2DKLT-based Color Image Watermarking

Paweł Forczmański

West Pomeranian University of Technology, Szczecin,
Faculty of Computer Science and Information Technology,
Żołnierska Str. 52, 71–210 Szczecin, Poland
pforczmanski@wi.zut.edu.pl

Summary. The paper presents a digital image watermarking algorithm realized by means of two-dimensional Karhunen-Loeve Transform (2DKLT). The information embedding is performed in the two-dimensional spectrum of KLT. Employed two-dimensional approach is superior to standard, one-dimensional KLT, since it represents images respecting their spatial properties, resulting in a lower noise and better adaptation to the image characteristics. The principles of 2DKLT are presented as well as sample implementations and experiments, which were performed on benchmark images. We propose a measure to evaluate the quality and robustness of the watermarking process. Finally, we present a set of experiments related to the color-space, embedding variants and their parameters.The results show that the 2DKLT employed in the above application gives obvious advantages in comparison to certain standard algorithms, such as DCT, FFT and wavelets.

1 Introduction

The problem of hiding certain content into some other data is called *steganography* and its modern applications include embedding watermarks and copyright information into music, images and videos (so called digital rights management – DRM), hiding secret messages in order to transfer them over the Internet, protect data against alteration, etc.[1, 2]. Many algorithms have been developed so far, however, not all all of them are usable and robust enough in practical applications. The main issue is if the carrier (cover object) is altered, the watermark should be preserved. This problem often exists in cases when data have to be protected against unauthorized usage or modification. In literature there are many general approaches aimed at still image steganography and watermarking [1, 2, 3, 4, 5, 6]. Almost all such methods work in either space domain [4] or in certain spectrum domain (FFT, DFT, DCT, Wavelets [3]).

One of the most popular yet not very effective method is a least significant bit (LSB) insertion [2], which alters the least important part of the cover object. The main problem with LSB and other methods is the low robustness to typical visual manipulations, e.g. changes of intensity, contrast and focus as well as lossy re-compression and framing. On the other hand, typical transformation-based methods introduce high level of distortions into the cover image and are sensitive to lossy compression. It means we need a compromise between high robustness to attacks and low visibility.

The natural solution to above stated problems is the usage of transformation that adapts to the characteristics of input data. Many transformation methods have been proposed in the literature, most of them based on operations such as Karhunen-Loeve Transform (KLT), Discrete Cosine Transform (DCT), Discrete Fourier Transform (DFT), Linear Discriminant Analysis (LDA), Discrete Wavelet Transform (DWT), etc. The literature survey shows that in case of images presenting real scenes one of the most promising is KLT since it uses basis functions optimally adjusted to actual data characteristic. The KLT also known as Principal Components Analysis (PCA) is a technique widely used to reduce multidimensional data sets to lower dimensions for analysis, compression or classification [7]. The PCA/KLT involves the computation of eigenvalue decomposition or singular value decomposition of a data set, usually after mean centering [8]. However, it should be noted that using PCA/KLT for tasks such as pattern recognition or image processing can be challenging because it treats data as one-dimensional, when in fact they are two-dimensional. That is why almost all developed algorithms employ some sort of dimensionality pre-reduction discarding, in many cases, the spatial relations between pixels. One of the possible solutions is using two-dimensional transformation based on PCA (and KLT). The first algorithm from this group was presented in [9], where a novel, two-dimensional version of KLT for face recognition task was developed. An extension of this method (known as PCArc - for row and column representation) was presented in [10, 11]. Many current publications show that two-dimensional KLT (2DKLT) can be directly applied for high-dimensional data [10, 11] because it does not require any preliminary processing or additional reduction of dimensionality of input images. There are several works related to watermarking in the KLT domain, however they deal with one-dimensional representation [6, 12, 13] or process gray-scale images only [6].

In the further parts of the paper we show the main principles of 2DKLT together with the discussed application area - information embedding (watermarking). The experiments performed on the standard benchmark images gave satisfactory results, comparable to the other known approaches [1, 6, 12, 13].

2 Algorithm description

2.1 Processing scheme

Developed algorithm uses a two-dimensional version of Karhunen-Loeve Transform and redundant pattern encoding [4]. It consists of the following steps: 1. preparation of data – input image is being decomposed into blocks of equal size, a message is expanded into bit-wise sequence, 2. calculation of two-dimensional basis functions by means of PCA row/column, 3. transformation of blocks (2DKLT), 4. inserting a binary representation of a message according to the specified key, 5. inverse 2DKLT transformation, 6. joining blocks into a single image. The reverse algorithm used for extracting the message is based on the same general rules. In the beginning, the carrier image is being decomposed into blocks, which are transformed by means of known eigenvectors (calculated previously). Then the message fragments are being extracted from specific blocks' elements according to the known key. Finally, the bit-wise extended message is compacted into graphical form for a convenient readout.

In order to unnoticeably embed a message into the cover image, it should be modified in a way that it will contain small differences similar to the original object. That is why we operate in KLT spectrum, in order to spread the change in several pixels. Thus, the watermark is being expanded into a longer sequence of small values e.g. binary ones. Then we have to generate a key which will determine where in the transformed blocks the elements of expanded watermark have to be placed. The key is a sequence of offsets from the origin of each block. It is important to place the watermark into the "middle-frequency" part of each block as a compromise between output image quality and robustness to intentional manipulations (see first matrix in the Fig. 1). This rule is especially significant in a case of images containing large number of small details, which helps keeping all the changes imperceptible. The only noticeable difference can be observed in the areas of uniform color.

2.2 2DKLT principles

Let us assume that the image I being processed is stored in a structure of $P \times Q \times R$ elements, containing R channels of $P \times Q$ matrices. We decompose each r-th channel into a set of K subimages (blocks) X_k with equal, constant dimensions $M \times N$, where $(M << P, N << Q)$. These blocks do not have common parts and do not overlap each other. They cover the whole image area. This approach is similar to JPEG/JFIF, where $M = N = 8$, which is a compromise between compression ratio and performance [14]. In a case being discussed here it is possible to use block of any size, even not square [11]. The algorithm uses PCA for row/column representation (PCArc) in order to calculate the eigenvectors and KLT in order to transform them [10]. In the first step, for all blocks in the dataset we calculate mean block $\bar{X}$ and remove it from each individual block $\hat{X}_k = X_k - \bar{X}$.

In the next step we calculate two covariance matrices for both row and column representations of input blocks, which correspond to the variation of blocks in the set [9]: $\Sigma^{(r)} = \sum_k \hat{X}_k \hat{X}_k^T$ and $\Sigma^{(c)} = \sum_k \hat{X}_k^T \hat{X}_k$.

After solving the following equations $\Lambda^{(r)} = (V^{(r)})^T \Sigma^{(r)} V^{(r)}$ and $\Lambda^{(c)} = (V^{(c)})^T \Sigma^{(c)} V^{(c)}$ we get eigenvectors V and eigenvalues Λ matrices which will further serve as the transformation basis. Usually, this step requires a computer-based algorithm for computing eigenvectors and eigenvalues. Finally, for each block X_k in order to get its transformation Y_k the following operation is performed [10]: $Y_k = V^{(r)}(X_k - \bar{X})V^{(c)}$.

It should be noted that we do not perform any kind of dimensionality reduction here, hence the KLT coefficients preserve total information of blocks X_k. The transformation on blocks is the main operation for further steps described in the following sections.

2.3 Watermark embedding/recovering

The maximal size of a watermark is directly linked to the number of blocks (K), which are calculated from the proportion of image dimensions and block size. If we assume to have binary watermark, then one bit is embedded in one image subblock. So the size of the watermark is equal to the number of blocks. If the watermark is smaller than the number of blocks then it can be inserted in the cover image repeatedly. This approach is called redundant pattern encoding [4] and increases the robustness of the watermark to the intentional manipulations like compressing of filtering. There is also a possibility of inserting more than one bit in each block, however it decreases the robustness of the method.

The recovery algorithm, in a comparison to the embedding phase, is performed in the following order. In the first step we decompose the cover image into sub-blocks, which are further transformed using known eigenvectors (2DKLT) and finally, the appropriate part of a watermark is being extracted from KLT spectrum of each block. It is important to know the right eigenvectors as well as the key used to extract the exact watermark. If we do not know the key we still are able to extract the watermark (with the help of eigenvectors calculated straight form the cover image), but its exact content will be hard to guess.

As it was already written, there are several variants of information embedding. We investigated three variants, namely:

1. **Direct Bit Once** - one bit put in each block according to the key;
2. **Direct Bit Twice** - one bit put twice in each block according to the key;
3. **Direct Two Bits** - two bits put in each block according to the key;

The areas of alternation in each block are presented in Fig.1. The first block presents an area of middle frequencies which are especially suitable for alternation, since they are responsible for least noticeable changes in the image. The second block in Fig.1 represents a variant of the embedding called Direct Bit Once, which changes single element in the chosen area according to the key. The change is done by substitution of original element $X(i,j)$ with a value equal to:

$$\tilde{X}(i,j) = \begin{cases} SX(i,j) & \text{if} \quad X(i,j) = 1 \\ -SX(i,j) & \text{if} \quad X(i,j) = 0 \end{cases} \tag{1}$$

The extraction of the watermark is done by checking the sign of the element in the KLT spectrum. If the extracted element is more (or equal) than zero, then the extracted bit of watermark is set to one. Otherwise, it is set to zero.

The third block in Fig.1 represents a variant of the embedding called Direct Bit Twice, that changes two elements in the chosen area according to two independent keys. It was developed in order to increase the robustness of the embedding. The change is done by substitution of original elements $X(i_1, j_1)$ and $X(i_2, j_2)$ with a value calculated according to the Eq.1. The extraction of the watermark is done in the following manner. Both elements are extracted (according to the keys) and then added. If the sum is more (or equal) than zero, than the extracted bit is set to one. Otherwise it is set to zero.

The method called Direct Two Bits is similar to the above one. The main difference is the fact of coding two independent bits of the watermark in each block. Instead of doubling the embedded information, in each block, we put there two successive bits of the watermark. The extraction algorithm is also similar.

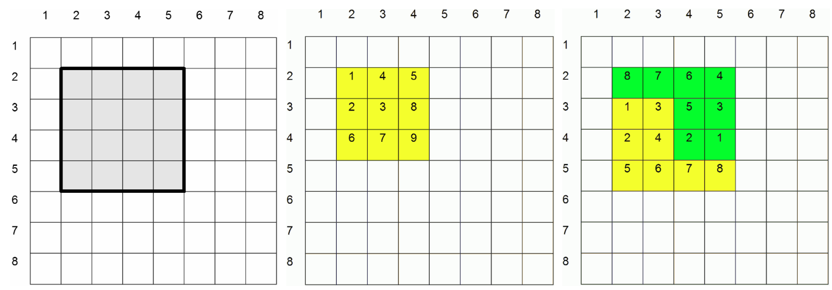

Fig. 1. KLT spectrum area being altered (left), Subarea chosen in Direct Bit Once (middle) and in Direct Bit Twice (right), respectively

2.4 Watermarking quality/robustness evaluation

Three are two main measures of evaluating the quality and robustness of watermarking. They include Peak Signal-to-Noise Ratio and Cross-Correlation. Both are used in an independent manner [6, 12, 13] and can not give an unequivocal answer. In order to evaluate the influence of parameters of presented process on image quality as well as the robustness of the watermarking procedure simultaneously we introduce the combined metrics. It employs cross-correlation coefficient calculated for original W and extracted watermark image $\tilde{W}$ and the Peak Signal-to-Noise Ratio of modified image $\tilde{I}$ vs. original one I:

$$D = PSNR(I, \tilde{I}) * log_{10}(Corr(W, \tilde{W})). \quad (2)$$

Such measure is capable of capturing both characteristics of watermarking process and help finding its practically useful parameters. In future, some more advanced image quality metrics (like [15]) may be used in order to reproduce the subjective characteristic of quality estimation.

3 Experiments

Developed algorithm and its prototype implementation have been investigated on sample color images of different origin and characteristic (see Fig.2). To prove the robustness of the algorithm, several tests, according to the *de facto* standards, have been performed [1, 2]. They included 13 operations: geometrical transforms (scaling and rotation), lossy JPEG compression, noising (Gaussian, Poisson), changing the brightness and contrast of an image and finally convolution filtering with different masks.

Fig. 2. Images used for experiments: *Lena*, *Griffin*, *Ladybug* and the watermark

We investigated two aspects of the method:

1. selection of color space (single color channel) in which the watermark is embedded: RGB, HSB, YCbCr;
2. selection of watermark power $S \in 10, 25, 50, 75$;

The combined results of the experiments (mean results for all 13 attack types) are shown in Fig.3. The lower row of the figure represents the normalized quality values for 4 different watermarking powers (S). As it can be seen, the best results were obtained for Y, G and V channels. However, when we consider the robustness to attacks, the results are quite different. The first row of the figure presents the combined metric D in dB. The higher the value, the better quality and the higher robustness to attacks, thus the best results give *Green* and *Saturation* channels (depending on the characteristics of images).

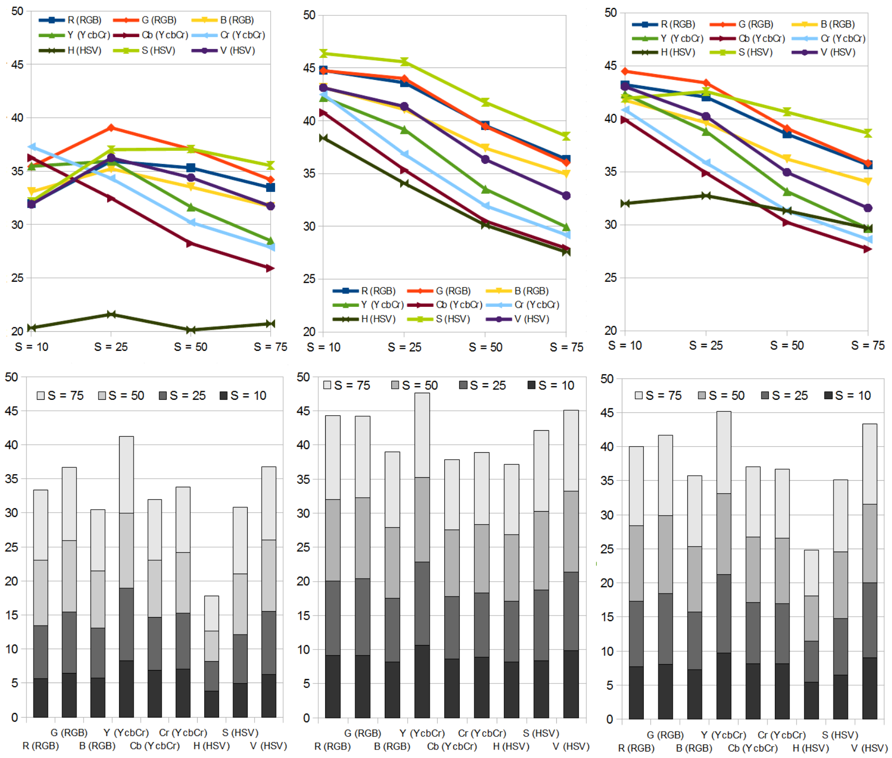

Fig. 3. The combined quality/robustness metric (upper row) and detailed color-space comparision in terms of joint PSNR (lower row) for *Lena*, *Griffin*, *Ladybug*, respectively

4 Summary

Algorithm presented here and its application area showed that the 2DKLT method can be a valuable basis for efficient color image watermarking. This approach makes it possible to hide a watermark in the so called cover image with high resistance to image distortions by using optimal representation in the eigenvectors space. During the experiments high robustness of 2DKLT-based algorithm to typical steganographic attacks has been proved. The quality and strength is similar to the ones presented in [6, 12, 13], however the proposed approach is more flexible in the aspect of color-space and quality settings. The bottom line is that during investigations the hidden message was not detected by one of the most popular steganalysis tools - stegdetect v0.4 [2], which makes the proposed method highly promising.

Acknowledgement

The experimental results presented in this paper are a part of Bachelor's Thesis prepared under a supervision of the author by Pawel Marczeski at The Faculty of Computer Science and Information Technology, West Pomeranian University of Technology, Szczecin in 2012.

References

1. Podilchuk, C.I., Delp, E.J.: Digital Watermarking Algorithms and Applications. Proc. IEEE Signal Processing Magazine, 18(4), pp. 33-46 (2001)
2. Provos, N., Honetman P.: Hide and Seek: An Introduction to Stegangography, IEEE Security & Privacy Magazine, May/June, (2003)
3. Kaarna, A., Toivanen P.: Digital Watermarking of Spectral Images in PCA/Wavelet Transform Domain, Proc. IGARSS, 6: pp. 3564–3567 (2003)
4. Cheung, W. N., Digital Image Watermarking in Spatial and Transform Domains, Proc. TENCON, 3: pp. 374–378 (2000)
5. Artz, D.: Digital Steganography, Hiding Data within Data, Proc. IEEE Internet Computing, 5(3): pp. 75–80 (2001)
6. Dafas, P., Stathaki, T., Digital image watermarking using block-based Karhunen-Loeve transform, Image and Signal Processing and Analysis, 2003. ISPA 2003. Proceedings of the 3rd International Symposium on, vol.2, pp.1072–1075 Vol.2 (2003)
7. Jolliffe, I. T.: Principal Component Analysis. Springer Verlag, NY (1986)
8. Fukunaga, K.: Introduction to Statistical Pattern Recognition, second edition, New York: Academic Press, (1990)
9. Tsapatsoulis, N., Alexopoulos V., Kollias S.: A Vector Based Approximation of KLT and Its Application to Face Recognition, Proc. of The IX European Signal Processing Conference EUSIPCO-98, Island of Rhodes (1998)
10. Kukharev, G., Forczmański P.: Data Dimensionality Reduction for Face Recognition, Machine Graphics and Vision. Vol. 13, No. 1/2 (2004)
11. Kukharev, G., Forczmański P.: Facial images dimensionality reduction and recognition by means of 2DKLT. Machine Graphics and Vision. vol. 16, no. 3/4, pp. 401–425 (2007)
12. Stanescu, D., Stratulat, M., Ciubotaru, B., Chiciudean, D., Cioarga, R., Borca, D.: Digital Watermarking using Karhunen-Loeve transform, Applied Computational Intelligence and Informatics. SACI '07. 4th International Symposium on. pp.187–190, (2007)
13. Botta, M., Cavagnino, D., Pomponiu, V.: KL-F: Karhunen-Loeve based fragile watermarking, Network and System Security (NSS), 2011 5th International Conference on, pp.65–72, 6-8 Sept. (2011)
14. Wallace, G. K.: The JPEG Still Picture Compression Standard, Communications of the ACM, April (vol. 34, no. 4) (1991)
15. Okarma, K.: Combined image similarity index, Optical Review, vol. 19, issue 5, pp. 349–354 (2012)