GEOMETRIC HASHING TECHNIQUES FOR WATERMARKING

H.Z. Hel-Or, Y. Yitzhaki

Department of Computer Science
Haifa University
Haifa 31905, Israel

Y. Hel-Or

Department of Computer Science
The InterDisciplinary Center
Herzeliya, Israel

ABSTRACT

In this paper we introduce the idea of using Computer Vision techniques for improving and enhancing watermarking capabilities. Specifically, we incorporate Geometric Hashing techniques into the watermarking methodology. Geometric Hashing was developed to detect objects in a visual scene under a class of geometric transformations. The technique is incorporated into watermarking to detect watermarks encoded under transformations. This allows randomization of the watermark code without the need of maintaining the Random Generator seed. In turn, this randomization increases robustness under attacks such as collusion (determining the watermark from multiple watermarked examples). Depending on the embedding domain, robustness of the watermark under geometric attacks can be achieved.

1. INTRODUCTION

Digital Watermarking is a technique for marking and labeling digital images. The various techniques are based on image processing, channel communication, spectrum spreading, and others. We introduce the idea of exploiting Computer Vision techniques to enhance watermarking capabilities. Specifically, techniques that deal with object recognition under various views and transformations can be exploited by considering the watermark code as an object model. This allows a given watermark to be embedded under a large number of possible transformations, without the need of maintaining the transformed watermark nor the transformation parameters for decoding.

Many watermarking techniques embed a watermark code by perturbing image values. The perturbation is performed either in the image domain or in some other image representation domain such as frequency or local frequency domains. Perturbing image values at fixed, predefined locations produces watermarks that are susceptible to collusion attacks. Under collusion, the watermark can be removed or mimicked by combining information from several watermarked images. To prevent such attacks, watermarking techniques introduce randomness and variability into the selection of locations at which image values are varied. Thus, either various predefined patterns of locations are chosen for each image, or a random generator seed (RNGS) is chosen for each image which produces the random image locations. This introduces the need for maintaining the RNGS or the specific location pattern for each image in order to decode the watermark. In this paper, we introduce a new technique which eliminates the need of maintaining these randomization parameters. Specifically, we introduce Computer Vision techniques into the watermarking process, in which the watermark can be detected under a large number of transformation parameters. These transformation parameters serve as the randomization factor in watermarking and are not required for watermark extraction. When combined with watermarking techniques that inherently include randomization, involving Computer Vision techniques can increase the variability and randomization.

Additionally, using such Computer Vision techniques, while embedding a watermark in an appropriate domain, may increase robustness of the watermark under geometric attacks, since, transforming an image in which a model is embedded can be viewed as applying the inverse transformation on the model itself.

As an example of this approach we incorporate Geometric Hashing into watermarking, which allows detection of object models under a number of 2D transformations such as translation, rotation, scale and shearing. The technique can be incorporated into many different watermarking techniques. We demonstrate the approach using a specific DCT domain watermarking technique.

2. GEOMETRIC HASHING

Geometric Hashing is a Computer Vision tool developed to recognize object models in a visual scene. The models may appear in the scene under a number of transformations. Details of the Geometric Hashing technique can be found in [1]. A short review is provided here.

Restricting the discussion to 2D models and 2D transformations, a *model* is a set of 2D points given as a set of 2D coordinates $M = \{P_i = (x_i, y_i)\}$ $(i = 1 \ldots n)$. The model may undergo a number of transformations including translation, rotation, scale and shear. The goal is to find the model in a collection of 2D points called the *measurement* points. The measurement points may include many irrelevant points (noise) and may be missing some model points (occlusion).

Geometric Hashing recognizes a model using a voting scheme, where each measurement point votes for all suitable models. The model with the highest vote is given as output. To deal with models under varying transformations, the model and measurement are represented in a transformation invariant representation prior to comparison and voting.

Preprocessing:
Given a set of models $M_1, \ldots, M_t$, a Hash Table is built. The models are inserted into the Hash table after normalization in which the points of each model are represented in a transformation invariant representation as follows (Figure 1):

1. Two points of the model are chosen, P_0 and P_1. These are denoted the *base points*.
2. A 2D transform is found that maps P_0 to the origin and P_1 to the coordinates (0,1).
3. All model points are transformed using the 2D transformation found in Step 2.

All coordinates of the normalized model points are inserted into the Hash table and marked with the model number and the base points. The process is repeated for all possible selection of pairs of base points in Step 1.

Recognition:
Given a set of measurement points (e.g. extracted from a visual scene), $\{Q_i = (x_i, y_i)\}$ $(i = 1 \ldots m)$, represent the set in a transformation invariant representation by randomly choosing 2 measurement points and applying the normalization procedure as described above for the model points. For each, normalized measurement point, test whether that point is in the Hash table. If it is in the table add a vote for all models and base points associated with that Hash table entry. The model and base points having the greatest number of votes is determined as the recognized model for the measurement points.

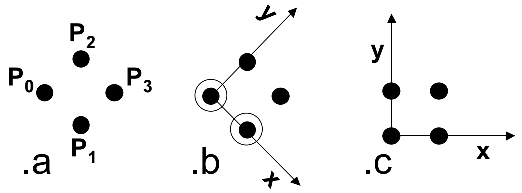

Fig. 1. Normalizing a set of model points. a) Model points b) 2 base points are chosen (marked with a ring) c) model points are transformed so that the base points are mapped to (0,0) and (1,0). The normalized model points: (0,0),(1,0),(0,1),(1,1).

Several issues must be considered during the preprocessing and recognition stages. These include digitization and quantization of the continuous 2D plane into quantization bins for the Hash table. The size of the bins, affect robustness under transformations, noise and occlusion. When noisy measurement points are in the set, the choice of 2 measurement points as the basis for normalization, may be incorrect and will produce incorrect recognition results. Thus the recognition process must, in practice, be applied several times, choosing different measurement base points. When model points are missing (occlusion), not all model points will be recognized and voting results might not be significant. Thresholds on the voting results and testing for compliance of the model to the measurements is required. These and other issues are discussed in [2].

3. GEOMETRIC HASHING AND WATERMARKING

To combine Geometric Hashing with watermarking, we require that the code embedding method affects image values that can be associated with 2D coordinates. Furthermore, in the decoding process, the set of possibly marked values (and their associated coordinates) should be detectable. The embedded code at specific coordinates, serves as the transformed model and the detected set of possibly marked coordinates, serves as the measurement set. There are many watermarking methods that fulfill these requirements (see [3, 4] for a review on watermarking techniques). These vary in the image domain in which the code is actually embedded, such as spatial, DCT, or Wavelet domains. They also vary in the embedding technique such as incremental changes to image values [5], specific settings of values [4] or setting values to comply with inter value relationships [6]. The principle of incorporating Geometric Hashing can be applied to any of these techniques. For demonstration, we implement Geometric Hashing on a specific watermarking technique which embeds a code in the DCT domain [4].

3.1. Watermarking in the DCT domain

A novel algorithm was suggested in [4] in which code embedding is performed in the local frequency domain - the block DCT transform. In this watermarking technique, randomly chosen 8×8 blocks are altered in their low frequency bands by nullifying certain DCT coefficients. Decoding is based on statistical estimations of the appropriate coefficients. The output of the decoder is a decision whether a watermark exists or not in the image. The watermarking method is very robust under attacks in the frequency domain such as JPEG compression and LowPass filtering.

In the encoding process, the image is divided into blocks, of size 8x8. Blocks are chosen randomly, according to a random number generator seed (RNGS). The DCT trans-

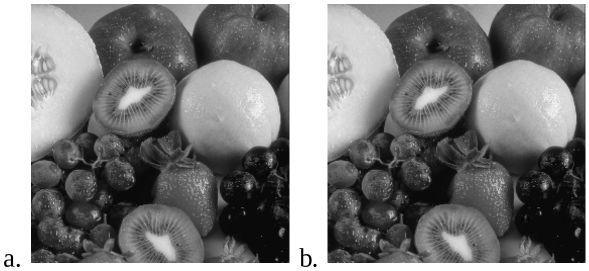

Fig. 2. a) Original Image b) Watermarked image

form is computed for each of the chosen blocks. In each of the chosen blocks, several predefined low frequency coefficients are nullified. Nullification is performed only if the absolute value of the coefficient is smaller than a given visual threshold. A correct threshold ensures that the image will not be visually affected. After changing the appropriate coefficients, the inverse DCT transform is performed and the values scaled to within the image value range so as to obtain the watermarked image.

There is a trade-off, depending on the number of blocks that were chosen, between robustness of the watermark and visibility of the watermark. There is also a trade-off between the number of nullified coefficients in each block and the visibility of the watermark. Increasing the number of the selected blocks and/or the number of nullified coefficients, increases the robustness of the method, but causes the watermark to become more visible. It was found, experimentally, that selecting 5% of the blocks in the image and nullifying 4 low frequency coefficients, gives good results.

Decoding of the watermark is based on statistical evaluation of the DCT coefficients of the watermarked image. There is no need for the original image for decoding. Decoding is performed by computing the positions of the blocks that were modified in the encoding stage, based on the RNGS. In each block the maximum of the absolute values of the coefficients that were modified in the block is determined. The sum of all the maximum values over all the chosen blocks is computed (in the summation, only those values smaller than the given visual threshold are considered). The expected value of the sum in an unattacked watermarked image using the correct RNGS, is approximately zero (slight deviation from zero is expected due to rounding and scaling operations in the encoding stage). However, in a watermarked image, computing the sum value using an incorrect RNGS that chooses many blocks that were not modified in the encoding stage, results in a value which deviates from zero and is expected to be large especially when the nullified coefficients are low frequency coefficients. Thus, obtaining a sum close to zero determines that there is a watermark in the image. In practice the proximity to zero is statistically evaluated as a deviation from the average value expected using incorrect RNGSs. For details on this watermarking technique see [7, 4].

3.2. Geometric Hashing and DCT Watermarking

Instead of randomly choosing image blocks for encoding, the blocks are chosen according to a predefined pattern which has undergone a random transformation. Each chosen block is marked as described above, by nullifying specific DCT coefficients of the block. The predefined pattern of image blocks is considered a model. The model is defined by 2D coordinates representing the positions of the image blocks. Similarly any transformed version of the model is also given as a set of image block coordinates.

Whereas in the original watermarking scheme, the decoder recognizes the embedded blocks using the RNGS, in the Geometric Hashing approach, the decoder, must extract all image blocks which may possibly be marked. These extracted blocks form the measurement set. The decoder traverses all image blocks and tests whether, the specific coefficients were nullified. Statistically based testing is performed to allow robustness under quantization of model coordinates and under attacks (see below). Given the measurement set of image blocks, the Geometric Hashing procedure tests whether the model (watermark pattern of blocks) is found within the measurement set. If it is found, the image is determined to be watermarked.

Issues that must be considered:

- The measurement set of possibly marked image blocks, may consist of many noise blocks which may deteriorate performance. The number of noise blocks is dependent on the specific coefficient values that are nullified in each block and on the statistics and characteristic of the specific image. It was found that nullifying only low frequency DCT coefficients, tends to minimize the noise in the measurement set.
- Since image block coordinates are discrete, the transformed model coordinates are rounded to whole units. This adds digitization noise to the decoding process. Also, certain transformation parameters produce model coordinates that are inaccurate following digitization (e.g. rotation by an angle not a multiple of $90 \deg$). If the robustness of the statistical testing is insufficient, it is possible to limit the set of allowed transformation parameters to be discrete rather than continuous.

4. RESULTS

The approach was tested on images of size 256×256. As described above (Section 3.1) good results are obtained using the DCT watermarking technique by nullifying 4 coefficients in about 5% of the image blocks. Thus, given the size of the image, we use models containing 40 points. Figure 2 Shows an example of a watermarked image.

We used a single model as the watermarking pattern, although the technique based on Geometric Hashing can deal

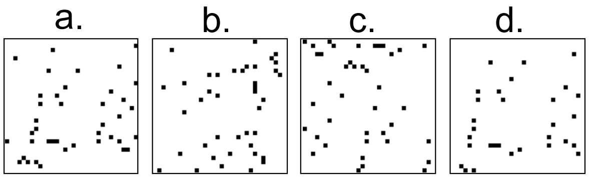

Fig. 3. a) Image block pattern - the model b) Model rotated $90\deg$ c) Cyclically translated model d) Scaled model

with a number of models. We designed the model to contain image block coordinates that are spread over the whole image block domain. In the encoding process, transformation parameters were randomly chosen for translation, rotation and scaling. The watermark was embedded in the original image using the transformed model (image block pattern). Figure 3 shows an example model and several of its transformations. Decoding success rate degraded as the scale parameter increased and as the translation (non cyclic) parameter increase. This is expected since model points exceed image boundaries at large transformations. Figure 4 shows the decrease in performance as the translation parameter increases. These results show that within a large range of transformation parameters, watermark decoding performs well. It is expected that using more robust watermarking techniques, embedding more than one model and including robust estimation and detection techniques, will increase performance even more.

The watermarking technique was tested under various attacks including JPEG compression, lowpass filtering and addition of Salt&Pepper noise. Figure 5 shows the deterioration of the decoding rate as a function of attack strength. Comparing with [7] it can be seen that performance is similar to that obtained under the basic DCT watermarking technique.

Finally, robustness of the technique was tested under geometric attacks, specifically, cropping. Figure 5d shows that the method successfully decodes at above 90% under cropping of the image up to more than 50% of the image.

5. CONCLUSION

The approach presented here, combines Geometric Hashing with watermarking as an example of using Computer Vision tools to enhance performance of watermarking techniques. The method presented can be applied together with a number of different existing watermarking techniques. Performance is dependent on the robustness and characteristic of the watermarking technique chosen. The advantage of using Computer Vision techniques in general, and Geometric Hashing in particular, is that randomization of the watermark can be introduced which does not require maintaining information (such as the random generator seed, or the specific random watermark) for decoding. Additionally, using an appropriate watermarking technique, Computer vision techniques canincrease robustness under geometric attacks.

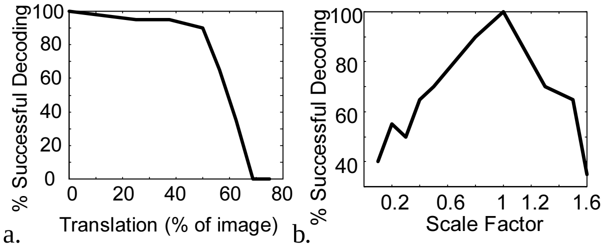

Fig. 4. Percentage of successful watermark decoding as a function of model transformation a) Translation b) Scale

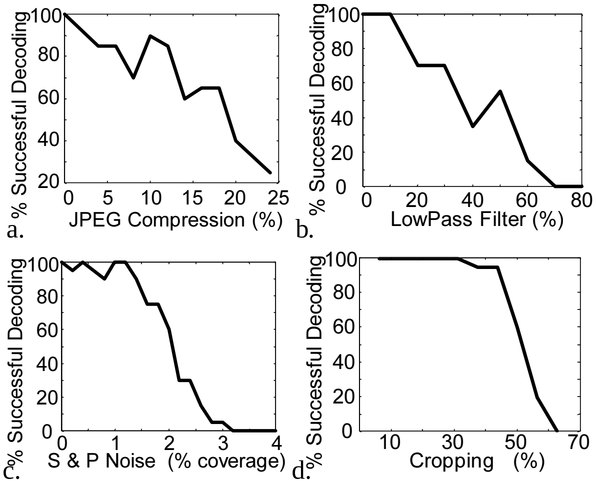

Fig. 5. Percentage of successful watermark decoding as a function of attack strength. a) JPEG compression b) Low-Pass Filtering c) Salt & Pepper noise d) Cropping

6. REFERENCES

[1] H.J. Wolfson and I. Rigoutsos, "Geometric hashing: An overview," *IEEE Comp. Sci. and Eng.*, vol. 4(4), pp. 10–21, 1997.

[2] Y. Lamdan and H.J. Wolfson, "On the error analysis of geometric hashing," in *CVPR91*, 1991, pp. 22–27.

[3] G. Langelaar, I. Setyawan, and R. Lagendijk, "Watermarking digital image and video data a state-of-the-art overview," *IEEE Sig. Proc. Mag.,17(5)*, 20–46,2000.

[4] M. Butman and H. Hel-Or, "Multi-level watermarking with independent decoding," *Submitted to ICIP01.*

[5] W. Bender, D. Gruhl, N. Morimoto, and A. Lu, "Techniques for data hiding," *IBM Systems Journal*, vol. 35, no. 3/4, pp. 313–336, 1996.

[6] A. G. Bors and I. Pitas, "Image watermarking using dct domain constraints," in *IEEE Int. Conf. on Image Processing*, Sept 1996, vol. 3, pp. 231–234.

[7] M. Butman, "Stand-alone multi-level watermarking," M.S. thesis, Bar-Ilan University, Dept of Computer Science, Ramat-Gan, Israel, 2000.