partition the picture into rectangular regions. The HEVC is expected to be ratified as a standard in 2013.

Having provided a brief overview of the history and development of the MPEG video compression standards, the next section will cover the basics of the MPEG-2 encoding.

2.3 MPEG-2 encoding

According to [JW01], video signals exist in 4 dimensions, namely the attributes of the pixels, the horizontal and vertical spatial axes of the image and the time (or temporal) axis. Compression can be applied in any or all of these 4 dimensions. When an image is compressed without reference to any other images, the time axis naturally does not come into play and therefore, this type of compression is referred to as “intra-coded” (intra meaning within) compression. Alternatively, it can also be referred to as “spatial coding”. However, given that in a video, the scene change relatively slowly, an even greater compression can be obtained if the compression takes into account the redundancy from one image to another. This would involve the time axis and therefore is referred to as “inter-coded” (inter meaning between) compression or “temporal coding” (refer to [Chapter 1 of JW01] for more information on intra and inter-coding). MPEG-2 involves both inter and intra-coding. When temporal compression is used, the current picture/image is not sent in its entirety. Rather, the difference between the current picture and the previous one is sent.

In an MPEG-2 encoded video sequence, there exists 3 types of frames, namely the intra (or known as the I-frame), predictive (known as the P-frame) and the bi-directional (known as B-frame) frames. Each of these frames offer different degrees of compression. The frames are also arranged in a frame pattern sequence which repeats periodically. For example, a possible frame pattern sequence is as follows:

“**I_1 B_2 B_3 P_4 B_5 B_6 P_7 B_8 B_9 I_{10} B_{11} B_{12}**…” where the subscript denotes the time axis. In this frame sequence, a GOP (Group of Picture) refers to the frames starting from an I-frame to the last frame (it can be B or P-frames) before another I-frame. So, the frame pattern sequence of the just mentioned GOP is “I BB P BB P BB”. It can be seen that there are a total of 9 frames and 2 of these are P-frames. In MPEG-2 terminology, the number of frames within a GOP is commonly denoted with the variable, N. The number of P-frames within a GOP is commonly denoted with the variable M. Hence, the just mentioned GOP of “I BB P BB P BB” is sometimes known as a GOP with M = 2 and N = 9.

The I-frame is encoded without reference to any other frames in a GOP. The P-frames are encoded with respect to a previous I or P-frame while the B-frames are encoded with respect to the previous and next (and hence known as bi-directional) I or P-frames.

The I-frame is the anchor or reference frame for the other B and P-frames. The I-frame is encoded using the standard JPEG compression mode – see previous Section 2.1 (JPEG compression standard) on the JPEG compression standard. As the I-frame is an independent JPEG compressed picture, compression is only achieved at the spatial level and this is why the I-frame is referred to as an intra-coded frame. Compared to the other B or P-frames, it is the highest quality frame that offers the least amount of compression.

The B and P-frames are intended to exploit the temporal redundancies across frames and are known as “inter-coded” frames. Both of them offer a better compression rate than the I-frame. The B and P-frames employ motion estimation to capture the differences between the B/P-frame and another previous and/or next I or P-frame. The MPEG-2 encoder contains a motion estimator that computes the direction and distance of the motion between the pictures and captures this as the motion vector. However, in real images, objects do not necessarily maintain their appearance as they move. For example, the objects may move into shade or light. As such, the encoder needs to compensate for this difference by sending another variable known as the motion error. When decoding an MPEG-2 B or P-frame, both the motion vectors and the motion errors are applied together to create the final picture.

Figures 2-2a, 2-2b and 2-2c below show how the motion estimation using the motion vectors and motion errors are applied in a B or P-frame. The pictures are reproduced from [pp. 2 of WW06].

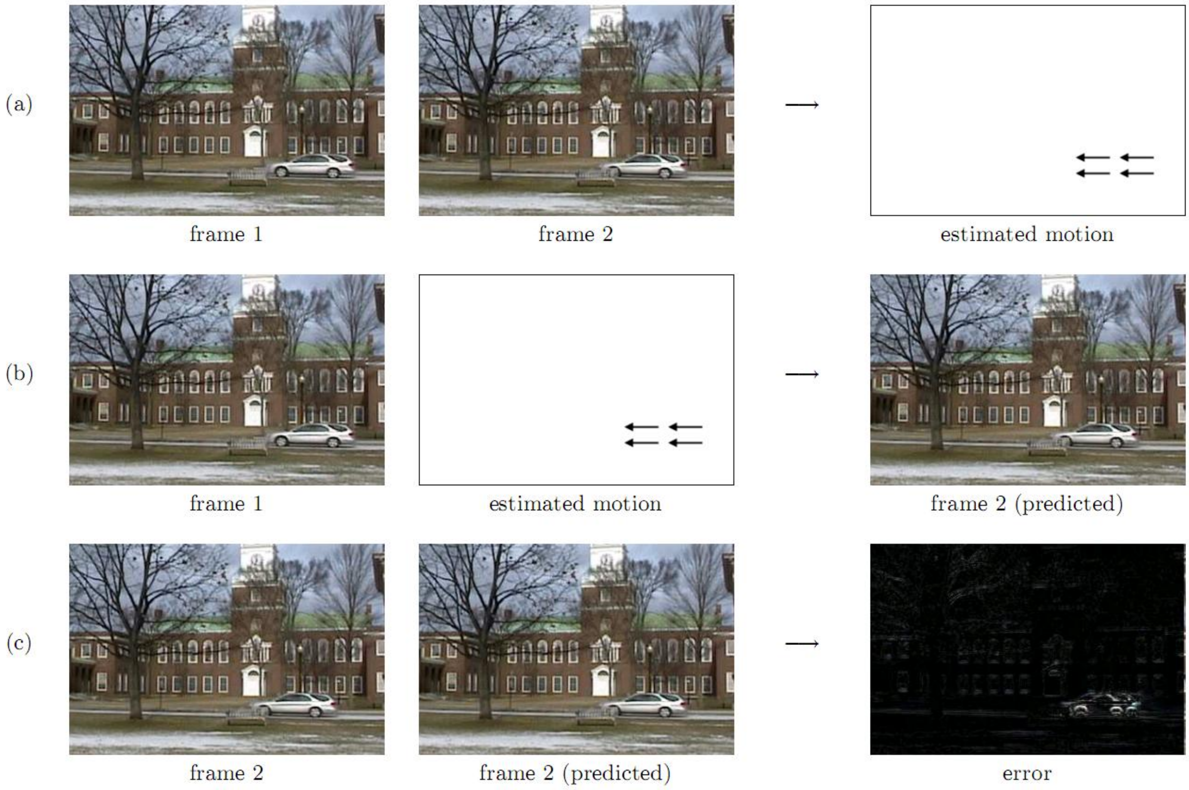


Figure 2-2

Firstly, as can be seen in Figure 2-2a above, the MPEG encoder will compute the estimated motion between 2 frames, namely frame #1 and frame #2. The output is the motion vector (indicated with the 4 arrows in Figure 2-2a). Hence, by applying the estimated motion (containing the motion vectors), the decoder would be able to generate a predicted frame #2 as seen in Figure 2-2b. However, as mentioned earlier, the appearance of the objects (e.g. due to lighting conditions) would change when the objects move. Hence, if the decoder just strictly applies the motion vector on frame #1, the output would possibly be a blurry or "blocky" image. In order to compensate for this, the encoder must also include a motion error together with the motion vector in its encoding. As can be seen in Figure 2-2c, the motion error is obtained by comparing the difference between the expected frame #2 (available in the original pre-encoded video) and the predicted frame #2 (after applying the motion vector on frame #1). Therefore, when a B or P-frame is encoded, both the motion vector and motion error are included as part of the MPEG-2 encoded bit stream.

In practice, a single motion vector is not sufficient to accurately capture motion in most natural video (there could be many objects moving at the same time). As such, the motion estimation is applied at a smaller level. In MPEG-2, a 16 by 16 pixel is defined as a macroblock and the motion estimation is applied at the macroblock level.

The B-frame is almost similar to the P-frame in that both frames employ motion estimation (unlike the I-frame which is an independent JPEG encoded picture). However, the B-frame has an even higher compression rate compared to the P-frame. This is due to the fact that the B-frame relies on a past, future and both of its neighbouring I or P-frame for motion estimation. This allows the B-frame to achieve a higher compression rate. As such, the B-frame is also known as a bi-directionally predictive frame while the P-frame is known as a forward predictive frame. More details regarding the MPEG-2 encoding scheme are available in [WW06] and [JW01].

This concludes the relevant background material for understanding the fundamentals of images and videos as well as relevant concepts in MPEG-2 encoding. The next chapter, Chapter 3 (Related Works) will first cover the general research trends and challenges in digital image and video forensics followed by a discussion on the current research approaches in MPEG-2 video forensics.

Chapter 3: Related Works

The previous chapter, Chapter 2 (Background) covered certain important key concepts that aided the understanding of how images and videos are encoded. It also covered a brief description of the relevant MPEG standards. As highlighted in Chapter 1 (Introduction), given the need to preserve storage space and transmission bandwidth, most videos that are produced are compressed in one form or another using the most popular video compression standard which is MPEG-2. As such, this report focuses on the detection of MPEG-2 videos that are being tampered, specifically on MPEG-2 videos that are deleted and cloned/duplicated.

In this chapter, this report will first cover the research on the general trends and challenges facing digital image and video forensics. It will then look at the more specific trends and latest research in MPEG-2 video forensics. Following that, the next chapter, Chapter 4 (Problem Statement) will then cover the aims and objectives of this report as well as introduce the key research question that this report will cover.

3.1 Trends and Challenges in Digital Image and Video Forensics

A recent paper published in 2011 by [AR11] covered a survey of the current trends and challenges in digital image and video forensics. The paper started off by describing the motivation of attackers in wanting to tamper digital images and videos – much like what was explained in Chapter 1 (Introduction). It then went on to define digital image and video forensics research, which is to uncover and analyse underlying facts about an image or video and that the main objectives of the research is to detect tampering, hidden data as well as source identification of the images and videos [see pp. 2 of AR11]. The report then went on to describe the following common image manipulation techniques [see pp. 5 to 7 of AR11], such as:

- Composition or Splicing – the composition or merging of an image using one or more parts of the images.
- Retouching, Healing and Cloning – the alteration of parts of an image or video using parts or properties of the same image or video.
- Content Embedding or Steganography – conveying a secret message through a cover media without affecting the cover's statistical properties.

Subsequently, [AR11] went on to highlight that most techniques for digital image and video forensics are blind and passive, i.e., blind because the techniques does not use the original content for analysis and passive because it does not require any digital watermarking. Digital watermarking solutions require a form of active/direct implementation (i.e., watermark) directly into the acquisition sensor. [AR11] further argued that the current image and video forensics approaches in literature are broken up into the following 3 categories [see pp. 8 of AR11]:

- Category 1 – Camera sensor fingerprinting or source identification.
- Category 2 – Image and video tampering detection.
- Category 3 – image and video hidden content detection/recovery.

As explained earlier in this chapter, this report will focus on techniques related to Category 2, which is the detection of video tampering. As such, this chapter will just cover a brief description of the other 2 categories as reported in [AR11].

In Category 1, [AR11] described the following 2 categories of source camera identification [see pp. 8 to 19 of AR11]:

- Device Class Identification – the goal of the research here is to be able to identify the model and/or manufacture of the device producing the image in question.
- Specific Device Identification – the goal of the research here is to be able to identify the exact device that produced the image in question.

In Category 2, [AR11] explained that the current research approach for detecting image and video tampering relies on analysing certain properties of the image and video in question, e.g., detection of cloned region, analysis of feature variations (using the original and tampered image/video), inconsistencies in features, inconsistencies regarding the acquisition process or even structural inconsistencies present in the targeted attacks. [AR11] went on to present an overview of the following approaches in detecting tampering of image/video [see pp. 19 to 26 of AR11]:

- Image Cloning Detection – Cloning is one of the simplest and most common types of forgeries and is also known as the copy-and-paste approach. Often the main objective of cloning is to make certain object from a scene “disappear” using the properties of the same scene (e.g., neighbouring pixels, etc.). Several research papers [JF03 and JH06], relies on the use of the calculation of the DCT (Discrete Cosine Transform) of region of interests in the image to detect cloning. As explained in Chapter 2 (Background), in image encoding, DCT is used to separate an image into parts of differing importance with respect to their image quality.

- Video Splicing and Cloning Detection – Similar to image cloning detection, video splicing and cloning detection seeks to detect cloning in videos. There are several techniques proposed in this area of research. [WW07] relies on using computational algorithms to compare the correlation coefficients of various frames (inter-frame) and regions of interests within a frame (intra-frame) to detect duplicated frames and regions respectively. [WW06] relies on using statistical analysis of the FFT (Fast Fourier Transform) graphs of mean motion errors of P-frames of an MPEG video file to detect temporal artefacts of videos that have undergone a double compression (i.e., the encoded video was decoded and has undergone some form of editing such as deletion of frames before it is being re-encoded again) while [WW09] relies on the non-uniform distribution patterns of doubly quantised coefficients of I-frames in an MPEG file to detect the spatial artefacts of videos that have undergone a double compression. This report will focus on research in this area, i.e., the detection of video that are tampered through cloning/duplication.
- Variations in Image Features – This approach focuses on the fact that image doctoring typically involve multiple steps, which often demand a sequence of elementary image processing operations, such as rotation, scaling, smoothing, etc. As such, the research in this area focuses on techniques to detect such operations.
- Inconsistencies in Image Features – This approach focuses on the fact that when 2 images are spliced (i.e., merging regions of an image) to create a new composite image, the image has to be re-sampled which then introduces specific correlations that when detected, may mean that some form of tampering has been performed. For example, [AP05] described how re-sampling (e.g., scaling, rotation, etc.) causes statistical correlations in the Fourier transform domain which allow the detection of tampered uncompressed TIFF images and JPEG and GIF images with minimal compression.
- Lighting inconsistencies – This approach seeks to detect tampering by analysing lighting inconsistencies (e.g., direction of light, light reflection/refraction, etc.) in images in order to reveal traces of digital tampering. For example, [MJ05] analysed the direction of lighting sources to see if tampering through lighting inconsistencies can be detected in tampered images.
- Acquisition Inconsistencies – This approach looks at the camera response normality and response functions to detect tampered images. Cameras from different manufacturers provide different response functions and as such, images produced by these cameras

have a unique “fingerprint” that allows identification of images. These “fingerprints” then become distorted when the images are tampered and the tampering can then be detected.

- JPEG inconsistencies – This approach focuses on the creation of JPEG images and the associated JPEG properties. For example, when a JPEG image is doubly compressed (i.e., it is being edited and re-saved or re-compressed as a new JPEG image), certain artefacts will be present whereas such artefacts are not present in an original JPEG image which has undergone only a single compression [AP04].

In Category 3, [AR11] went on to describe how hidden image and video can be detected or recovered, i.e., forensics steganalysis. [AR11] described the following 3 categories of steganalysis [see pp. 27 to 36 of AR11]:

- Targeted Steganalysis – In this approach, the forensics investigator has certain ideas on the algorithms used to provide stenography and as such, the detection techniques proposed makes use of certain assumptions regarding the algorithms. Such techniques tend to be more accurate in detecting a targeted stenographic algorithm but it cannot be used against other algorithms.
- Blind Steganalysis – In this approach, the forensics investigator does not makes any assumptions regarding the algorithms in used and as such, the detection techniques proposed have to be able to detect any kind of stenographic algorithms that are being employed.
- Stegi@Work – the research in this area focuses on providing a scalable framework that is able to process a large volume of images, such as in photo-sharing websites such as Flickr and Picasa.

[AR11] then ended off by making several recommendations [see pp. 37 of AR11], such as the need to evaluate existing and new algorithms as the authors felt that nearly all the surveyed papers lack the rigor of other areas of digital image processing and computer vision. In addition, they also highlighted the challenge of fusing the various techniques researched in order to be able to yield more accurate results, especially when one does not precisely know what one is looking for in a tampered image or video. More troubling, [AR11] found that there are very few papers on counter-forensics (even on the supposedly more mature field of digital image forensics) techniques and as such, they are not confident on the robustness of the existing detection techniques when subjected to the tinkering of a clever manipulator.

In conclusion, [AR11] provided a very recent survey on the trends and challenges in digital image and video forensics. As can be seen from the above, most of the research tends to focus on digital image forensics. As mentioned previously in Chapter 1 (Introduction), this should not be surprising since a video is essentially a collection of images and due to incremental progress in processing and storage capacity, images were much readily produced and available historically than video. It can be seen however, that some of the techniques used in digital video forensics were a result of earlier research in digital image forensics. For example, the investigation on the statistical properties of videos that are doubly compressed [WW09] were inspired by the research on the statistical properties of JPEG images when they are doubly compressed [AP04, JH06]. In a way, the lack of research in digital video forensics as compared to digital image forensics presents a lot of opportunities to a researcher focusing on video forensics.

Following the discussion on the general trends and challenges on digital image and video forensics, the next section of this chapter will look at the findings of recent research on the tampering of MPEG-2 videos.

3.2 Current Research in Digital Video Forensics for MPEG-2

In general, when an attacker tries to tamper an MPEG-2 video, the attacker would typically try to tamper the video through the following manner:

- Using a video editor to decode the MPEG-2 encoded file and using the editor to edit the frames and then re-encode the modified video as a new MPEG-2 encoded file – this technique is known as double compression since the MPEG-2 videos are compressed twice (the first time during the original compression and the second time during the post-editing and re-encoding), OR,
- Directly editing the information stored in the MPEG-2 encoded bit stream, i.e., .mpeg file, without going through an intermediate video editor.

The first method mentioned above is typically the tampering technique used by an attacker because it allows the attacker to decode the MPEG-2 video to see what the frames look like before editing it in a manner that they want. The second method is seldom used because the attacker is unable to visualise how the video frames look like since the MPEG-2 file is in an encoded and compressed form.

As such, recent research in the MPEG-2 video forensics tends to focus on detecting tampering using the first method. Some of the research rely on the fact that when a MPEG-2 video (which is already compressed once) is subjected to a second compression (after it is decoded, edited and re-encoded again), there will be changes to the statistical properties of the re-encoded MPEG-2 video.

In [WW09], a technique based on analysing the spatial statistical properties of a doubly compressed video was proposed. In general, the research relies on observing the statistical distribution of the quantised DCT coefficients of a doubly quantised I-frame when the MPEG-2 video was doubly compressed. Figure 3-2-1a below shows the distribution of singly quantised coefficients with the quantisation level set to 5 while Figure 3-2-1b below shows the distribution of doubly quantised coefficients with the first quantisation level set to 5, followed by the second quantisation level which is set to 3. Both figures are reproduced from [see pp. 3 of WW09]. As can be seen in Figure 3-2-1b, an artefact is introduced when the step increases between quantisations. The artefacts due to double compression will be even more pronounced when the first quantisation level is greater than the second quantisation level. This technique allows a forensics investigator to detect whether an MPEG-2 video has been tampered or not. It is even capable of detecting localised tampering in regions of frames that are as small as 16 by 16 pixels. However, the limitation of this technique is that it is only effective when the second quantisation level is different from the first quantisation level.

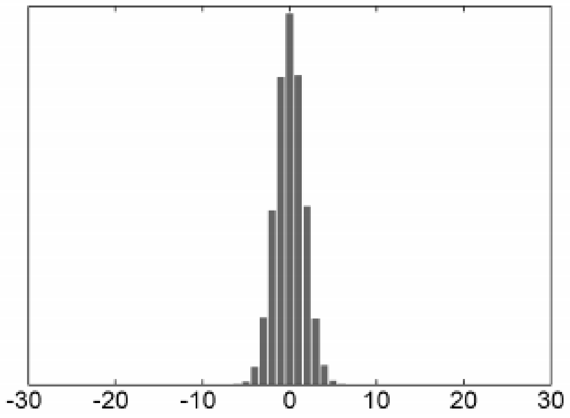


Figure 3-2-1a

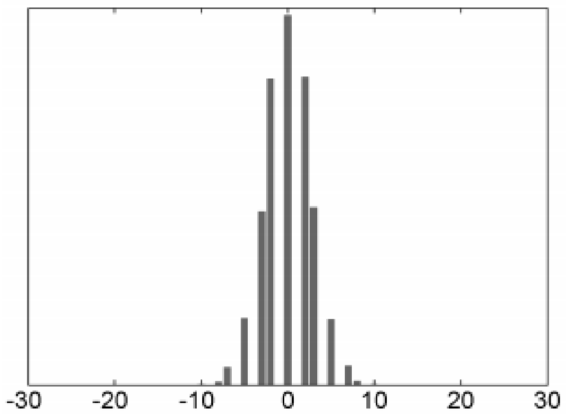


Figure 3-2-1b

[WW06] observed that a doubly compressed MPEG-2 video will also introduce temporal statistical effects. More specifically, the FFT graphs of the mean motion errors of P-frames for a doubly compressed video will show spikes when the frames are deleted. Please refer to Chapter 2 (Background) on the types of frames in an MPEG-2 video.

Figure 3-3-2a below shows the mean motion errors of the P-frames (upper half of a graph) and its accompanying FFT (lower half of a graph) for a video that has undergone a double compression. There are in total 12 graphs. The number on the top left hand corner of each graph showed the number of frames that has been deleted. Spikes in the graphs indicate double compression. The figure is reproduced from [pp. 9 of WW06].

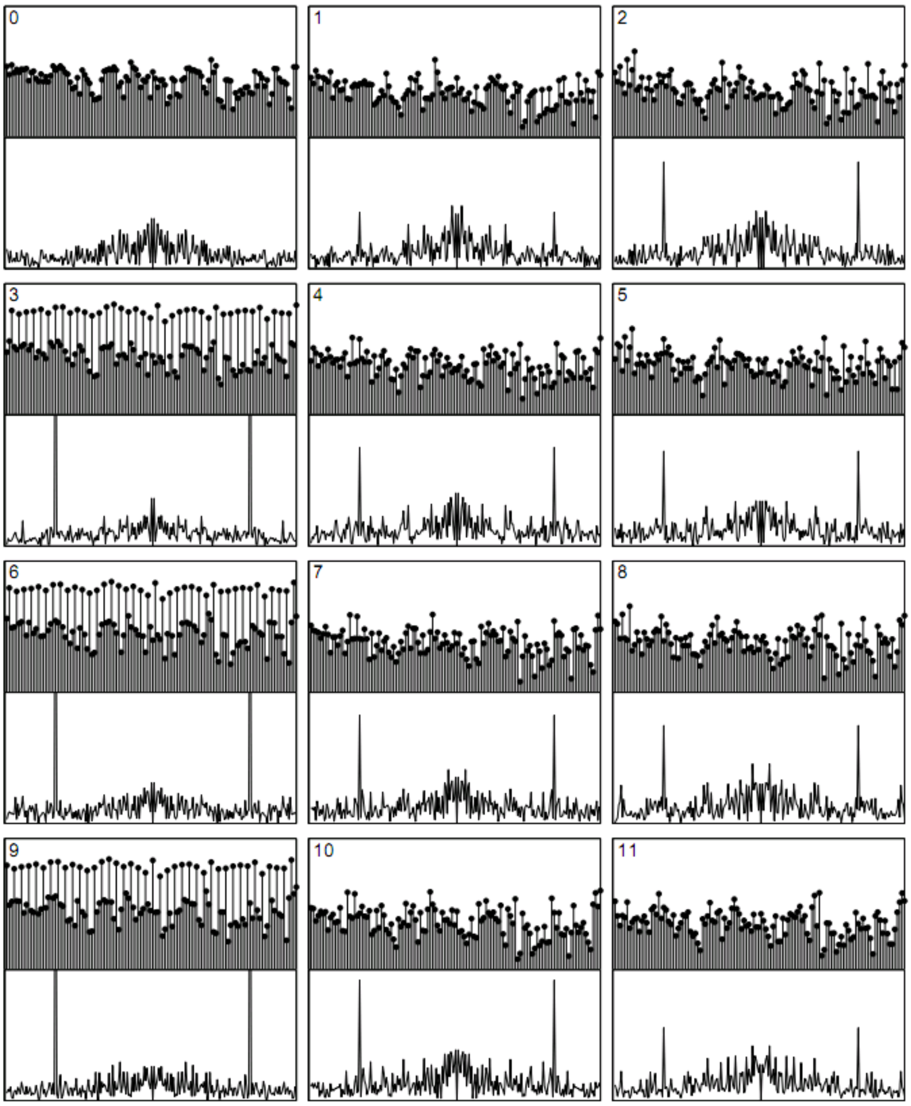


Figure 3-3-2a

According to [WW06], the reason for this change in motion error is that the P-frames within a single GOP are correlated to its initial I-frame. Due to the motion compensation encoding, these compression artefacts therefore propagate through the P-frames. As a result, the motion errors of each P-frame will be correlated to its neighbouring P-frame or I-frame. As can be seen in the original sequence in Figure 3-2-2b below, the motion error of P_2 will be correlated to I_1 while the motion error of P_3 will be correlated to P_2 and so on and so forth. The motion errors are expected to be small within a GOP since the first P-frame is always correlated to its initial I-frame in the GOP.

	GOP #1					**GOP #2**					**GOP #3**				
Original Sequence	I_1	P_2	P_3	P_4	P_5	I_6	P_7	P_8	P_9	P_{10}	I_{11}	P_{12}	P_{13}	P_{14}	P_{15}
Deleted Frames			P_3	P_4											
Re-encoded Sequence	I_1	P_2			P_3	P_4	P_5	I_6	P_7	P_8	P_9	P_{10}	I_{11}	P_{12}	P_{13}

Figure 3-2-2b

When the P-frames in an MPEG-2 video are deleted and the video is then re-encoded, the new neighbouring I and P-frames of the (in the region where the frames are deleted) will have a larger motion error since they originated from different GOPs. As can be seen in Figure 3-2-2b above, in the re-encoded sequence where the original P_3 and P_4 frames were deleted, the motion error of the new P_4 (which used to be I_5 in the original sequence) is expected to be larger compared to its preceding P-frames since the correlation is weaker as it originated from a different GOP (the new P_4 used to belong to GOP #2 in the original sequence). This increase in the motion errors will appear periodically throughout the GOPs as all the various I-frames and P-frames were being shifted (starting from the point where the frames are deleted or added). The addition of new frames will also have the same effect as deleted frames. The periodic increase in the mean motion errors for these P-frames will be reflected as “spikes” in the FFT graphs of the mean motion errors of P-frames of the MPEG-2 video.

Hence, when the FFT graphs of the mean motion errors of the P-frames for the re-encoded video (where a series of frames were deleted) are plotted, as seen in Figure 3-2-2a above, there will be visible “spikes” so long as the deletion (or addition) of P-frames is not a multiple of the GOP length. This property allows a forensic investigator to deduce that the video has been tampered. However, as acknowledged by [WW06], this technique will not work if the addition or deletion of frames is in the multiple of the GOP length. This is because if an entire group of GOP is deleted, there will not be any correlation gaps between the neighbouring I-frames or P-frames of any of the GOPs (since the frames still belong to their original GOPs) in the new re-encoded video. As such, the mean motion errors of P-frames for such re-encoded videos will not be significantly different from the original encoded video. A forensics investigator will therefore not be able to tell from this technique that the video actually has been tampered.

Building on the research by [WW06] in observing the temporal artefacts of frames, [YS11] proposed another new technique in detecting the deletion of frames in a doubly compressed MPEG-2 video. When frames are encoded in MPEG-2, [YS11] noted that the high frequency components of intra-coded frames (i.e., I-frames) are much lower than that of inter-coded frames (i.e., P-frames or B-frames). Please refer to [pp. 14 to 16 of JW01] for more information on inter and intra-coded compression. In a re-encoded video where some frames are deleted, some of the I-frames that are being re-encoded as B or P-frames will have a lower high frequency components compared to their neighbouring B or P-frames. This is because as mentioned earlier, the high frequency components of these I-frames have already been removed in in the first compression. Again, similar to the approach used by [WW06], a periodic artefact can then be observed in the DFT graphs of high frequency components of inter-coded frames. As can be seen from Figure 3-2-3, there is a “spike” in the DFT of the high frequency components of the inter-coded frames. Figure 3-2-3 is reproduced from [pp. 4 of YS11] – the top graph shows the DFT of the original video while the bottom graph shows the DFT of the tampered video.

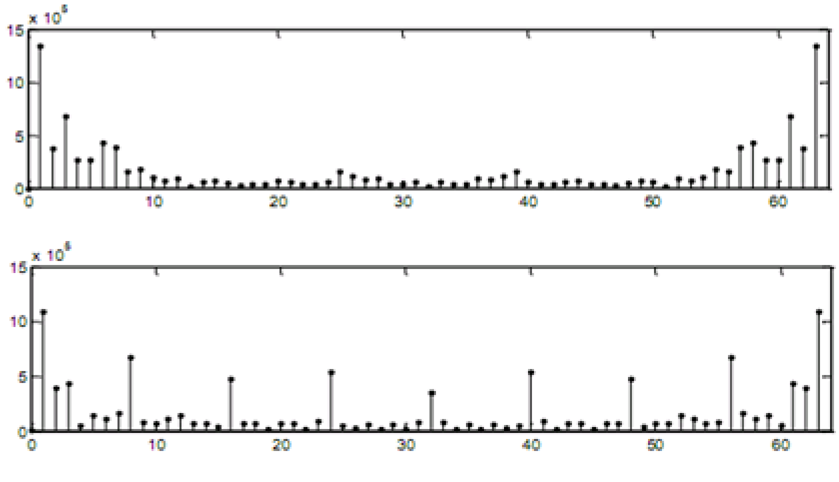


Figure 3-2-3

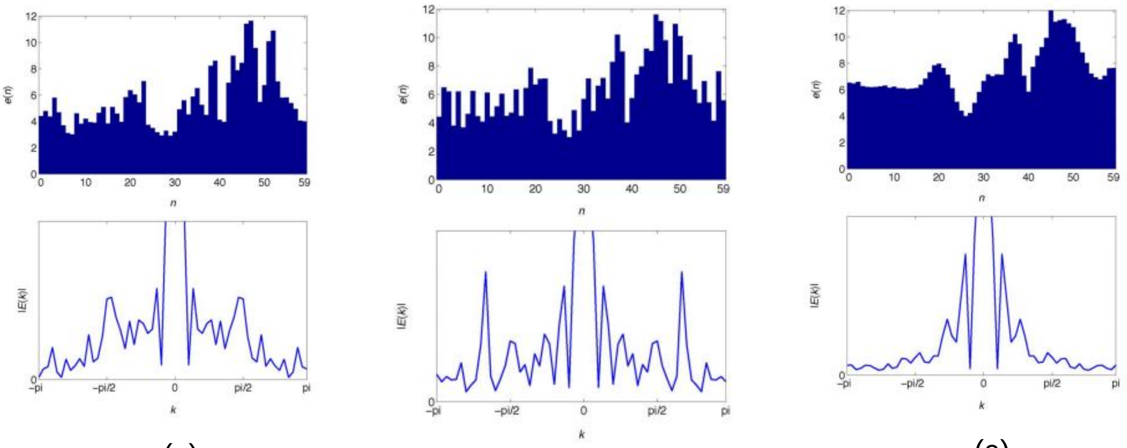
Using this technique, [YS11] is able to detect the deletion of frames in a tampered MPEG-2 video. Unfortunately, similar to the limitation in [WW06], [YS11] is unable to detect frame tampering operations which does not cause coding-type changes, such as the deletion of frames in the multiple of the GOP length.

Despite the relative effectiveness of [WW06} in detecting video tampering, a very recent paper published by [MS12] in August 2012 highlighted that the method has several weaknesses. One of the limitations of [WW06] is that it requires human interaction to observe whether there is a spike in the FFT graphs of the mean motion errors of the P-frames in a targeted video. Another limitation is that it is unable to detect video tampering of a doubly compressed MPEG-2 video if the deletion/addition of frames is not in a multiple of the GOP length of the video. In addition, [WW06] is also unable to be used on videos that are compressed using encoders that adaptively change the GOP length. In order to overcome the highlighted weaknesses of [WW06], [MS12] proposed a mathematical model of the temporal fingerprint which is able to take into account both fixed and variable GOP lengths and also be used in creating automated solutions that can spot “spikes” in the DFT graphs of tampered videos.

In addition, [MS12] proposed a counter-forensics algorithm in which the algorithm first constructs a target P-frame motion error sequence that is free from the temporal fingerprints and then selectively alters the video’s predicted frames so that the motion error sequence from the tampered video matches the targeted one. This is done by setting the P-frame’s motion vectors for certain macroblocks to zero and then recalculating the associated motion error values for the affected macroblocks. By doing so, [MS12] is able to manipulate the mean motion errors of P-frames.

Figures 3-2-4a, 3-2-4b, 3-2-4c below which are reproduced from [MS12] show the mean motion errors of the P-frames (top) and their associated DFT graphs (bottom) for an original video, the tampered version where 6 frames were deleted and the same tampered version with the counter-forensics algorithm applied respectively. Per [WW06], the original video does not show any “spikes” in the DFT graph in Figure 3-2-4a while a “spike” can be seen at the DFT graph of the tampered version in Figure 3-2-4b. Using the proposed counter-forensics algorithm, [MS12] was able to manipulate the mean motion errors of P-frames such that there are no visible spikes in the DFT graphs as can be seen in Figure 3-2-4c.

In addition, the manipulation of the mean motion errors of the P-frames was so precise that it produced a very smooth graph in the mean motion errors. The motion errors (top) of the original video in Figure 3-2-4a were rather jagged in the steps and produced an uneven looking graph but the tampered version with the counter-forensics algorithm applied in Figure 3-2-4c show a very smooth transition in the steps of mean motion errors.



(a)

(b)

(c)

Figure 3-2-4

Buoyed by their success in developing a counter-forensics algorithm, [MS12] further proposed a counter to their counter-forensics algorithm. Their proposed counter-forensics algorithm alters the motion vectors in order to cause the desired changes in the motion errors. However, the true motion present in the video is not changed. As such, there are discrepancies between many motion vectors stored in an original video and the video that has been tampered with counter-forensics algorithm applied. These discrepancies form “fingerprints” which may be subsequently detected.

[MS12] subsequently went on to show that the rate of detection of tampered videos (whether applied with counter-forensics technique or not) by a forensics investigator would depend on how careful the attacker is when crafting a tampered video and also how fine a threshold that a forensics investigator would set in detecting tampered videos. In the case of the counter-forensics algorithm proposed by [MS12], the discrepancies between the motion vectors can cause false positives if the threshold for detecting the discrepancies is set too low.

Finally, [MS12] proposed a set of game theoretic techniques to study the interplay between the attacker and a forensics investigator. [MS12] does so by formulating the functions of both the attacker and the forensics investigator in terms of probabilistic quantities associated with the performance of their detection and counter-forensics techniques.

In conclusion, this chapter first started off by covering the general trends in digital image and video forensics through [AR11]. [AR11] gave a description of the evolution of digital image and video forensics and also went on define the scope of the field. It also covered at length the general research approaches in the field, where it can be seen that most of the research techniques originated from digital image forensics. Finally, it highlighted some of the challenges that is currently facing the digital image and video forensics, such as the lack of rigour of the techniques proposed as well as the lack of counter-forensics in verifying the robustness of the various proposed detection techniques.

After covering the general trends in digital image and video forensics, the chapter went on to cover some of the recent and popular research in the detection of tampered MPEG-2 videos. In particular, recent research tends to rely on the fact that an attacker will use an intermediate video editor to tamper the video which in turn causes the original MPEG-2 video to undergo a second compression. The various research highlighted in the chapter rely on observing the spatial [WW09] and temporal [WW06, YS11 and MS12] statistical properties of a doubly compressed MPEG-2 video.

Following an understanding of the recent research in MPEG-2 forensics video, the next chapter, Chapter 4 (Problem Statement) will highlight the research aims and objectives of this report.

Chapter 4: Problem Statement

Chapter 1 (Introduction) covered the motivation of studying digital video forensics, especially on MPEG-2 videos. In Chapter 3 (Related Works), the report covered the general research trends of digital image and video forensics as well as presenting a summary of the recent and popular research in MPEG-2 video forensics. It also highlighted that there are 2 ways in tampering MPEG-2 videos. The first approach is to use intermediate software, such as video editors to modify the video while the second approach is edit the MPEG-2 encoded file directly. The former is obviously more user-friendly and therefore presumably attackers will favour this approach. Such an approach will result in the video undergoing double compression. As such, current research in MPEG-2 video forensics relies heavily on observing the spatial [WW09] and temporal [WW06, YS11 and MS12] statistical properties of a doubly compressed video. The research in observing the temporal statistical properties invariably relies on the observation of “spikes” in the DFT/FFT graphs in order to detect tampering.

However, what if an increasingly sophisticated attacker chooses to tamper the video by editing the MPEG-2 encoded file directly? Such an attack while presumably harder, may gain popularity since recent research [WW09, WW06, YS11, MS12] seems to suggest that tampered MPEG-2 videos can be reliably detected by observing the spatial and temporal statistical properties of a doubly compressed MPEG-2 video. Therefore, perhaps the direct editing of the MPEG-2 encoded file can present a sort of counter-forensics approach to defeating the various recent detection techniques proposed by [WW09, WW06, YS11, MS12].

Hence, this report will focus on directly tampering an MPEG-2 encoded file to see if the current research techniques in digital video forensics can detect the tampering. Specifically, this report will look at the methodology proposed by [WW06]. As covered in Chapter 3 (Related Works), according to [WW06], when frames are added or deleted from a video and then re-encoded again, the FFT graphs of the mean motion errors of P-frames will show large spikes that indicate the tampering of the video. However, [WW06] acknowledged that the tampering of the video cannot be detected if the number of frames inserted/deleted is a multiple of the GOP length.

The methodology by [WW06] is chosen to be investigated because the methodology was the first to look at the temporal statistical properties of a doubly compressed video and is often quoted in subsequent research [YS11 and MS12] in MPEG-2 forensics. It would be interesting to know if the direct tampering of MPEG-2 encoded file will leave any kind of temporal artefacts.

Usually, when an attacker tampers a video, the attacker typically seeks to manipulate the video frames in the following manner:

- Deleting frames, e.g., deleting a particular scene (that consists of a series of frames) that may contain critical evidence.
- Adding frames, e.g., adding a particular scene that did not happen in order to prove evidence otherwise.
- Duplicating frames, e.g., adding scenes that happened in the video to either add or replace other scenes – most famously the looping of CCTV footage in movies such as “Speed” and “Mission Impossible”.
- Modifying sections of a frame, e.g., adding or deleting an object within a frame.

This report will focus on the deletion and duplication of frames as the means of tampering a video. The direct addition of frames is ignored since the behaviour will be the same as the deletion of frames and in most cases, an attacker is more likely to delete incriminating frames rather than adding frames to a video, such as removing the presence of a person from CCTV scenes [MS12]. Furthermore, the duplication of frames involves deletion of certain frames in the video and replacing them (which is similar to the addition of frames) with other existing frames in the video.

As part of this investigation, a series of experiments will be conducted to investigate the effectiveness of the methodology proposed by [WW06]. Details of how the experiments are designed will be explained in Chapter 5 (Design of Experiments). The results of the experiments are available in Chapter 6 (Results of Experiments) and Chapter 7 (Discussion) will present an analysis of the results of the experiments. Finally, Chapter 8 (Conclusion & Summary) will summarise the findings of the experiments and discuss the effectiveness of the methodology proposed by [WW06] as well as highlighting other aspects of detection and counter-forensics and also presenting future areas of research and improvement.

Chapter 5: Design of Experiments

As discussed in Chapter 4 (Problem Statement), this report seeks to investigate whether the methodology used by [WW06] can be used to determine video tampering if an attacker were to tamper the MPEG-2 video stream directly. A series of experiments will be conducted as part of the investigation.

MATLAB, a popular programming environment for algorithm development, data analysis, visualization, and numerical computation, is used in the investigation. Firstly, an MPEG-2 style encoder/decoder (referred to as the "decoder") is implemented in MATLAB. This decoder is then used to generate MPEG-2 encoded videos from a series of test reference videos. The encoded MPEG-2 videos are then tampered by deleting and duplicating certain frames directly from the encoded MPEG-2 file using MATLAB. The tampered videos are then decoded by the same decoder and viewed using MATLAB's built-in video player to determine if the tampering can be detected visually. The FFT graphs for the mean motion errors of P-frames are also generated using MATLAB to determine whether the tampering can be detected per [WW06]. In addition, MATLAB is also used as the programming platform to develop algorithms to detect and analyse the video tampering.

An MPEG-2 style decoder that was implemented using MATLAB was obtained from [SH05]. This decoder is freely available and is used to encode/decode a series of videos for the purpose of the investigation. The decoder also had to be modified in order to investigate the effectiveness of the methodology proposed by [WW06] which looks at the effect of the FFT graphs of the mean motion errors of P-frames for a tampered video. The decoder is modified so that the mean motion errors for a frame are calculated by obtaining the arithmetic mean of the motion errors for each macroblock within a frame. Only I and P frames are generated using this decoder since the methodology proposed by [WW06] only looked at the mean motion errors of P-frames. I-frames are necessary as they provide a reference frame for the P-frames but B-frames are not necessary in the experiments.

The test reference videos were obtained from the following 2 sources, namely:

- A test reference video (called the "foreman" video) was obtained from [XO12] which is a non-profit organisation that seeks to put foundation standards of Internet audio and video into the public domain. [XO12] has a series of reference videos that is publicly available in its website.

- A built-in webcam from the author’s notebook (Acer Aspire 3820TG) is used to generate a test video (called the “hand” video”) for the purpose of the investigation.

Each video consists of 300 frames with a resolution of 352 by 288 pixels. A macroblock consists of 16 by 16 pixels. As such, each frame will have 18 by 22 macroblocks. The videos are encoded in using RGB and have to be stored in a MATLAB movie format.

The “foreman” video from [XO12] is an uncompressed video in Y4M format and therefore has to be first converted into the MATLAB movie format in order to be processed by MATLAB. There is a MATLAB script (“loadFileY4m.m”) that is available in [SH05] that provides this conversion. According to [MW12], the Y4M format is a format that holds uncompressed frames of YCbCr video for the purpose of encoding to other video standards, such as MPEG-2. The “hand” video is captured directly from the webcam using the MATLAB Image Processing Toolbox and hence is already stored in the MATLAB movie format. As such, no conversion is necessary.

In total, 4 main sets of experiments (as indicated in the sub-chapter headings below) will be conducted as part of the investigation. Firstly, the properties of the non-tampered encoded videos are observed. Next, the encoded videos are then tampered through the deletion and duplication of frames. The results of these experiments are then made available in Chapter 6 (Results of Experiments) and a discussion on the results of these experiments is presented in Chapter 7 (Discussion).

5.1 Non-tampered encoded videos

The 2 test reference videos being used, namely, the “foreman” and the “hand” video are encoded using the MPEG-2 style decoder and are encoded with the “IPPPP” frame pattern sequence.

A screen capture of selected key frames for both videos is first produced in order to give an idea of how the videos look like. Next, the mean motion errors of every frame of both videos are produced followed by the FFT graphs for the mean motion errors of the P-frames of both videos. The purpose of this experiment is to gain an understanding of the key characteristics (visual outlook, mean motion errors and FFT graphs) of the non-tampered videos.

5.2 Tampering encoded video – deleting frames

This section will investigate the deletion of frames in the "foreman" video where the video is encoded using the frame pattern sequence "IPPPP" and is chosen over the "hand" video simply because visual effects due to the deletion of the frames are more noticeable.

The "foreman" video shows a foreman who is talking and then subsequently pointing to a construction site behind him. The video started off by showing the foreman and then pans to the right as the foreman points to the construction site.

There are in total 5 experiments in this section. The frames are deleted in following manner:

- The deletion of an I-frame (frame #151) in a GOP in the video.
- The deletion of a P-frame (frame #152) in a GOP in the video.
- The deletion of a combination of I and P-frames within a GOP in the video:
    - Frames #151, #152
    - Frames #151, #152, #153
    - Frames #151, #152, #153, #154
- The deletion of an entire GOP (frames #151 to #155) in the video.
- The deletion of 1/3rd of the GOPs (frames #151 to #250) in the video.

A screen capture of selected key frames for the non-tampered and tampered videos are shown where relevant so as to allow observation as well as comparison of the visual aspects of the videos. Next, the mean motion errors of relevant frames will also be shown so as to observe the changes in the mean motion errors of the frames when the frames are being deleted. Finally, the FFT graphs for the mean motion errors of the P-frames of both the non-tampered and tampered videos will be shown in order to investigate the effectiveness of the methodology proposed by [WW06] in detecting the tampering of videos.

5.3 Tampering encoded video – deleting frames from different frame pattern sequences

The previous section investigated the effects of the deletion of frames using the frame pattern sequence "IPPPP". This section will investigate the effects of deletion of frames from videos that are encoded using 2 other frame pattern sequences. For ease of reference, the various frame pattern sequences will be known as the following:

- Pattern #1 – "I PPPP" – starting from frame #151 (I-frame).
- Pattern #2 – "I PPPP PPPP" – starting from frame #154 (I-frame).
- Pattern #3 – "I PPPP PPPP PPPP" – starting from frame #157 (I-frame).

The deletion of frames for each of the above frame pattern sequence starts from the initial I-frame of a GOP. Given the different encoding frame pattern sequences, the initial I-frame number for each of the patterns are obviously different. For Pattern #1, it starts from frame #151, while for Pattern #2 and #3, the frames are frame #154 and #157 respectively.

Initially, the first I-frame is deleted. Next, the I-frame and its neighbouring P-frame are deleted. After that, the I-frame and the next 2 P-frames are deleted and so on and so forth. The deletion of frames continues until an entire GOP is deleted. However, for Pattern #1, the deletion of frames was already performed in the previous section, namely Section 5.2 (Tampering encoded video – deleting frames). As such, the results will be obtained from that section and the experiment will not be repeated again. For Pattern #2 and #3, the experiments will be performed according to the description above.

For each of the above frame pattern sequences, a sample of the frame motion errors will be made available. In addition, the FFT graphs of the mean motion errors of P-frames for each of the deletion will also be presented. The objective for this set of experiments is to determine the effects of different encoding (i.e., the frame pattern sequences) on the methodology proposed by [WW06].

5.4 Tampering encoded video – duplicating frames

This section will investigate the duplication of frames that are already available in the targeted video to be tampered. The "hand" video is chosen over the "foreman" video due to its simplicity in scene changes. This will allow the visual aspects of the duplication to be more readily observed. The "hand" video is encoded with the frame pattern sequence "IPPPP".

The "hand" video attempts to simulate the recording of a stationary CCTV camera where there are no subject (in this case, the hand) in the scenes for most part of the video followed by the appearance (and subsequent disappearance) of a subject. The "hand" video is stationary and consists largely of the following 3 sets of scenes (refer to Figure 6-1-1b for the screen captures of key frames):

- Set 1 – first 1/3rd of the video showing an empty stationary scene.
- Set 2 – second 1/3rd of the video showing a hand entering the scene, the hand staying stationary in the scene and then finally exiting the scene.
  - The hand is first seen entering the scene at frame #98.
  - The hand is last seen leaving the scene at frame #203.
- Set 3 – final 1/3rd of the video showing again an empty stationary scene.

The hand starts to appear in the video starting from frame #98 and remains stationary in the video until frame #203. The objective of this experiment is to slowly reduce the number of frames that the hand appears in the video until the hand totally disappears from the video. This simulates an attacker trying to hide incriminating evidence (in this case, the hand).

A screen capture showing the following GOPs will be made available so that a visual idea of the video may be presented:

- GOP Set A – GOP with frames #91 to #95 – showing an empty stationary scene.
- GOP Set B – GOP with frames #96 to #100 – showing the hand entering the scene.
- GOP Set C – GOP with frames #201 to #205 – showing the hand leaving the scene.

During the duplication, the frames will be duplicated from GOPs to GOPs. GOP Set A with frames #91 to #95 (which is part of Set 1 that shows an empty stationary scene) is arbitrarily chosen as the GOP to replace other GOPs that form part of Set 2 (which contain the subject, i.e., the hand”).

However, if GOP Set A is simply duplicated over any other GOPs that contains the scene with the hand, then it will create an obvious visual effect where the hand just suddenly appears (since GOP Set B that shows the hand entering the scene is also overwritten). Hence, in order to prevent this abruptness, it is reasonable to assume that the attacker will have to duplicate all the targeted GOPs with GOP Set A followed by GOP Set B.

The experiments will be slowly repeated (with the duration of the video that shows the hand getting shorter and shorter) until the last experiment which will just show empty stationary scenes (simulating a suspect completely removing himself from the video). Figure 5-4 below shows the frames to be replaced and how they are being replaced. The FFT graphs of the mean motion errors of P-frames for each of the experiment in Figure 5-4 below will be presented in order to determine if the methodology by [WW06] can detect the duplication of frames. The results of the experiments presented in this chapter will be made available in the following chapter, Chapter 6 (Results of Experiments).

No.	**Frames to be duplicated**	**GOPs to be duplicated with GOP Set A (frames #91 to #95 which shows an empty stationary scene)**	**GOPs to be duplicated with GOP Set B (frames #96 to #100 which shows a hand entering the scene)**
1	#96 to #105	GOPs with frames #96 to #100	GOPs with frames #101 to #105
2	#96 to #110	GOPs with frames #96 to #105	GOPs with frames #106 to #110
3	#96 to #115	GOPs with frames #96 to #110	GOPs with frames #111 to #115
4	#96 to #120	GOPs with frames #96 to #115	GOPs with frames #116 to #120
5	#96 to #125	GOPs with frames #96 to #120	GOPs with frames #121 to #125
6	#96 to #130	GOPs with frames #96 to #125	GOPs with frames #126 to #130
7	#96 to #135	GOPs with frames #96 to #130	GOPs with frames #131 to #135
8	#96 to #140	GOPs with frames #96 to #135	GOPs with frames #136 to #140
9	#96 to #145	GOPs with frames #96 to #140	GOPs with frames #141 to #145
10	#96 to #150	GOPs with frames #96 to #145	GOPs with frames #146 to #150
11	#96 to #155	GOPs with frames #96 to #150	GOPs with frames #151 to #155
12	#96 to #160	GOPs with frames #96 to #155	GOPs with frames #156 to #160
13	#96 to #165	GOPs with frames #96 to #160	GOPs with frames #161 to #165
14	#96 to #170	GOPs with frames #96 to #165	GOPs with frames #166 to #170
15	#96 to #175	GOPs with frames #96 to #170	GOPs with frames #171 to #175
16	#96 to #180	GOPs with frames #96 to #175	GOPs with frames #176 to #180
17	#96 to #185	GOPs with frames #96 to #180	GOPs with frames #181 to #185
18	#96 to #190	GOPs with frames #96 to #185	GOPs with frames #186 to #190
19	#96 to #195	GOPs with frames #96 to #190	GOPs with frames #191 to #195
20	#96 to #200	GOPs with frames #96 to #195	GOPs with frames #196 to #200
21	#96 to #205	GOPs with frames #96 to #205	

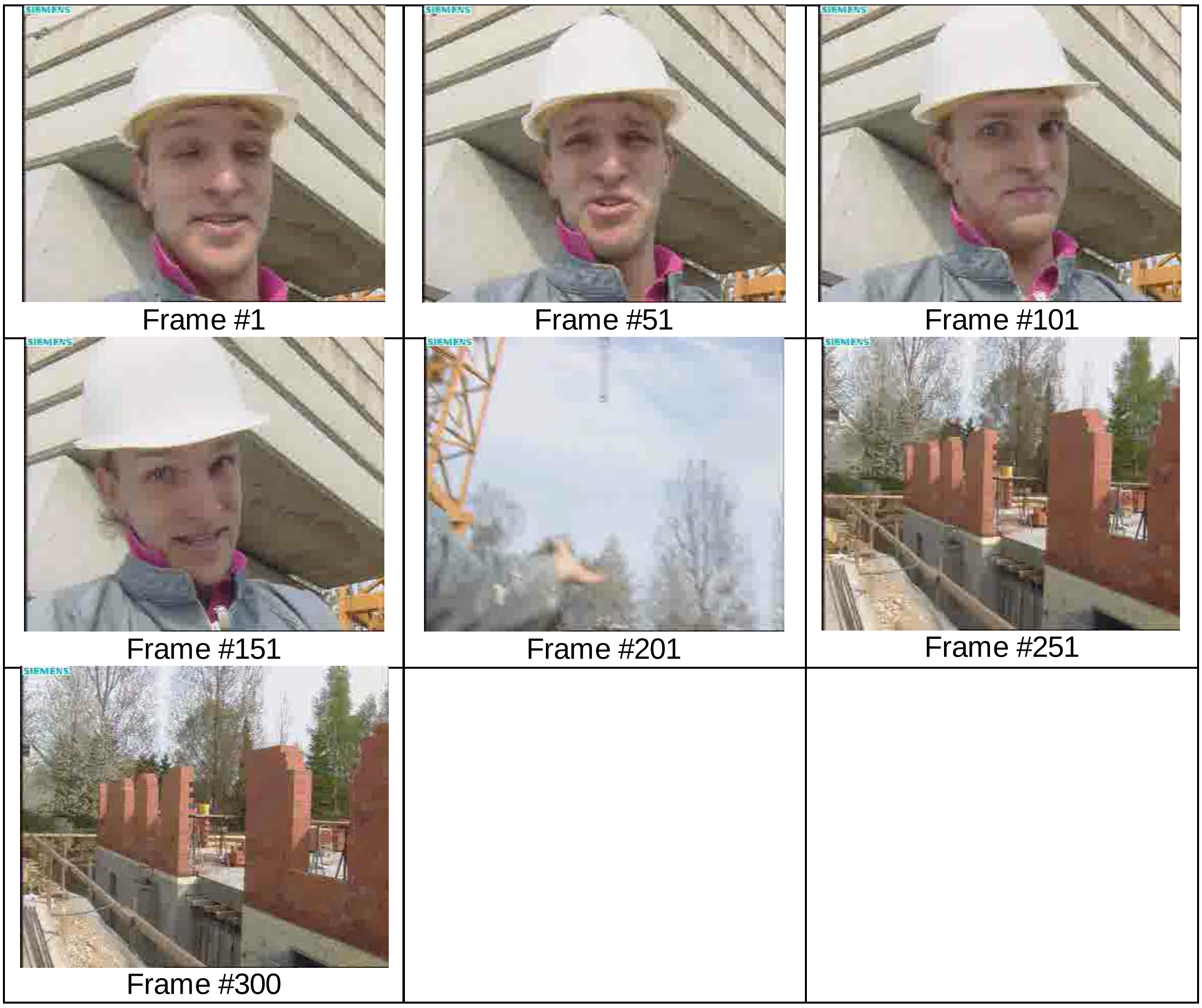
Figure 5-4

Chapter 6: Results of Experiments

This chapter shows the results of the experiments conducted in Chapter 5 (Design of Experiments).

6.1 Non-tampered encoded videos

Figure 6-1-1a below showed the key frames of the “foreman” video. As explained in Chapter 5 (Design of Experiments), the “foreman” video showed a foreman who is talking and pointing to a construction site with the video focusing first on the foreman and then panning to the right to show the construction site.



Frame #1

Frame #51

Frame #101

Frame #151

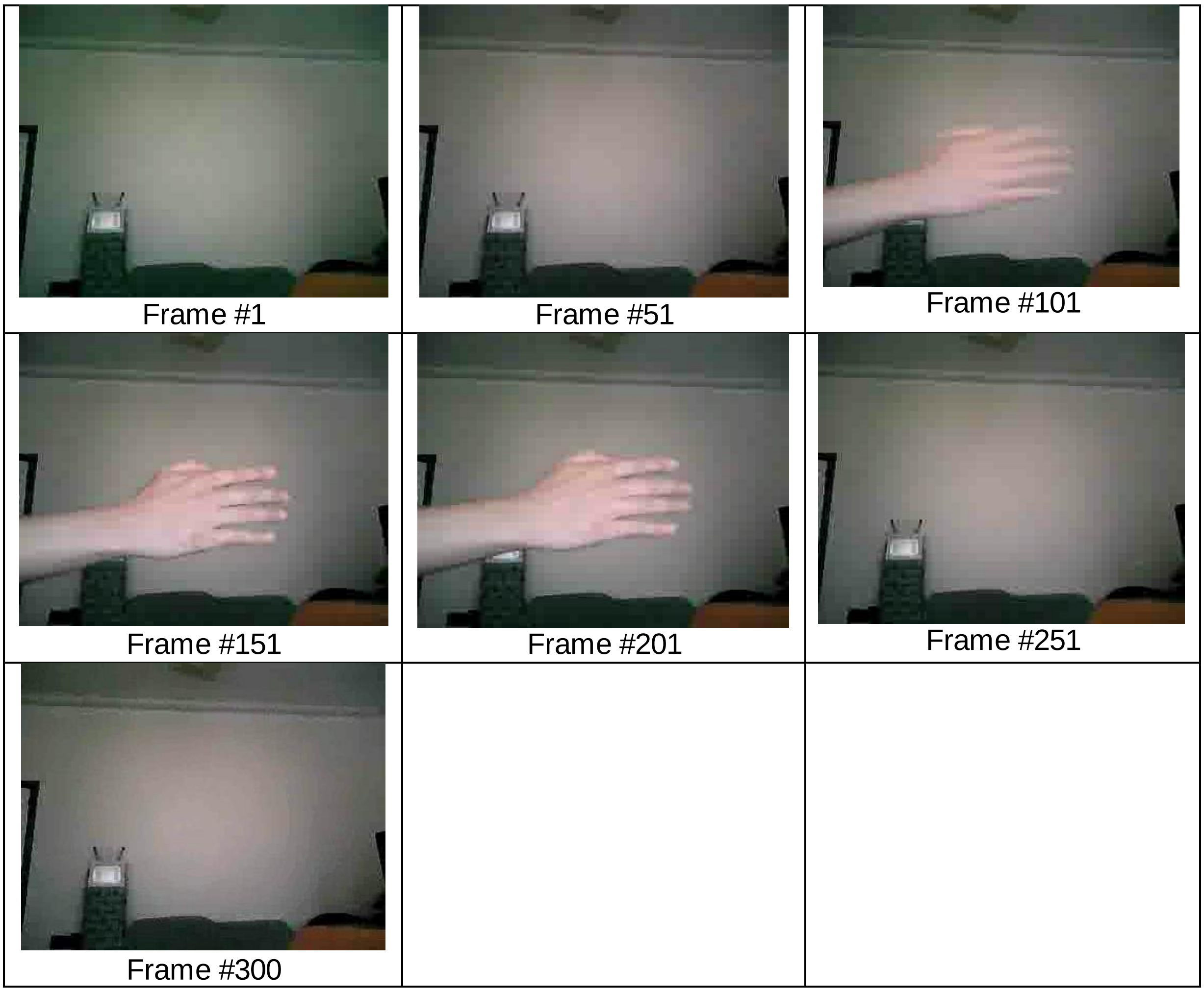
Frame #201

Frame #251

Frame #300

Figure 6-1-1a

Figure 6-1-1b below showed the key frames of the “hand” video. As explained in Chapter 5 (Design of Experiments), the “hand” video first show an empty stationary scene followed by a hand entering the scene and then staying in the video for 1/3rd of the duration of the video before leaving the scene and then showing the same empty stationary scene as seen in the beginning of the video.



Frame #1

Frame #51

Frame #101

Frame #151

Frame #201

Frame #251

Frame #300

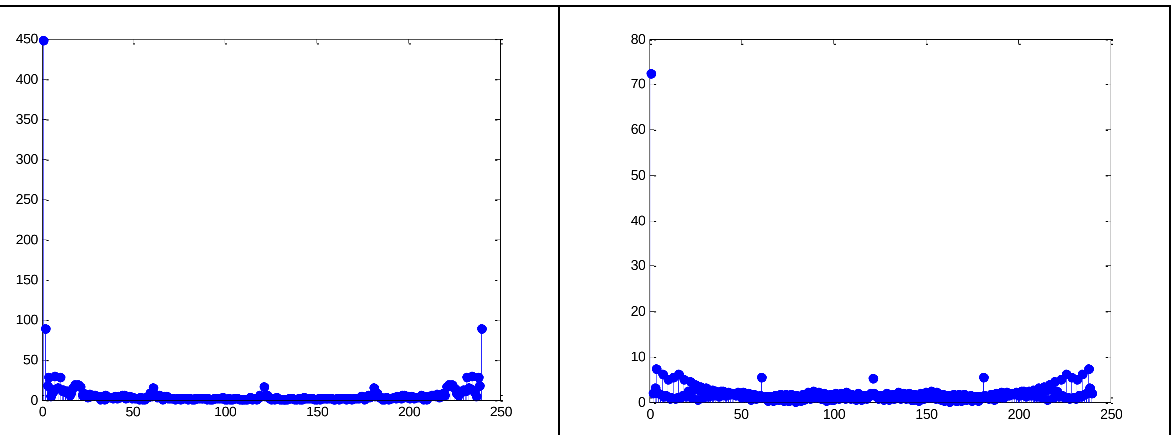
Figure 6-1-1b

Figure 6-1-2 below showed a sample of the mean motion errors of the I and P-frames of the non-tampered “foreman” and “hand” videos. The complete mean motion errors of all the 300 frames of each video are available in Figure A-1 of Appendix 1 (Mean Motion Errors of Frames).

Frame	Frame Type	Foreman	Hand	Frame	Frame Type	Foreman	Hand
1	I	60.3742	35.8367	151	I	58.5394	37.4103
2	P	1.4106	1.2123	152	P	1.6639	0.3675
3	P	1.2102	0.454	153	P	2.4054	0.2837
4	P	1.1111	1.0793	154	P	3.0604	0.2923
5	P	1.25	0.4629	155	P	2.805	0.2724
51	I	59.6046	35.7172	201	I	69.9585	37.664
52	P	1.5176	0.3146	202	P	2.3956	1.1133
53	P	1.1264	0.2364	203	P	2.4426	1.2724
54	P	1.1707	0.2067	204	P	2.3661	0.5001
55	P	1.426	0.1984	205	P	2.2615	0.3066
101	I	59.1438	37.1018	251	I	50.0767	36.0255
102	P	1.3943	0.6088	252	P	2.7897	0.3085
103	P	1.016	0.4508	253	P	2.063	0.2082
104	P	0.9848	0.3643	254	P	1.893	0.2541
105	P	1.0088	0.3422	255	P	2.2455	0.2138
145	P	1.5595	0.2798	295	P	2.1066	0.2355
146	I	59.5836	37.3195	296	I	50.1978	36.3251
147	P	1.5101	0.3337	297	P	2.6882	0.3034
148	P	1.2456	0.2949	298	P	2.3282	0.2431
149	P	1.2786	0.299	299	P	2.1198	0.2261
150	P	1.0139	0.2577	300	P	1.6994	0.2058

Figure 6-1-2

Figure 6-1-3 below showed the FFT graphs for the mean motion errors of P-frames of the non-tampered “foreman” and “hand” videos.



FFT of “foreman” video

FFT of “hand” video

Figure 6-1-3

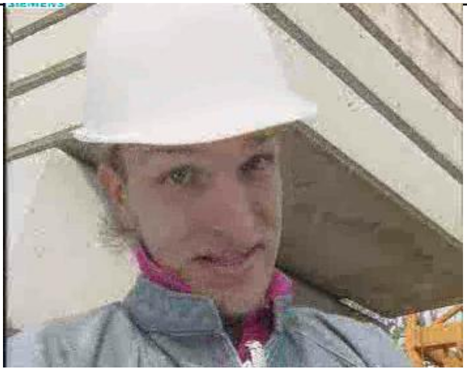
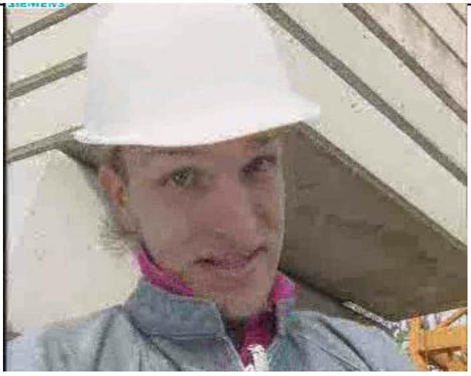
6.2 Tampering encoded video – deleting frames

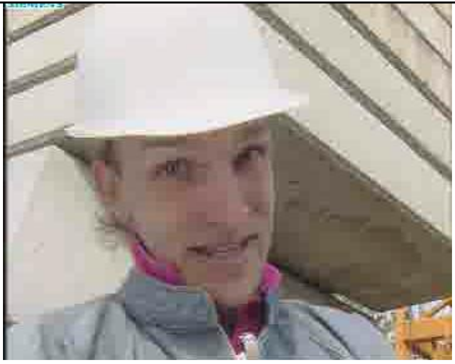
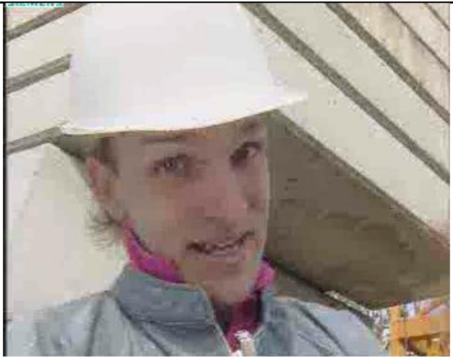
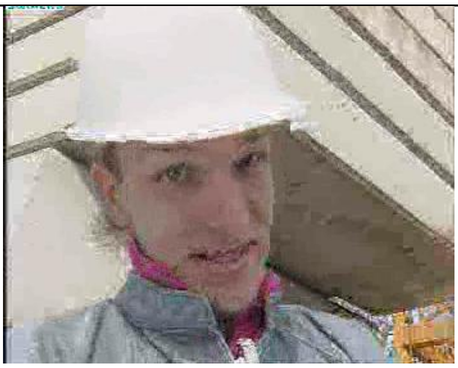
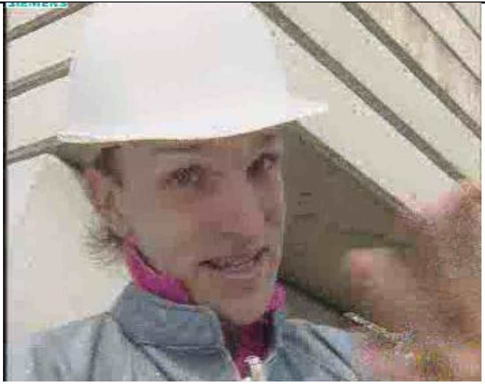
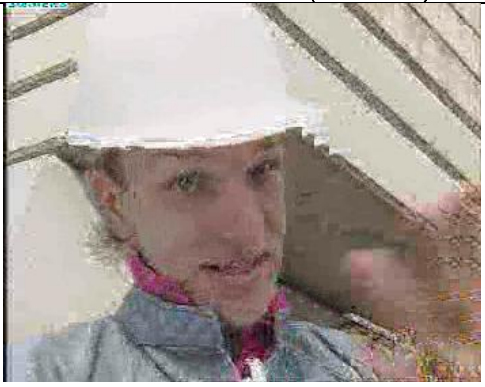
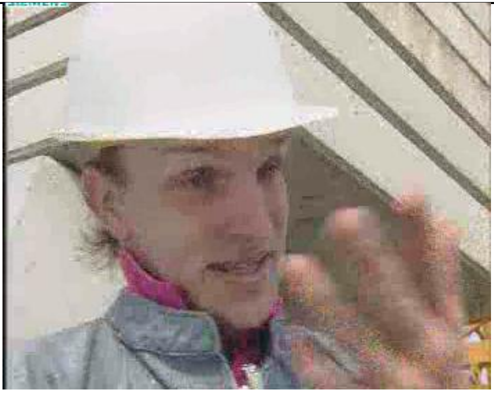
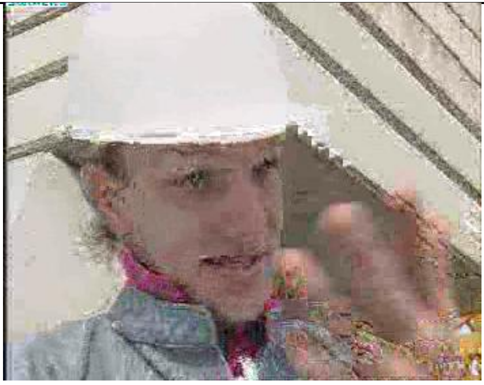
This section will show the results of the investigation of the deletion of frames in the "foreman" video. As explained in Chapter 5 (Design of Experiments), the frames are deleted in following manner:

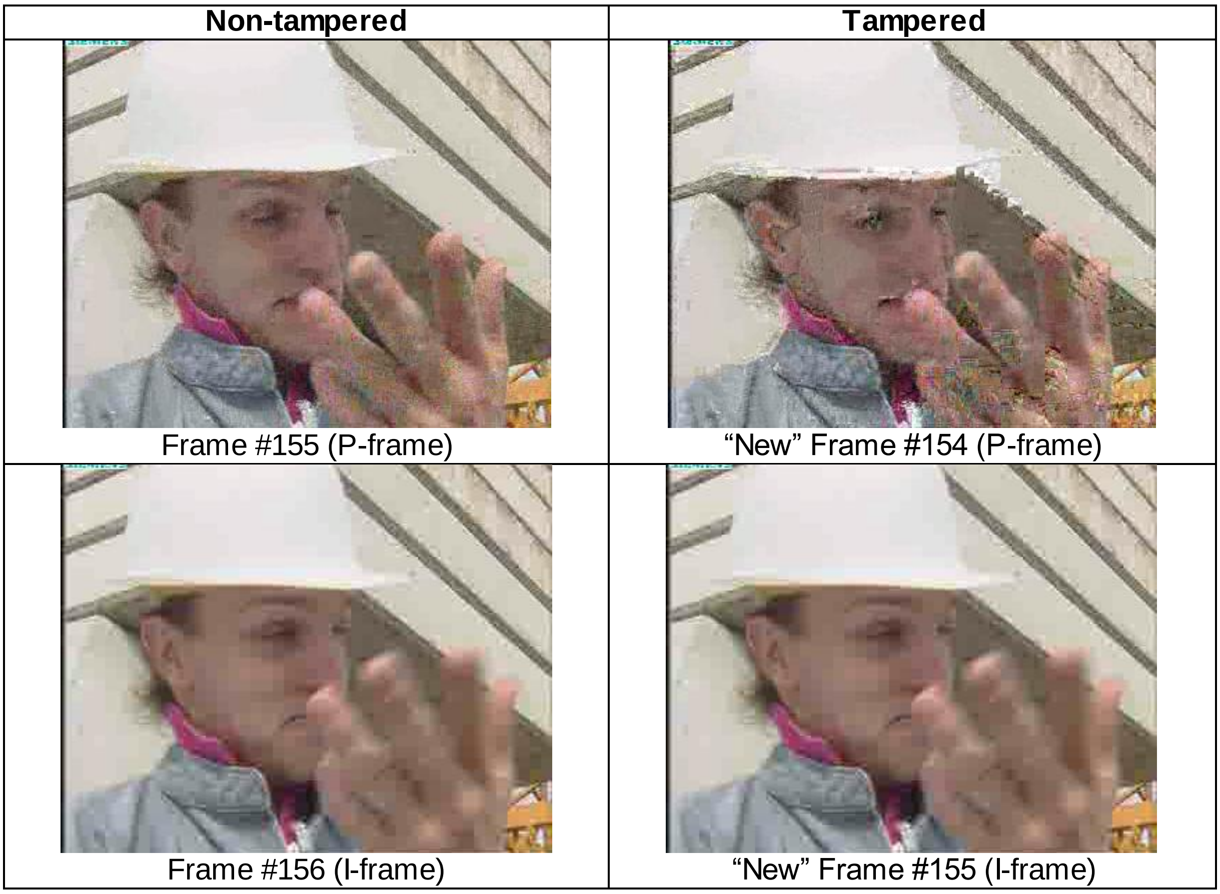
- The deletion of an I-frame (frame #151) in a GOP in the video.
- The deletion of a P-frame (frame #152) in a GOP in the video.
- The deletion of a combination of I and P-frames within a GOP in the video:
    - Frames #151, #152
    - Frames #151, #152, #153
    - Frames #151, #152, #153, #154
- The deletion of an entire GOP (frames #151 to #155) in the video.
- The deletion of 1/3rd of the GOPs (frames #151 to #250) in the video.

6.2.1 Deletion of an I-frame (frame #151) in a GOP in the video

Figure 6-2-1a below showed the screencaptures of frames #150 to #156 of the "foreman" video for both the non-tampered video version versus the tampered video version where an I-frame (frame #151) has been deleted.

Non-tampered	Tampered
 Frame #150 (P-frame)	 Frame #150 (P-frame)

Non-tampered	Tampered
 Frame #151 (I-frame)	Original I-frame (frame #151) deleted
 Frame #152 (P-frame)	 “New” Frame #151 (P-frame)
 Frame #153 (P-frame)	 “New” Frame #152 (P-frame)
 Frame #154 (P-frame)	 “New” Frame #153 (P-frame)



Non-tampered | Tampered

Frame #155 (P-frame)

“New” Frame #154 (P-frame)

Frame #156 (I-frame)

“New” Frame #155 (I-frame)

Figure 6-2-1a

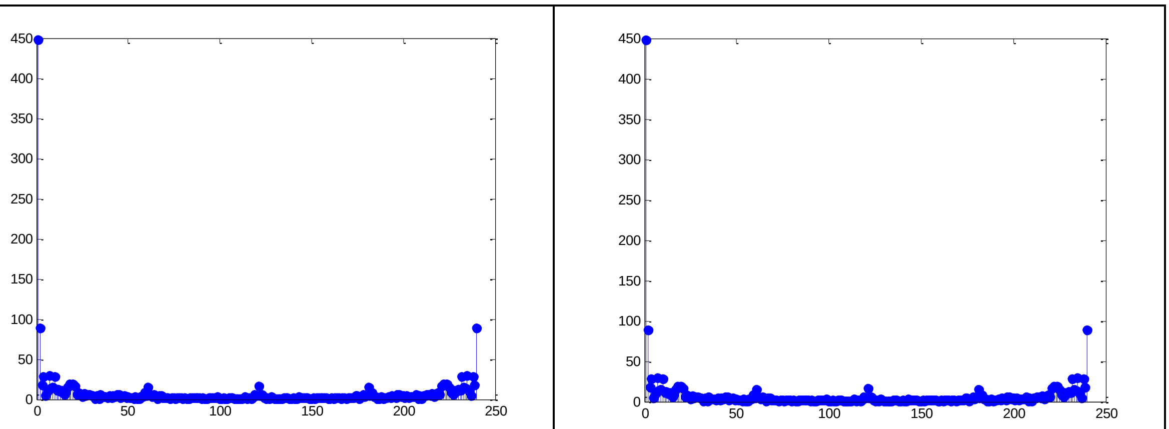
Figure 6-2-1b below shows a comparison of the relevant frames’ mean motion errors for the non-tampered and tampered video that has an I-frame (frame #151) deleted.

Frame	Frame Type	Non-tampered	Frame	Frame Type	Tampered	Remarks
146	I	59.5836	146	I	59.5836	
147	P	1.5101	147	P	1.5101	
148	P	1.2456	148	P	1.2456	
149	P	1.2786	149	P	1.2786	
150	P	1.0139	150	P	1.0139	
151	**I**	**58.5394**	151	P	1.6639	The old I-frame (#151) was deleted
152	P	1.6639	152	P	2.4054	
153	P	2.4054	153	P	3.0604	
154	P	3.0604	154	P	2.805	
155	P	2.805	155	I	59.5208	
156	I	59.5208	156	P	3.1054	
157	P	3.1054	157	P	2.6855	
158	P	2.6855	158	P	2.3992	

Frame	Frame Type	Non-tampered	Frame	Frame Type	Tampered	Remarks
159	P	2.3992	159	P	1.3642	
160	P	1.3642	160	I	57.426	

Figure 6-2-1b

Figure 6-2-1c below showed the comparison of the FFT graphs for the mean motion errors of P-frames of the non-tampered and tampered video that has an I-frame (frame #151) deleted.



FFT of non-tampered “foreman” video

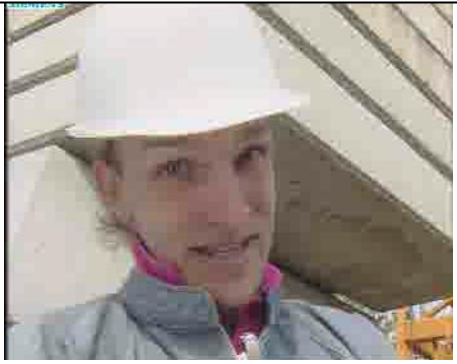
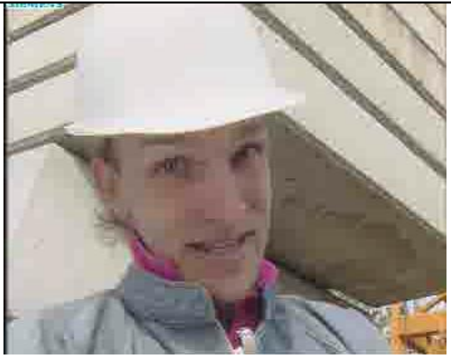
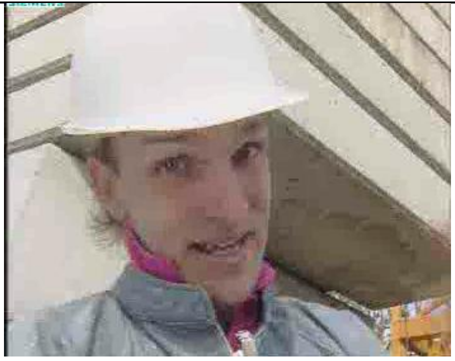
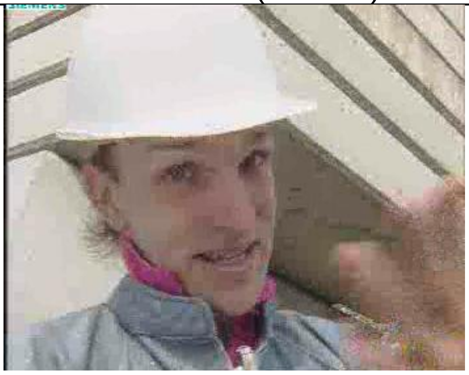
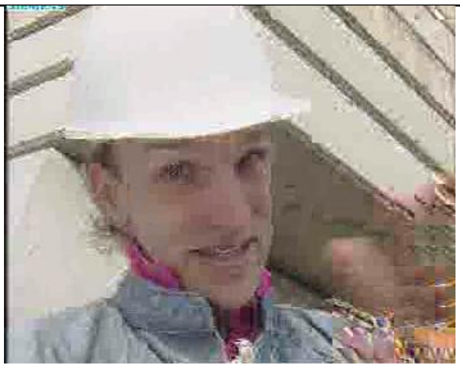
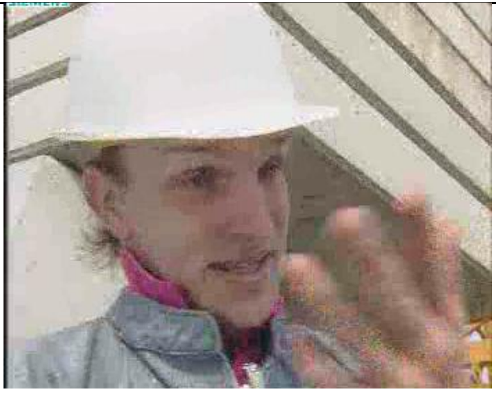
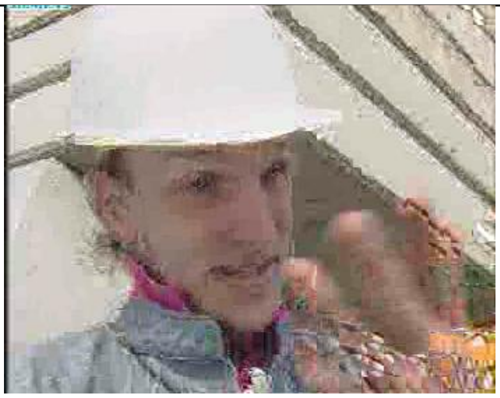
FFT of tampered “foreman” video (I-frame, #151 deleted)

Figure 6-2-1c

6.2.2 Deletion of a P-frame (frame #152) in a GOP in the video

Figure 6-2-2a below showed the screencaptures of frames #150 to #156 of the “foreman” video for both the non-tampered video version versus the tampered video version where a P-frame (frame #152) has been deleted.

Non-tampered	Tampered
 Frame #150 (P-frame)	 Frame #150 (P-frame)

Non-tampered	Tampered
 Frame #151 (I-frame)	 Frame #151 (I-frame)
 Frame #152 (P-frame)	Original P-frame (frame #152) deleted
 Frame #153 (P-frame)	 “New” Frame #152 (P-frame)
 Frame #154 (P-frame)	 “New” Frame #153 (P-frame)

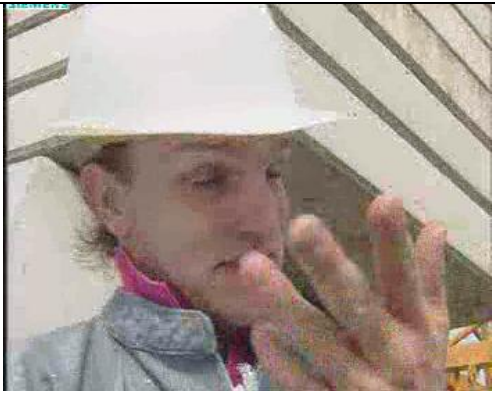
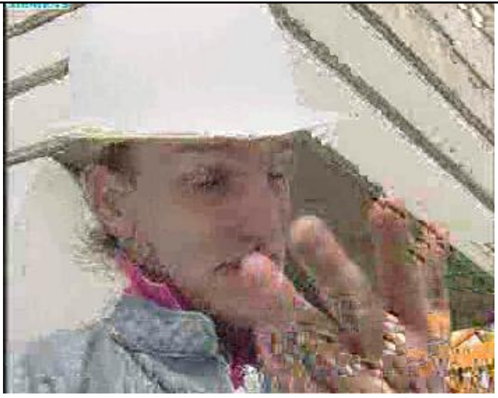
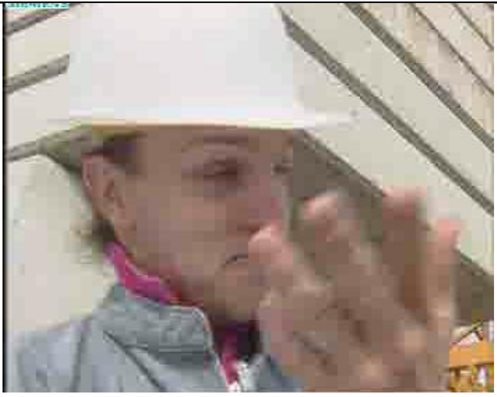
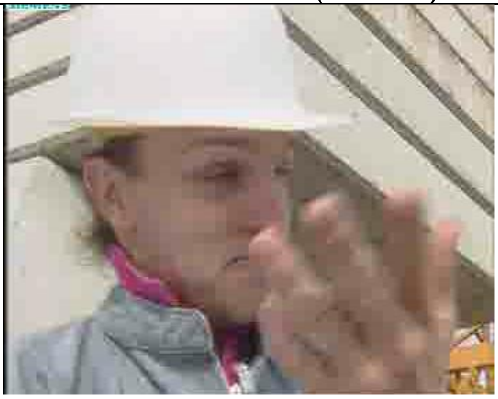
Non-tampered	Tampered
 Frame #155 (P-frame)	 “New” Frame #154 (P-frame)
 Frame #156 (I-frame)	 “New” Frame #155 (I-frame)

Figure 6-2-2a

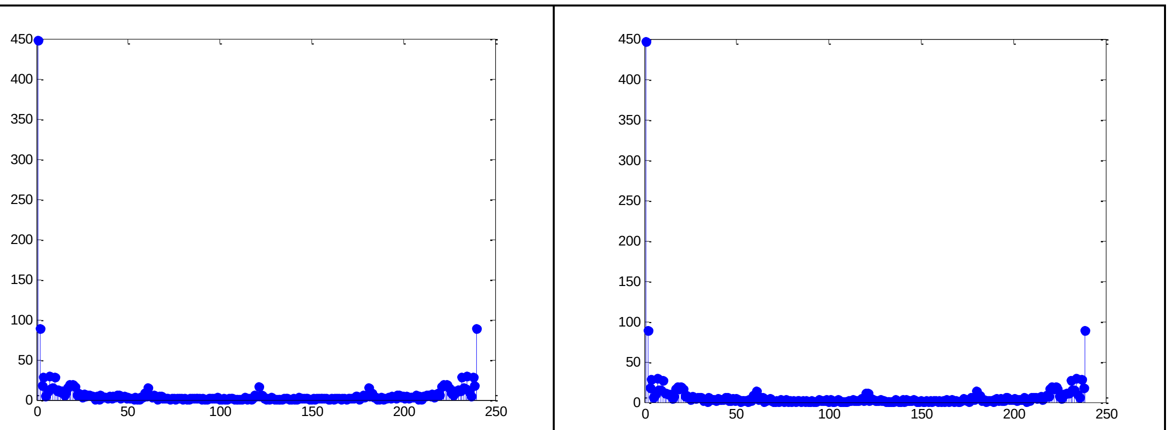
Figure 6-2-2b showed a comparison of the relevant frames’ mean motion errors for the non-tampered and tampered video that has a P-frame (frame #152) deleted.

Frame	Frame Type	Non-tampered	Frame	Frame Type	Tampered	Remarks
146	I	59.5836	146	I	59.5836	
147	P	1.5101	147	P	1.5101	
148	P	1.2456	148	P	1.2456	
149	P	1.2786	149	P	1.2786	
150	P	1.0139	150	P	1.0139	
151	I	58.5394	151	I	58.5394	
152	**P**	**1.6639**	152	P	2.4054	The old P-frame (#152) was deleted
153	P	2.4054	153	P	3.0604	
154	P	3.0604	154	P	2.805	
155	P	2.805	155	I	59.5208	
156	I	59.5208	156	P	3.1054	
157	P	3.1054	157	P	2.6855	
158	P	2.6855	158	P	2.3992	

Frame	Frame Type	Non-tampered	Frame	Frame Type	Tampered	Remarks
159	P	2.3992	159	P	1.3642	
160	P	1.3642	160	I	57.426	

Figure 6-2-2b

Figure 6-2-2c below shows the comparison of the FFT graphs for the mean motion errors of P-frames for both the non-tampered and tampered video that has a P-frame (frame #152) deleted.



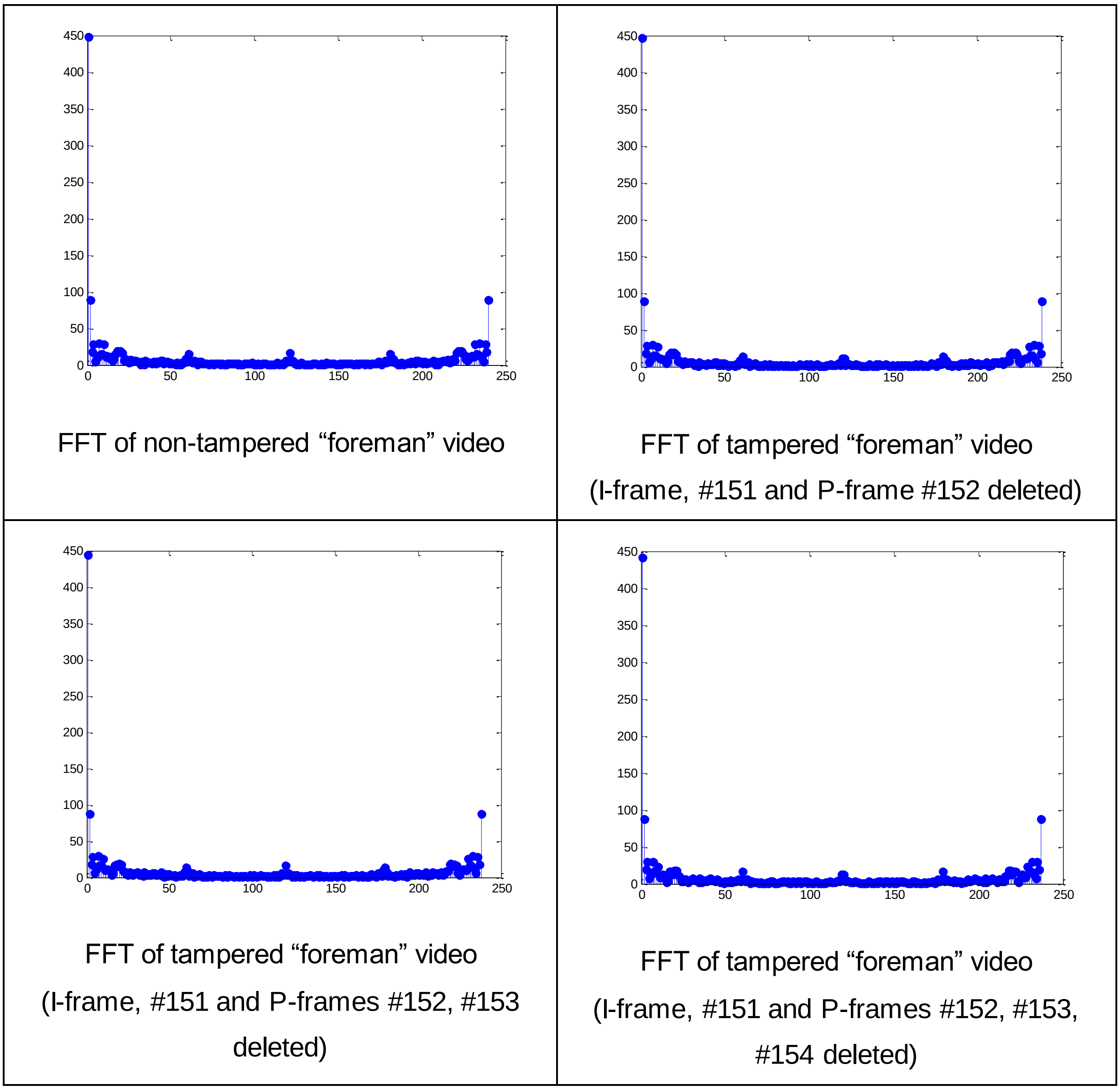
FFT of non-tampered "foreman" video

FFT of tampered "foreman" video (P-frame, #152 deleted)

Figure 6-2-2c

6.2.3 Deletion of a combination of I and P-frames within a GOP in the video

Figure 6-2-3 showed the FFT graphs for the mean motion errors of P-frames for the non-tampered and tampered videos (where a combination of I and P-frames were deleted).



FFT of non-tampered “foreman” video

FFT of tampered “foreman” video (I-frame, #151 and P-frame #152 deleted)

FFT of tampered “foreman” video (I-frame, #151 and P-frames #152, #153 deleted)

FFT of tampered “foreman” video (I-frame, #151 and P-frames #152, #153, #154 deleted)

Figure 6-2-3

6.2.4 Deletion of a GOP (frames #151 to #155) in the video

Figure 6-2-4a below shows the screencaptures of frames #150 to #156 of the “foreman” video for both the non-tampered video version versus the tampered video version where an entire GOP (frames #151 to #155) has been deleted.

Non-tampered	**Tampered**
 Frame #150 (P-frame)	 Frame #150 (P-frame)
 Frame #151 (I-frame)	A GOP (frames #151 to #155) deleted
 Frame #152 (P-frame)	A GOP (frames #151 to #155) deleted
 Frame #153 (P-frame)	A GOP (frames #151 to #155) deleted

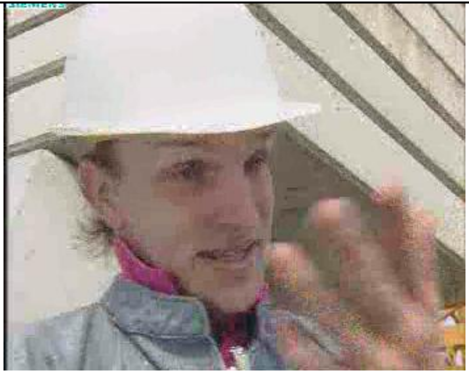
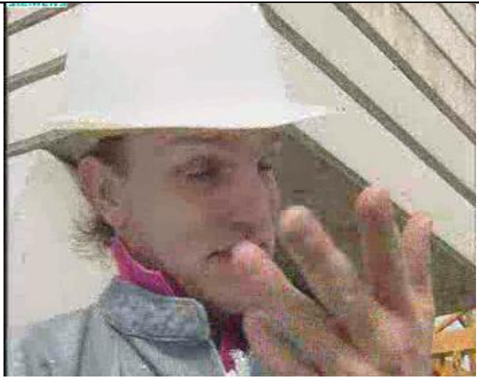
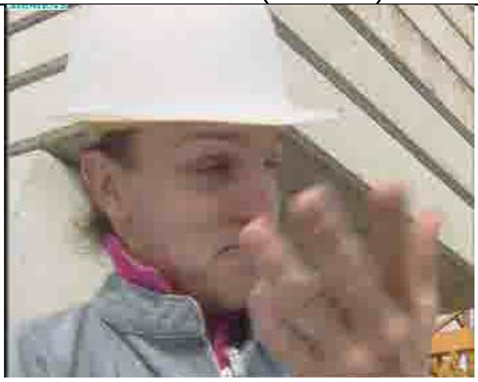
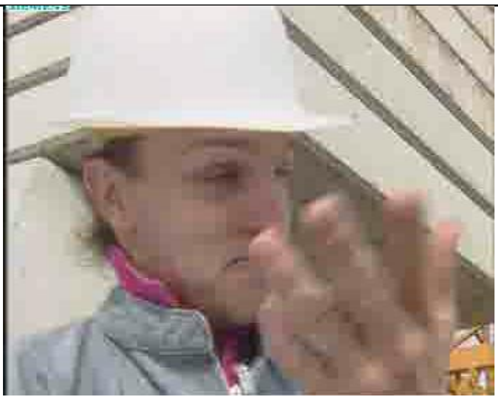
Non-tampered	Tampered
 Frame #154 (P-frame)	A GOP (frames #151 to #155) deleted
 Frame #155 (P-frame)	A GOP (frames #151 to #155) deleted
 Frame #156 (I-frame)	 “New” Frame #151 (I-frame)

Figure 6-2-4a

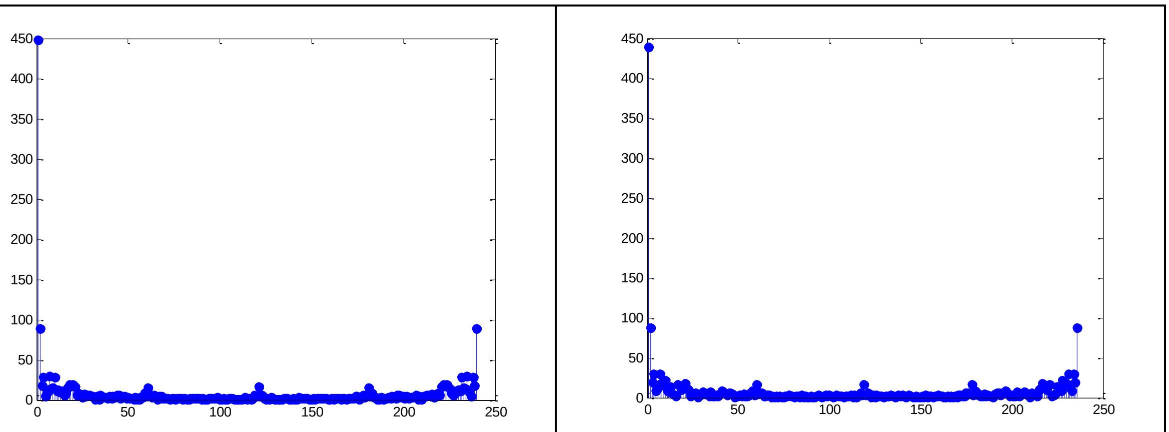
Figure 6-2-4b below showed a comparison of the relevant frames’ mean motion errors for the non-tampered and tampered video that has an entire GOP (frames #151 to #155) deleted.

Frame	Frame Type	Non-tampered	Frame	Frame Type	Tampered	Remarks
146	I	59.5836	146	I	59.5836	
147	P	1.5101	147	P	1.5101	
148	P	1.2456	148	P	1.2456	
149	P	1.2786	149	P	1.2786	
150	P	1.0139	150	P	1.0139	

Frame	Frame Type	Non-tampered	Frame	Frame Type	Tampered	Remarks
151	**I**	**58.5394**	151	I	59.5208	An entire GOP (frames #151 to #155) was deleted
152	**P**	**1.6639**	152	P	3.1054	
153	**P**	**2.4054**	153	P	2.6855	
154	**P**	**3.0604**	154	P	2.3992	
155	**P**	**2.805**	155	P	1.3642	
156	I	59.5208	156	I	57.426	
157	P	3.1054	157	P	1.4764	
158	P	2.6855	158	P	1.2569	
159	P	2.3992	159	P	1.0323	
160	P	1.3642	160	P	1.2345	

Figure 6-2-4b

Figure 6-2-4c below showed the comparison of the FFT graphs for the mean motion errors of P-frames of the non-tampered and tampered video that has an entire GOP (frames #151 to #155) deleted.



FFT of non-tampered “foreman” video

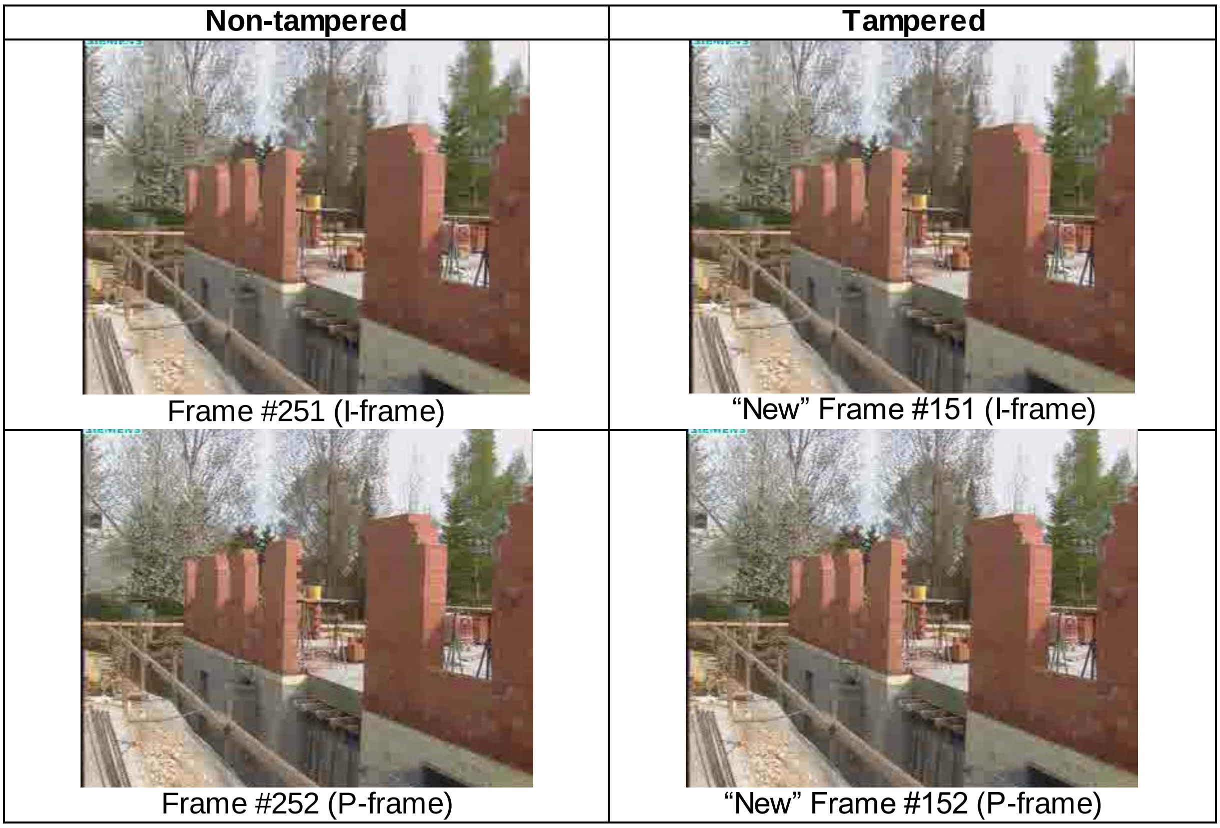
FFT of tampered “foreman” video (GOP, #151 to #155 deleted)

Figure 6-2-4c

6.2.5 Deletion of 1/3rd GOPs (#151 to #250) of the video

Figure 6-2-5a below showed the screencaptures of frames #150 to #152 and frames #250 to #252 of the “foreman” video for both the non-tampered video version versus the tampered video version where a significant number of GOPs (frame #151 to #250) has been deleted. In total, one third of the GOPs in the video were deleted.

Non-tampered	Tampered
 Frame #150 (P-frame)	 Frame #150 (P-frame)
 Frame #151 (I-frame)	1/3rd of GOP (frames #151 to #250) deleted
 Frame #152 (P-frame)	1/3rd of GOP (frames #151 to #250) deleted
 Frame #250 (P-frame)	1/3rd of GOP (frames #151 to #250) deleted



Non-tampered | **Tampered**

Frame #251 (I-frame)

“New” Frame #151 (I-frame)

Frame #252 (P-frame)

“New” Frame #152 (P-frame)

Figure 6-2-5a

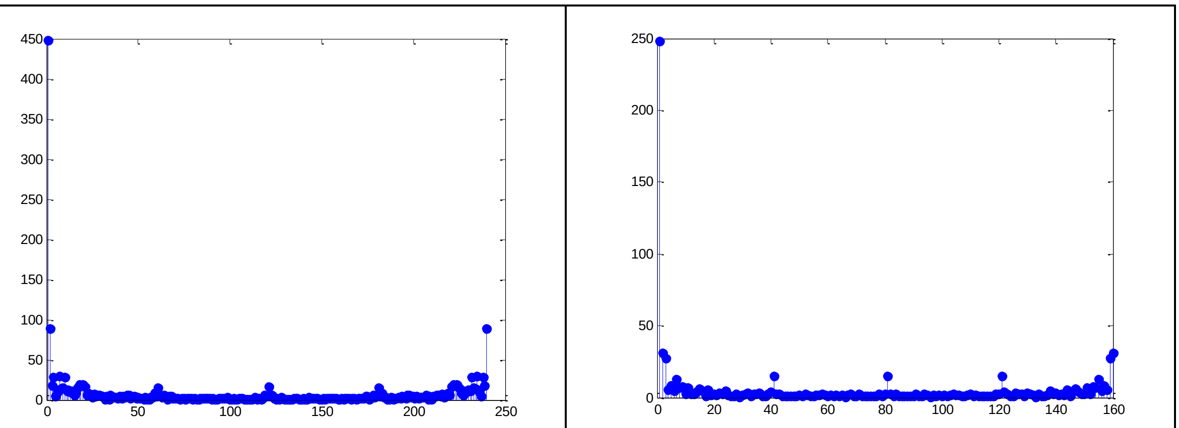
Figure 6-2-5b below showed a comparison of the relevant frames’ mean motion errors for the non-tampered and tampered video that has 1/3rd of the GOP (frame #151 to #250) in the video deleted.

Frame	**Frame Type**	**Non-tampered**	**Frame**	**Frame Type**	**Tampered**	**Remarks**
146	I	59.5836	146	I	59.5836	
147	P	1.5101	147	P	1.5101	
148	P	1.2456	148	P	1.2456	
149	P	1.2786	149	P	1.2786	
150	P	1.0139	150	P	1.0139	
151	**I**	**58.5394**	151	I	50.0767	1/3rd of GOP (frames #151 to #250) in the video was deleted
152	**P**	**1.6639**	152	P	2.7897	
153	**P**	**2.4054**	153	P	2.063	
154	**P**	**3.0604**	154	P	1.893	
155	**P**	**2.805**	155	P	2.2455	
…	**…**	**…**	…	…	…	
251	I	50.0767	196	I	50.1978	
252	P	2.7897	197	P	2.6882	
253	P	2.063	198	P	2.3282	

Frame	Frame Type	Non-tampered	Frame	Frame Type	Tampered	Remarks
254	P	1.893	199	P	2.1198	
255	P	2.2455	200	P	1.6994	

Figure 6-2-5b

Figure 6-2-5c below showed the comparison of the FFT graphs for the mean motion errors of P-frames of the non-tampered and tampered video that has 1/3rd of the GOP (frames #151 to #250) in the video deleted.



FFT of non-tampered “foreman” video

FFT of tampered “foreman” video (1/3rd of GOPs, #151 to #250 deleted)

Figure 6-2-5c

6.3 Tampering encoded video – deleting frames with different frame pattern sequences

This section will show the results of deleting frames (from the initial I-frame to the entire GOP) from the “foreman” video that is encoded with the following frame pattern sequence:

- Pattern #1 – “I PPPP” – starts from frame #151.
- Pattern #2 – “I PPPP PPPP” – starts from frame #154.
- Pattern #3 – “I PPPP PPPP PPPP” – starts from frame #157.

As mentioned in Section 5.3 (Tampering encoded video – deleting frames with different frame pattern sequences), the deletion of frames for each of the above frame pattern sequence starts from the initial I-frame of a GOP. Given the different encoding frame pattern sequences, the initial I-frame number for each of the patterns will be different. The deletion of frames will first start from the initial I-frame and then continue with the I-frame and its neighbouring P-frame and so on and so forth until the entire GOP is deleted.

However, for Pattern #1, the deletion of frames was already performed in the previous section, namely Section 5.2 (Tampering encoded video – deleting frames). Hence, the experiment will not be performed again for Pattern #1. Instead, the results of Pattern #1 are available in the previous section as shown in the following:

- Figure A-1 in Appendix 1 (Mean Motion Errors of Frames) showed the entire mean motion errors for all the frames for the "foreman" video.
- Figure 6-2-1c showed the FFT graph of the mean motion error of P-frames for the "foreman" video with an I-frame (frame #151) deleted.
- Figure 6-2-3 shows the FFT graphs of the mean motion error of P-frames for the "foreman" video with the deletion of a combination of I and P-frames:
  - Deletion of I and P-frames (frames #151 and #152).
  - Deletion of I and 2 P-frames (frames #151, #152 and #153).
  - Deletion of I and 3 P-frames (frames #151, #152, #153 and #154).
- Figure 6-2-4c shows the FFT graph of the mean motion error of P-frames for the "foreman" video with the deletion of the entire GOP (frames #151 to #155).

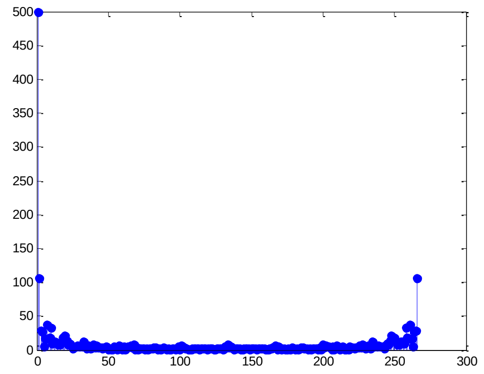
Figure A-2 and A-3 in Appendix 1 (Mean Motion Errors of Frames) showed the mean motion errors of the frames for the "foreman" video that is encoded with the frame pattern sequence "IPPPP PPPP" and "IPPPP PPPP PPPP" respectively. Figure 6-3-1 shows a sample of the mean motion errors of the frames for both pattern sequences.

"IPPPP PPPP"			**"IPPPP PPPP PPPP"**		
Frame	**Frame Type**	**Foreman**	**Frame**	**Frame Type**	**Foreman**
151	P	1.2702	151	P	1.3595
152	P	1.6149	152	P	1.6625
153	P	2.5784	153	P	2.6107
154	**I**	**59.6739**	154	P	3.1799
155	**P**	**2.7775**	155	P	3.0006
156	**P**	**2.1256**	156	P	2.5035
157	**P**	**3.1512**	157	**I**	**58.1066**
158	**P**	**2.6859**	158	**P**	**2.7027**
159	**P**	**2.4297**	159	**P**	**2.3956**
160	**P**	**1.4427**	160	**P**	**1.236**
161	**P**	**1.3721**	161	**P**	**1.1817**

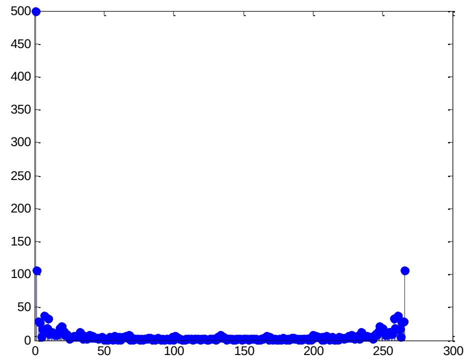
162	**P**	**1.3215**	162	**P**	**1.1916**
163	I	57.5552	163	**P**	**1.2923**
164	P	1.5058	164	**P**	**1.1505**
165	P	1.2765	165	**P**	**1.3065**
166	P	1.058	166	**P**	**1.1528**
167	P	1.202	167	**P**	**1.4267**
168	P	1.439	168	**P**	**1.5613**
169	P	1.4605	169	**P**	**1.6602**
170	P	1.6599	170	I	57.1312

Figure 6-3-1

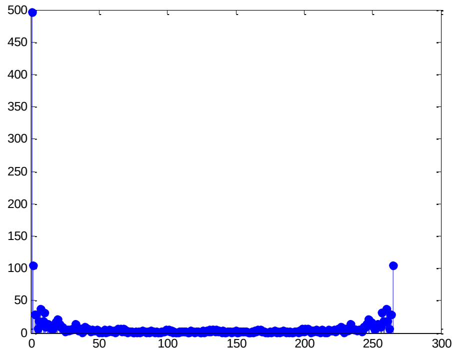
Figure 6-3-2 below showed the FFT graphs of the mean motion errors of P frames for the deletion of frames from the video that is encoded with the frame pattern sequence "IPPPP PPPP".



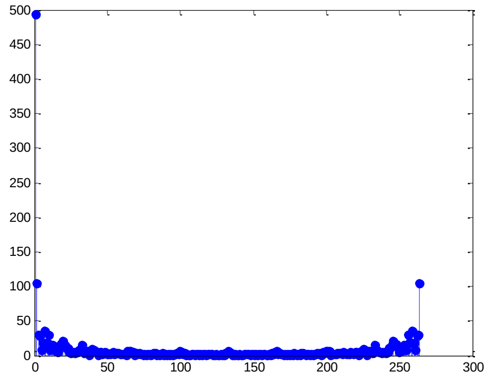
FFT of non-tampered "foreman" video



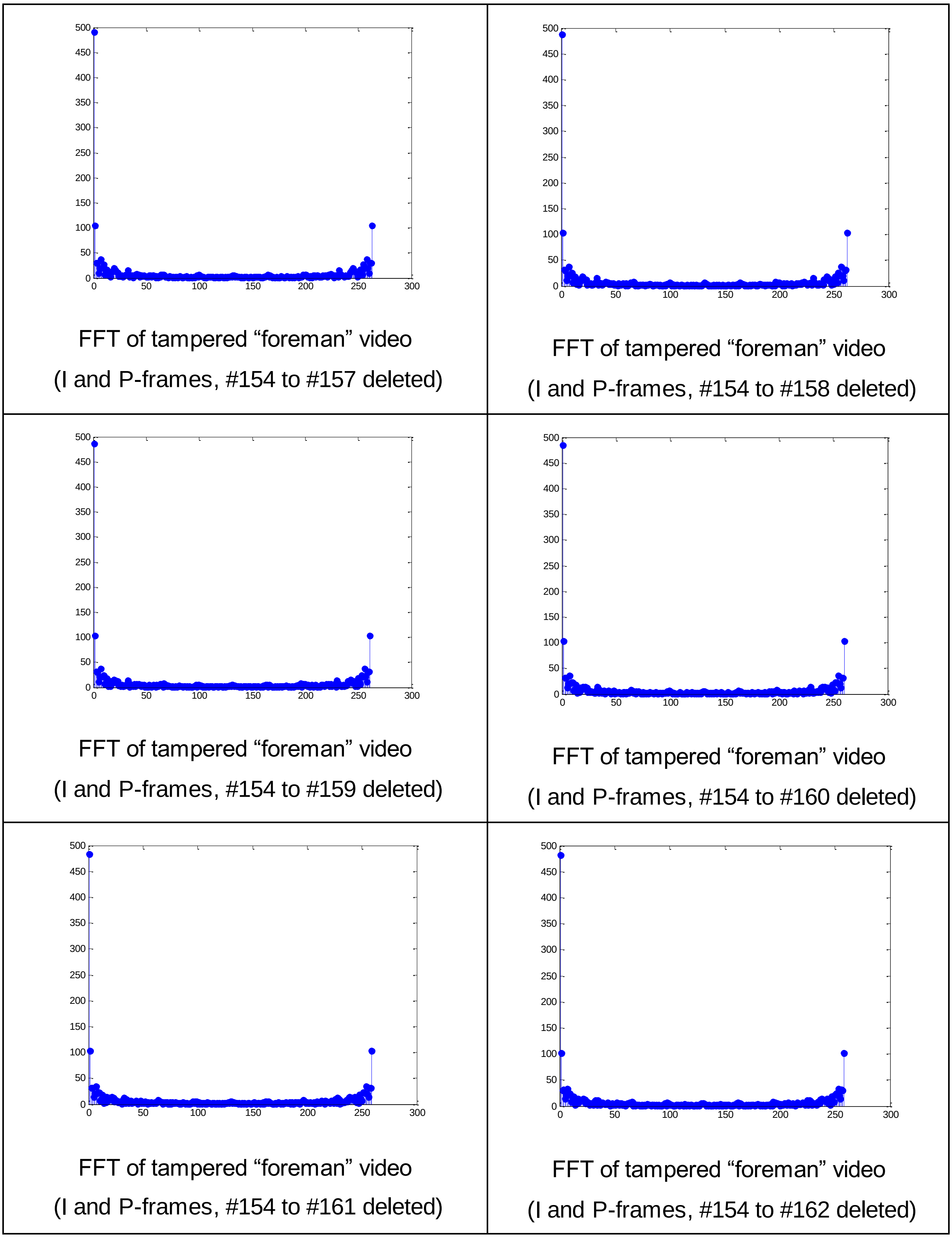
FFT of tampered "foreman" video
(I-frame, #154 deleted)



FFT of tampered "foreman" video
(I and P-frames, #154 to #155 deleted)



FFT of tampered "foreman" video
(I and P-frames, #154 to #156 deleted)



FFT of tampered “foreman” video

(I and P-frames, #154 to #157 deleted)

FFT of tampered “foreman” video

(I and P-frames, #154 to #158 deleted)

FFT of tampered “foreman” video

(I and P-frames, #154 to #159 deleted)

FFT of tampered “foreman” video

(I and P-frames, #154 to #160 deleted)

FFT of tampered “foreman” video

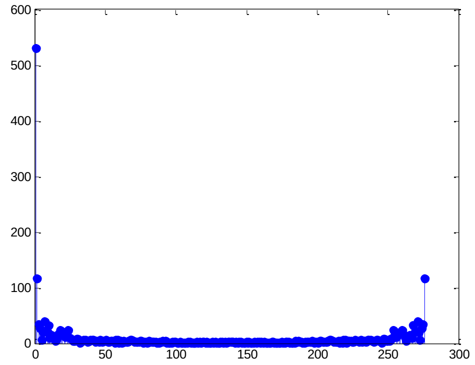
(I and P-frames, #154 to #161 deleted)

FFT of tampered “foreman” video

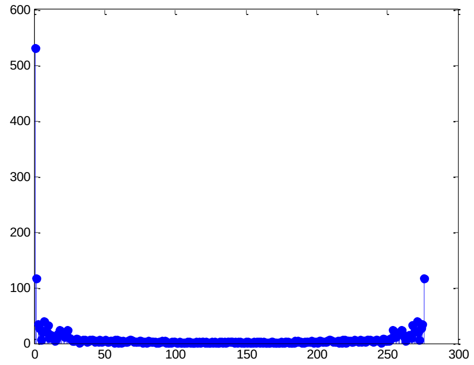
(I and P-frames, #154 to #162 deleted)

Figure 6-3-2

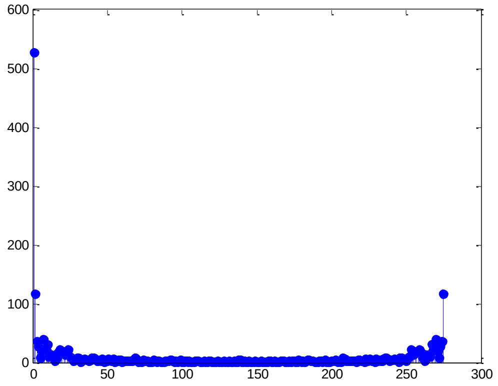
Figure 6-3-3 below showed the FFT graphs of the mean motion errors of P frames for the deletion of frames from the video that is encoded with the frame pattern sequence “IPPPP PPPP PPPP”.



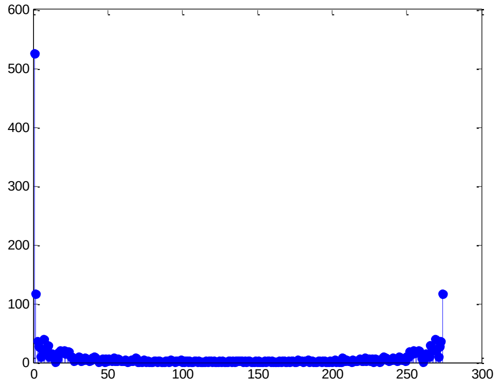
FFT of non-tampered “foreman” video



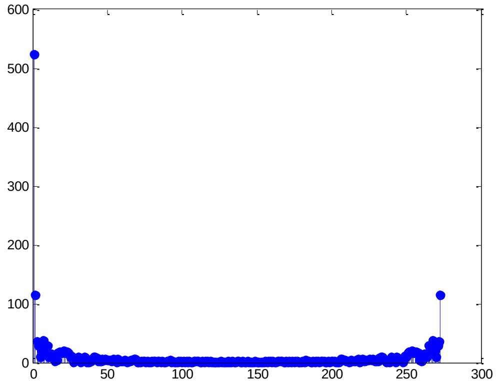
FFT of tampered “foreman” video (I-frame, #157 deleted)



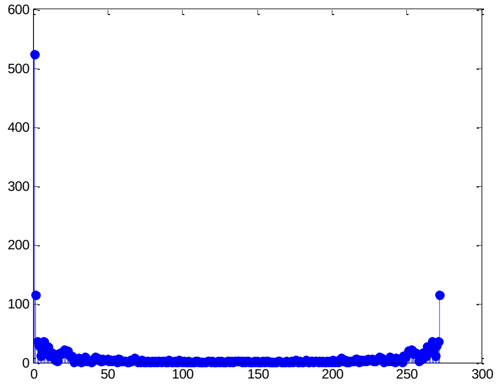
FFT of tampered “foreman” video (I and P-frames, #157 to #158 deleted)



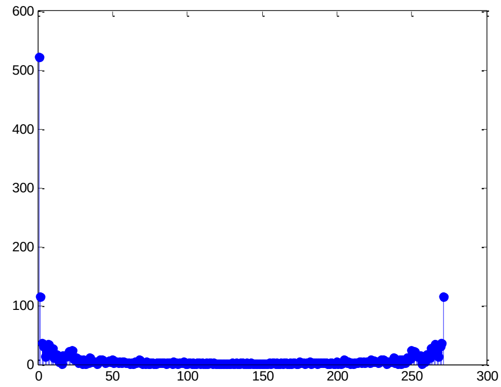
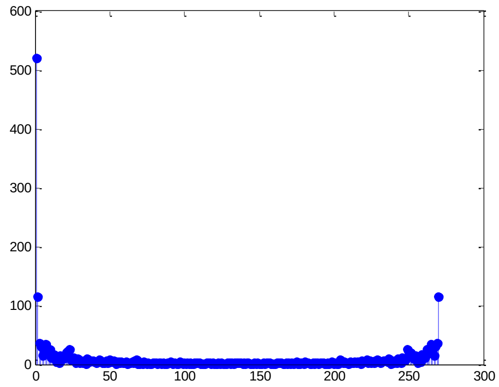
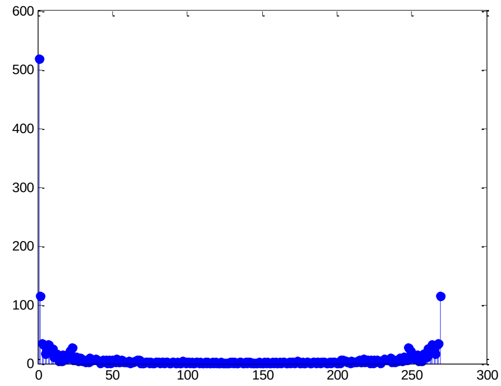
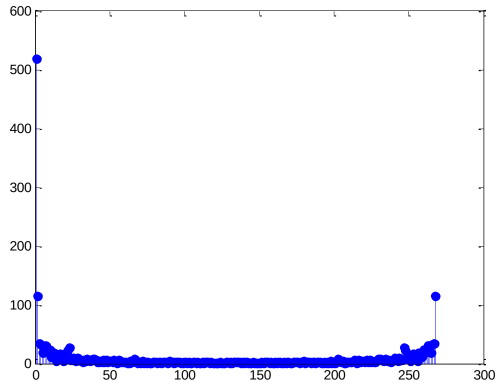
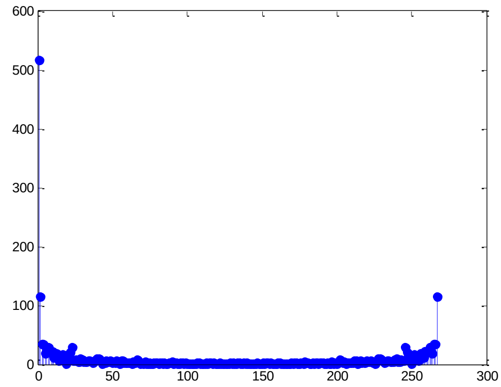
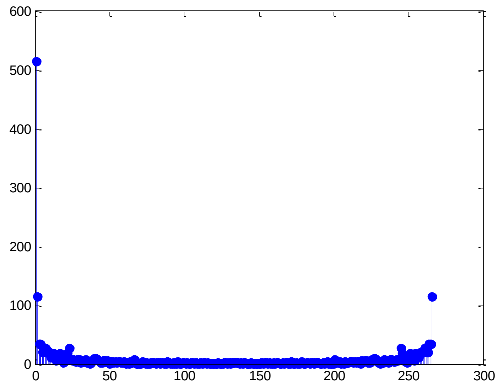
FFT of tampered “foreman” video (I and P-frames, #157 to #159 deleted)

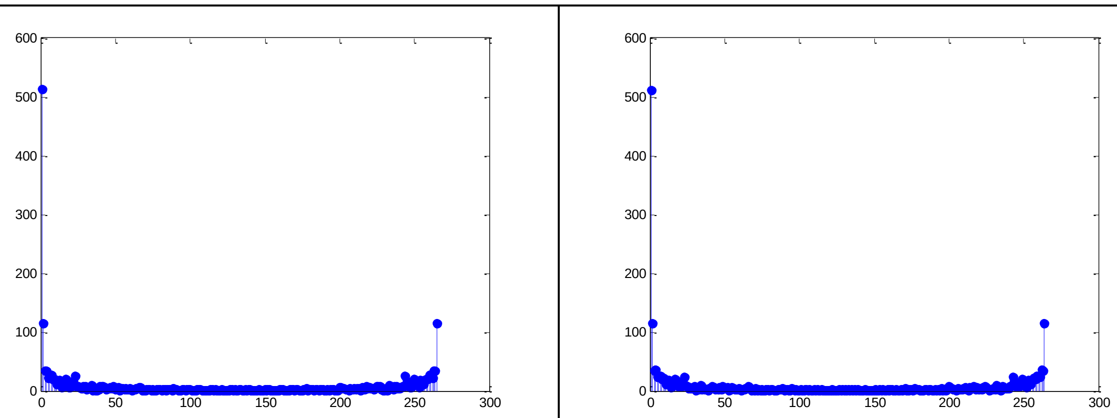


FFT of tampered “foreman” video (I and P-frames, #157 to #160 deleted)



FFT of tampered “foreman” video (I and P-frames, #157 to #161 deleted)

 FFT of tampered "foreman" video (I and P-frames, #157 to #162 deleted)	 FFT of tampered "foreman" video (I and P-frames, #157 to #163 deleted)
 FFT of tampered "foreman" video (I and P-frames, #157 to #164 deleted)	 FFT of tampered "foreman" video (I and P-frames, #157 to #165 deleted)
 FFT of tampered "foreman" video (I and P-frames, #157 to #166 deleted)	 FFT of tampered "foreman" video (I and P-frames, #157 to #167 deleted)



FFT of tampered “foreman” video
(I and P-frames, #157 to #168 deleted)

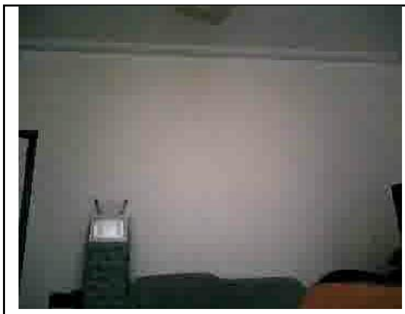
FFT of tampered “foreman” video
(I and P-frames, #157 to #169 deleted)

Figure 6-3-3

6.4 Tampering encoded video – duplicating frames

As mentioned in Chapter 5 (Design of Experiments), the following showed the screen captures of the following GOPs so as to allow a visual understanding of the GOPs in the “hand” video that will be involved in the following experiment to duplicate frames.

- Figure 6-4-1a: GOP Set A with frames #91 to #95 – empty stationary scene.
- Figure 6-4-1b: GOP Set B with frames #96 to #100 – the hand entering the video.
- Figure 6-4-1c: GOP Set C with frames #201 to #205 –the hand leaving the video.



Frame #91

Frame #92

Frame #93

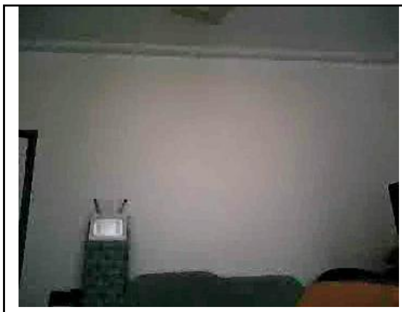
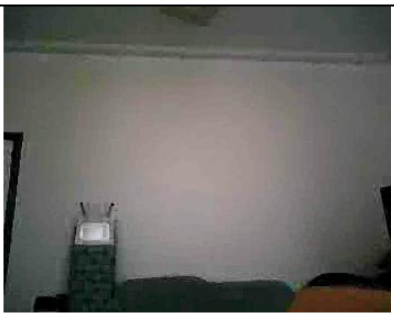
 Frame #94	 Frame #95	

Figure 6-4-1a

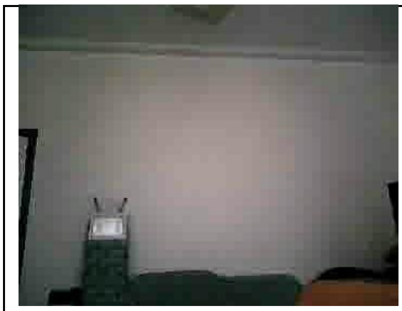
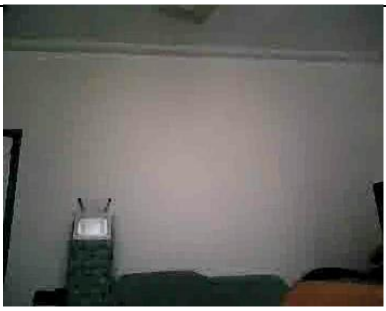
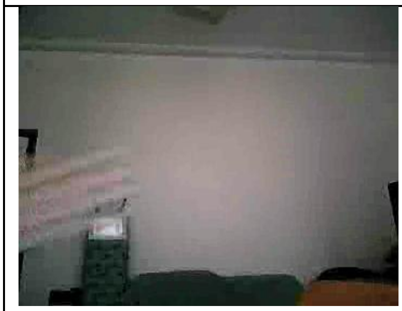
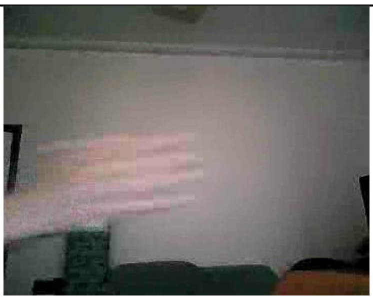
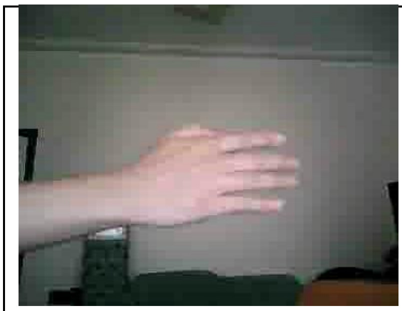
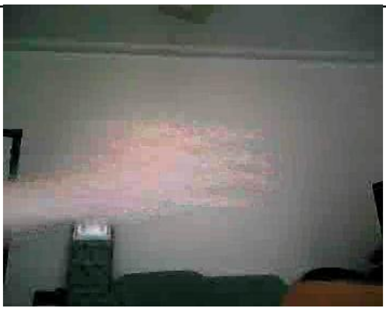
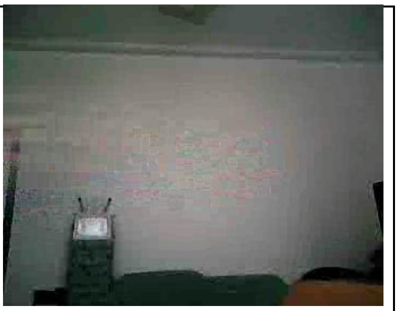
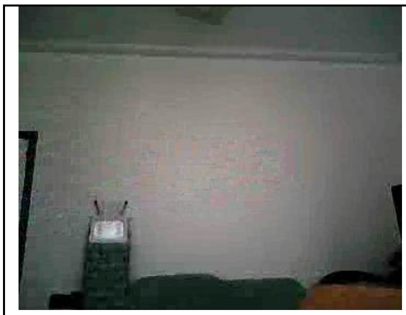
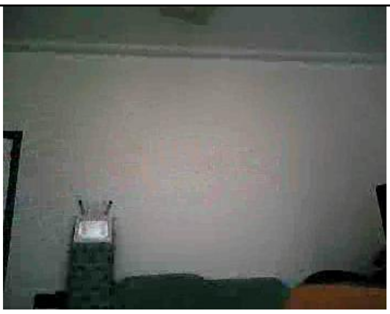
 Frame #96	 Frame #97	 Frame #98
 Frame #99	 Frame #100	

Figure 6-4-1b

 Frame #201	 Frame #202	 Frame #203



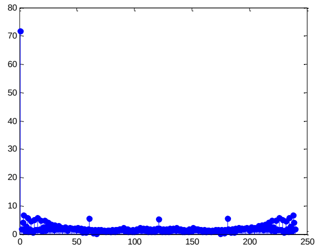
Frame #204



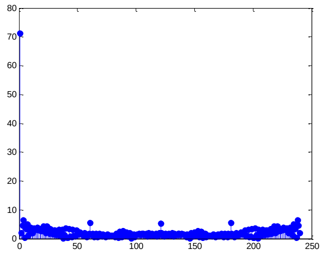
Frame #205

Figure 6-4-1c

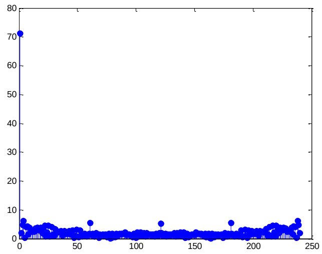
Figure 6-4-2 showed the FFT graphs of the mean motion errors of P-frames for the duplication of frames/GOPs with the empty stationary scene (GOP with frames #91 to #95) as mentioned in Figure 5-4 of Chapter 5 (Design of Experiments).



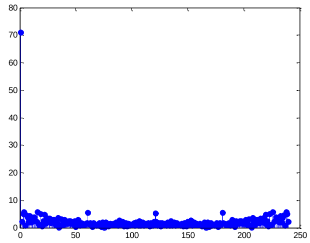
FFT of frames #96 to #105



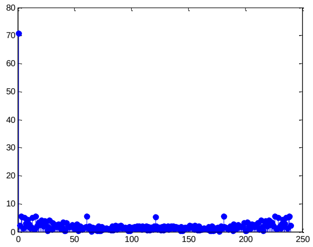
FFT of frames #96 to #110



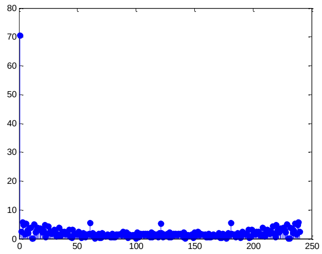
FFT of frames #96 to #115



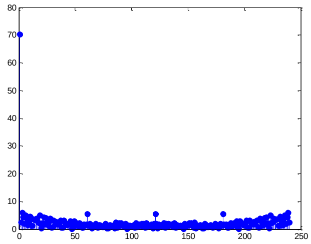
FFT of frames #96 to #120



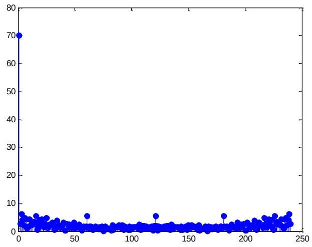
FFT of frames #96 to #125



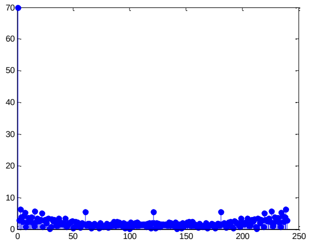
FFT of frames #96 to #130



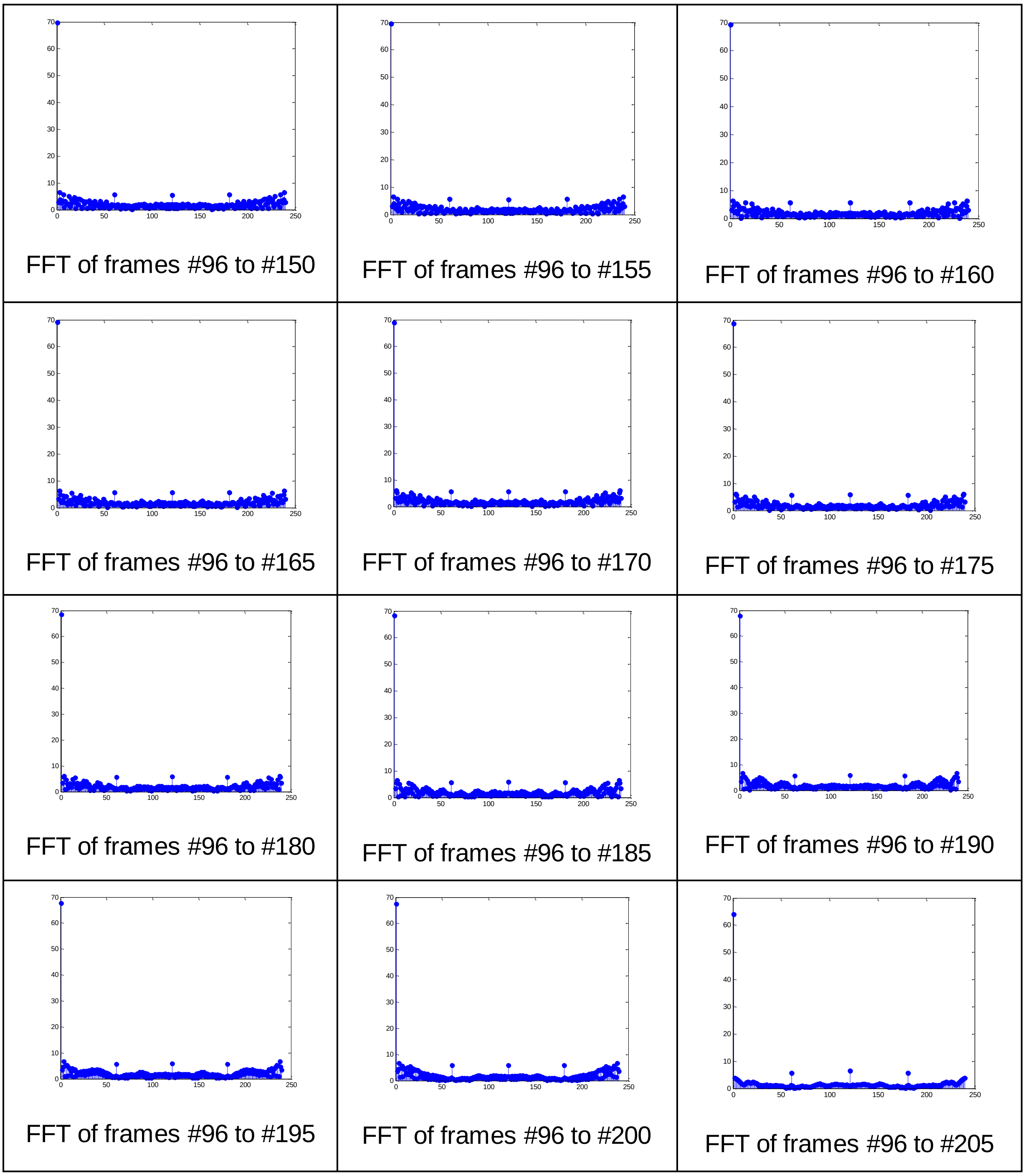
FFT of frames #96 to #135



FFT of frames #96 to #140



FFT of frames #96 to #145



FFT of frames #96 to #150

FFT of frames #96 to #155

FFT of frames #96 to #160

FFT of frames #96 to #165

FFT of frames #96 to #170

FFT of frames #96 to #175

FFT of frames #96 to #180

FFT of frames #96 to #185

FFT of frames #96 to #190

FFT of frames #96 to #195

FFT of frames #96 to #200

FFT of frames #96 to #205

Figure 6-4-2

The analysis of the results of the various experiments will be presented in the next chapter, Chapter 7 (Discussion).

Chapter 7: Discussion

This chapter will present an analysis of the results of the experiments discussed in Chapter 6 (Results of Experiments).

7.1 Non-tampered encoded videos

In this section, 2 of the videos, namely, the “foreman” and “hand” videos were introduced because the tampering of videos will be based on these 2 videos. Figure 6-1-1a and Figure 6-1-1b showed the key frames of the “foreman” and “hand” videos respectively in order to give an idea of how the videos look like visually. In the subsequent experiments, certain tampering will result in an obvious change in the visual aspect of the videos, thereby alerting a forensics investigator to the tampering of the videos.

In Figure 6-1-1a, the “foreman” video showed a foreman who is talking and pointing subsequently to a construction site which is located behind the foreman. The video started off by showing the foreman and then pans to the right to show the construction site. In Figure 6-1-1b, the “hand” video shows an empty stationary scene (for the first one-third of the video) followed by a hand entering the scene (for the first second-third of the video) and then an empty stationary scene (for the final one-third of the video). It attempts to simulate the recording of a CCTV camera where there are no subject (in this case, it’s a hand) in the scenes for most part of the recordings followed by the appearance (and subsequent disappearance) of the subject (i.e., the hand).

As seen in Figure 6-1-2, the mean motion errors of the I-frames for the “foreman” and “hand” videos are in the range of “50-60” and “30-40” respectively. In contrast, the mean motion errors of the P-frames for the “foreman” and “hand” videos are in the range of “1-4” and “0.1-2” respectively. The mean motion errors of the P-frames are very small compared to the I-frames because the motion errors of the P-frames are made in reference to the nearest preceding I or P-frames and the movement of both videos are relatively smooth and thus the scene change for the frames are not very drastic. The mean motion errors of the P-frames for the “foreman” video is much higher than that for the “hand” video because there is more movement in the “foreman” video (e.g. foreman talking, camera pans to construction site, etc.) compared to the “hand” video which is largely stationary (except for the hand entering and leaving the video). The mean motion error for the I-frames are much larger as they form the point of reference for the P-frames and are independent JPEG pictures which means that their motion errors will be large since they are not referenced against any frames.

Figure 6-1-3 showed the FFT graphs of the mean motion errors of P-frames for both videos. As can be seen from Figure 6-1-3, there were no large spikes in the FFT graphs for both the non-tampered videos. These graphs will serve as a point of reference when the videos are subsequently tampered through the deletion and duplication of frames. According to [WW06], the FFT graphs will show large spikes when the videos are re-encoded again as long as the deletion of frames is not a multiple of the GOP length.

7.2 Tampering encoded video – deleting frames

This section will discuss the results of the investigation over the deletion of frames in the “foreman” video where the frames are deleted in following manner:

- The deletion of an I-frame (frame #151) in a GOP in the video.
- The deletion of a P-frame (frame #152) in a GOP in the video.
- The deletion of a combination of I and P-frames within a GOP in the video:
    - Frames #151, #152
    - Frames #151, #152, #153
    - Frames #151, #152, #153, #154
- The deletion of a GOP (frames #151 to #155) in the video.
- The deletion of 1/3rd of the GOPs (frames #151 to #250) in the video.

7.2.1 Deletion of an I-frame (frame #151) in a GOP in the video

Figure 6-2-1a showed the screencaptures of the frames #150 to #156 for both the non-tampered (left column) and tampered (right column) videos. It can be seen that in the tampered video, the “new” frame #151 showed a “blockier” image compared to the similar frame in the non-tampered video. This is because the original I-frame (frame #151) which was used as the point of reference was deleted. The motion vectors in the P-frame ((the “new” frame 151) were applied on frame #150 which was a P-frame instead of an I-frame (which was deleted). The result is a decoded P-frame (the “new” frame #151) with a more “inaccurate” picture and hence producing a “blockier” image. The same effect happened to the rest of the P-frames in the GOP (i.e., the “new” frames #152 to #154). The “new” frame

#155 which is a new I-frame stopped the effect of this “blocky” image. The rest of the frames in the video appear normally from this point onwards.

Figure 6-2-1b showed the mean motion errors of the frames for both the non-tampered and tampered videos. It can be seen that the values of the motion errors of the frames for the “new” frame #151 (in the tampered video) is identical to that of frame #152 (in the non-tampered video). The values of the rest of the motion errors in the frames in the tampered video after frame #151 is the same as that from frame #152 onwards (see shaded cells) in the non-tampered video. This is because in the tampered video, the “old” frame #151 (an I-frame – bold text) was deleted. As such, the decoder simply interprets the next frame (the “old” frame #152 which is a P-frame) as the replacement or “new” frame #151.

Figure 6-2-1c showed the FFT graphs of the mean motion errors of the P-frames for both the non-tampered and tampered videos. Since the values of the mean motion errors of the FFT graphs are taken only from P-frames, this means that the FFT graphs for both the non-tampered and tampered videos in Figure 6-2-1c are actually identical as only an I-frame was deleted and I-frames are ignored in the FFT graphs. The FFT graphs are similar to that presented in Figure 3-3-2a of [WW06]. Therefore, the deletion of a single I-frame cannot be detected at all using the methodology described by [WW06] since the methodology does not involve I-frames.

However, visually, as seen in Figure 6-2-1a, the “new” frame #151 is “blockier” in image quality and as such, this tampering can be easily detected by a forensics investigator with a sharp eye. In addition, a forensics investigator can also easily notice that the frame pattern sequence in the GOP is not being observed in the tampered video. The “foreman” video was encoded using the frame pattern sequence of “IPPPP”. However, as can be seen in Figure 6-2-1b, this frame pattern sequence in the tampered video is now changed to “PPPP” (see the “new” frames #151 to #154) and does not follow the expected frame pattern sequence of “IPPPP”. A forensics investigator can easily conclude that an I-frame was deleted away and this caused the change in the image quality as well as the change in the frame pattern sequence.

7.2.2 Deletion of a P-frame (frame #152) in a GOP in the video

Figure 6-2-2a showed the screencaptures of the frames #150 to #156 for both the non-tampered (left column) and tampered (right column) videos. It can be seen that similar to the experiment where an I-frame was deleted in the tampered video, the "new" frame #152 showed a "blockier" image compared to the similar frame in the non-tampered video. This is because the original P-frame (frame #152) which was used as the point of reference by the successive P-frame was deleted. The motion vectors in the "new" P-frame #152 were then erroneously applied on the I-frame #151 when it should have been applied on the deleted P-frame #152. This erroneous decoding of the "new" P-frame #152 caused the image to appear "blocky". The error is propagated to the rest of the P-frames in the GOP. The "new" frame #155 which is a new I-frame stopped the effect of this "blocky" image. The rest of the frames in the video appear normally from this point onwards.

Figure 6-2-2b showed the mean motion errors of the frames for both the non-tampered and tampered videos. It can be seen that the values of the motion errors of the frames for the "new" frame #152 (in the tampered video) is identical to that of frame #153 (in the non-tampered video). The values of the rest of the motion errors in the frames in the tampered video after frame #152 is the same as that from frame #153 onwards (see shaded cells) in the non-tampered video. This is because in the tampered video, the "old" frame #152 (a P-frame – bold text) was deleted. As such, similarly to the previous experiment where an I-frame was deleted, the decoder simply interprets the next frame (the "old" frame #153 which is a P-frame) as the replacement or "new" frame #152.

Figure 6-2-2c showed the FFT graphs of the mean motion errors of the P-frames for both the non-tampered and tampered video that has a P-frame (frame #152) deleted. Unlike the FFT graphs in Figure 6-2-1c, the FFT graphs in Figure 6-2-2c are not exactly the same since in the case of the deletion of a P-frame (frame #152), the mean motion error of a P-frame (frame #152) is now included in the calculation of the FFT. However, Figure 6-2-2c showed that the omission of a single P-frame from the FFT graph is not significant enough to show any difference from the FFT graph of a non-tampered video. Again, there are no unusual spikes as indicated in Figure 3-3-2a of [WW06]. Therefore, similar to the deletion of an I-frame (per previous experiment), the deletion of a single P-frame cannot be detected using the methodology described by [WW06].

However, visually, the deletion of an I-frame or that of a P-frame both showed obvious visual differences (e.g., "blocky" images) and as such, this kind of tampering can be easily detected by a forensics investigator. In addition, similar to the previous experiment, the frame pattern sequence in the GOP is again not being observed in the tampered video. In this case, as seen in Figure 6-2-2b, the frame pattern sequence from frames #151 to #154 is now "IPPP" instead of the expected frame pattern sequence of "IPPPP". The forensics investigator can easily conclude that a P-frame was deleted away and this caused the change in the image quality as well as the change in the frame pattern sequence.

7.2.3 Deletion of a combination of I and P-frames within a GOP in the video

Figure 6-2-3 showed the FFT graphs for the mean motion errors of the P-frames for the non-tampered video as well as the tampered videos where a combination of I and P-frames within a GOP were deleted. It can be seen that the FFT graphs are visually similar to one another. There are no unusual spikes as indicated in Figure 3-3-2a of [WW06]. Therefore, the deletion of a combination of frames, cannot be detected using the methodology described by [WW06].

Visually, the screen captures are not provided but it should be obvious that the affected P-frames within the GOP of frames #151 to #155 would show degradation in the image quality. As such, a forensics investigator would be able to detect that the videos have been tampered. In addition, the frame sequence patterns for the affected GOP would also be affected, i.e., instead of expecting the "IPPPP" frame pattern sequence, the tampered videos would show "PPP", "PP" and "P" frame pattern sequences for the respective deletion of frames in the combination below:

- Deletion of frames #151, #152
- Deletion of frames #151, #152, #153
- Deletion of frames #151, #152, #153, #154

Similarly, the mean motion errors of P-frames are not provided here but again, it should be obvious that the deletion of the frames simply means that the decoder will interpret the next available frame as the "new" frame as evident in Figure 6-2-1b and 6-2-2b where an I-frame and P-frame were deleted respectively.

7.2.4 Deletion of a GOP (frames #151 to #155) in the video

Figure 6-2-4a showed the screencaptures of the frames #150 to #156 for both the non-tampered (left column) and tampered (right column) videos. It can be seen that in the tampered video, unlike the previous 3 sections (Section 7.2.1 to 7.2.3) where a combination of I-frame and/or P-frames were deleted, the image quality of the “new” frame #151 is identical to its counterpart in the non-tampered video. This is because while an entire GOP (frames #151 to #155) were deleted, the “new” frame #151 in the tampered video is an I-frame which is an independent JPEG picture and as such, there is no loss in image quality although a series of prior frames (frames #151 to #155) were deleted.

Nevertheless, despite the lack of “blocky” images, a forensics investigator with a sharp eye may be able to notice that the video sequences in the tampered video are not smooth. In particular, in the non-tampered video, the foreman is seen to slowly raise his hand (frames #151 to #155). However, in the tampered video, frame #150 shows the foreman talking while the “new” frame #151 shows the foreman’s hand suddenly appearing without any prior hand movements.

Figure 6-2-4b showed the mean motion errors of the frames for both the non-tampered and tampered videos. It can be seen that the values of the motion errors of the frames for the “new” frame #151 (in the tampered video) is identical to that of frame #156 (in the non-tampered video). Again, similar to the results of the previous 3 sections (Section 7.2.1 to 7.2.3) the decoder simply interprets the next frame (the “old” frame #156) as the replacement or “new” frame #151. However, unlike the previous 3 sections, in terms of the frame pattern sequences, there are no “odd” frame pattern sequences since an entire GOP (frame pattern sequence of “IPPPP”) was deleted.

Figure 6-2-4c showed the FFT graphs of the mean motion errors of the P-frames for both the non-tampered and tampered video that has an entire GOP (frames #151 to #155) deleted. Despite the deletion of 4 P-frames, the FFT graphs for the non-tampered and tampered videos are quite similar. This is in line with the expected results indicated in [WW06] where an acknowledged weakness of his proposed methodology is that it cannot detect tampering when the number of inserted/deleted frames is a multiple of the GOP length.

Therefore, it seems that whether a frame (I or P-frame) or a combination of frames or an entire GOP are being deleted, there are no obvious changes in the FFT graphs of the mean motion errors for the P-frames for the tampered videos. This is contrary to the results reported by [WW06].

Visually, the detection can be obvious to a forensics investigator as the image will either appear “blocky” (for deletion of frames within a GOP) or there is an abrupt appearance of the subject (for deletion of GOP). However, in the latter case, such abruptness may not be necessarily noticeable in all videos, e.g., the subject in focus was already in the scene and did not have any prior movements or perhaps the subject together with its entering and leaving the scene was also removed. In such cases, a forensics investigator would not be able to detect any abnormalities in the video.

For example, as mentioned in Chapter 5 (Design of Experiments), the “hand” video (which simulates a stationary CCTV) consists of largely 3 sets of scenes (refer to Figure 6-1-1b):

- Set 1 – first 1/3rd of the video showing an empty stationary scene
- Set 2 – second 1/3rd of the video showing a hand entering the scene, the hand staying motionless in the scene and finally exiting the scene.
- Set 3 – final 1/3rd of the video showing again an empty stationary scene as seen earlier in the video.

If the attacker were to completely delete Set 2, then visually, the forensics investigator would not be able to notice any visual abnormalities (e.g., not “block” images or subjects suddenly appearing in the scenes, etc.). However, in such cases, more than 1 GOP are being deleted and as such, the FFT graphs might not be as similar as reported in Figure 6-2-4c.

The next part of the experiment will investigate the results of the FFT graphs when a significant number of GOPs are being deleted in a video.

7.2.5 Deletion of 1/3rd GOPs (#151 to #250) of the video

Figure 6-2-5a showed the screencaptures of the frames #150 to #156 for both the non-tampered (left column) and tampered (right column) videos. In this experiment, 1/3rd of the GOPs of the video were deleted compared to only a single GOP in the previous experiment. Again, visually, there are no “blocky” images since the deletion of frames were done in groups of GOPs. However, the change in the video scenes was even more abrupt compared to Figure 6-2-4a. The tampered video shows the foreman talking midway and suddenly the scene change abruptly to the construction site. A forensics investigator may notice such abruptness and realise that the video may have been tampered with.

Figure 6-2-5b showed the mean motion errors of the frames for both the non-tampered and tampered videos. Again, similar to the other previous experiments, the "old" frame #251 was interpreted as the "new" frame #151 since frames #151 to #250 were deleted. Again, similar to the previous experiment, there are no "odd" frame pattern sequences since the deletion was done on a group of GOPs.

Figure 6-2-5c showed the FFT graphs of the mean motion errors of the P-frames for both the non-tampered and tampered video that has 1/3rd of the GOPs (frames #151 to #250) deleted. It can be seen that the FFT graphs for the tampered video shows peaks that are a slightly higher from the peaks seen in the non-tampered video. Nevertheless, despite the increase in the height of the peaks, the peaks are not as abnormally large or obvious and this is again in line with the observation made by [WW06] that his methodology are unable to detect video tampering as long as the number of deleted frames is a multiple of the GOP length. Furthermore, a forensics investigator may not necessarily have access to a non-tampered video and certain non-tampered videos, such as the non-tampered "hand" video exhibits relatively large peaks in its FFT graphs (see Figure 6-1-3). Hence, the height of the peaks seen in Figure 6-2-5c is not conclusively enough to state that there is some form of video tampering.

While the FFT graphs do not show any obvious changes or spikes, deleted frames can usually be detected in the following manner:

- Visually – there must not be any "blocky" images and the scenes must flow seamlessly so that it cannot easily detected. Otherwise, as seen in Figure 6-2-1a, 6-2-2a, 6-2-4a, 6-2-5a, a forensics investigator can easily detect the tampering of video through simple visual observation.
- GOP frame pattern sequence – as long as frames are not deleted in the multiple of the GOP lengths, such as seen in Figure 6-2-1b and Figure 6-2-2b, a forensics investigator can detect changes in the GOP frame pattern sequences.
- Length of video – a forensics investigator would also be able to detect the tampering of the video if the video that was received has a shorter than expected video duration. E.g.,

if a CCTV was expected to show 60 minutes of footage, the forensics investigator would be suspicious if only 50 minutes of footage were available.

Therefore, while the deletion of frames can indeed remove evidences and are not detectable through the methodology proposed by [WW06], it may be easily detected through other means as described above. A more insidious approach would be the replace the deleted frames so that the characteristics of the video (e.g., visual quality, consistency of GOP frame pattern sequences, length of video, etc.) can be maintained.

7.3 Tampering encoded video – deleting frames with different frame pattern sequences

The previous section proved that the deletion of frames from an MPEG-2 video that is encoded with the frame pattern sequence "IPPPP" cannot be detected using the methodology by [WW06]. But what if the video was encoded using a different or "longer" frame pattern sequence?

In this experiment, the "foreman" video was encoded with 2 other frame pattern sequences, i.e., Pattern #2 ("IPPPP PPPP") and Pattern #3 ("IPPPP PPPP PPPP"). Each of the frame pattern sequences had their frames deleted – starting from an initial I-frame and then continued with the I-frame and its neighbouring P-frame and so on and so forth.

The screen captures of the frames for both sequences were not shown since it should be obvious that the deletion of frames will result in either a "blocky" image or an abrupt change of scene that can be visually detected. In addition, as long as the deletion of the frames is not done in a multiple of the GOP length, the forensics investigator can detect tampering through the "odd" and unexpected frame pattern sequence (similar to the previous experiments).

Figure 6-3-1 showed a sample of the mean motion errors of P-frames for the 2 frame pattern sequences. The mean motion errors are different from one another. This is expected since the videos are encoded differently from one another using frame pattern sequences that are not the same.

Figure 6-3-2 showed the FFT graphs for the deletion of combination of frames (up to the entire GOP) for the video encoded with "IPPPP PPPP" while Figure 6-3-3 showed the FFT graphs for the deletion of combination of frames (up to the entire GOP) for the video encoded with "IPPPP PPPP PPPP". Just like the results of the FFT graphs for the deletion of frames for Pattern #1 ("IPPPP") – see Figures 6-2-1c, 6-2-3 and 6-2-4c – there are no obvious "spikes" in any of the graphs for the 2 other frame pattern sequences. This means that even if the videos are encoded in different (longer) frame pattern sequences, the methodology by [WW06] is unable to detect tampering.

The next set of experiments will look into how deleted frames can be replaced via the duplication of frames that already exist in the video to be tampered. While it is possible to use frames from other similar videos to replace the deleted frames in the targeted video, the use of frames that are already available in the video is a much simpler approach since the attacker need not hunt for matching scenes from other videos (attacker just need to copy and paste frames!). Visually, it is also more likely to fool a forensics investigator since the conditions of the images (e.g., lighting, position of objects, etc.) are much more correlated to the targeted video than using another video. An example as mentioned in Chapter 4 (Problem Statement) is the famous looping of CCTV footages in Hollywood movies such as "Speed" and "Mission Impossible".

7.4 Tampering encoded video – duplicating frames

Figures 6-4-1a, 6-4-1b and 6-4-1c which shows the screen captures of the GOP of an empty stationary scene, the hand entering the video and hand leaving the video respectively, provides a visual picture on how the frames are to be duplicated. The arbitrarily selected GOP Set A (frames #91 to #95) of an empty stationary scene in Figure 6-4-1a which will be duplicated to other GOPs, is almost identical visually to the rest of the GOPs in first 1/3rd of the video (see Figure 6-1-1b for the screen captures of the non-tampered "hand" video as a comparison) as described in Section 5.3 of Chapter 5 (Design of Experiments).

Therefore, in terms of visual detection, it can be very difficult for a forensics investigator to detect any form of tampering when the hand is removed from the video via the duplication of the GOP Set A (frames #91 to #95) with the empty stationary scene. The task is made even more difficult as the scenes showing the hand entering the video (GOP Set B) are also duplicated accordingly. Thus visually, there are no “blocky” images, no abrupt appearance or disappearance of the hand and no obvious changes to the empty stationary scene.

Figure 6-4-2 shows the FFT graphs of the mean motion errors of P-frames for a series of frames/GOPs which were duplicated with the empty stationary scene (GOP with frames #91 to #95). As can be seen from Figure 6-4-2, the peaks in the FFT graphs become relatively higher as more GOPs were being duplicated. This is similar to the results seen in Figure 6-2-5c and the discussion in the preceding Section 7.2.5 (on the deletion of 1/3rd of the GOPs in the “foreman” video). The height of the peaks is still not as obvious or as large as spikes indicated in Figure 3-3-2a as highlighted in [WW06]. Furthermore, this is consistent with the findings in [WW06] that his methodology is unable to detect video tampering if the frames added/deleted is in the multiple of the GOP length.

In addition to the inability to detect video tampering for a duplicated GOP through visual means and the FFT graphs, the following tell-tale characteristics mentioned in the earlier Section 7.2.5, such as consistency of GOP frames or duration of video are also addressed:

- GOP frame pattern sequence – since the duplication makes use of GOPs instead of frames, the GOP frame pattern sequences (i.e., “IPPPP”) remains the same throughout the video.
- Length of video – since the frames are duplicated instead of being deleted, the length of the video remains the same.

The next chapter, Chapter 8 (Conclusion & Summary), will summarise the findings of the investigation in this report and also provide a short discussion on the further areas of improvement and research.

Chapter 8: Conclusion & Summary

Chapter 1 (Introduction) of this report first introduced the motivation of studying digital video forensics. Chapter 2 (Background) then presented some of the important key concepts in understanding images and videos, especially how videos are encoded according to the MPEG-2 standard. In addition, it also covered a brief description of the relevant MPEG standards. Chapter 3 (Related Works) then covered the current research trends and challenges in digital image and video forensics, followed by a discussion on the recent and popular research approaches in MPEG-2 video forensics. It also highlighted the 2 ways that an attacker can tamper a video, namely, either using intermediate software such as video editors or to tamper the encoded MPEG-2 file directly.

Chapter 4 (Problem Statement) then highlighted that the recent research approach on MPEG-2 video forensics tends to rely on the fact that attackers use an intermediate software to tamper effects. These researches therefore focused on observing the spatial and temporal statistical effects of a doubly compressed video. Chapter 4 (Problem Statement) then pose the research question on whether recent research techniques, such as [WW06] are able to detect video tampering if the ever increasingly sophisticated attacker chooses to tamper the encoded MPEG-2 video file directly. [WW06] was chosen because the methodology was the first to look at the temporal statistical properties of a doubly compressed video and is often quoted in subsequent research [YS11 and MS12] in MPEG-2 forensics.

Chapter 5 (Design of Experiments) then discussed how the experiments will be setup and designed in order to validate the research question asked in Chapter 4 (Problem Statement). Essentially, the experiments focused on the deletion and duplication/cloning of frames on the test videos and observing the FFT graphs of the mean motion errors of P-frames of the original and tampered video.

The results of the experiments were made available in Chapter 6 (Results of Experiments) and an analysis of the results was made in Chapter 7 (Discussion). Finally, this chapter, Chapter 8 (Conclusion & Summary) will provide a summary of the discussion and discuss whether the research question has been adequately answered as well as provide a short overview on other possible detection and counter-forensics techniques. It will also briefly cover future areas of research and improvements.

8.1 Effectiveness of FFT graphs of mean motion errors of P-frames

[WW06] relies on the observation of “spikes” in the FFT graphs of the mean motion errors of P-frames to detect tampered videos as these tampered videos would undergo a second compression which introduced temporal artefacts that subsequently show up as “spikes”. The technique is regarded as fairly reliable [MS12] but is not unable to detect tampered videos if the deletion or addition of frames in the tampered video is in a multiple of the GOP length. Chapter 3 (Related Works) discussed the improvement of [WW06] by [MS12] in that [MS12] can detect tampered videos even if the modification of frames is done in a multiple of the GOP length. Nevertheless, [MS12] still relies on observing the temporal effects of a doubly compressed video.

Chapter 5 (Design of Experiments) designed a series of experiments in which frames are deleted and later on duplicated directly on the encoded MPEG-2 file instead of going through an intermediate video editor and as such, prevented (in theory) the appearance of the temporal artefacts. The experiments were conducted such that an individual I-frame and P-frames were first deleted and then the number of frames deleted was increased until an entire GOP was deleted and finally a full 1/3rd of the GOPs in the video were also deleted.

Chapter 6 (Results of Experiments) showed that the mean motion errors of the P-frames of these deleted frames (whether done singly or until the entire GOP) remained essentially the same. This was because the tampered video did not undergo a second compression and as such, the motion errors of the frames in the tampered video remained the same as the original video. As such, unsurprisingly, the FFT graphs of the mean motion errors of P-frames for the tampered video did not show any obvious or noticeable “spikes”. In addition, no “spikes” were observed regardless of whether the deletion of frames was done in a multiple of the GOP length or not. In addition, even if the videos are encoded differently, there are also no obvious “spikes” when the frames are deleted from the video. *In other words, by deleting frames directly from the encoded MPEG-2 file, the attacker is able to prevent a forensics investigator from detecting the tampered video if the forensics investigator relied on the technique used by [WW06] or its subsequent research (in theory) such as [YS11 and MS12]*. This is due to the lack of temporal (or spatial) artefacts in the video.

However, while the deletion of frames directly via an encoded MPEG-2 video seems to be a good counter-forensics technique against recent MPEG-2 forensics research which focuses on doubly compressed videos, Chapter 6 (Results of Experiments) showed that deletion of frames can be quite easily detected visually since the tampered video either showed “blocky” images or an abrupt change of scene. On the other hand, a doubly compressed video would not show any “blocky” images since the frames are re-encoded although the abrupt change in scenes can still be visible. In addition, for the deletion of frames that is not in a multiple of the GOP length (e.g. deletion of a single frame), a forensics investigator can also tell that the videos are tampered through the inconsistent frame pattern sequences. Videos that have their frames deleted can also arouse suspicion if the videos were expected to have certain number of frames or video duration.

Therefore, to overcome such suspicions (e.g., visual check, frame pattern sequences, video length/duration, etc.) that may arise from the direct deletion of frames from an encoded MPEG-2 file, Chapter 5 (Design of Experiments) also included several subsequent experiments (refer to Section 5.3 Tampering encoded video – duplicating frames) where GOPs were duplicated across other GOPs. The duplication of frames using GOPs allowed the frame sequence pattern to remain consistent and also to maintain the video length/duration. A careful attacker can also re-use existing scenes in the video to fool the forensics investigator and pass the visual test.

In the experiments to duplicate frames, the test video uses the “hand” video which simulates the presence of a subject (e.g., a criminal suspect) in a CCTV and the experiments seek to make the “hand” disappear from the video – akin to a criminal trying to erase his presence in a CCTV video. In the experiments, the number of GOPs being duplicated was slowly increased until the “hand” which appeared in almost 1/3rd of the video completely disappeared.

Similar to the findings in the earlier experiments where frames are deleted, Chapter 6 (Results of Experiments) showed that there were no obvious or noticeable “spikes” in the tampered videos where these scenes were being duplicated. However, this time, unlike the deletion of frames, the duplication of frames did not cause any obvious visual tell-tale signs, such as “blocky” images or abrupt change of scene.

Therefore, this report can conclude that by editing the encoded MPEG-2 file directly and using careful duplication of frames, an attacker can tamper a video such that the tampering cannot be detected by the existing detection methodologies proposed by current research approaches [WW06, YS11 and MS12] that rely on observing temporal artefacts.

In addition, the duplication of frames also may not be easily detected through other means such as checking the frame pattern sequences, visual checks or observing the length/duration of the movie. Hence, it seemed that while editing encoded MPEG-2 videos is certainly less user-friendly, sophisticated attackers may actually turn to this method as it offers a viable counter-forensics solution to current MPEG-2 forensics detection techniques.

8.2 Alternative Methods of Detection and Counter-Forensics

Despite the success in evading the detection techniques in recent MPEG-2 forensics research by editing the MPEG-2 encoded videos directly, there might be other detection strategies that may detect the tampering of the "hand" video, such as the following:

- Spatial artefacts of doubly compressed MPEG-2 videos – This report focused on the methodology by [WW06] which relied on the existence of temporal artefacts caused by double compression. [WW09] introduced another method in detecting doubly compressed MPEG-2 videos by looking at the spatial artefacts that is caused by the double compression. The spatial artefacts that are a result of double compression rely on the fact that there is a dual layer of quantisation as discussed in Section 3.2 (Current Research in Digital Video Forensics for MPEG-2).

  This report suggests that the direct editing of MPEG-2 video should also evade the detection methodology proposed by [WW09] since by editing the encoded video directly, the attacker can avoid generating the second quantisation level which is a tell-tale spatial artefacts of double compression. Again, it would appear that the direct editing of the encoded file presents a very viable counter-forensics technique against contemporary research in MPEG-2 video forensics.

- Hashing of video to provide a cryptographic fingerprint – A very common way to detect changes in the integrity of an object, such as a video, is to hash the object with a hashing algorithm to obtain a cryptographic fingerprint [KM12]. If the video was tampered (e.g., addition, deletion, duplication, editing of frames, etc.), the cryptographic fingerprint for the tampered video will be different from the non-tampered version and this will indicate to the forensics investigator that the video has been tampered.

However, the hashing of the video has its limitations. Firstly, the video has to be hashed in its entirety. If the video is long (very possible in scenarios such as CCTVs videos, etc.), then the hashing of videos may be computationally intensive. In addition, the MPEG-2 standards do not specify how encoding should be done. As such, a raw video can be encoded differently depending on devices and their configurations. Hence, there could be multiple versions of the same video but with different encoding and therefore different cryptographic fingerprints as well. It would be argued in court that a suspected tampered video with a different fingerprint (compared to the original video) is actually the same video but differently encoded.

More importantly, most videos are generated without any cryptographic fingerprints in the first place. Many video sources, such as webcam, digital cameras or camcorders do not automatically generate cryptographic fingerprints (presumably it causes delays since it takes time to compute the hash for videos and this may be perceived as not user-friendly). As such, without access to the original video as well as its accompanying cryptographic fingerprint, a forensics investigator would not be able to tell if a suspected video has been tampered or not via the cryptographic fingerprint (since none are available in the first place!). For example, a video containing child pornography was seized from a suspected paedophile. The suspected paedophile could actually claim the video in possession was actually a tampered video where the child in question is a computer generated character (very believable given the advances in computer graphics and imagery). A forensics investigator has no way to prove whether the video has been tampered or not since there were no cryptographic fingerprint of the original video that could be compared against. Therefore, this technique cannot be effectively used to detect the tampering of the “hand” video described in this report.

- Hashing of video frames to detect duplication of frames – The previous point argued that detecting tampered videos through the hashing of videos may be difficult due to the fact that the cryptographic fingerprints of the videos may not be available in the first place. However, what if the hashing was done over the frames rather than the entire video? Unlike the need to obtain the cryptographic fingerprint of the original video, the forensics investigator simply need to detect that the frames are not being duplicated/cloned in the targeted video. The forensics investigator can hash a frame and compare its hash value against other frames. If the hash values matches, then it could mean that the frames are actually duplicated/cloned.

If this method is indeed applied to the vanilla duplication of the “hand” video in this report, then this method can indeed detect that the frames were being copied since the hash values of the copied frames are the same as the original frame.

However, this method can also be very easily defeated. This is because the attacker only needs to cause a slight change in encoded bit stream of the copied frames to ensure that the hash values of these frames would be different from the original frames. Such slight changes would not make very visible change to the visual quality of the frame. For example, as indicated by [MS12] in Chapter 3 (Related Works), the mean motion errors of a frame is very much correlated to the motion vectors in producing the final image of the frame. An attacker can simply set the motion vector of a particular macroblock to be zero while compensating the change by changing the associated motion error of the macroblock. Such a move can ensure that the image quality remain almost imperceptible to the human eye but can totally change the structure of the frame such that the hash value is totally different from the original frame. Alternatively, the attacker can simply introduce additive noise to each of the copied frames. Again, the noise would not change the visual image and so remains imperceptible to the human eye but the noise would be enough to ensure that the hash values are unique and different from the original frame. Hence, the use of hash on either the entire video or frames would not be able to reliably detect the duplication of frames in the “hand” video if such techniques were to be applied.

- Digital watermarking of video – Digital watermarking an image or video is a common technique to ensure the copyright protection of videos. However, this technique requires videos to be actively inserted with digital watermarks. Just like hashing of a video, such feature may not be readily available in many video sources. For example, webcams in notebooks, digital camcorders or even some of the commercial CCTVs do not provide digital watermarking as a standard product feature. Therefore, similar to the limitation of using hashes mentioned in the second point, digital watermarking is not a viable solution to detect video tampering if the video source do not provide digital watermarks in the first place. In the case of the “hand” video, the digital watermarks are certainly not available as a product feature in the author’s webcam.

- Video fingerprinting – This technique is similar to hashing as well as digital watermarking in that it seeks to detect tampering. But unlike hashing, it relies on certain video characteristics such as colour or motion changes to generate a unique fingerprint instead of using the entire video to generate a hash. Also, unlike digital watermarking, it does not actively insert any additional data. Unfortunately, similar to the limitation of hashing and digital watermarks, if the video source does not provide a video fingerprint in the first place together with the original video, the forensics investigator has no way to tell if the video has been tampered or not since there are no fingerprints to be compared to.

This section discussed some of the possible strategies that can be used to detect the tampering of the “hand” video which is manipulated through the direct editing of an encoded MPEG-2 file. The next section will discuss the possible areas of research as well as future improvements.

8.3 Future Work

The results of the experiments suggested that tampering a video by directly editing the encoded MPEG-2 file may actually be a more forensics-resistant approach since recent research focused heavily on assuming that attackers would tamper a video by using intermediate software such as video editors. Nevertheless, there are certainly room for improvements in the current work in this report.

In the earlier section, Section 8.2 (Alternative Methods of Detection and Counter-Forensics), it was highlighted that the direct editing of encoded videos should be able to defeat detection methodology of [WW09] which looks at the spatial artefacts of the tampered video since there is no dual layer of quantisation which the methodology relied on. Still, it may be interesting to conduct further research (such as performing experiments) in this area to prove conclusively that the direct editing of encoded videos can indeed foil detection using this methodology.

In addition, Section 8.2 (Alternative Methods of Detection and Counter-Forensics) also mentioned the use of additive noise as well as the manipulation of the relationship between the motion errors and motion vectors (as indicated by [MS12]) to mask the duplication of the frames. The work in this report could be extended further by exploring these 2 factors and to see if existing tools, used in current research approaches, such as looking at the quantisation levels seen in [WW09] or the FFT graphs of the motion vectors of frames seen in [WW06] or the frequency components of frames seen in [YS11] can be used to detect such tampering.

Chapter 4 (Problem Statement) highlighted that an attacker can tamper videos by deleting, adding, duplication or changing parts of a frame. However, this report only looked at the deletion/addition and duplication of frames. As such, the work in this report could also be extended to include the modification of certain parts of a frame. This could be potentially more difficult than the deletion and duplication of frames which is relatively simple to perform since it is not difficult to identify the start and end of a frame in the encoded video. However, it is likely to be more challenging to identify which parts of the encoded bit stream belongs to which part of the frame.

Given that there are millions of videos, it is very possible that some of these videos are so different from other (e.g., have radical and many scenes changes or encoded very “oddly”) that the FFT graphs of the mean motion errors of P-frames for these videos can actually show “spikes” even when they are not tampered in the first place. This could be possible, since the FFT graphs will show spikes if the mean motion errors of the P-frames have highs or lows (compared to its neighbouring frames) that occurs at a periodic interval. This report can be extended to look into a statistical analysis of a sample size of videos to provide a certain degree of confidence on whether such spikes occur naturally or not.

Another possible area of improvement is to use a real compliant MPEG-2 decoder rather than just using the MPEG-2 style decoder from [SH05] which is used to encode/decode the MPEG-2 videos in this report. The MPEG-2 style encoder/decoder obtained from [SH05] can generate an MPEG-2 file that can be decoded by any MPEG-2 compliant decoder but it is not able decode every MPEG-2 videos. Hence, although the codes for the MPEG-2 style decoder is available and the design allows it to be easily understood and modified, the work here may have different effects or results on a real compliant MPEG-2 decoder.

Currently, while MPEG-2 videos remain possibly the most popular video compression standard, technology is always improving and there are signs that H.264 or MPEG-4 Part 10 may be slowly but surely replacing MPEG-2 as the more preferred video compression standard, especially when low-cost H.264 hardware decoders are getting cheaper. Per Chapter 2 (Background), H.264 has employs object-based coding with yields better compression rates. The object encoding mechanism could also possibly make the tampering of an H.264 video easier since it might be possible for the attacker to manipulate the properties of the objects directly through the encoded bit stream. Such manipulation might present an even finer control over the visual aspects of the video. For example, if the ball in the football video mentioned in Section 2.2 (Overview of the MPEG video compression standards) can be easily erased, then it should also be equally easy to erase the presence of a suspected criminal in a H.264 encoded video.

Lastly, the next generation of video compression standard, HEVC or MPEG-H Part 2 is slated to be ratified as a standard in 2013. Again, there are improvements in HEVC, such as the use of tree-like structure for prediction, etc., which allows finer granularity in manipulating objects in the images. With the passage of time, it is inevitable that digital doctoring tools will eventually be made available. Once this happens, as mentioned in Chapter 1 (Introduction), it will be easy for ordinary people with nefarious aims to tamper images and videos. The difference is that advanced compression standards may actually be easier for them to tamper videos and possibly make it harder to detect such tampering. Still, it is exciting to continue research in this area as the research will have tremendous value and impact in countering video tampering.

With this, the report has successfully concluded and satisfactorily answered the research question posed in Chapter 4 (Problem Statement).

Bibliography

[AP04] A. Popescu, H. Farid, Statistical tools for Digital Forensics, *IH'04 Proceedings of the 6th international conference on Information Hiding*, pp. 128 – 147, ISBN:3-540-24207-4 978-3-540-24207-9, doi: 10.1007/978-3-540-30114-1_10

[AP05] A. Popescu, H. Farid, Exposing Digital Forgeries by detecting Traces of Resampling, 5*IEEE Transactions on Signal Processing*, Volume 53 Issue 2, pp. 758 – 767, ISSN: 1053-587X, INSPEC Accession Number: 8252153, doi: 10.1109/TSP.2004.839932(410) 53

[AR11] A. Rocha, W. Scheirer, T. Boult, S. Goldenstein, Vision of the Unseen: Current Trends and Challenges in Digital Image and Video Forensics, *ACM Computing Surveys (CSUR)*, Volume 43 Issue 4, October 2011, Article No. 26, doi: 10.1145/1978802.1978805

[DL91] D. LeGall, MPEG: A Video Compression Standard for Multimedia Applications, *Communications of the ACM – Special issue on digital multimedia systems*, Volume 34 Issue 4, April 1991, pp. 46 – 58, doi: 10.1145/103085.103090

[JF03] J. Fridrich , D. Soukal , J. Lukas, Detection of Copy-Move Forgery in Digital Images, *Proceedings of Digital Forensic Research Workshop*, 2003, http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.121.1962

[JH06] J. He, Z. Lin, L. Wang, X. Tang, Detecting Doctored JPEG Images via DCT Coefficient analysis, *ECCV'06 Proceedings of the 9th European conference on Computer Vision - Volume Part III*, pp. 423 – 435, ISBN:3-540-33836-5 978-3-540-33836-9, doi: 10.1007/11744078_33

[JM99] J. Miano, *Compressed Image File Formats: JPEG, PNG, GIF, XMB, BMP*, Addison Wesley, 1999, ISBN: 0-201-604434-4

[JW01] J. Watkinson, *The MPEG Handbook: MPEG-1, MPEG-2, MPEG-4*, Focal Press, 2001, ISBN 0-240-51656-7

[KM12] K. Martin, *Everyday Cryptography: Fundamental Principles and Applications*, OUP Oxford, March 2012, ISBN: 0-199-69559-8, 978-0-199-69559-1

[KO12] K. Ooi, BBC News Asia, Viewpoint: January 2012, http://www.bbc.co.uk/news/world-asia-16465164

[LH04] L. Hempel, E. Topfer, CCTV in Europe, Urbaneye Project Working Paper No. 15, 5th Framework Programme of the European Commission, August 2004, http://www.urbaneye.net/results/ue_wp15.pdf

[LS04] L. Simmonds, MPEG – The Standards and History, 2004, www.lessimmonds.com.au/pdf/0412-MPEG-01.pdf

[MJ05] M. Johnson, H. Farid, Exposing Digital Forgeries by detecting Inconsistencies in Lighting, *MM&Sec '05 Proceedings of the 7th workshop on Multimedia and security*, pp. 1 – 10, ISBN:1-59593-032-9, doi: 10.1145/1073170.1073171

[MM02] M. McCahill, C. Norris, CCTV in London, Urbaneye Project Working Paper No. 6, 5th Framework Programme of the European Commission, June 2002, http://www.urbaneye.net/results/ue_wp6.pdf

[MS12] M. Stamm, W. Lin, K. Liu, Temporal Forensics and Anti-Forensics for Motion Compensated Video, *IEEE Transactions on Information Forensics and Security*, Volume 7, Issue 4, pp. 1315 - 1329, August 2012, ISSN: 1556-6013, INSPEC Accession Number: 12850457, doi: 10.1109/TIFS.2012.2205568

[MW12] Multimedia Wikipedia: Y4M, Accessed on 19 August 2012, http://wiki.multimedia.cx/index.php?title=YUV4MPEG2

[SB12] S. Boh, Asiaone, Police roll out CCTVs in 7 Neighbourhoods, April 2012, http://www.asiaone.com/News/Latest+News/Singapore/Story/A1Story20120420-340864.html

[SH05] S. Hoelzer, MPEG-2 Style Encoder/Decoder, April 2005, http://www.cs.cf.ac.uk/Dave/Multimedia/Lecture_Examples/Compression/mpegproj/

[WK12a] Wikipedia: http://en.wikipedia.org/wiki/Anwar_Ibrahim, Accessed on 18 August 2012

[WK12b] Wikipedia: H.265, http://en.wikipedia.org/wiki/H.265, Accessed on 19 August 2012

[WK12c] Wikipedia: MPEG, http://en.wikipedia.org/wiki/MPEG, Accessed on 19 August 2012

[WW06] W. Wang, H. Farid, Exposing Digital Forgeries in Video by Detecting Double MPEG Compression, *MM&Sec '06 Proceedings of the 8th workshop on Multimedia and security*, pp. 37 - 47, ISBN:1-59593-493-6, doi: 10.1145/1161366.1161375

[WW07] W. Wang, H. Farid, Exposing Digital Forgeries in Video by Detecting Duplication, *Proceedings of the 9th workshop on Multimedia & security*, pp. 35 – 42, ISBN: 978-1-59593-857-2 doi: 10.1145/1288869.1288876

[WW09] W. Wang, H. Farid , Exposing Digital Forgeries in Video by detecting Double Quantization, *MM&Sec '09 Proceedings of the 11th ACM workshop on Multimedia and security*, pp. 39 – 48, ISBN: 978-1-60558-492-8, doi: 10.1145/1597817.1597826

[XO12] Xiph.org Video Test Media, Accessed on 7 July 2012, http://media.xiph.org/video/derf/y4m/foreman_cif.y4m

[YS11] Y. Su, W. Nie, C. Zhang, A Frame Tampering Detection Algorithm for MPEG Videos, *Information Technology and Artificial Intelligence Conference (ITAIC)*, 2011 6th IEEE Joint International, Volume 2, pp. 461 – 464, August 2011, ISBN: 978-1-4244-8622-9, INSPEC Accession Number: 12272642, doi: 10.1109/ITAIC.2011.6030373

Appendix 1 – Mean Motion Errors of Frames

Figure A-1 below shows the mean motion errors for the I and P-frames of the non-tampered “foreman” and “hand” videos which were encoded using the frame pattern sequence “IPPPP”.

Frame	Frame Type	Foreman	Hand	Frame	Frame Type	Foreman	Hand
1	I	60.3742	35.8367	151	I	58.5394	37.4103
2	P	1.4106	1.2123	152	P	1.6639	0.3675
3	P	1.2102	0.454	153	P	2.4054	0.2837
4	P	1.1111	1.0793	154	P	3.0604	0.2923
5	P	1.25	0.4629	155	P	2.805	0.2724
6	I	60.7325	35.8721	156	I	59.5208	37.3557
7	P	1.4501	0.3712	157	P	3.1054	0.3351
8	P	1.1642	0.4457	158	P	2.6855	0.3166
9	P	1.1782	0.3688	159	P	2.3992	0.2808
10	P	1.2132	0.2538	160	P	1.3642	0.3036
11	I	61.2275	35.4241	161	I	57.426	37.472
12	P	1.5599	0.3223	162	P	1.4764	0.3807
13	P	1.3074	0.2564	163	P	1.2569	0.3058
14	P	1.4192	0.2566	164	P	1.0323	0.2768
15	P	1.4714	0.4097	165	P	1.2345	0.3164
16	I	60.8975	35.9172	166	I	57.3727	37.2695
17	P	1.561	0.3817	167	P	1.4998	0.3501
18	P	1.3665	0.486	168	P	1.4045	0.2818
19	P	1.1928	0.3859	169	P	1.4441	0.2361
20	P	1.1463	0.269	170	P	1.6172	0.2817
21	I	60.7689	36.0891	171	I	57.0789	37.2448
22	P	1.4809	0.5801	172	P	2.4534	0.3514
23	P	1.3214	0.3389	173	P	2.6103	0.2857
24	P	1.4309	0.2421	174	P	2.7918	0.2671
25	P	1.3846	0.2364	175	P	2.6954	0.2936
26	I	60.3568	35.7598	176	I	56.5034	37.2298
27	P	1.4998	0.3141	177	P	2.6499	0.3765
28	P	0.7723	0.253	178	P	2.5505	0.2906
29	P	0.8808	0.25	179	P	2.4872	0.306
30	P	0.8355	0.2309	180	P	2.4383	0.2822
31	I	60.615	35.6373	181	I	58.2105	37.3567
32	P	1.457	0.3243	182	P	2.1026	0.3687
33	P	1.2522	0.2169	183	P	2.2267	0.3097
34	P	1.312	0.218	184	P	2.8734	0.3261
35	P	1.313	0.2062	185	P	3.739	0.3025

Frame	Frame Type	Foreman	Hand	Frame	Frame Type	Foreman	Hand
36	I	60.4901	35.7814	186	I	62.5617	37.466
37	P	1.5301	0.2949	187	P	4.1375	0.3589
38	P	1.283	0.2454	188	P	4.3202	0.313
39	P	1.2498	0.2265	189	P	4.4053	0.2888
40	P	1.1281	0.1981	190	P	4.6577	0.3103
41	I	60.0761	35.7072	191	I	69.6168	37.5794
42	P	1.4942	0.3093	192	P	3.7603	0.3915
43	P	1.2292	0.2437	193	P	3.2291	0.2563
44	P	1.014	0.2652	194	P	3.223	0.2678
45	P	1.1572	0.2394	195	P	3.0545	0.2801
46	I	59.8401	35.684	196	I	70.4975	37.361
47	P	1.4398	0.3233	197	P	2.36	0.36
48	P	1.2511	0.2542	198	P	1.9017	0.2536
49	P	1.4102	0.2462	199	P	1.9477	0.2722
50	P	1.36	0.2534	200	P	1.9331	0.3188
51	I	59.6046	35.7172	201	I	69.9585	37.664
52	P	1.5176	0.3146	202	P	2.3956	1.1133
53	P	1.1264	0.2364	203	P	2.4426	1.2724
54	P	1.1707	0.2067	204	P	2.3661	0.5001
55	P	1.426	0.1984	205	P	2.2615	0.3066
56	I	60.17	35.6814	206	I	64.7894	35.7435
57	P	1.6667	0.3008	207	P	1.9665	0.33
58	P	1.3349	0.2238	208	P	2.0432	0.2383
59	P	1.3205	0.2087	209	P	2.0043	0.2271
60	P	1.2547	0.2172	210	P	2.2178	0.1956
61	I	60.4248	35.7732	211	I	62.1197	35.8968
62	P	1.567	0.2826	212	P	1.7608	0.3171
63	P	1.246	0.2179	213	P	1.5421	0.2496
64	P	1.1854	0.221	214	P	1.9098	0.2502
65	P	1.1725	0.2136	215	P	2.4338	0.2179
66	I	59.8637	35.6873	216	I	57.1158	35.9814
67	P	1.5132	0.3197	217	P	2.7655	0.3069
68	P	1.3058	0.2219	218	P	2.607	0.2272
69	P	1.1758	0.2071	219	P	2.603	0.2271
70	P	1.1527	0.2019	220	P	2.9621	0.2196
71	I	59.2888	35.7585	221	I	53.243	35.9939
72	P	1.5296	0.3082	222	P	2.9037	0.31
73	P	1.1778	0.2406	223	P	3.0089	0.244
74	P	1.0767	0.2142	224	P	2.8828	0.2418
75	P	1.1825	0.196	225	P	2.9688	0.2363
76	I	59.0031	35.6857	226	I	51.1742	35.9882

Frame	Frame Type	Foreman	Hand	Frame	Frame Type	Foreman	Hand
77	P	1.4974	0.2992	227	P	2.5481	0.3303
78	P	1.3877	0.2595	228	P	2.3272	0.2189
79	P	1.5067	0.2658	229	P	2.7358	0.2165
80	P	1.5532	0.2203	230	P	2.7731	0.2186
81	I	59.0394	35.7058	231	I	48.9118	36.0138
82	P	1.8512	0.3256	232	P	2.4502	0.3083
83	P	1.5204	0.2189	233	P	2.184	0.2391
84	P	1.3962	0.2184	234	P	2.3317	0.2319
85	P	1.5804	0.2033	235	P	2.2146	0.2088
86	I	59.8282	35.5302	236	I	48.7067	36.0382
87	P	1.7284	0.2856	237	P	2.6018	0.3242
88	P	1.3052	0.2199	238	P	2.273	0.2621
89	P	1.6578	0.223	239	P	2.4813	0.2326
90	P	1.7028	0.2203	240	P	2.2317	0.2456
91	I	59.1717	35.6496	241	I	49.2592	36.1085
92	P	2.0777	0.327	242	P	2.8417	0.3498
93	P	1.8487	0.2183	243	P	2.88	0.2527
94	P	2.046	0.2222	244	P	3.0467	0.2416
95	P	1.7785	0.2384	245	P	3.0924	0.2109
96	I	58.9026	35.6037	246	I	49.8999	36.0745
97	P	1.5424	0.3032	247	P	2.8746	0.3144
98	P	1.0354	0.372	248	P	2.7596	0.2561
99	P	1.1527	0.6031	249	P	2.4468	0.2567
100	P	1.0567	1.0999	250	P	2.1004	0.2455
101	I	59.1438	37.1018	251	I	50.0767	36.0255
102	P	1.3943	0.6088	252	P	2.7897	0.3085
103	P	1.016	0.4508	253	P	2.063	0.2082
104	P	0.9848	0.3643	254	P	1.893	0.2541
105	P	1.0088	0.3422	255	P	2.2455	0.2138
106	I	59.2119	36.6993	256	I	49.9934	36.098
107	P	1.3381	0.3724	257	P	2.6855	0.2992
108	P	0.7704	0.3747	258	P	2.4945	0.2603
109	P	0.9873	0.3331	259	P	2.2131	0.2341
110	P	0.9954	0.2954	260	P	2.1545	0.2255
111	I	58.9642	37.0461	261	I	49.9501	36.0691
112	P	1.3973	0.3588	262	P	2.686	0.3243
113	P	1.0989	0.2882	263	P	2.5413	0.2472
114	P	1.0627	0.2563	264	P	2.338	0.2425
115	P	1.0523	0.2495	265	P	2.0558	0.2322
116	I	58.7977	37.0838	266	I	49.9412	36.0158
117	P	1.3433	0.3835	267	P	2.6748	0.3206

Frame	Frame Type	Foreman	Hand	Frame	Frame Type	Foreman	Hand
118	P	0.7487	0.2666	268	P	2.1066	0.2253
119	P	0.8416	0.2735	269	P	2.1901	0.1947
120	P	0.885	0.2883	270	P	2.0022	0.2025
121	I	58.8278	37.0484	271	I	49.7777	36.0191
122	P	1.3937	0.3609	272	P	2.6204	0.308
123	P	1.0112	0.3149	273	P	2.0616	0.2441
124	P	1.0166	0.2736	274	P	1.7621	0.2299
125	P	1.0816	0.29	275	P	1.5708	0.2216
126	I	58.906	36.9269	276	I	49.7051	36.1498
127	P	1.5959	0.363	277	P	2.5425	0.3242
128	P	1.2664	0.2991	278	P	1.7537	0.2688
129	P	1.2495	0.2295	279	P	1.9004	0.249
130	P	1.2411	0.2722	280	P	1.9703	0.2332
131	I	59.67	37.089	281	I	49.7226	36.1673
132	P	1.6999	0.3764	282	P	2.5158	0.3077
133	P	1.3782	0.3057	283	P	1.989	0.2247
134	P	1.3115	0.2777	284	P	2.0872	0.2127
135	P	1.2888	0.288	285	P	1.8834	0.1921
136	I	60.5768	37.1255	286	I	49.9169	36.2071
137	P	1.7248	0.3398	287	P	2.7224	0.3107
138	P	1.5452	0.291	288	P	2.2367	0.238
139	P	1.6947	0.2889	289	P	2.0527	0.2225
140	P	1.8207	0.2854	290	P	2.1444	0.2009
141	I	59.7021	36.9724	291	I	50.0615	36.2194
142	P	1.7994	0.3445	292	P	2.6898	0.3229
143	P	1.568	0.2996	293	P	2.1808	0.2373
144	P	1.4314	0.2666	294	P	2.0991	0.2371
145	P	1.5595	0.2798	295	P	2.1066	0.2355
146	I	59.5836	37.3195	296	I	50.1978	36.3251
147	P	1.5101	0.3337	297	P	2.6882	0.3034
148	P	1.2456	0.2949	298	P	2.3282	0.2431
149	P	1.2786	0.299	299	P	2.1198	0.2261
150	P	1.0139	0.2577	300	P	1.6994	0.2058

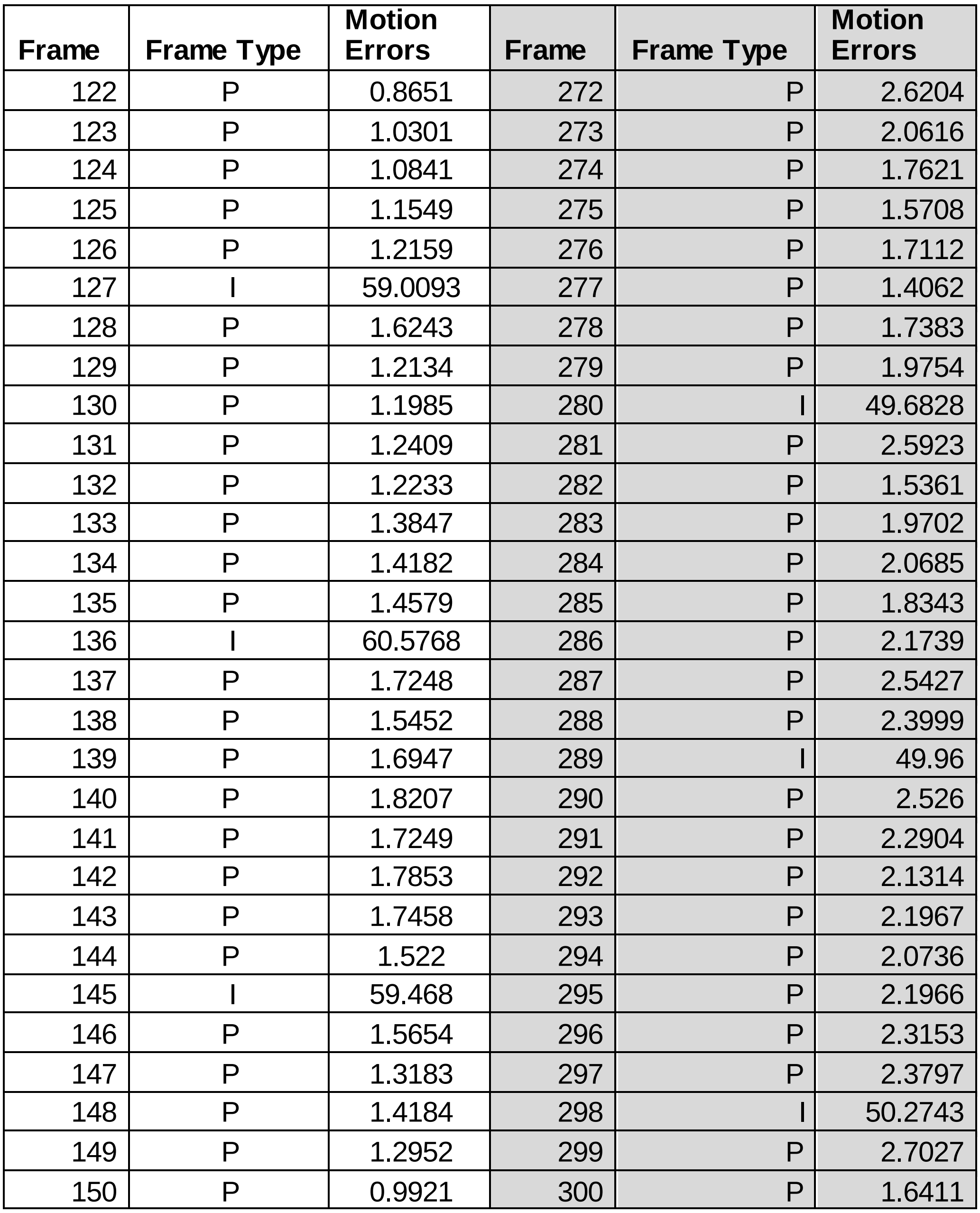
Figure A-1

Figure A-2 below shows the mean motion errors for the I and P-frames of the “foreman” video that is encoded with the frame pattern sequence “IPPPP PPPP”.

Frame	Frame Type	Motion Errors	Frame	Frame Type	Motion Errors
1	I	60.3742	151	P	1.2702
2	P	1.4106	152	P	1.6149
3	P	1.2102	153	P	2.5784
4	P	1.1111	154	I	59.6739
5	P	1.25	155	P	2.7775
6	P	1.1878	156	P	2.1256
7	P	0.9607	157	P	3.1512
8	P	1.1778	158	P	2.6859
9	P	1.2926	159	P	2.4297
10	I	61.1533	160	P	1.4427
11	P	1.4549	161	P	1.3721
12	P	1.2314	162	P	1.3215
13	P	1.2825	163	I	57.5552
14	P	1.4343	164	P	1.5058
15	P	1.5207	165	P	1.2765
16	P	1.4908	166	P	1.058
17	P	1.425	167	P	1.202
18	P	1.4512	168	P	1.439
19	I	60.9033	169	P	1.4605
20	P	1.4788	170	P	1.6599
21	P	1.0586	171	P	2.0241
22	P	1.1411	172	I	56.9428
23	P	1.3333	173	P	2.6389
24	P	1.4284	174	P	2.708
25	P	1.3568	175	P	2.5802
26	P	1.4007	176	P	2.5965
27	P	1.2253	177	P	2.724
28	I	60.5093	178	P	2.7942
29	P	1.4851	179	P	2.6748
30	P	0.8413	180	P	2.6519
31	P	0.9957	181	I	58.2105
32	P	1.0184	182	P	2.1026
33	P	1.2049	183	P	2.2267
34	P	1.2921	184	P	2.8734
35	P	1.3339	185	P	3.739
36	P	1.3729	186	P	4.0887
37	I	60.384	187	P	4.4851
38	P	1.4883	188	P	4.633
39	P	1.2651	189	P	4.533

Frame	Frame Type	Motion Errors	Frame	Frame Type	Motion Errors
40	P	1.0968	190	I	68.5634
41	P	1.1061	191	P	4.0398
42	P	1.2318	192	P	3.8858
43	P	1.2104	193	P	3.5275
44	P	1.0165	194	P	3.4433
45	P	1.2309	195	P	3.3416
46	I	59.8401	196	P	2.9067
47	P	1.4398	197	P	2.422
48	P	1.2511	198	P	2.3181
49	P	1.4102	199	I	70.3053
50	P	1.36	200	P	1.8005
51	P	1.2977	201	P	2.0652
52	P	1.2149	202	P	2.47
53	P	1.2712	203	P	2.5334
54	P	1.228	204	P	2.4767
55	I	59.9488	205	P	2.328
56	P	1.6826	206	P	2.3597
57	P	1.3415	207	P	2.2869
58	P	1.3198	208	I	63.0233
59	P	1.3614	209	P	1.9155
60	P	1.2779	210	P	2.0374
61	P	1.3154	211	P	2.0241
62	P	1.3717	212	P	1.7667
63	P	1.2953	213	P	1.6072
64	I	60.1424	214	P	2.2602
65	P	1.4999	215	P	2.602
66	P	1.2236	216	P	2.9141
67	P	1.2706	217	I	56.049
68	P	1.3588	218	P	2.7172
69	P	1.2136	219	P	2.5396
70	P	1.2171	220	P	2.7572
71	P	1.3467	221	P	2.9701
72	P	1.3351	222	P	3.145
73	I	59.2885	223	P	3.3787
74	P	1.4273	224	P	3.2223
75	P	1.1264	225	P	3.3388
76	P	1.105	226	I	51.1742
77	P	1.2718	227	P	2.5481
78	P	1.3836	228	P	2.3272
79	P	1.552	229	P	2.7358
80	P	1.5161	230	P	2.7731

Frame	Frame Type	Motion Errors	Frame	Frame Type	Motion Errors
81	P	1.6586	231	P	2.7366
82	I	59.1057	232	P	2.4716
83	P	1.7341	233	P	2.3794
84	P	1.4311	234	P	2.5362
85	P	1.5172	235	I	48.6419
86	P	1.5198	236	P	2.5957
87	P	1.5003	237	P	2.4032
88	P	1.3326	238	P	2.4499
89	P	1.7267	239	P	2.6602
90	P	1.7825	240	P	2.4481
91	I	59.1717	241	P	2.562
92	P	2.0777	242	P	3.2169
93	P	1.8487	243	P	3.315
94	P	2.046	244	I	49.7873
95	P	1.7785	245	P	2.9994
96	P	1.4157	246	P	2.6388
97	P	1.3069	247	P	2.8677
98	P	1.0804	248	P	2.9724
99	P	1.2066	249	P	2.5561
100	I	59.0778	250	P	2.2552
101	P	1.436	251	P	2.1233
102	P	1.0092	252	P	2.5706
103	P	0.9849	253	I	50.0637
104	P	0.9693	254	P	2.6535
105	P	1.008	255	P	2.2863
106	P	0.9751	256	P	2.4497
107	P	0.9292	257	P	2.4124
108	P	0.6808	258	P	2.4733
109	I	59.0812	259	P	2.1704
110	P	1.3486	260	P	2.0542
111	P	1.0594	261	P	2.2609
112	P	1.0463	262	I	49.9735
113	P	1.0574	263	P	2.7667
114	P	1.0727	264	P	2.3093
115	P	1.0539	265	P	2.0709
116	P	1.0596	266	P	2.1578
117	P	1.0097	267	P	2.0415
118	I	58.8572	268	P	2.108
119	P	1.3647	269	P	2.2528
120	P	0.9384	270	P	2.0711
121	P	0.8823	271	I	49.7777



Frame	Frame Type	Motion Errors	Frame	Frame Type	Motion Errors
122	P	0.8651	272	P	2.6204
123	P	1.0301	273	P	2.0616
124	P	1.0841	274	P	1.7621
125	P	1.1549	275	P	1.5708
126	P	1.2159	276	P	1.7112
127	I	59.0093	277	P	1.4062
128	P	1.6243	278	P	1.7383
129	P	1.2134	279	P	1.9754
130	P	1.1985	280	I	49.6828
131	P	1.2409	281	P	2.5923
132	P	1.2233	282	P	1.5361
133	P	1.3847	283	P	1.9702
134	P	1.4182	284	P	2.0685
135	P	1.4579	285	P	1.8343
136	I	60.5768	286	P	2.1739
137	P	1.7248	287	P	2.5427
138	P	1.5452	288	P	2.3999
139	P	1.6947	289	I	49.96
140	P	1.8207	290	P	2.526
141	P	1.7249	291	P	2.2904
142	P	1.7853	292	P	2.1314
143	P	1.7458	293	P	2.1967
144	P	1.522	294	P	2.0736
145	I	59.468	295	P	2.1966
146	P	1.5654	296	P	2.3153
147	P	1.3183	297	P	2.3797
148	P	1.4184	298	I	50.2743
149	P	1.2952	299	P	2.7027
150	P	0.9921	300	P	1.6411

Figure A-2

Figure A-3 below shows the mean motion errors for the I and P-frames of the “foreman” video that is encoded with the frame pattern sequence “IPPPP PPPP PPPP”.

Frame	Frame Type	Motion Errors	Frame	Frame Type	Motion Errors
1	I	60.3742	151	P	1.3595
2	P	1.4106	152	P	1.6625
3	P	1.2102	153	P	2.6107
4	P	1.1111	154	P	3.1799
5	P	1.25	155	P	3.0006
6	P	1.1878	156	P	2.5035
7	P	0.9607	157	I	58.1066
8	P	1.1778	158	P	2.7027
9	P	1.2926	159	P	2.3956
10	P	1.2238	160	P	1.236
11	P	1.3154	161	P	1.1817
12	P	1.4245	162	P	1.1916
13	P	1.4384	163	P	1.2923
14	I	61.0413	164	P	1.1505
15	P	1.6819	165	P	1.3065
16	P	1.4505	166	P	1.1528
17	P	1.4017	167	P	1.4267
18	P	1.3885	168	P	1.5613
19	P	1.2741	169	P	1.6602
20	P	1.2023	170	I	57.1312
21	P	1.183	171	P	2.1066
22	P	1.2354	172	P	2.3308
23	P	1.4488	173	P	2.6231
24	P	1.581	174	P	2.8983
25	P	1.5207	175	P	2.6834
26	P	1.5123	176	P	2.6357
27	I	60.4138	177	P	2.7923
28	P	1.4801	178	P	2.9149
29	P	0.9921	179	P	2.7372
30	P	0.8528	180	P	2.7429
31	P	0.9766	181	P	2.7941
32	P	0.9791	182	P	2.6593
33	P	1.166	183	I	59.5932
34	P	1.2802	184	P	2.8005
35	P	1.295	185	P	3.5297
36	P	1.3535	186	P	4.0756
37	P	1.3888	187	P	4.253
38	P	1.4371	188	P	4.4881
39	P	1.374	189	P	4.5123

Frame	Frame Type	Motion Errors	Frame	Frame Type	Motion Errors
40	I	60.0461	190	P	4.652
41	P	1.4304	191	P	4.5232
42	P	1.1703	192	P	4.1409
43	P	1.1703	193	P	3.8957
44	P	1.0042	194	P	3.7848
45	P	1.1388	195	P	3.5414
46	P	1.1845	196	I	70.4975
47	P	1.2475	197	P	2.36
48	P	1.3443	198	P	1.9017
49	P	1.4374	199	P	1.9477
50	P	1.4423	200	P	1.9331
51	P	1.3765	201	P	2.1817
52	P	1.2678	202	P	2.6253
53	I	59.6719	203	P	2.6415
54	P	1.7378	204	P	2.6464
55	P	1.4711	205	P	2.4595
56	P	1.3929	206	P	2.4539
57	P	1.4059	207	P	2.4409
58	P	1.4213	208	P	2.5503
59	P	1.44	209	I	62.6636
60	P	1.3436	210	P	2.0847
61	P	1.34	211	P	1.9285
62	P	1.3877	212	P	1.7302
63	P	1.3723	213	P	1.618
64	P	1.3198	214	P	2.1761
65	P	1.3772	215	P	2.5347
66	I	59.8637	216	P	2.9062
67	P	1.5132	217	P	3.1849
68	P	1.3058	218	P	2.9824
69	P	1.1758	219	P	3.0545
70	P	1.1527	220	P	3.2492
71	P	1.2829	221	P	3.3349
72	P	1.2425	222	I	52.8162
73	P	1.2759	223	P	3.0606
74	P	1.1469	224	P	2.847
75	P	1.2695	225	P	2.9138
76	P	1.2551	226	P	3.0529
77	P	1.4357	227	P	2.7443
78	P	1.4989	228	P	2.6886
79	I	58.9522	229	P	3.309
80	P	1.6203	230	P	3.246

Frame	Frame Type	Motion Errors	Frame	Frame Type	Motion Errors
81	P	1.4146	231	P	3.1188
82	P	1.5187	232	P	2.7674
83	P	1.4515	233	P	2.5796
84	P	1.3167	234	P	2.7256
85	P	1.5755	235	I	48.6419
86	P	1.6161	236	P	2.5957
87	P	1.5873	237	P	2.4032
88	P	1.4202	238	P	2.4499
89	P	1.7647	239	P	2.6602
90	P	1.9099	240	P	2.4481
91	P	2.0373	241	P	2.562
92	I	59.0189	242	P	3.2169
93	P	2.0149	243	P	3.315
94	P	1.9805	244	P	3.4733
95	P	1.7789	245	P	3.3718
96	P	1.35	246	P	3.2678
97	P	1.288	247	P	3.4361
98	P	1.0704	248	I	49.9226
99	P	1.1353	249	P	2.6892
100	P	1.0762	250	P	2.1399
101	P	1.0954	251	P	1.9748
102	P	1.0582	252	P	2.3283
103	P	1.1626	253	P	2.0839
104	P	1.1352	254	P	1.9555
105	I	59.1263	255	P	2.3098
106	P	1.363	256	P	2.5963
107	P	0.9388	257	P	2.6498
108	P	0.6779	258	P	2.7223
109	P	0.9314	259	P	2.3169
110	P	1.0046	260	P	2.1758
111	P	1.0263	261	I	49.9501
112	P	1.1096	262	P	2.686
113	P	1.0988	263	P	2.5413
114	P	1.0526	264	P	2.338
115	P	1.0903	265	P	2.0558
116	P	1.1211	266	P	2.1305
117	P	0.9651	267	P	2.0906
118	I	58.8572	268	P	2.1116
119	P	1.3647	269	P	2.2766
120	P	0.9384	270	P	2.0773
121	P	0.8823	271	P	1.6242

Frame	Frame Type	Motion Errors	Frame	Frame Type	Motion Errors
122	P	0.8651	272	P	2.138
123	P	1.0301	273	P	2.2648
124	P	1.0841	274	I	49.7526
125	P	1.1549	275	P	2.5838
126	P	1.2159	276	P	1.7301
127	P	1.3697	277	P	1.4313
128	P	1.3305	278	P	1.7456
129	P	1.3444	279	P	1.9268
130	P	1.2993	280	P	2.014
131	I	59.67	281	P	1.8965
132	P	1.6999	282	P	1.4959
133	P	1.3782	283	P	1.9704
134	P	1.3115	284	P	2.1041
135	P	1.2888	285	P	1.8796
136	P	1.3518	286	P	2.2483
137	P	1.4278	287	I	49.9526
138	P	1.6632	288	P	2.6375
139	P	1.804	289	P	2.0357
140	P	1.8317	290	P	2.1125
141	P	1.8225	291	P	2.2395
142	P	1.9161	292	P	2.1867
143	P	1.8179	293	P	2.1747
144	I	59.3241	294	P	2.1301
145	P	1.673	295	P	2.2757
146	P	1.4372	296	P	2.3169
147	P	1.3837	297	P	2.5481
148	P	1.4266	298	P	2.5935
149	P	1.3873	299	P	2.3752
150	P	1.1045	300	I	50.2771

Figure A-3