APPLICATIONS OF INDEPENDENT COMPONENT ANALYSIS TO IMAGE CODING

Zhenhuan Sun, Ziyi Xu

z98sun@uwaterloo.ca - z297xu@uwaterloo.ca
University of Waterloo
Department of Electrical and Computer Engineering
200 University Ave W, Waterloo, ON N2L 3G1

ABSTRACT

In communication systems, fast and secured transmissions have always been the ultimate goal that people want to achieve. The speed, accuracy and security of image transmissions are even more in demands for any existing communication systems. Because of the enormous demand, various image coding methods including image compression and decompression, image encryption and decryption have been introduced in the past 20 years. Among those coding or decoding techniques, independent component analysis (ICA) has caught our eyes for its flexibility and diversity. In this article, we will briefly introduce the ICA algorithm and its functions, then focus on explaining and discussing its role and impact on different image coding methods, we then conclude this article by sharing our critical insights about using ICA in the field of image coding.

Keywords: Independent Component Analysis (ICA), image compression, image decompression, image encryption, image decryption.

1. INTRODUCTION

Due to the evolution of digital image processing techniques, digital images now have much higher resolutions and contain more visual information than before. In exchange, digital images now require more bits to be stored and transmitted, thus require more sophisticated methods to manipulate the visual data stored in them. Therefore, in order to make the image transmission fast and secured, research on finding effective and secure image coding methods to represent image data has become essential. In the context of image coding, there are two types of coding method, lossless and lossy. The lossless coding methods represent image data with the minimum possible number of bits, and the receiver will be able to reconstruct the original image from this received minimum bits representation without any error theoretically. The lossy image coding, on the other hand, allow errors to occur, however these errors are constrained within a tolerable range. Although lossy image coding methods can not reconstruct original image perfectly, lossy image coding has more applications in image coding field because in many cases lossy image coding can achieve a far better compression ratio than lossless image coding. Conventionally, Discrete Fourier transform (DFT) and Discrete Cosine Transform (DCT) are widely used as lossy image coding methods [1], but in this article unconventional lossy image coding methods which are based on independent component analysis (ICA) [2] are considered.

Independent component analysis (ICA) was originally introduced as a statistical tool to address the blind source separation (BSS) [3] problems, but recently independent component analysis (ICA) has gained a lot of attention due to its great potential in many other applications such as face recognition, analysis of electroencephalography (EEG) signal [4], audio signal processing, and data mining [5]. ICA models the observed data x as a linear combination of n independent sources:

$$x = As$$

Where $s \in \mathbf{R}^n$ is a vector of independent sources and A is an unknown square matrix called the mixing matrix. The goal of ICA is to recover the independent sources s by finding the unmixing matrix which can be defined as $W = A^{-1}$ and perform matrix multiplication between x and W:

$$s = Wx$$

However, the recovered unmixing matrix is not always perfect, there are some ambiguities to it, for example, the permutation of original sources is ambiguous as well as the scaling of W. Fortunately, these ambiguities do not matter for most applications, however, one point worth noting is that we need to make sure sources s are non-Gaussian since Gaussian density is rotationally symmetric which means there is no way to determine whether the sources were mixed with A or A^T.

To apply ICA to two dimensional data such as images, block-based ICA [6] approaches are usually used to avoid having a huge vector of x and s. In practice, in order to ensure fast convergence, FastICA algorithm [7] are used to compute the unmixing matrix W.

2. FROM ICA TO IMAGE CODING

Independent Component Analysis (ICA) has potential in many aspects of image coding field such as image compression and image encryption. Its ability to separate combined signal to individual independent signal sources give researchers tremendous amount of freedom to apply this algorithm at different stages of the image coding system. In this section we elaborate on several aspects of image coding in which independent component analysis (ICA) can make significant impact.

2.1. Image Compression and Decompression

The purpose of image compression is to find a new representation of image data using fewer bits from which original image data can be reconstructed with or without error. The result of Independent Component Analysis (ICA) can be lossy or lossless depends on the number of independent components, if all independent components are used then the reconstruction will be lossless, however there is no compression at all, thus ICA is often used as a lossy image compression method, it can achieve reasonable and controllable compression ratio with very fast compression speed due to the present of FastICA algorithm [7].

2.1.1. Block-Based ICA on Color Image Overview

In block-based ICA [6], the color image of size $M \times M$ is first separated into its three color components R, G and B, each color component image is further divided into $N \times N$ subblocks where each subblock of the image is denoted as $x^{(i)}$ and it represents a linear combination of n independent image subblock sources. We then feed $x^{(i)}$ to FastICA algorithm as training examples to compute the unmixing matrix $W^{(i)}$ for each image subblock. From the computed unmixing matrix a vector of n independent sources $s^{(i)}$ associated with the current image subblock is obtained by performing matrix multiplication: $s^{(i)} = W^{(i)}x^{(i)}$. Unmixing matrix $W^{(i)}$ is then reversed to obtain the mixing matrix $A^{(i)}$. As a consequence, the original image is now decomposed to two parts which are mixing matrix:

$$A = \{A^{(i)}; i = 1, 2, ..., m\}$$

and independent sources:

$$s = \{s^{(i)}; i = 1, 2, ..., m\}$$

Where $m = \frac{M^2}{N^2}$. These two parts are the results of ICA compression and the data will be transmitted over the channel. Prior to transmission, arithmetic coding will be applied to both mixing matrix and independent sources to convert them into binary format thus encoding the messages for security purpose. The receiver will be able to recover the transmitted information by doing arithmetic decoding and reconstruct the original image by multiplying A and s.

2.1.2. Block-Based ICA Compression Algorithm

1. Decompose original image to RGB components.
2. Choose the size of subblock.
3. Divide each RGB component to $m = \frac{M^2}{N^2}$ subblocks.
4. Choose the number of independent components.
5. Use subblocks as training examples for FastICA.
6. Obtain mixing matrix A and independent sources s.
7. Apply arithmetic encoding.
8. Transmit the data.

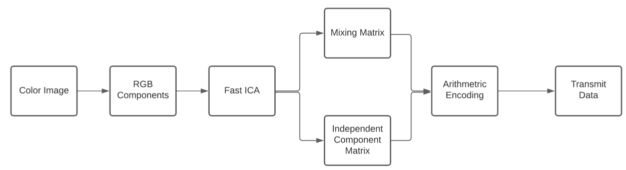

Fig. 1. Block-Based ICA Compression Flow Diagram [6]

2.1.3. Block-Based ICA Decompression Algorithm

1. Apply arithmetic decoding to received data.
2. Obtain mixing matrix A and independent components s.
3. Recover the original data x by multiplying A and s.
4. Repeat step 3 for all color components.
5. Reconstruct the original image.

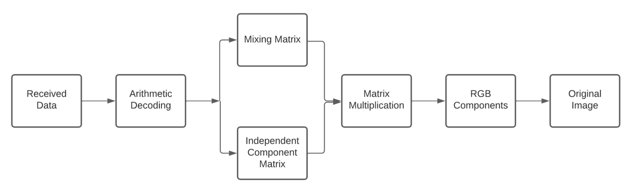

Fig. 2. Block-Based ICA Decompression Flow Diagram [6]

2.1.4. Simulation Result

The performance of Block-Based ICA Compression method is tested using Lenna image. To determine the optimal size of subblock which will result in the best overall quality and compression ratio, several different subblock size including 4×4, 8×8, 16×16, 32×32 are used for testing. It can be observed that subblocks with size 8×8 can give us better compression ratio and better image quality after reconstruction than any other subblock sizes [6]. Apart from determining the size of each subblock, a more important task is to determine the number of independent components that would be sufficient to represent the original image and allow us to use as few bits as possible.

The number of independent components used for the compression is highly application-dependent, however we can visualize the effect of compression with different numbers of independent components by doing the following steps. Since the subblock size is 8×8, we assume there are 8 mixed signal which each one of them is composed by 8 independent sources. These 8 mixed signals are 8 columns of the block and they are denoted as $x^{(i)}$ where $i = 1, 2, ..., 8$. After $x^{(i)}$'s are fed to FastICA algorithm, mixing matrix and 8 independent sources can be obtained, the image is then reconstructed with different number of independent components. The number of independent components that is sufficient to represent the original image can be chosen by comparisons between the quality of reconstructed image and original image by means of various quality metrics and compression ratio, different applications may require different number of independent components to implement different functions.

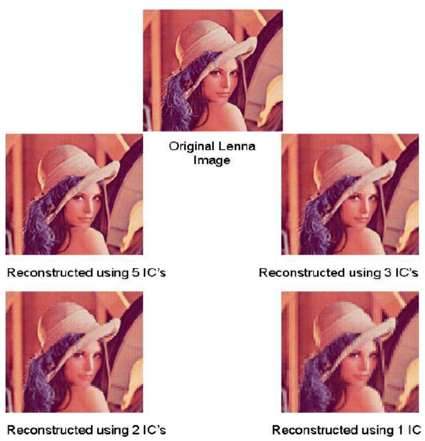

Fig. 3. Comparison between reconstructed image and original image using different quality metrics [6]

Number of Independent Components	Mean Square Error (MSE)	Signal to Noise Ratio (SNR)	Peak Signal to Noise Ratio (PSNR)	Compression Ratio
1	288.9443	18.189	23.0682	88.5284
2	182.3956	20.189	25.0662	47.2906
3	140.9408	21.3068	26.1859	32.9019
5	85.5514	23.4749	28.354	20.4505

Fig. 4. Comparison between reconstructed image and original image using different quality metrics [6]

As shown in figure 4, the degree of compression and the quality of reconstructed image are completely controlled by the number of independent components, if the quality of reconstructed image is the major factor that application need to consider then compress with more independent component will certainly result in more accurate reconstruction, however the compression ratio may be reduced. On the other hand if we need to minimize the number of bits that represent the image then use less independent components lead us to higher compression ratio but less accurate reconstruction.

2.2. Image Encryption and Decryption

In the communication system, besides the fast transmission of compression, information security is another important factor to be considered. In order to achieve secure information transmission, a feasible and effective algorithm to encrypt the image from the transmitter and decrypt the image from the receiver is very important. As a method to separate several independent source signals from some unknown source signals and transmission channels based on statistic independent principle [8], Independent Component Analysis (ICA) provides a possible way to encrypt and decrypt the image. In this paper, a new ICA-based image encryption and compression method is considered[9].

2.2.1. ICA-Based Image Encryption and Decryption Overview

In last few decades, independent component analysis algorithms have been proposed and developed from blind source separation (BSS) problem. These ICA algorithms aims at searching for those unknown source components in the mixed signals that are statistically as independent from each other as possible[10]. The general purpose of the new ICA-based image encryption and compression method comes from the ability of independent components splitting from the mixtures, which means in encryption, a mixed image can be designed and generated based on the original image, and in the decryption, the original image can be recovered by the means of ICA algorithms.

2.2.2. ICA-Based Image Encryption Method

In the encryption stage, one simple way to make a mixed image is to add independent images to the original image. Before that, for better information safety performance however, the new method divides the original images into several small subblocks, and each of which is applied to two-dimensional Discrete Cosine Transform (DCT). The DCT has an excellent energy-compaction property, which concentrates the major amount of signal energy in the left-corner of the frequency/energy spectrum[11]. For natural images, energy concentrates at low frequencies, containing the important information of the images. Therefore, using lowpass filters to throw high frequencies away can reduce the amount of information to be transmitted without losing too much image quality.

Because we need to use ICA to retrieve these DCT components in receiver, it is necessary to make all DCT components independent to each other before transmission. For natural images, there are high dependency between two neighboring pixels, and thus between two subblocks. In order to eliminate such high dependency as possible, the rotation operation that randomly rotate each DCT subblocks will be used and rotation information (such as rotation degree) will be stored and added into an encryption key, which will also be sent to receiver during transmission.

The last step in encryption is to make a mixed image. We can add our DCT components with some random images. Therefore, the image is totally encrypted, and it is unrecognizable for non-authorized people without ICA algorithms and an encryption key.

2.2.3. *ICA-Based Image Decryption Method*

In the decryption stage, the authorized receiver will receive a mixture and an encryption key. The receiver will use ICA algorithm (i.e., FastICA algorithm) to demix the mixtures of the DCT components. The result of demixing will get DCT components from other random images. In the following step, the encryption key will be used to rotate these DCT components back to the normal positions, in which low frequencies are retained in the left-corner of the subblocks. Finally, we apply inverse discrete cosine transform (IDCT) to the normal DCT components and thus gain the estimated original images from the encrypted mixtures.

2.2.4. *ICA-Based Image Encryption Algorithm*

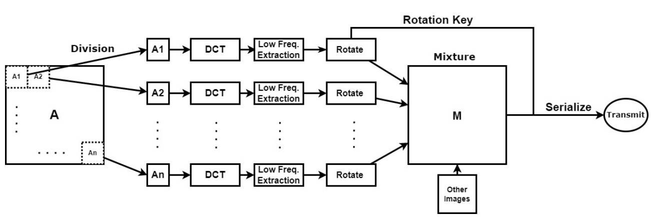

Fig. 5. ICA-based Image Encryption Flow Diagram

In the algorithm, only grayscale images are used as original images, however color images can also be applied in the same way. For the rigor and simplicity of the paper, we only consider grayscale images as our inputs.
1. Choose the size of subblocks.
2. Divide the original image into different subblocks based on the subblock size.
3. Use Discrete Cosine Transform (DCT) to obtain DCT components of each block.
4. Apply lowpass filter to drop higher frequency components for the fast transmission.
5. Rotate each DCT component randomly to eliminate dependency of each subblock.
6. Mix DCT components with some random images.
7. Apply arithmetic encoding to the image mixtures and encrypted rotation key.
8. Transmit the data to receivers.

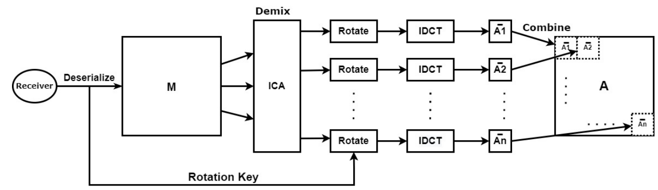

Fig. 6. ICA-based Image Decryption Flow Diagram

2.2.5. *ICA-Based Image Decryption Algorithm*

1. Apply arithmetic decoding to the data from transmitter.
2. Use FastICA to demix the mixtures.
3. Rotate the DCT components back to normal positions with received encrypted rotation key.
4. Use Inverse Discrete Cosine Transform (IDCT) to restore the original images.

2.2.6. *Simulation Result*

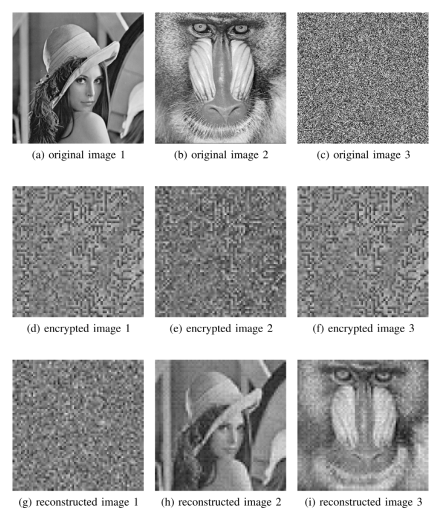

Fig. 7. Example of the simulation; 1st row: original images; 2nd row: IDCT of mixed DCT components; 3rd row: Restored images [9]

3. CRITICAL THINKING

ICA-based image compression methods are often compared with another statistical image compression method called Principal Component Analysis (PCA) [12]. In many cases the performance of ICA-based compression methods on color images exceeds the performance of PCA-based compression methods on both compression ratio and quality parameters [6], also the speed advantage of ICA-based image compression method is sometimes crucial when image transmission time is the subject that need to be minimized.

4. CONCLUSION

...

5. REFERENCES

[1] K Ramamohan Rao and Ping Yip, *Discrete cosine transform: algorithms, advantages, applications*, Academic press, 2014.

[2] Pierre Comon, "Independent component analysis, a new concept?," *Signal processing*, vol. 36, no. 3, pp. 287–314, 1994.

[3] Shun-ichi Amari, Andrzej Cichocki, and Howard Yang, "A new learning algorithm for blind signal separation," *Advances in neural information processing systems*, vol. 8, 1995.

[4] Julie Onton, Marissa Westerfield, Jeanne Townsend, and Scott Makeig, "Imaging human eeg dynamics using independent component analysis," *Neuroscience & biobehavioral reviews*, vol. 30, no. 6, pp. 808–822, 2006.

[5] Jayanta Basak, Anant Sudarshan, Deepak Trivedi, and MS Santhanam, "Weather data mining using independent component analysis," *The Journal of Machine Learning Research*, vol. 5, pp. 239–253, 2004.

[6] V Singh, SK Yadav, NK Singh, and PK Kalra, "Color image compression using block-based independent component analysis," in *2007 5th International Symposium on Image and Signal Processing and Analysis*. IEEE, 2007, pp. 288–292.

[7] Aapo Hyvarinen, "Fast and robust fixed-point algorithms for independent component analysis," *IEEE transactions on Neural Networks*, vol. 10, no. 3, pp. 626–634, 1999.

[8] Zheng-Hong Huang, "Image denoising by independent component analysis based on dyadic wavelet transform," in *2009 International Conference on Wavelet Analysis and Pattern Recognition*, 2009, pp. 351–354.

[9] Masanori Ito, Noboru Ohnishi, Ayman Alfalou, and Ali Mansour, "New image encryption and compression method based on independent component analysis," in *2008 3rd International Conference on Information and Communication Technologies: From Theory to Applications*, 2008, pp. 1–6.

[10] Chao Ma and Lian-min Wang, "Review of ica based fixed-point algorithm for blind separation of mixed images," in *2010 4th International Conference on Bioinformatics and Biomedical Engineering*, 2010, pp. 1–3.

[11] K V Suma, K Indira, and Bheemsain Rao, "Classification of nailfold capillary images using wavelet and discrete cosine transform," in *International Conference on Circuits, Communication, Control and Computing*, 2014, pp. 105–108.

[12] Clifford Clausen and Harry Wechsler, "Color image compression using pca and backpropagation learning," *pattern recognition*, vol. 33, no. 9, pp. 1555–1560, 2000.