CS 888 Project: Steganographic Watermarking Using Stereo-3D Disparity Channels

Karl Knopf
kknopf@uwaterloo.ca
University of Waterloo
Waterloo, Ontario, Canada

ABSTRACT

As the demand for stereo-3D images grows, so to does the need to protect those images from unauthorized use. In this work, we present a watermark embedding pipeline that takes advantage of the unique disparity property between stereo-3D image pairs. A prototype system implementing this pipeline was developed and evaluated, with limited success. However, this work can still serve as a template for future schemes using disparity as the embedding channel.

KEYWORDS

Stereo-3D, Steganography, Disparity, Watermarking

ACM Reference Format:
Karl Knopf. 2020. CS 888 Project: Steganographic Watermarking Using Stereo-3D Disparity Channels. In *Woodstock '18: ACM Symposium on Neural Gaze Detection, June 03–05, 2018, Woodstock, NY*. ACM, New York, NY, USA, 5 pages. https://doi.org/10.1145/1122445.1122456

1 INTRODUCTION

As stereo-3D images become accepted as image medium, so to will the transmission of stereo-3D information over public channels. One may want to embedded additional information into an image they send, either to watermark their work or to hide a message. When using a stereo-3D image pair, embedding becomes a more difficult problem, as the final images must still be a valid stereogram.

A common approach for hiding data in another medium is known as steganography [18]. With this technique, the goal is to prevent an adversary from learning of the existence of the message, while still allowing the indented recipient access to the hidden information. Steganography has uses in secret communication and the watermarking of proprietary information. The data should be invisible, so no independent observer should be able to tell that it is there, and robust, where limited manipulation of the cover image does not destroy the meaning of the hidden message. The capacity of the steganographic scheme is also important, as it dictates the size the message.

Prior work on image steganography has focused on manipulating the colour information of the pixels to hide information [6]. With stereo-3D image pairs, the perceived difference in the image's location otherwise known as binocular disparity, is another potential channel. This information can be represented as a disparity map between the two images.

In this work, we introduce the notion of steganographic embedding into the disparity channel of the stereo-3D image pair. As common approaches to steganoanalysis [11] rely on direct investigation of the cover images, we believe that a scheme that takes advantage of this channel unique to stereo-3D image pairs may prove to be more robust in the face of well-known adversarial attacks. Our contributions can be summarized as follows:

- We present a pipeline for embedding and recovering information into a stereo-3D disparity map.
- We have developed a prototype implementation of this pipeline and present initial results.
- We describe how we can evaluate this prototype in comparison to current state-of-the-art stereo-3D watermarking schemes.

The rest of the paper is organized as follows: First there is a discussion of necessary background material. Then the steganographic pipeline methodology is presented as well as a description of the design of the prototype. Initial results created by the prototype are then described, before a method of evaluation for the scheme is given. Finally, a discussion of future directions and a summary are provided.

2 BACKGROUND

A popular medium for steganographic techniques are digital images, where information is embedded into an image file. There are three major categories of digital image steganographic techniques [6]. Cover modification (CMO) involves directly changing the information of a cover image to include the secret data. The cover selection (CSE) technique allows the steganographer to choose an image from a database that corresponds to the chosen message. This approach is known for its low theoretical capacity. Cover synthesis (CSY) has the steganographer generate a cover image containing the secret information. The generation of convincing steganographic images is thought of as a difficult problem, however recent advances in GAN approaches have shown to be promising [12].

The most popular technique is cover modification [1], which has two key approaches. They are known as least significant bit modification (LSB) and frequency domain modification. LSB works by replacing the least significant bits of random pixels in the cover image with the message [11]. Frequency domain modification involves changing the coefficients of the encoding scheme used by the image compression algorithm, so that during decompression the coefficients can be used to reconstruct the message [1].

Woodstock '18, June 03–05, 2018, Woodstock, NY

ACM ISBN 978-1-4503-XXXX-X/18/06…$15.00
https://doi.org/10.1145/1122445.1122456

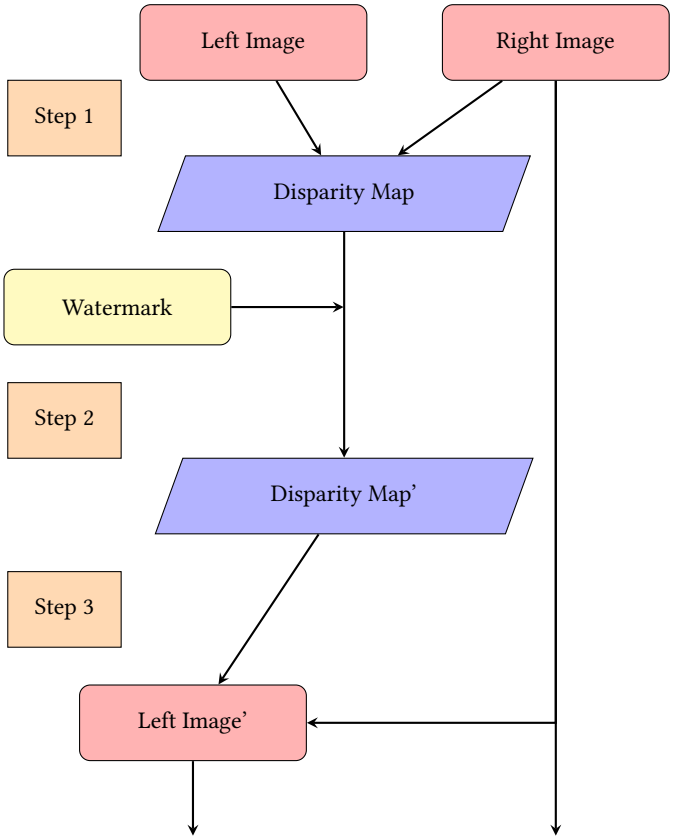

Figure 1: Embedding Pipeline

Steganographic techniques have been applied not just to monocular digital images, but to stereoscopic-3D image pairs as well. Generally, watermarking of anaglyph image pairs is done using a frequency domain approach, primarily the discrete wavelet approach, to embed information into one of the colour channels (red or cyan [15]) in both images of the pair [5]. Some techniques have been created to make use of one of the images in the pair [14], while others have extended prior work to anaglyph-3D videos [4, 13].

3 METHODOLOGY

The following is our proposed pipeline for watermarking stereo-3D image pairs by embedding the message into the disparity information between them. Figure 1 demonstrates the flow of how the message is embedded and Figure 2 demonstrates the flow of how the message is recovered. There are 5 key steps to this process.

3.1 Step 1: Generate the Disparity Map

The first step for embedding information into the disparity of the stereo image pair is to generate a disparity map using the two images. A disparity map is a projection of the apparent difference in location between two corresponding pixels in the stereo image pair. This difference is computed using a stereo matching algorithm. There have been many approaches for stereo matching [7, 17], including recent approaches based on machine learning algorithms [19].

If it was not done prior to the start of this pipeline, the input images will also need to be rectified before the generation of the disparity map. Otherwise there may be errors introduced by vertical disparities.

In our implementation, we use a block matching algorithm [10] to generate the disparity map. Our implementation is based on the algorithms from the OpenCV library [2], as they were easy to adapt. Using a more sophisticated technique will lead to more accurate depth estimations, but it may make the design of an embedding recovery algorithm more difficult as intentional perturbations of the disparity may be eliminated. The sum of absolute differences (SAD) is used as the metric to determine to matching blocks in our implementation.

3.2 Step 2: Embed the Watermark

The second step of this pipeline is the embedding of the watermark into the disparity information. This is done by applying a digital image steganographic CMO algorithm to modify the values of the disparity map.

In our implementation, described in Algorithm 1,we use a simple linear modification scheme where the value of the disparity is linearly increased in pixels that are assigned to be cover pixels. The changes to the disparity values need to be large enough such that they will be able to be recovered, but not large enough that they cause obvious errors in Step 3. Thus the size of the shift may need to be tuned to the particular stereo-image pair. The cover pixels are chosen to be the centres of the blocks used in the block matching algorithm.

Algorithm 1: Disparity Embedding Algorithm

```
input  : Disparity Map, Watermark Image
output : Modified Disparity Map
for Each pixel in Watermark Image do
    if pixel value is equal to 0 then
        increase corresponding pixel value by a significant
          amount;
    end
end
```

3.3 Step 3: Synthesize New Left Image

The final step of the embedding portion of the pipeline is the creation of a new left image. Using the right image, one can project the corresponding pixels from the disparity map to generate a left image. This will lead to holes forming in the new image, in places where there is no corresponding value, so an in-painting algorithm will need to be applied. The left image was chosen arbitrarily, as one could follow the same approach to generate right images.

In our implementation, we use a simple interpolation algorithm for hole-filling in the synthetic image. Here, following a linear search for holes, the largest of the surrounding pixel values is selected as the new value for the hole. This will not lead to very

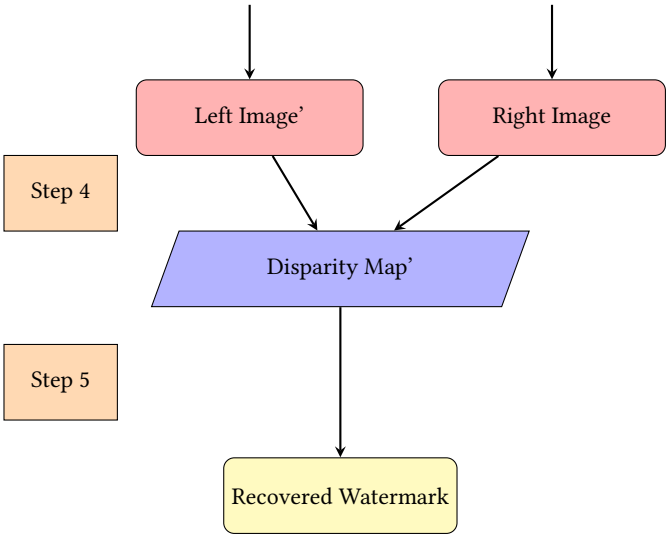

Figure 2: Extraction Pipeline

convincing results, however the goal of our implementation is to demonstrate the feasibility of the pipeline. In a real-world setting a better in-painting algorithm, such as the one from [9], would be more appropriate.

3.4 Step 4: Recreate the Disparity Map

This step of the pipeline involves generating a disparity map from a stereo image pair. It is an similar process to the one that was discussed in Step 1. There is likely to be slight differences between the original map constructed and the one constructed in this step, as the stereo-matching algorithms will likely introduce some noise.

In our implementation, the block matching process described in Step 2 is used again, except it is modified to not make use of the centre pixel in the block when determining the best matching block. Instead, the value of the centre pixel in the disparity map is assigned the value of the disparity when it is included in the matching process. Thus the disparity map generating algorithm is run twice; once for the block and once for the centre pixel. This is done to make our implementation of Step 5 easier.

3.5 Step 5: Recover the Watermark

The final step of the pipeline is the recovery of the watermark. This is done by applying the decoding algorithm corresponding to the embedding algorithm used in Step 2. Due to potential errors stemming from the conversion process, it is advised that the algorithm used for recovery be robust.

In our implementation of this pipeline, to invert the encoding scheme from Step 2 we need to generate an almost identical disparity map as with Step 1 but we do not include the centre pixels in our calculation for the block matching. This is generated as part of the disparity map from Step 4. The idea behind this inversion is that the centre pixel should have a large disparity value if a message was encoded in that location, so it should have a strong influence on choosing the corresponding block. Thus we can see by the shifts in calculated disparity if there was a message. This process is described in Algorithm 2.

Figure 3: Test Watermark

Algorithm 2: Watermark Recovery Algorithm

```
input  : Recovered Disparity Map
output : Recovered Watermark
for Each pixel that could have an embedding do
    if difference in disparity when including centre pixel is
      large then
        Set value of corresponding pixel in watermark to 0;
    else
        Set value of corresponding pixel in watermark to
          255 ;
    end
end
```

4 RESULTS

We have implemented the proposed pipeline in Python3, and using some functions from the OpenCV library [2]. This prototype has been designed to work on gray-scale images only, as colour channels may weaken the signal from the embedding. The following is a qualitative evaluation of the steps of the pipeline using an example stereo-image pair.

4.1 Test Case: Baby1

Foe this test case, we have chosen to use Baby1, a stereo-3D image pair from the 2006 Middlebury bench-marking data set [8]. This image pair was accompanied by a ground truth disparity map, which was used instead of generating a disparity map for Step 1. A standard 32x32 watermark, as seen in Figure 3 was borrowed from [3] to allow for comparison to other stereo-3D watermarking techniques. The results for each step of the pipeline are shown in Figure 4.

In Figure 4, the first column shows the original left image of the stereo-3D pair. The second column are the left images synthesized in Step 3, where the top image contains the watermark information while the bottom image does not. These images appear almost identical, which suggests that the embedding procedure does not affect the visual perception of the generated image.

The third column shows the recovered disparity map from Step 4. Again, the map including the embedding and not including the embedding are very similar. Finally the fourth column represents the recovered watermark from Step 5. These results are still similar, which suggests the watermarking scheme is not very robust.

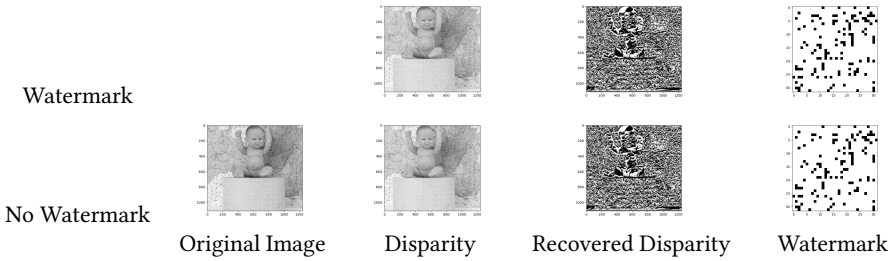

Figure 4: Baby 1 Pipeline

Aloe	73.61303804916142
Baby1	76.47164814571376
Bowling	79.63363796244104
Cloth	73.7014074198925
Plastic	78.1929892112964
Average	76.322544157701

Figure 5: PSNR for five test image-pairs (dB)

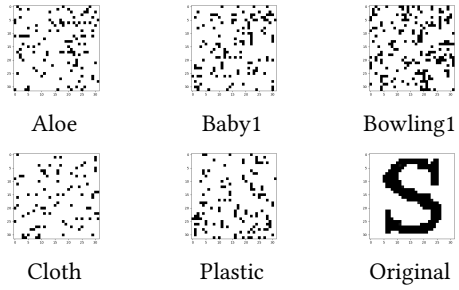

Figure 6: Comparison of Recovered Watermarks

This pipeline was applied to 4 other Middlebury image pairs (Aloe,Bowling1,Cloth,Plastic) and the results were all consistent with Baby1. The failure to recover the watermark suggests that actual disparity values will overpower the signal created by the embedding process.

5 EVALUATION

For a given watermarking scheme, there are three criteria that it can be evaluated on. First is invisbility, or how imperceptible the watermark is when embedded in the cover image. Second is robustness, or how well the watermark can be recovered when the cover image has been manipulated by an adversary. Finally, we can evaluate a scheme on capacity, or how large of a watermark can be successfully recovered using this scheme.

5.1 Invisibility

There are two important objectives when testing for the invisibility of a watermark. The first is that it can convince a human observer that there has been no embedding. The second is that it can convince a program looking for noisy values that it does not contain a signal.

The first objective appears to be achieved, as seen in Figure 4 column 2. To confirm the imperceptibility of the watermark to human, as user study would need to be carried out. This is not feasible at this time.

The second objective can be tested using a peak signal to noise ratio (PSNR) [16]. This is commonly expressed as the decibel value of the mean squared error between the pixel values of the image with watermark and the image without. The PSNR for this scheme for the five test images can be seen in Figure 5. Given that a watermarking scheme with a PSNR greater than 51dB is considered invisible [3], the proposed scheme will meet this requirement with an average PSNR of 76.322544157701 dB.

5.2 Robustness

To evaluate robustness, we must first demonstrate that we can recover the watermark successfully without the presence of an adversary. Figure 6 contains the results of watermarking scheme when used on 5 test stereo-image pairs. It is clear that the scheme does not recover the watermark very well, if at all.

Increasing the shift in disparity in Step 2 does not seem to meaningfully affect the quality of the watermark. We have also tried changing the threshold for significance in algorithm 2, and changing the block size but this appears to have no affect. Earlier results showed more promise when using larger watermarks, but a bug in the code was discovered which erroneously allowed for recovered left image to be included in the extraction process. Once correct, similar results to what is shown in Figure 6.

Given that the quantitative results are not good, there is no reason to investigate more formal measures of robustness. The most frequent measure is the normalized correlation coefficent [3] between the original watermark (OW) and the recovered watermark (RW). This can be expressed in the following equation:

$$\phi(OW, RW) = \frac{\sum_i [OW_i - OW_{mean}] * [RW_i - RW_{mean}]}{\sum_i [OW_i - OW_{mean}]^2 * [RW_i - RW_{mean}]^2} \quad (1)$$

If this coefficient shows good results, one can then evaluate the recovery rate of the watermark against known attacks such as mean filtering, rotation, and scaling [16].

5.3 Capacity

To evaluate the capacity of the watermarking scheme, one should try to find the maximum size of the watermark that the scheme can successfully embed. Given that the prototype uses a block matching

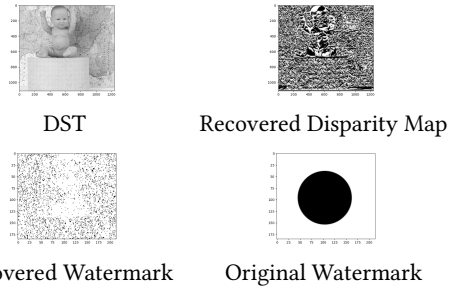

Figure 7: Result from higher capacity watermark

scheme generate the disparity map, the maximum number size of the watermark is thus the number of blocks.

To test this, a larger (640 pixels x 466 pixels) circular watermark was created and embedded into the Baby1 image pair. Figure 7 shows some of the results of this embedding. As seen with the smaller watermarks, the generated left image and recovered disparity map appear to be useful but the recovered watermark is not close to the original. In fact, the scheme seems to include more of the cover image than the watermark in the recovered image. This suggests again that the technique used to recover the watermark can not successfully separate the true disparities from the watermark signal.

6 FUTURE WORK

Given some of the lackluster results from the prototype, we believe that it is important to address the future directions for this work. The first important step is to develop a better watermark recovery approach than Algorithm 2. From the invisibility experiments, it appears that the embedding procedure used maybe useful but the robustness and capacity experiments show that the watermark recovery algorithm is not. This new technique will need to be more sensitive to changes in the disparity of the recovered left image, suggesting that a better algorithm for generating the disparity map will be needed. Thus a critical next step will be to determine an appropriate stereo-correspondence algorithm and determine a new scheme based on its properties.

Another future direction should be the expansion of this work to colour images. This will allow us to directly compare the results of this scheme with the prior work done in stereo-3D steganography. We may also want to include a better in-painting algorithm to fill the holes in the generated left image from Step 3. This will help with the invisibility of the scheme, as ideally a human should not be able to determine that there has been any modification to the left image.

7 CONCLUSION

In this work we have presented a watermarking pipeline for embedding images into the disparity information of stereo-3D image pairs. A prototype implementation of this pipeline was developed, and initial tests were not lackluster but promising. With the ideas and methodology presented in this paper to build upon, we can hopefully design a more novel and robust way for watermarking stereo-3D image pairs. We then hope it will be possible to extend this idea to other stereo-3D content, such as stereo-3D video to further demonstrate the applicability of this work.

REFERENCES

[1] Samer Atawneh, Ammar Almomani, and Putra Sumari. 2013. Steganography in digital images: Common approaches and tools. *IETE Technical Review* 30, 4 (2013), 344–358.
[2] G. Bradski. 2000. The OpenCV Library. *Dr. Dobb's Journal of Software Tools* (2000).
[3] Hidangmayum Saxena Devi and Khumanthem Manglem Singh. 2017. A novel, efficient, robust, and blind imperceptible 3D anaglyph image watermarking. *Arabian Journal for Science and Engineering* 42, 8 (2017), 3521–3533.
[4] Dorra Dhaou, Saoussen Ben Jabra, and Ezzeddine Zagrouba. 2019. An Efficient Anaglyph 3D Video Watermarking Approach Based on Hybrid Insertion. In *International Conference on Computer Analysis of Images and Patterns*. Springer, 96–107.
[5] Dorra Dhaou, Saoussen Ben Jabra, and Ezzeddine Zagrouba. 2019. A review on anaglyph 3D image and video watermarking. *3D Research* 10, 2 (2019), 13.
[6] Jessica Fridrich. 2009. *Steganography in digital media: principles, algorithms, and applications*. Cambridge University Press.
[7] Rostam Affendi Hamzah and Haidi Ibrahim. 2016. Literature survey on stereo vision disparity map algorithms. *Journal of Sensors* 2016 (2016).
[8] Heiko Hirschmuller and Daniel Scharstein. 2007. Evaluation of cost functions for stereo matching. In *2007 IEEE Conference on Computer Vision and Pattern Recognition*. IEEE, 1–8.
[9] Lesley Istead, Paul Asente, and Craig S Kaplan. 2016. Layer-based disparity adjustment in stereoscopic 3D media. In *Proceedings of the 13th European Conference on Visual Media Production (CVMP 2016)*. 1–9.
[10] Andreas Koschan, Volker Rodehorst, and Kathrin Spiller. 1996. Color stereo vision using hierarchical block matching and active color illumination. In *Proceedings of 13th International Conference on Pattern Recognition*, Vol. 1. IEEE, 835–839.
[11] Bin Li, Junhui He, Jiwu Huang, and Yun Qing Shi. 2011. A survey on image steganography and steganalysis. *Journal of Information Hiding and Multimedia Signal Processing* 2, 2 (2011), 142–172.
[12] Jia Liu, Yan Ke, Zhuo Zhang, Yu Lei, Jun Li, Minqing Zhang, and Xiaoyuan Yang. 2020. Recent Advances of Image Steganography with Generative Adversarial Networks. *IEEE Access* 8 (2020), 60575–60597.
[13] Zhihan Lu, Shafiq ur Réhman, Muhammad Sikandar Lal Khan, and Haibo Li. 2013. Anaglyph 3d stereoscopic visualization of 2d video based on fundamental matrix. In *2013 International Conference on Virtual Reality and Visualization*. IEEE, 305–308.
[14] Fuminori Matsuura and Nobuyuki Fujisawa. 2008. Anaglyph stereo visualization by the use of a single image and depth information. *Journal of visualization* 11, 1 (2008), 79–86.
[15] Ruchika Patel and B Parth. 2015. Robust watermarking for anaglyph 3D images using DWT techniques. *International Journal of Engineering and Technical Research (IJETR)* 3, 6 (2015), 55–58.
[16] Fabien AP Petitcolas. 2000. Watermarking schemes evaluation. *IEEE signal processing magazine* 17, 5 (2000), 58–64.
[17] Daniel Scharstein and Richard Szeliski. 2002. A taxonomy and evaluation of dense two-frame stereo correspondence algorithms. *International journal of computer vision* 47, 1-3 (2002), 7–42.
[18] CP Sumathi, T Santanam, and G Umamaheswari. 2014. A study of various steganographic techniques used for information hiding. *arXiv preprint arXiv:1401.5561* (2014).
[19] Feihu Zhang, Victor Prisacariu, Ruigang Yang, and Philip HS Torr. 2019. Ga-net: Guided aggregation net for end-to-end stereo matching. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 185–194.