

TÁCH ẢNH DÙNG BIẾN ĐỔI WAVELET VÀ PHÂN TÍCH THÀNH PHẦN ĐỘC LẬP

Võ Minh Sơn, Nguyễn Hữu Phương

Trường Đại học Khoa học Tự nhiên, ĐHQG-HCM

(Bài nhận ngày 29 tháng 03 năm 2007, hoàn chỉnh sửa chữa ngày 01 tháng 03 năm 2008)

TÓM TẮT: Phân tích thành phần độc lập (Independent Component Analysis – ICA), thuộc các thuật toán học không giám sát, được nghiên cứu và ứng dụng nhiều trong vài chục năm trở lại đây. Một lĩnh vực ứng dụng của ICA là tách nguồn mù (Blind Source Separation – BSS), trong đó từ các tín hiệu trộn lẫn ta tìm lại các tín hiệu nguồn nguyên thủy trong lúc không biết được chính sự trộn. Thường các tín hiệu trộn quan sát phải được tiền xử lý bởi một, hoặc hơn, phương pháp phù hợp trước khi đưa vào ước lượng bởi mô hình ICA. Bài báo này trình bày việc tách ảnh bằng tiền xử lý biến đổi wavelet rời rạc (DWT) kết hợp với ước lượng ICA dùng thuật toán InfoMax và FastICA. Chúng tôi thử nghiệm trên nhiều loại ảnh dùng các DWT và thuật toán ICA khác nhau để so sánh về hiệu quả.

Từ khóa: Phân tích thành phần độc lập, tách nguồn mù, tách ảnh, biến đổi wavelet rời rạc.

1. GIỚI THIỆU

Xem trường hợp có nhiều tín hiệu nguyên thủy, gọi tín hiệu nguồn, được trộn lẫn thành nhiều tín hiệu trộn. Từ các tín hiệu trộn quan sát được này ta muốn phục hồi các tín hiệu nguồn riêng rẽ. Đây là bài toán tách nguồn mù (Blind Source Separation – BSS) mà phân tích thành phần độc lập (Independent Component Analysis – ICA) là một phương pháp hiệu quả. Tín hiệu trong bài này là các ảnh. Các tín hiệu nguồn là các thành phần độc lập và việc phục hồi chúng thường được gọi là ước lượng (estimation) ICA. Bài toán ICA không thể giải bằng toán học thông thường vì số lượng ẩn nhiều hơn số phương trình, mà được giải bằng các phương pháp thống kê. Để tăng hiệu quả, ICA thường được hỗ trợ bởi ít nhất một tiền xử lý mà trong bài này là biến đổi wavelet rời rạc (Discrete Wavelet Transform – DWT).

Tổ chức tiếp theo bài báo như sau: Mục 2 là mô hình ICA cơ bản, mục 3 tóm lược về DWT, mục 4 tách ảnh và chuẩn đánh giá, mục 5 mô phỏng và kết quả, và sau cùng, mục 6 là kết luận.

2. MÔ HÌNH ICA CƠ BẢN [1] [2] [3] [4]

Nhiều tín hiệu nguồn s_1, s_2, \dots, s_n được trộn (tổng hợp) tuyến tính thành các tín hiệu trộn x_1, x_2, \dots, x_n mô tả bởi hệ phương trình sau:

$$x_i = a_{i1}s_1 + a_{i2}s_2 + \dots + a_{in}s_n, \quad i = 1, 2, \dots, n \quad (1)$$

trong đó a_{ij} , $i, j = 1, 2, \dots, n$, là các hệ số trộn được giả sử là thực, và chưa biết. Thường dạng vector và ma trận được dùng:

$$\mathbf{x} = \mathbf{A}\mathbf{s} \quad (2)$$

với $\mathbf{x} = [x_1, x_2, \dots, x_n]^T$ là vector tín hiệu trộn

$\mathbf{s} = [s_1, s_2, \dots, s_n]^T$ là vector tín hiệu nguồn

$\mathbf{A} = [a_{ij}]$ là ma trận trộn

Ma trận trộn A gồm các cột gọi ai, nên vectơ x còn có thể viết như

$$\mathbf{x} = \sum_{i=1}^n \mathbf{a}_i s_i \quad (3)$$

Số lượng tín hiệu nguồn và tín hiệu trộn được giả sử bằng nhau (cùng là n) để việc tính toán được đơn giản. Đây là mô hình ICA cơ bản. Lúc bấy giờ A là ma trận vuông $n \times n$.

2.1 Bài toán ICA

Khi giải bài toán ICA ta ước lượng được các nguồn s nguyên thủy nhưng không hoàn toàn. Cụ thể là chỉ có thể xác định được dạng sóng của các thành phần độc lập, còn yếu tố thang tỉ lệ (biên độ tương đối), dấu (pha) và thứ tự của chúng thì không rõ. Trong phần lớn trường hợp thực tế các yếu tố bất định này ít quan trọng. Thật ra người ta đã và đang phát triển các cách để khắc phục các bất định.

2.2 Ước lượng ICA

Từ mô hình (2) bài toán ngược là

$$\mathbf{s} = \mathbf{A}^{-1} \mathbf{x} \quad (4)$$

\mathbf{A}^{-1} là ma trận giải trộn (demixing/unmixing matrix). Thật ra vì A không biết nên ta chỉ có thể ước lượng \mathbf{A}^{-1} và s, gọi chung là ước lượng ICA. Có nhiều cách như phương pháp khả năng cực đại (maximum likelihood) bao gồm các thuật toán như entropy cực đại, thông tin hỗ tương cực tiểu, tính phi gauss cực đại (maximum nongaussianity), và Informax.

3. BIẾN ĐỔI WAVELET VÀ PHÂN TÍCH ĐA PHÂN GIẢI [11] [12]

Vì biến đổi wavelet đã rất phổ biến nên mục này trình bày rất tóm lược về phân tích wavelet mà sẽ được làm một tiền xử lý cho các tín hiệu trộn trước khi đưa vào ước lượng ICA.

3.1. Biến đổi wavelet liên tục

Bằng cách lấy thang tỉ lệ (scaling) và dịch chuyển một hàm thời gian $\psi(t)$ gọi wavelet mẹ hay wavelet cơ sở, ta được một họ wavelet:

$$\Psi_{ab}(t) = \frac{1}{\sqrt{a}} \psi\left(\frac{t-b}{a}\right) \quad (5)$$

trong đó a là thông số thang tỉ lệ chỉ sự co giãn của wavelet, b là thông số dịch chuyển chỉ vị trí thời gian của wavelet. Dạng sóng tổng quát của các wavelet trong cùng họ được bảo toàn trong mọi co giãn và tịnh tiến.

Biến đổi wavelet liên tục (CWT) của một hàm thời gian (tín hiệu) x(t) được định nghĩa như

$$W_x(a, b) = \int_{-\infty}^{\infty} x(t) \psi_{ab}^*(t) dt = \langle x(t), \psi_{ab}(t) \rangle \quad (6)$$

trong đó * chỉ liên hiệp phức, $\langle \cdot \rangle$ chỉ tích nội. Biến đổi wavelet $W_x(a, b)$ diễn tả sự tương quan giữa tín hiệu x(t) và wavelet $\psi_{a, b}(t)$. Biến đổi thuận ở trên là phân tích, ngược lại là tổng hợp để phục hồi tín hiệu thời gian.

3.2. Biến đổi wavelet rời rạc

Biến đổi wavelet liên tục chứa nhiều trùng lặp và đòi hỏi tính toán công phu nên ít được dùng. Cả hai trở ngại trên được giải quyết đồng thời bằng cách rời rạc hóa thông số a, b:

$$a = a_0^m, \quad b = nb_0 a_0^m \quad (7)$$

trong đó m, n là số nguyên. Họ wavelet ở (5) trở thành

$$\Psi_{m,n}(t) = a_0^{-m/2} \Psi(a_0^{-m} t - nb_0) \quad (8)$$

Thông dụng nhất là rời rạc hóa dạng bát phân (octave) hay lũy thừa của 2 (dyadic) với $a_0 = 2, b_0 = 1$, kết quả

$$\Psi_{m,n}(t) = 2^{-m/2} \Psi(2^{-m} t - n) \quad (9)$$

Tập wavelet trên trực giao khi

$$\langle \Psi_{m,n}(t), \Psi_{k,l}(t) \rangle = \delta_{mk} \delta_{nl} \quad (10)$$

cho tất cả m, n, k, l nguyên, và δ là ký hiệu Kronecker.

Với sự chọn lựa thông số a, b như trên ta có biến đổi wavelet rời rạc (DWT) có các hệ số wavelet là

$$c_{m,n} = \langle x(t), \Psi_{m,n}(t) \rangle = \int_{-\infty}^{\infty} x(t) \Psi_{m,n}^*(t) dt \quad (11)$$

Việc tổng hợp sẽ cho lại tín hiệu thời gian:

$$x(t) = \sum_m \sum_n c_{m,n} \Psi_{m,n}(t) \quad (12)$$

3.3 Phân tích đa phân giải wavelet

Phân tích đa phân giải (multiresolution analysis – MRA) phân tích tín hiệu thời gian $x(t)$ ra các dải tần số khác nhau bởi các lọc thông thấp và thông cao liên tiếp. MRA dùng hai hàm bổ túc nhau là hàm tỉ lệ (scaling function) và hàm wavelet (thường chỉ gọi wavelet) liên kết lẫn lượt với các lọc thông thấp và thông cao. Hình 1a là cây phân tích wavelet 3 mức. Ra ở lọc thông thấp là thành phần xấp xỉ (approximation-A) và ra ở lọc thông cao là thành phần chi tiết (detail-D).

Từ các hệ số wavelet nhận được bởi sự phân ly, ta có thể phục hồi tín hiệu $x(n)$ nguyên thủy. Đây là biến đổi wavelet rời rạc ngược (IDWT).

Trong phân tích bằng con ở trên các tần số cao chứa các chi tiết tinh bị bỏ sót nên chỉ thích hợp với các tín hiệu có năng lượng tập trung ở vùng tần số thấp. Tuy nhiên ở nhiều tín hiệu, năng lượng tập trung ở các tần số giữa hoặc trải rộng khắp. Lúc đó cần phải phân chia ra các dải tần số thấp lẫn các dải tần số cao. Đây là sự phân tích wavelet packet (gói sóng con).

3.4. Wavelet packet

Các pixel của ảnh phân bố theo hai chiều (2-D) là ngang và dọc, khác với tín hiệu thông thường chỉ có chiều thời gian. Do đó trong xử lý ảnh ta dùng wavelet 2 chiều hoặc wavelet packet 2 chiều. Các hàm cơ sở wavelet packet 1-D được dùng để tạo ra các hàm cơ sở wavelet packet 2-D nhờ tích tensor theo chiều ngang và dọc.

4. TÁCH ẢNH VÀ CHUẨN ĐÁNH GIÁ [2] [4] [8]

4.1 Tách ảnh

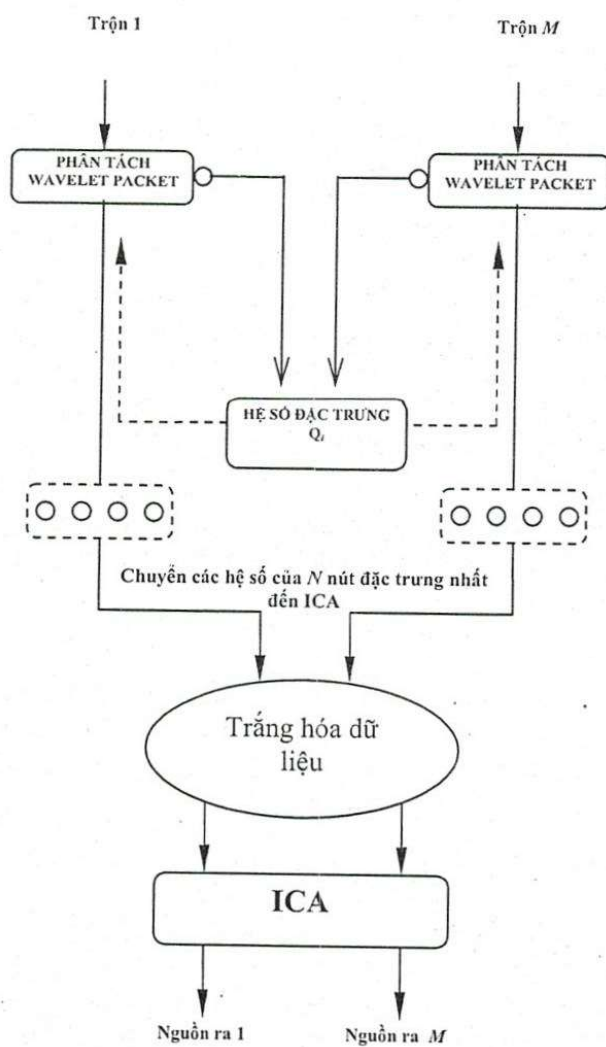
Các nhóm ảnh gốc được sẵn vào máy tính, sau đó ta thực hiện trộn tín hiệu theo Hình 1a bằng ngôn ngữ Matlab để tạo ra các trộn (hỗn hợp). Hình 1b là sơ đồ ước lượng ICA. Các hỗn hợp tạo ra từ mô hình 2a được đưa đến đây để thực hiện việc tách ảnh.

Trước tiên các ảnh trộn được tiền xử lý với wavelet packet 2D. Sau đó các hệ số đặc trưng của các cây wavelet packet được làm trắng hóa (whitened) rồi dựa vào phân tích ICA, ở đây

hai thuật toán được sử dụng là Infomax [1] [7] và FastICA [1] [2]. Kết quả ở ngõ ra là các ảnh ước lượng được của các ảnh gốc.



Hình 2a. Mô hình trộn ảnh



Hình 2b. Mô hình ước lượng ICA

4.2. Chuẩn đánh giá

Mỗi thực nghiệm được thực hiện 5 lần để đánh giá các chỉ tiết:

Độ hội tụ của thuật toán thông qua thời gian thực hiện thuật toán.

Sai số giữa ma trận trộn A và ma trận ước lượng W: $E(A, W)$ như sau:

$$E(A, W) = \min_{M \in \Gamma} \|A - WM\|, \quad (13)$$

với P là nhóm các ma trận $n \times n$ thực khả nghịch mà chỉ một phần tử trong mỗi cột khác không.

Hiệu quả của tiền xử lý cho mô hình ICA.

PSNR (Peak Signal to Noise Ratio), giả sử các ảnh nguồn đã biết, theo định nghĩa

$$PSNR = 10 \log \left(\frac{255}{\delta} \right)^2 = 48 - 20 \log \delta \quad (14)$$

với

$$\delta = \sqrt{\frac{1}{n} \sum_{i=1}^n (g_i - g'_i)^2}$$

trong đó

g_i là giá trị pixel xám của ảnh nguồn.

g'_i là giá trị pixel xám của ảnh ước lượng.

n là số pixel của ảnh.

Các ký hiệu của các thông số đánh giá như sau:

T1: Thời gian thực hiện ICA với dữ liệu không dùng wavelet packet làm tiền xử lý.

Error1: Sai số giữa ma trận trộn A và ma trận ước lượng W với dữ liệu không dùng wavelet packet làm tiền xử lý.

T2: Thời gian thực hiện ICA với dữ liệu có dùng wavelet packet làm tiền xử lý.

Error2: Sai số giữa ma trận trộn A và ma trận ước lượng W với dữ liệu có dùng wavelet packet làm tiền xử lý.

TP: Thời gian thực hiện tiền xử lý wavelet packet 2D.

5. MÔ PHỎNG VÀ KẾT QUẢ

5.1 Các nhóm ảnh thực hiện

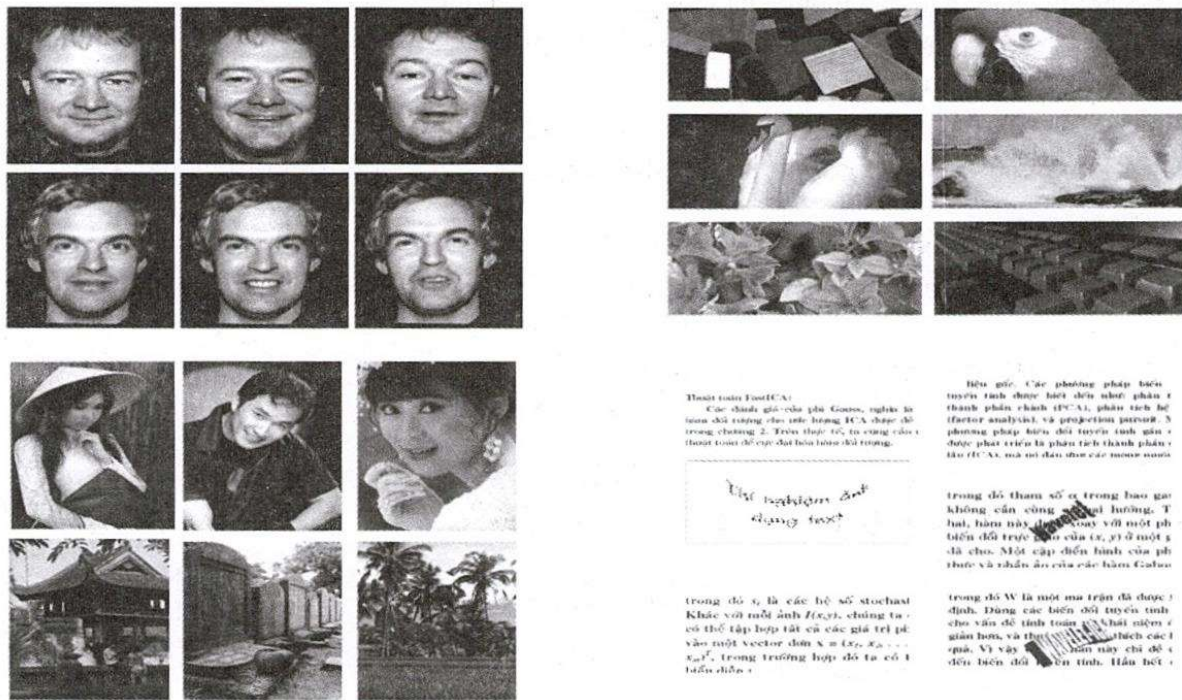
Chúng tôi thực hiện việc mô phỏng tách ảnh trên 4 nhóm ảnh khác nhau, mỗi nhóm gồm ảnh có nhiễu và ảnh không có nhiễu, trong đó nhóm ảnh đầu là các ảnh benchmark (cơ sở dữ liệu có sẵn) còn 3 nhóm ảnh sau là do chúng tôi nhập vào.

5.2 Kết quả

33 thực nghiệm ở 4 nhóm ảnh nêu ra ở mục 4.1 được thực hiện trên các ảnh không nhiễu và có nhiễu bằng các tiền xử lý dùng các wavelet packet (db4, db8, Haar, và sym2) được phân tích mức 2, mức 3, và hai thuật toán ICA là InfoMax và FastICA. Đánh giá các trường hợp từ thực nghiệm như sau.

5.2.1. Phân tiền xử lý

Thời gian thực hiện tiền xử lý T1 với các wavelet:



Hình 2. 4 nhóm ảnh khác nhau cho việc thực nghiệm

db4:

T1 = (14.22s ÷ 14.53s) ở mức 3
 T1 = (5.60s ÷ 7.23s) ở mức 2

db8:

T1 = (18.58s ÷ 18.74s) ở mức 3
 T1 = (5.40s ÷ 7.43s) ở mức 2

Haar:

T1 = (10.18s ÷ 10.49s) ở mức 3
 T1 = (4.14s ÷ 4.17s) ở mức 2

sym2:

Tp = (11.31s ÷ 11.56s) ở mức 3
 Tp = (4.57s ÷ 4.46s) ở mức 2

Từ các kết quả trên ta thấy nếu dùng phân ly wavelet packet mức 3 thì thời gian chiếm khoảng gấp đôi so với mức 2.

5.2.2. Phân ICA (Hình 3):

- Thuật toán InfoMax:

Thời gian thực hiện thuật toán và sai số E(A,W) với dữ liệu được tiền xử lý wavelet packet:

db4:

T2 = (1s ÷ 1.01s), Error2 = (0.72 ÷ 0.93) ở mức 3

	$T2 = (2.35s \div 3.09s), Error2 = (0.21 \div 0.31)$	ở mức 2
db8:	$T2 = (1.03s), Error2 = (0.66 \div 1.39)$	ở mức 3
	$T2 = (2.61s \div 2.64s), Error2 = (0.33 \div 0.46)$	ở mức 2
Haar:	$T2 = (0.83s \div 0.84s), Error2 = (0.28 \div 0.41)$	ở mức 3
	$T2 = (2.21s \div 2.23s), Error2 = (0.15 \div 0.28)$	ở mức 2
sym2:	$T2 = (0.83s \div 0.85s), Error2 = (0.28 \div 0.42)$	ở mức 3
	$T2 = (2.25s \div 2.26s), Error2 = (0.25 \div 0.31)$	ở mức 2

Bảng đánh giá PSNR (Peak Signal Noise Ratio): nhóm ảnh thứ nhất

	S1	S2	S3	S4	S5	S6
PSNR	94.53	88.09	90.58	88.97	86.06	87.31

- Thuật toán FastICA:

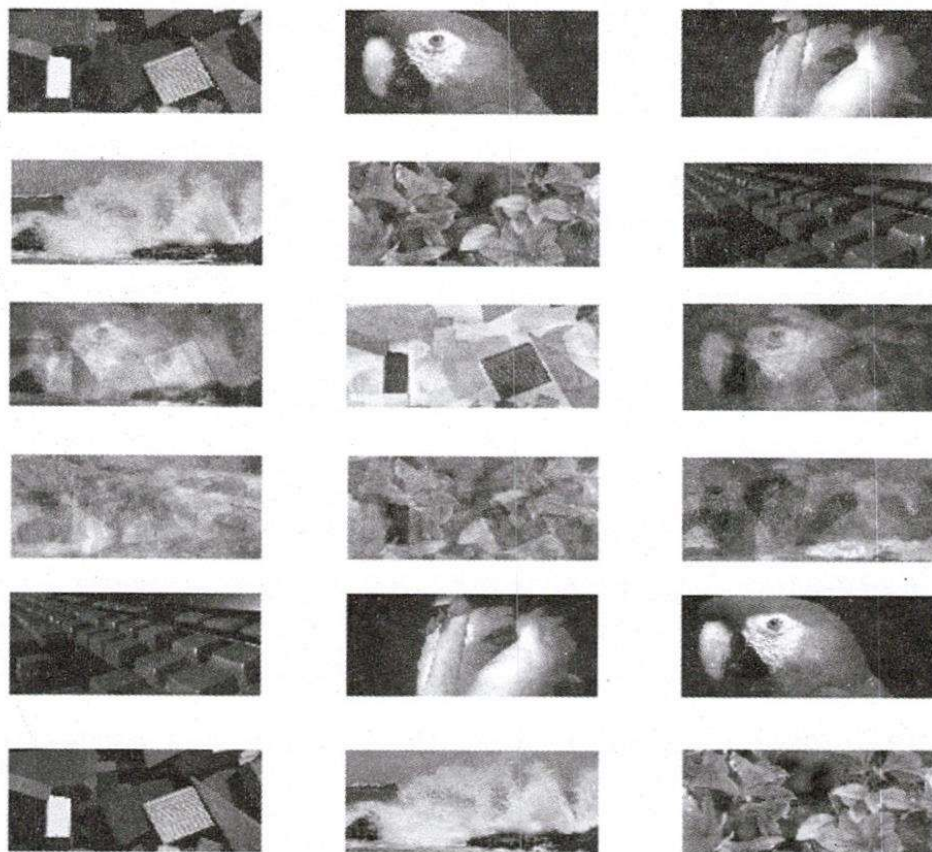
Thời gian thực hiện thuật toán và sai số E (A, W) với dữ liệu được tiền xử lý wavelet packet:

db4:	$T2 = (0.13s \div 0.25s), Error2 = (1.35e-14 \div 2.33e-14)$	ở mức 3
	$T2 = (0.24s \div 0.38s), Error2 = (6.82e-15 \div 1.20e-14)$	ở mức 2
db8:	$T2 = (0.14s \div 0.17s), Error2 = (9.73e-15 \div 9.73e-15)$	ở mức 3
	$T2 = (0.28s \div 0.34s), Error2 = (7.94e-15 \div 1.42e-14)$	ở mức 2
Haar:	$T2 = (0.13s \div 0.16s), Error2 = (1.01e-14 \div 1.49e-14)$	ở mức 3
	$T2 = (0.24s \div 0.32s), Error2 = (7.58e-15 \div 1.29e-14)$	ở mức 2
sym2:	$T2 = (0.13s \div 0.15s), Error2 = (1.16e-14 \div 1.47e-14)$	ở mức 3
	$T2 = (0.25s \div 0.49s), Error2 = (7.12e-15 \div 2.44e-14)$	ở mức 2

Bảng đánh giá PSNR (Peak Signal Noise Ratio):

	S1	S2	S3	S4	S5	S6
PSNR	95.03	93.37	92.06	74.67	83.90	83.79

Kết quả trên cho phép ta có một số nhận xét về hiệu quả của các loại wavelet packet, số mức, thuật toán ICA. Thuật toán FastICA có độ hội tụ nhanh và sai số giữa hai ma trận nhỏ hơn thuật toán InfoMax. Trong các trường hợp tiền xử lý khác nhau thì wavelet packet db4 và Haar cho kết quả ở ngõ ra tốt hơn số với trường hợp db8 nhưng nếu chúng ta đánh giá bằng chủ quan (nhìn vào kết quả) thì không thể phân biệt được thuật toán nào có sai số nhỏ hơn và ngõ ra ở trường hợp nào tốt hơn. Riêng trường hợp tiền xử lý với wavelet packet sym2 kết quả ngõ ra không tốt nếu phân ly mức 3.



Hình 3. Các ảnh được ước lượng từ các ảnh gốc không có nhiễu

5.3. Trường hợp tách ảnh có nhiễu [1] [7]:

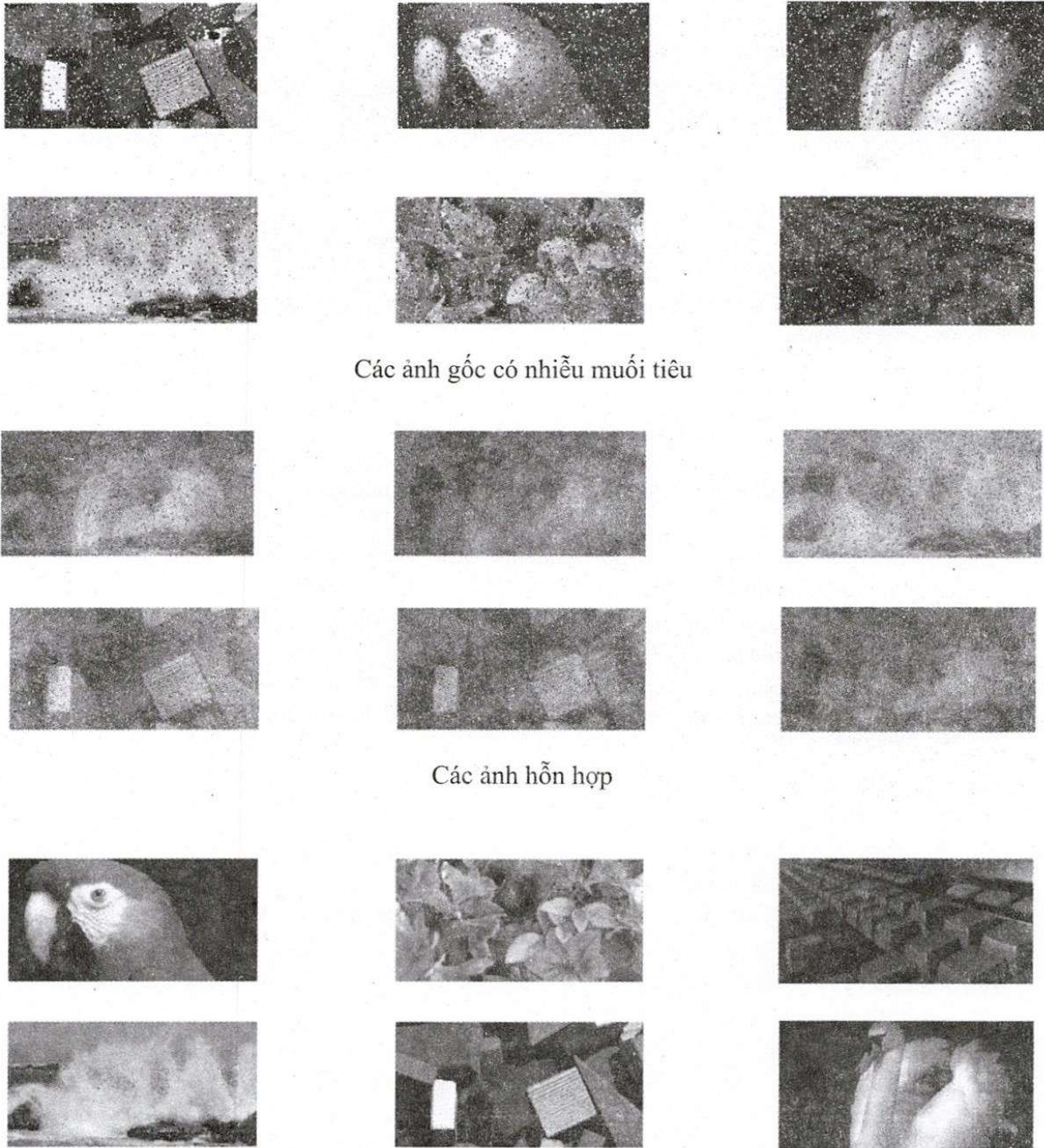
- Trường hợp ảnh gốc có nhiễu muối tiêu và gauss, thuật toán ICA vẫn tách được nhưng ảnh còn nhiễu.

- Trường hợp lọc nhiễu: nếu lọc nhiễu trên các ảnh đã ước lượng thì kết quả tốt (Hình 5) do quá trình lọc nhiễu không ảnh hưởng đến tín hiệu đưa vào thuật toán ICA. Nếu lọc nhiễu trên các ảnh hỗn hợp, tín hiệu bị ảnh hưởng do lọc nên khi đưa vào thuật toán ICA, kết quả ước lượng không tốt.

6. KẾT LUẬN

Các kết quả mô phỏng cho thấy ICA hiệu quả trong việc phục hồi lại các ảnh gốc sau khi các ảnh này đã được trộn tuyến tính. Sự hiệu quả tùy thuộc có dùng biến đổi wavelet làm tiền

xử lý hay không, loại wavelet, mức phân tích, và dùng thuật toán ICA nào (InfoMax hay FastICA ...). Kết quả của mô hình ICA cơ bản trình bày trong bản này là bước đầu để chúng tôi phân tích các tình huống khó khăn hơn (trộn phi tuyến, số trộn ít hơn số nguồn ...). Thật ra đã có nhiều công bố trên các tạp chí khoa học kết quả nghiên cứu của tác giả khác đối với các mô hình ICA phức tạp hơn mô hình cơ bản [8] [9] [10], tuy nhiên vẫn còn nhiều vấn đề cần được tiếp tục nghiên cứu.



Các ảnh gốc có nhiễu muối tiêu

Các ảnh hỗn hợp

Hình 4. Các ảnh tách từ ước lượng các ảnh gốc có nhiễu

IMAGE SEPARATION USING WAVELET TRANSFORM AND INDEDEDENT COMPONENT ANALYSIS

Vo Minh Son, Nguyen Huu Phuong
University of Natural Sciences, VNU-HCM

ABTRACT: *Independent Component Analysis (ICA), which belongs to the general class of unsupervised learning algorithm, has been studied and applied for a few decades. One of the application field is Blind Source Separation (BSS), in which from the mixed signals we would like to recover the original source signal, whereas the mixing is unknown. In our work, the mixed images are preprocessed by the Discrete Wavelet Transform (DWT) multiresolution analysis, followed by the ICA estimation using InfoMax and FastICA algorithms.*

We do not contribute any new theoretical or algorithmic development. Instead, we carry out extensive simulations on many different types of images using various DWTs and ICA algorithms for comparison of their effectiveness.

Keywords: *Independent component analysis, blind source separation, image separation, discrete wavelet transform.*

TÀI LIỆU THAM KHẢO

- [1]. Hyvrinen A., Karhunen J., Oja E., *Independent Component Analysis*, John Wiley & Sons, (2001).
- [2]. Hyvrinen, A., Oja, E., *Independent component analysis: algorithms and Applications*, Neural Networks, 13(4-5), 411- 430, (2000).
- [3]. Hyvrinen A., *Survey on Independent Component Analysis*, Helsinki University of Technology, Finland, (1999).
- [4]. Choi S., Cichocki A., Park H-M., Lee S-H., *Blind Source Separation and Independent Component Analysis: A Review*, Neural Information Processing, Vol. 6, No. 1, (2005).
- [5]. Cichocki A., Amari S-I., *Adaptive Blind Signal and Image Processing*, John Wiley & Sons, (2002).
- [6]. Almeida L.B., *Separating a Real-Life Nonlinear Image Mixture*, Journal of Machine Learning Research 6 1199-1229, (2005).
- [7]. Hyvrinen A., *Gaussian Moments for Noise Independent Component Analysis*, Helsinki University of Technology, Finland, (1999).
- [8]. Cardoso J.-F., *Infomax and maximum likelihood for source separation*, IEEE Letters on Signal Processing, 4:112-114, (1997).