Optics and Lasers in Engineering 101 (2018) 113–121

Contents lists available at ScienceDirect

Optics and Lasers in Engineering

journal homepage: www.elsevier.com/locate/optlaseng

Multiple image encryption scheme based on pixel exchange operation and vector decomposition

Y. Xiong, C. Quan*, C.J. Tay

Department of Mechanical Engineering, National University of Singapore, 9 Engineering Drive 1, Singapore, 117576, Singapore

ARTICLE INFO

Keywords:
Multiple image encryption
Pixel exchange
Vector decomposition

ABSTRACT

We propose a new multiple image encryption scheme based on a pixel exchange operation and a basic vector decomposition in Fourier domain. In this algorithm, original images are imported via a pixel exchange operator, from which scrambled images and pixel position matrices are obtained. Scrambled images encrypted into phase information are imported using the proposed algorithm and phase keys are obtained from the difference between scrambled images and synthesized vectors in a charge-coupled device (CCD) plane. The final synthesized vector is used as an input in a random phase encoding (DRPE) scheme. In the proposed encryption scheme, pixel position matrices and phase keys serve as additional private keys to enhance the security of the cryptosystem which is based on a 4-*f* system. Numerical simulations are presented to demonstrate the feasibility and robustness of the proposed encryption scheme.

1. Introduction

With the rapid development of modern communication, information security has drawn increasing attention. In recent years, optical techniques due to their inherent characteristics of high-speed and multidimensional signal processing capabilities [1,2] have been widely used for image encryption. Since DRPE was proposed by Refregier and Javidi [3], various optical image encryption structures and algorithms have been studied [4–10]. Several methods, such as fractional Fourier [11,12], fractional Mellin [13], Fresnel [14,15] and gyrator transforms [16,17] for converting pixel value, have been utilized in image encoding schemes.

Recently, multiple image encryption (MIE) algorithms have been studied owing to improvement in encryption capacity and efficient transmission and storage of ciphertext. Since a MIE scheme with wavelength and position multiplexing was proposed by Situ and Zhang [18,19], other architectures and schemes have followed [20–26]. One vital issue in the MIE system is cross-talk noise which degrades the quality of a decrypted image and limits the number of images to be encrypted. In this regards, MIE schemes based on phase retrieval algorithm [27–29] and interference [30–33] have been proposed to decrease the effects of cross-talk.

A pixel scrambling operation is an encryption method used to enhance the security of image information by removing silhouette problem which occurs in a conventional interference-based cryptosystem [34–37].

In this paper, a basic vector decomposition in a decryption process is employed to separate original images from an interfused image. We also present a new pixel exchange operator which is regarded as a scrambling method to alter pixel sequence in an encryption scheme. Original images are imported into a pixel exchange operator, and the pixels of the original images are exchanged with other images using a preset rule.

2. Pixel exchange operator

An illustration of a pixel exchange process based on Bubble sorting is shown in Fig. 1 where functions I_1, $I_2 \cdots I_k$ represent original images to be encrypted and variables m and n are indices of the images. Functions M_1, $M_2 \cdots M_k$ represent intermediate matrices, which record the intensity of original images and identify the pixels in each original image simultaneously. *Temp* represents an intermediate variable. Functions P_1, $P_2 \cdots P_k$ represent pixel position matrices. In the algorithm, we first compare the values in each pair of pixels at position (m, n) in adjacent images M'_j and M'_{j-1}. If $M'_j(m,n,1) < M'_{j-1}(m,n,1)$, the pixels are exchanged with each other in the matrices. If $M'_j(m,n,1) \geq M'_{j-1}(m,n,1)$, the pixels remain as they are in the original matrices. The above procedure is repeated until no exchanges are needed in the original images. When all the pixel positions have been dealt with in the above procedure, scrambled image I'_k and pixel position matrix P_k are considered as the outputs of the operation.

The inverse process of the pixel exchanging operation is implemented as shown in Fig. 2. In the algorithm, every matrix $P_i (i = 1, 2 \cdot \cdot \cdot k)$ at position (m, n) is scanned. If $P_i(m,n) = j$, the pixel value of $I_j(m,$

* Corresponding author.
E-mail address: mpeqcg@nus.edu.sg (C. Quan).

https://doi.org/10.1016/j.optlaseng.2017.10.010
Received 23 June 2017; Received in revised form 15 September 2017; Accepted 14 October 2017
0143-8166/

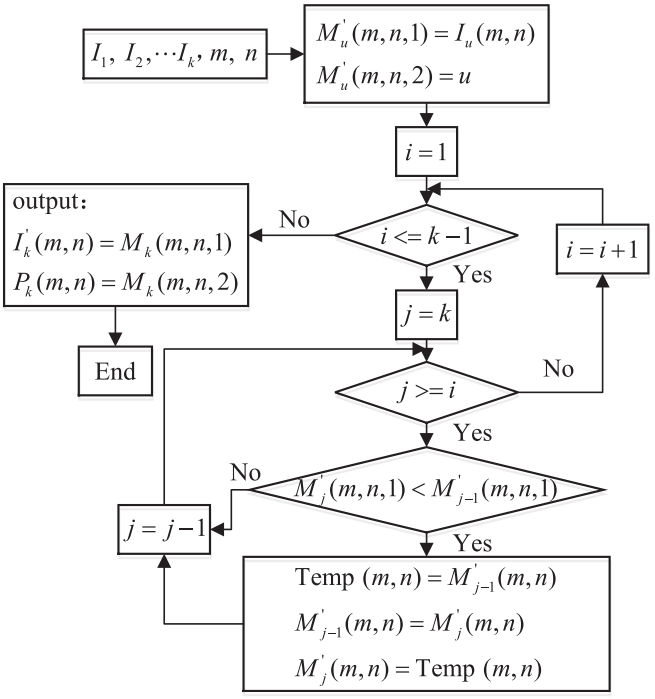

Fig. 1. Pixel exchange process.

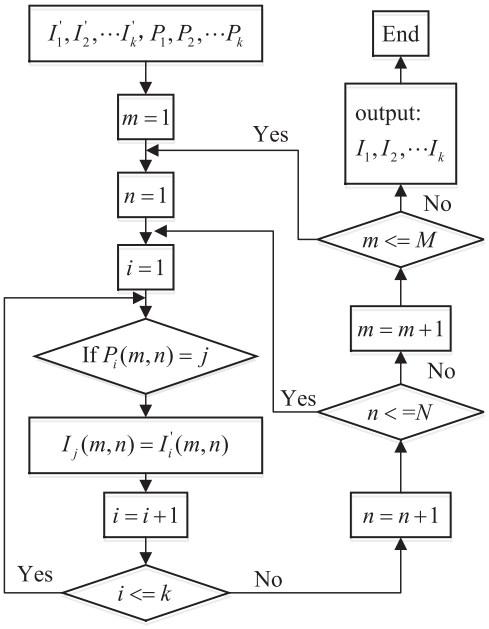

Fig. 2. Inverse pixel exchange process.

n) is stored in $I_i'(m,n)$. Then, the proposed algorithm reads the pixel value of $I_i'(m,n)$ and puts it into$I_j(m, n)$. When all pixel values of position matrices ($P_1, P_2 \cdots P_k$) at position (m, n) are dealt with, the above procedure is repeated for pixel position $(m, n+1)$. When all pixel positions in the position matrices have been completed, image I_k is recovered and considered as an output of the inverse operation.

An example of the pixel exchange operation is shown in Fig. 3. Four 256 × 256 pixels original images are as shown in Fig. 3(a)–(d) and four corresponding images obtained using the pixel exchange operator are shown in Fig. 3(e)–(h). The respective pixel position matrices are as shown in Fig. 3(i)–(l). The final retrieved images are as shown in Fig. 3(m)–(p).

3. Multiple image encryption

3.1. Image encryption algorithm

To encrypt multiple images, we consider combining phase encoding with a common vector decomposition. The optical implementation of a multiple image encryption algorithm is shown in Fig. 4. Two spatial light modulators, SLM_1 and SLM_2, are used for modulating phase while a third modulator SLM_3 is used for modulating amplitude. The intensity of synthesized vectors are recorded by a CCD camera CCD_1 while their phase values are displayed by SLM_2. The amplitude of final encrypted data is recorded by a camera CCD_2 and the phase values are displayed by SLM_2. As shown in Fig. 4, the original images $I_k(x, y)$ are imported into the pixel exchange operator and scrambled images $I_k'(x, y)$ are obtained in the pixel exchange process. Images I_1' and I_2' are imported into SLM_1 and SLM_2, respectively. The output of Beam Splitter 1 (BS_1) is expressed as follows:

$$\begin{aligned} C_1(x,y) &= \exp[i \cdot I_1'(x,y)] + \exp[i \cdot I_2'(x,y)] \\ &= A_1(x,y)\exp[i \cdot \varphi_1(x,y)] \end{aligned} \tag{1}$$

where $A_1(x, y)$ and $\varphi_1(x, y)$ denote respectively the amplitude and phase of the complex number $C_1(x, y)$. The phase information $\varphi_1(x, y)$ is displayed by SLM_2 and recorded while the amplitude information is captured by CCD_1 and displayed by SLM_3 initially.

The phase key θ_1 is calculated as $\theta_1 = I_2' - \varphi_1$ and serves as one of the private phase keys. Image I_3' is then imported to SLM_1 and combined with$C_1(x, y)$ and the output of BS_1 is given by

$$\begin{aligned} C_2(x,y) &= \exp[i \cdot I_3'(x,y)] + A_1(x,y)\exp[i \cdot \varphi_1(x,y)] \\ &= A_2(x,y)\exp[i \cdot \varphi_2(x,y)] \end{aligned} \tag{2}$$

The above procedure is repeated until all scrambled images have been encrypted. The amplitude of the final encrypted data is recorded by CCD_2 while the phase information is displayed by SLM_2. Here phase key θ_{k-1} calculated as $\theta_{k-1} = I_k' - \varphi_{k-1}$ serves as a private key.

$$\begin{aligned} C_{k-1}(x,y) &= \exp(i \cdot I_k'(x,y)) + A_{k-2}(x,y)\exp[i \cdot \varphi_{k-2}(x,y)] \\ &= A_{k-1}(x,y)\exp[i \cdot \varphi_{k-1}(x,y)] \end{aligned} \tag{3}$$

where C_{k-1} represents the final resultant vector, functions A_{k-1} and φ_{k-1} are the amplitude and phase of complex number C_{k-1}, respectively. A Fourier transform is further performed on complex number C_{k-1} and an inverse Fourier transform is performed on $C_k \exp(i2\pi R_2)$ as follows

$$\begin{cases} C_k(u,v) = \mathrm{FFT}\{C_{k-1}(x,y) \cdot \exp[i \cdot 2\pi \cdot R_1(x,y)]\} \\ E(x,y) = \mathrm{IFT}\{C_k(u,v) \cdot \exp[i \cdot 2\pi \cdot R_2(u,v)]\} \end{cases} \tag{4}$$

where C_k represents the complex number obtained by a Fourier transform, $E(x, y)$ represents the final encrypted result and $R_1(x, y)$ and $R_2(u, v)$ denote two random phase masks RPM_1 and RPM_2, respectively, FFT{} represents a Fourier transform, IFT{} represents an inverse Fourier transform.

3.2. Image decryption algorithm

The digital image decryption process is carried out as follows:

1. A Fourier transform is performed on an encrypted image $E(x, y)$ and the output is multiplied by a complex conjugate $R_2^*(u, v)$ to obtain a complex function followed by an inverse Fourier transform on the function. The output is then multiplied by a complex conjugate $R_1^*(x, y)$ to obtain a complex function $C_{k-1}(x, y)$.

$$\begin{cases} C_{k-1}(x,y) = \mathrm{IFT}\{[\mathrm{FFT}(E(x,y))] \cdot R_2^*(u,v)\} \cdot R_1^*(x,y) \\ A_{k-1}(x,y) = \mathrm{abs}[C_{k-1}(x,y)] \qquad (k \geq 2) \\ \varphi_{k-1}(x,y) = arg[(C_{k-1}(x,y))] \end{cases} \tag{5}$$

where symbol '*' denotes a complex conjugate, symbols 'abs' and 'arg' are the absolute value and complex angle of the complex number.