Encoding Robust Digital Watermarks Within a Video

Joshua Stallings

Integrated Computer Science

February 15, 2021

Introduction:

Piracy of digital media, such as images, videos, audio files, and so on has become an increasingly common crime with the rise in popularity of digital media. It is not uncommon for pirates to download or record videos off of platforms such as YouTube, Instagram, or other media sharing platforms and repost them on others as their own work. Furthermore, as most of the world has become reliant on digital media it is becoming more and more common for videos, images, or online documents to be wrongly leaked. One way to combat this activity is to encode identifying information, or watermarks, within the files. Hiding data within files is called steganography. By hiding a watermark within a file, specifically a video, the creator of the video can prove that the video belongs to them if it is stolen and reposted on another platform. Furthermore, in the scenario that a video is not meant to be made public, it is possible to give a copy of a video a unique watermark that identifies it to a specific individual. That way if a video is wrongly leaked, it is then possible to view the leaked video's watermark and trace it back to whoever leaked it.

While there exists many forms of image and even audio watermarking, there is relatively little research into the area of digitally watermarking a video.[2] Although watermarking a video can be viewed as an extension of still-image watermarking, which has several robust methods in place, there are key differences that can make image watermarking techniques useless in a video.[4] This includes details such as video compression and decompression techniques in various codecs, redundant data in the frames, and the difference between motion and motionless regions of the video. However, this does not mean that current methods present in normal image watermarking could not be used or adapted to work within a video.

Current popular image watermarking techniques can be classified into using the spatial, wavelet, or frequency domains. The spatial domain deals directly with the pixels of an image whereas wavelet and frequency domains represent images as a sum, amplitude, or frequency. Popular image watermarking methods take advantage of the Discrete Cosine Transform (DCT) , Fourier Transform, and others.[3] However, these watermarks can often be easily destroyed using specific forms of compression; JPEG compression destroys watermarks encoded with DCT, for instance.[6] Other methods that use the spatial domain include using the least significant bit, which is the smallest change in an image as to result in the least amount of perceptible change, or modification to brightness, specific color channels, and so on.[1] Some of these methods split the image into sections, and encode data into each section of the image. In still images, these can often be resistant to cropping and rotating attacks, making them preferable for their robustness.[6] Although watermarking an image is not the same as watermarking a video, they are similar enough that branching out of currently known image-encoding methods may be useful.

The concepts used to encode video watermarks are already similar to image methods; many take use of either the spatial or frequency domain, for instance. As a video is technically a different form of media than an image, there are also ways to encode data within a video other than its frames. Data could be theoretically encoded within the video based on the codec or format used.[2] While spatial based encodings are more conceptually simpler then frequency domain ones, they tend to be more limited in terms of resistance to various attacks.[4] Furthermore, video based encoding methods must be resistant to the types of compression and decompression that naturally occur with various video and codec formats. Video watermarks must also have a sufficient payload, or capacity to store information, while staying imperceptible to the end-user.[5]

Despite the fact that frequency domain based watermarks could potentially be more resistant to cropping, rotating, or other video attacks, there is little evidence or research that a spatial domain based watermark could not be made just as robust while remaining imperceptible and being able to encode significant amounts of data. Furthermore, the simplicity of a watermark made this way would make it easier to implement on a large scale. A potentially robust watermarking method based in the spatial domain could be based on a video's brightness, similar to previous and similar methods.

In order to test the viability of a new watermarking method based on brightness, the approach by which to encode was chosen. The easiest way to do this was to base binary bits on even or odd numbers so that a one bit corresponds to an odd number, while a zero corresponds to an even number. The pixel values of a video change depending on the codec and format used to compress it, and so therefore the amount by which the brightness of a video changes based on format had to be measured. After measuring, the brightness of either whole frames or specific sections of a frame could be changed by an amount greater than the video's compression, making it possible to recover what the intended brightness was. By doing this, data could be possibly encoded and decoded within a video.

Procedures:

Initially, each pixel in the video had its brightness evaluated. Brightness was represented as an average of the intensities of each color space: blue, green, and red. The brightness was compared to the next bit that was being encoded, and depending on the value of the bit, the color intensities of the pixel were increased by one or zero. A one-bit was represented by pixels with an odd brightness, and a zero-bit was represented by a pixel with even brightness; if the brightness was not even or odd and it should have been, then each color space in the pixel was modified to make it even or odd. This was only done for one frame, or one image of the video, as the number of pixels per frame would have been sufficient enough to encode a large amount of data. However, this was done before the encoding process of the video was taken into account; various video codecs compress and decompress videos as needed, which thus changes the pixel values of a frame by different amounts, meaning changing the pixel values by one would not be sufficient.

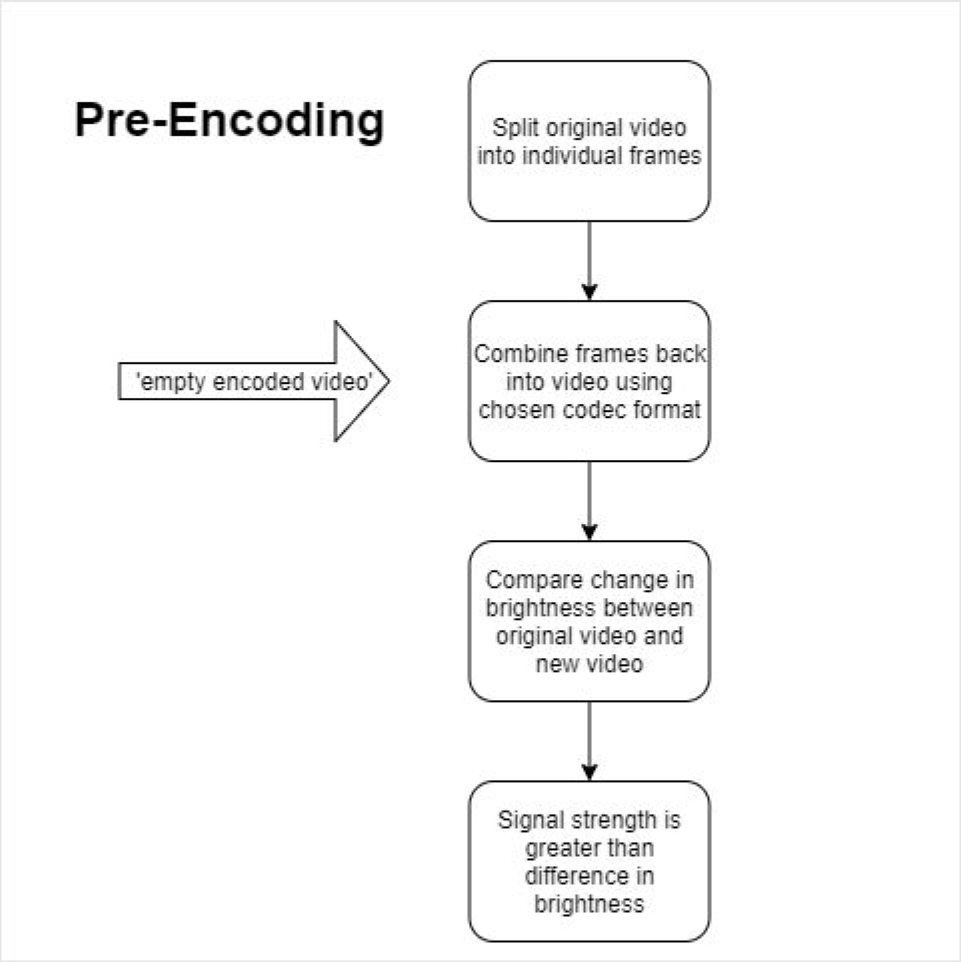
Pre-Encoding
Split original video into individual frames
'empty encoded video'
Combine frames back into video using chosen codec format
Compare change in brightness between original video and new video
Signal strength is greater than difference in brightness

Figure 1

In order to measure the amount by which pixel values changed according to a certain video codec, the original video had to be broken down into individual images before being put back together as a video. After this type of empty-encoding was performed on the video, the values of pixels in the original and empty-encoded video were compared. As going through every pixel in every frame of the video would have taken a long time, each frame was split into sixteen by sixteen blocks of pixels. The brightness of these blocks in both original and empty-encoded video was recorded and the difference taken into account. Once this was done, the maximum difference between the brightnesses could be found (fig 1). For the DivX codec, for instance, this was found to be approximately three.

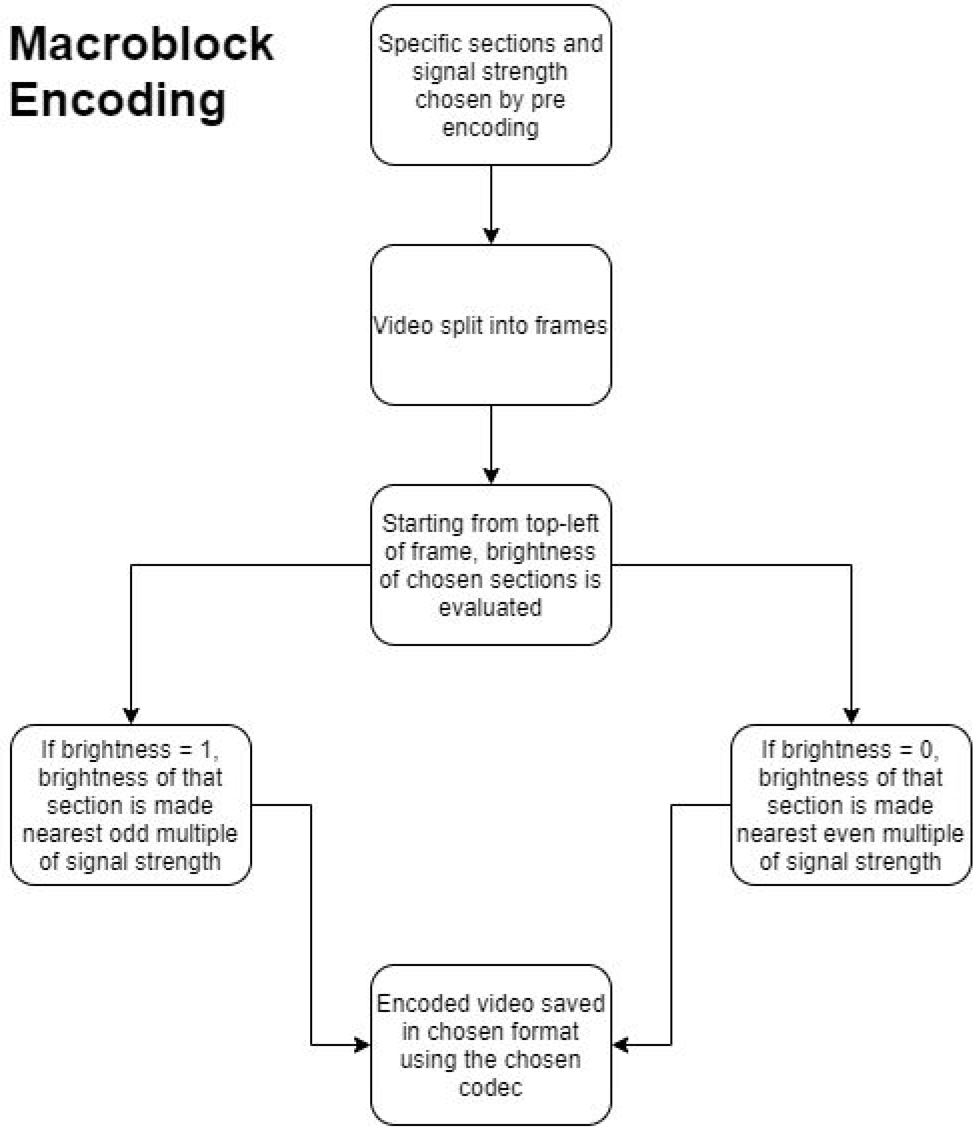
Macroblock Encoding
Specific sections and signal strength chosen by pre encoding
Video split into frames
Starting from top-left of frame, brightness of chosen sections is evaluated
If brightness = 1, brightness of that section is made nearest odd multiple of signal strength
If brightness = 0, brightness of that section is made nearest even multiple of signal strength
Encoded video saved in chosen format using the chosen codec

Figure 1.1

However, in order to minimize the amount by which each frame is modified, the blocks of each frame that changed the least could be specifically encoded in. Alternatively, the average difference in brightness could be recorded, and the sections of video that changed by or below the average could be encoded in (fig 1.1). The benefits of this could be twofold: if the amount by which each frame is changed is small, and the signal that is used to encode data is also small, then the changes in brightness could be virtually imperceptible. However, as the data is encoded to a specific chunk of the video, this method would not be robust against cropping attacks.

With the original method not being robust against the video being cropped, which would destroy the encoded data, another one was developed. The alternative method uses the brightness of the whole frame rather than specific chunks of the video. Besides that, most of the previous procedure was adapted to work with this. The change in brightness between the original video and the empty encoded video was recorded, and the maximum change in brightness was also noted. The brightness of each frame was changed by a value greater than that when encoding data, so that way the codec's compression method would not destroy the data. Furthermore, since the data was not encoded into specific parts of the video, but the whole frame, the video was tested against cropping attacks. In order to test this, the encoded video was cropped by five percent from all four sides in Adobe Premier and saved in the same file format. Then, the video was decoded as it would have been if it was not cropped.

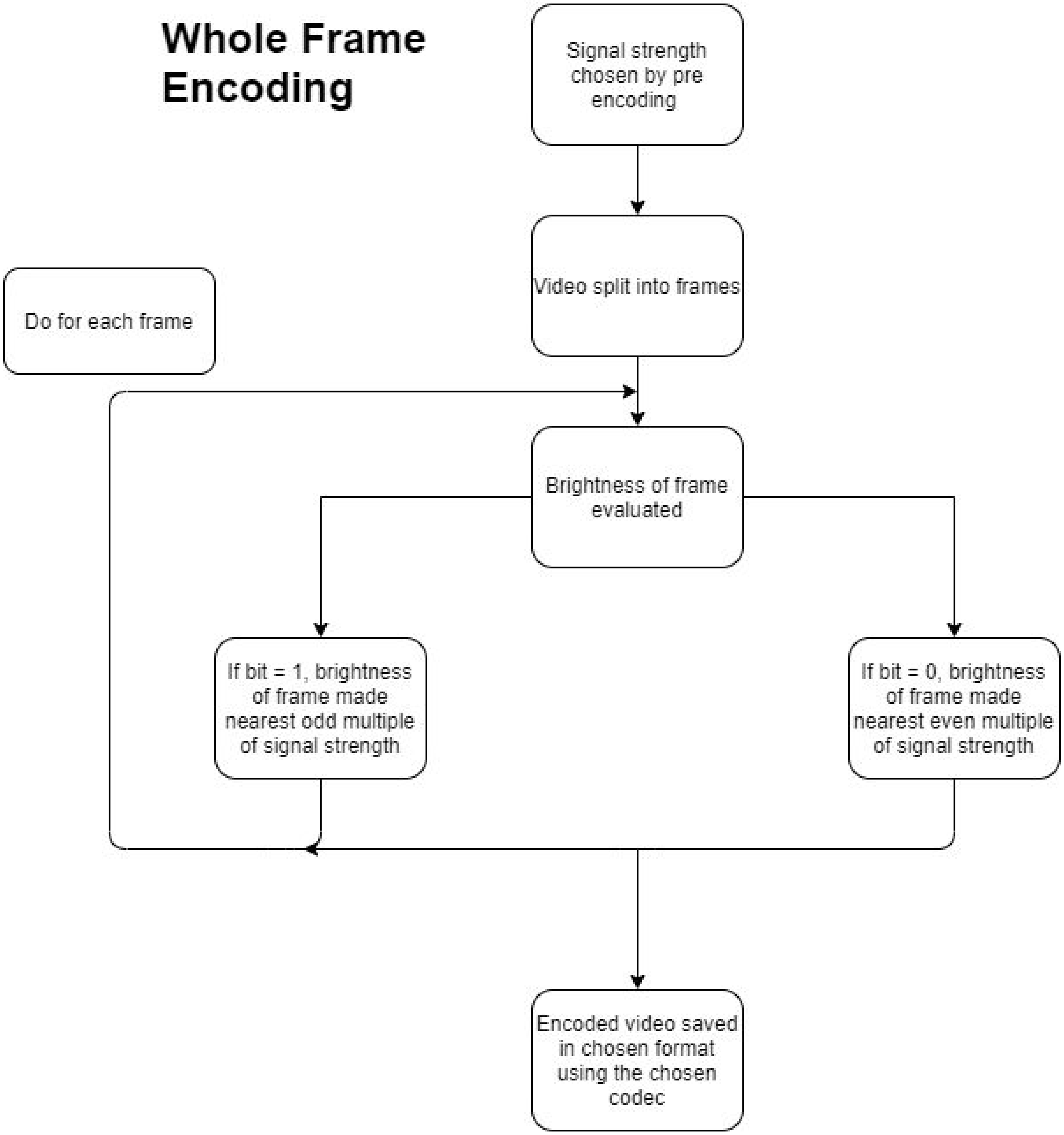
Whole Frame Encoding
Signal strength chosen by pre encoding
Video split into frames
Do for each frame
Brightness of frame evaluated
If bit = 1, brightness of frame made nearest odd multiple of signal strength
If bit = 0, brightness of frame made nearest even multiple of signal strength
Encoded video saved in chosen format using the chosen codec

Figure 1.2

When encoding data in either of the two methods, the same logic was used; a number by which the pixel values would be modified was chosen, called the signal strength, according to the difference in brightness between the original video and the empty-encoded videos. The brightness of either the frame or section of the video would be evaluated and modified depending on whether or not the bit was one or zero (fig 1.2). For instance, if the bit was zero then the brightness would be changed to the nearest even multiple of the signal strength, and if the bit was one it would be to the nearest odd multiple.

To decode the data, the brightness of the same section or frame is evaluated and since the signal strength is larger than the amount by which the codec's compression changes the video, the brightness prior to being compressed can be recovered. A recovered brightness that is odd signals a one was encoded, or a zero if it is even. Once that was done, the binary data could be converted into whatever form it was originally, such as text or an image (fig 1.3).

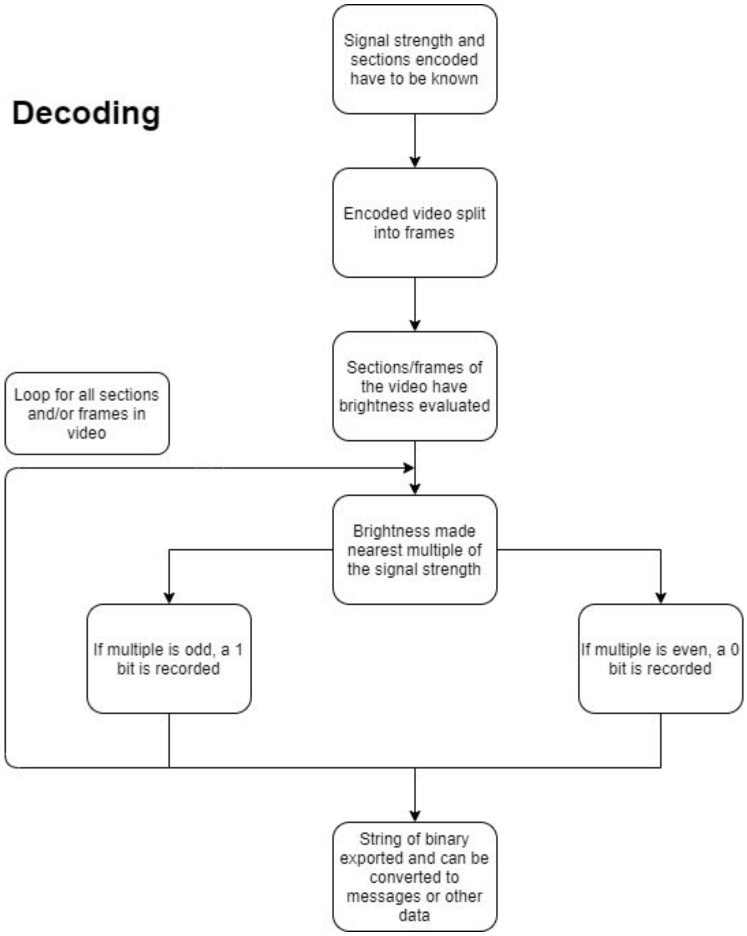
Decoding
Signal strength and sections encoded have to be known
Encoded video split into frames
Sections/frames of the video have brightness evaluated
Loop for all sections and/or frames in video
Brightness made nearest multiple of the signal strength
If multiple is odd, a 1 bit is recorded
If multiple is even, a 0 bit is recorded
String of binary exported and can be converted to messages or other data

Figure 1.3

Data and Results:

For both methods used, a binary message of varying lengths was encoded into the video and then decoded. Binary was used because both messages and images can be broken down into ones and zeros and later put back together. A technique perfectly resistant to the video codec's compression would mean that when the video was decoded, all the bits would match the ones encoded. However, it did not need to be perfectly resistant as the message could still be discernible if the data is roughly the same.

For the method that split the video's frames into sixteen-by-sixteen sections, the decoded messages were almost the same as the encoded message. In a message forty-eight bits long, only one bit was off. For a message eighty-eight bits long, the decoded data was inaccurate by two to four bits, depending on the signal strength that was used. Although the data seems to become more inaccurate as the length of the message increases, this can be resolved by splitting the data across multiple frames.

Encoding in the whole frame of the video rather than sections of it was not as accurate as expected. The accuracy depended largely on the signal strength used. For instance, signal strengths of fifteen and thirteen had eight and thirteen bits wrong, respectively. A signal strength of seven only had five bits of a one-hundred and forty-four-bit message wrong. Moreover, encoding in whole-frames was not resistant to cropping attacks.

In terms of visibility, both methods were virtually imperceptible to a viewer so long as a low enough signal strength was used. A smaller signal strength led to a higher chance of data being lost between the encoding and decoding process, but the data lost was still relatively small. Furthermore, encoding in whole frames was more visible when compared to encoding in specific, small sections of the video.

Conclusion:

Due to the nature of the increasingly digital world it would be useful to have a way to copyright videos while keeping the watermark in the video imperceptible to the end user. By doing this, potential pirates would have a harder time identifying and removing the watermark, and end-users, or viewers, of the video would have a more enjoyable experience watching the video. It would also make it easier for creators to prove stolen content was theirs, as the watermark encoded within the video could be specifically identified to them using an encrypted key or similar. Lastly, imperceptible yet robust digital watermarks could also make it possible to trace back where a wrongly leaked video originated from, thus increasing the security of confidential affairs.

Although research into digitally watermarking videos is lacking relative to that of still-image watermarking, it is possible to expand image watermarking methods to videos. The most common concepts used to encode images with a watermark is either through the spatial or frequency domain; the spatial domain utilizes the actual pixels of the image, whereas the frequency domain is based on amplitudes and frequencies of various values within the image. Popular spatial domain techniques utilize splitting videos into chunks or modifying brightness to encode information, thus making them simpler than frequency domain techniques, which use methods such as Discrete Cosine Transform (DCT).

It would be preferable to identify a technique that can robustly and imperceptibly encode data within a video using a spatial domain technique, as methods based on this are often easier to understand, which can make them easier to work with and understand. Furthermore, there is existing research in brightness based techniques on still images that mean they could be resistant to various attacks, such as video cropping, rotating, and compression.

Two similar methods were utilized to encode binary data in a video. One technique split the video's frames into sections of sixteen by sixteen pixels called macroblocks. The brightness of a particular section was made either an even or odd multiple of a chosen signal strength. The other method used the entire frame rather than specific sections of the frame, and encoded with the same logic; the brightness of the frame was made an even or odd multiple of a particular number, depending on whether or not the bit being encoded was a zero or one. The video was cropped in order to test how resistant the second method was to cropping attacks. Decoding worked the same way for both methods; the brightness of the macroblock or frame, depending on the method, was evaluated. Since the encoded brightness was originally a multiple of a particular number, the brightness of the video could be set to the nearest multiple of that number. Either a one or zero was output depending on whether or not that multiple was even or odd.

Overall, both methods proved effective in reliably and unnoticeably storing data within a video. Although the visibility of the video manipulation as a result of encoding largely depended on the signal strength, or by how much the brightness was modified, the encoded videos were qualitatively similar. Data encoded in the videos could be reliably decoded, with only one to four bits being inaccurate. However, cropping attacks, or cropping sections of the video, ultimately destroyed the watermarks.

The two methods were successful in that they could imperceptibly store potentially large amounts of data while retaining an acceptable amount of accuracy. However, the watermarks were ultimately destroyed by cropping attacks. Future research based on a brightness based technique resistant to cropping, rotating, and possibly even color grading would be the most beneficial. It would also be interesting and helpful to know whether or not frequency domain based techniques could be applied as easily as this one while being more robust to various attacks meant to destroy the watermark.

References:

1. Bassil, Youssef. "Image Steganography Method Based on Brightness Adjustment." In *Advances in Computer Science and its Applications,* Vol. 2, No. 2 (2012). arxiv.org/ftp/arxiv/papers/1212/1212.5801.pdf.

2. Griberman David and Pavel Rusakov. "Comparison of Video Steganography Methods for Watermark Embedding." *Applied Computer Systems* (2016). doi.org/10.1515/acss-2016-0007.

3. Harsh Verma, Abhishek Singh, and Raman Kumar. "Robustness of the Digital Image Watermarking Techniques against Brightness and Rotation Attack." *International Journal of Computer Science and Information Security* Vol. 5 No. 1 (2009) arxiv.org/ftp/arxiv/papers/0909/0909.3554.pdf.

4. Jayamalar and V. Radha. "Survey on Digital Video Watermarking Techniques and Attacks on Watermarks." In *International Journal of Engineering Science and Technology,* Vol. 2 No. 12 (2010). www.researchgate.net/publication/50384243_Survey_on_Digital_Video_Watermarking_Techniques_and_Attacks_on_Watermarks.

5. Mahbuba Begum and Mohammad Uddin. "Digital Image Watermarking Techniques: A Review." 17 February 2020. www.researchgate.net/publication/339348660_Digital_Image_Watermarking_Techniques_A_Review.

6. Verma, Bhupendra, Sanjeev Jain, D. P. Agarwal, and Amit Phadikar. "A New Color Image Watermarking Scheme". *INFOCOMP Journal of Computer Science* (2016) http://infocomp.dcc.ufla.br/index.php/infocomp/article/view/141.