Comparing Multiresolution SVD with Other Methods for Image Compression*

Ryuichi Ashino[†‡] Akira Morimoto[†§]

Michihiro Nagase[¶] Rémi Vaillancourt[‖]

CRM-2987

December 2003

*This research was partially supported by the Japanese Ministry of Education, Culture, Sports, Science and Technology, Grant-in-Aid for Scientific Research (B) 14340045(2002–2004), (C) 15540170(2003–2004), NSF Grant DMS-0139261, the Natural Sciences and Engineering Research Council of Canada and the Centre de recherches mathématiques of the Université de Montréal

[†]Osaka Kyoiku University, Kashiwara, Japan

[‡]ashino@cc.osaka-kyoiku.ac.jp

[§]morimoto@cc.osaka-kyoiku.ac.jp

[¶]Osaka University, Toyonaka, Osaka, Japan; nagase@math.wani.osaka-u.ac.jp

[‖]University of Ottawa, Ottawa, Canada; remi@mathstat.uottawa.ca

Abstract

Digital image compression with multiresolution singular value decomposition is compared with discrete cosine transform, discrete 9/7 biorthogonal wavelet transform, Karhunen–Loève transform, and a hybrid wavelet-svd transform. Compression uses SPIHT and run-length with Huffmann coding. The performances of these methods differ little from each other. Generally, the 9/7 biorthogonal wavelet transform is superior for most images that were tested for given compression rates. But for certain block transforms and certain images other methods are slightly superior.

To appear in *Proc. Fourth ISAAC Congress* (York University), H. Begehr, R. P. Gilbert, M. Muldoon, and M. W. Wong, eds., Kluwer.

Keywords. multiresolution singular value decomposition, biorthogonal wavelet, SPIHT, image compression

Résumé

On compare la compression d'image par décomposition multirésolution en veleurs singulière (dvs) avec la transformation en cosinus discrète, la transformation en ondelettes 9/7 biorthogonale, la transformation de Karhunen–Loève et une transformation ondelette-dvs hybride. La compression se fait par SPIHT et le code d'Huffman. La performance de ces méthodes diffrent peu entre elles, mais la transformation en ondelettes 9/7 biorthogonale les supplante quelque peu sur la plupart des images testées à compression égale. Pour certaines tranformations blocs et certaines images, d'autres méthodes sont quelque peu supérieures.

1 Introduction

Image compression is important in digital image transmission and storage. Comparative studies of compression methods are found in [5] and [1]. In [3], image compression with multiresolution singular value decomposition [6] is compared with discrete cosine transform, discrete 9/7 biorthogonal wavelet transform, Karhunen–Loève transform, and a hybrid wavelet-svd transform. Compression uses Set Partitioning in Hierarchical Trees (SPIHT) [7] and run-length with Huffmann coding. These methods are briefly reviewed and their performance is tested through numerical experiments on several well-known images. It is found that these methods differ little from each other at moderate compression ratio. Generally, the 9/7 biorthogonal wavelet transform is superior for most images that were tested for given compression rates. But for certain block transforms and certain images other methods are slightly superior.

Section 2 summarizes multiresolution analysis (MRA) and block algorithms. Section 3 describes the coding methods. In Section 4, we propose a hybrid method using the 9/7 biorthogonal wavelets with singular value decomposition (SVD). Table 1 lists the results of numerical experiments with these methods on five images and Table 2 in Section 5 list the results on a fingerprint image, for which visual inspection is done in Section 6.

2 Multiresolution Processing and Block Algorithms

The analysis stage of a two-dimensional separable discrete wavelet transform produces the matrix $X = UAV^T$, where the upper and lower half-parts of the orthogonal matrices U and V correspond to lowpass and highpass filters, respectively. The discrete wavelet transform divides the image into four parts as in the following **procedure**:

(P1) The scaling function $\varphi(x)\varphi(y)$ produces the top left part.

(P2) The vertical wavelet function $\psi(x)\varphi(y)$ produces the top right part.

(P3) The horizontal wavelet function $\varphi(x)\psi(y)$ produces the bottom left part.

(P4) The diagonal wavelet function $\psi(x)\psi(y)$ produces the bottom right part.

The top left part is called an *approximation* because it is smooth and has large values. The other three parts are called *details* because they emphasize horizontal, vertical, and diagonal edges, respectively. These three parts have small absolute values except for edges. A multi-level decomposition is obtained by applying this decomposition to successive approximations.

Similar decompositions are achieved by the discrete cosine transform (DCT) and the SVD by means of the following **block algorithm**:

(BA1) A given image matrix $X \in \mathbb{R}^{m\times n}$ is divided into $b \times b$ submatrices $X^{(k,\ell)}$, $1 \le k \le m/b$, $1 \le \ell \le n/b$.

(BA2) Each submatrix $X^{(k,\ell)}$ is transformed into $X_1^{(k,\ell)}$ by the DCT or the SVD.

(BA3) The matrix $X_1^{(k,\ell)}$ is rearranged into an $(m/b) \times (n/b)$ matrix $X_2^{(i,j)}$.

(BA4) The $X_2^{(i,j)}$ matrices are put in the (i, j) position to produce the $m \times n$ matrix X_3 which contains b^2 parts and is similar to the matrix obtained by the DWT.

The **Kakarala–Ogunbona's algorithm** [6] is a kind of multiresolution algorithm. We explain here the two-dimensional algorithm for level 1.

(KO1) Each $b \times b$ submatrix $X^{(k,\ell)}$ of a given matrix $X \in \mathbb{R}^{m\times n}$ is reshaped into a $b^2 \times 1$ column vector.

(KO2) These column vectors are collected into a $b^2 \times (mn/b^2)$ matrix T.

(KO3) T is factored into its reduced singular value decomposition in the form $T = USV^T$, where $U \in \mathbb{R}^{b^2\times(mn/n^2)}$ and $V \in \mathbb{R}^{(mn/b^2)\times b^2}$ have orthonormal columns, and $S \in \mathbb{R}^{4\times4}$ is diagonal.

(KO4) Calculate the $b^2 \times (mn/b^2)$ matrix $A = U^TT = SV^T$.

(KO5) Each column vector of A is reshaped into a $b \times b$ matrix $X_1^{(k,\ell)}$.

(KO6) All the matrices $X_1^{(k,\ell)}$ are rearranged into an $m \times n$ matrix X_1.

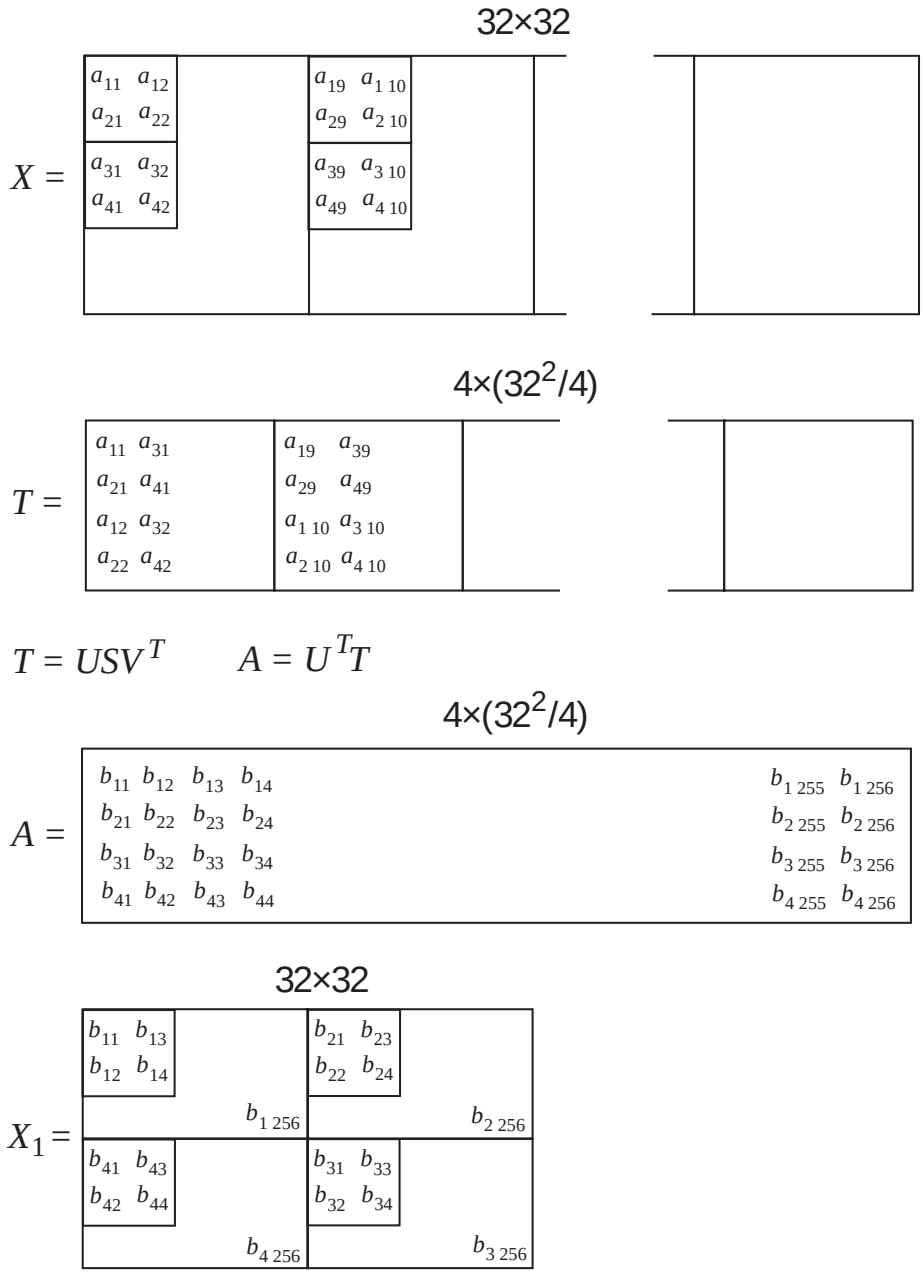

Figure 1: Level 1 SVD MRA for a 32×32 matrix.

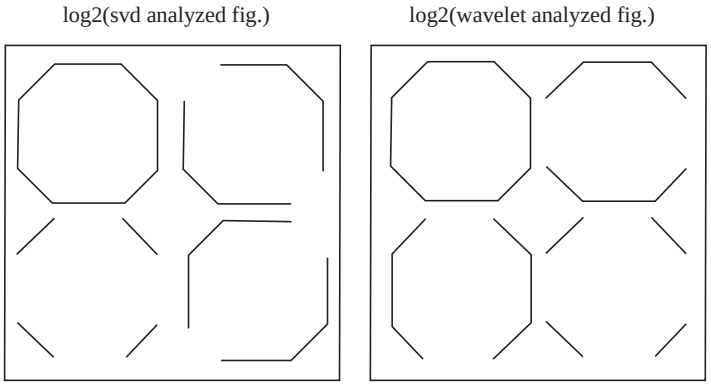

Figure 2: Negative image of level-1 approximation and detail subimages of the octagon figure produced with SVD and 9/7 wavelet MR, respectively. The level-1 approximation is in the top left subimages.

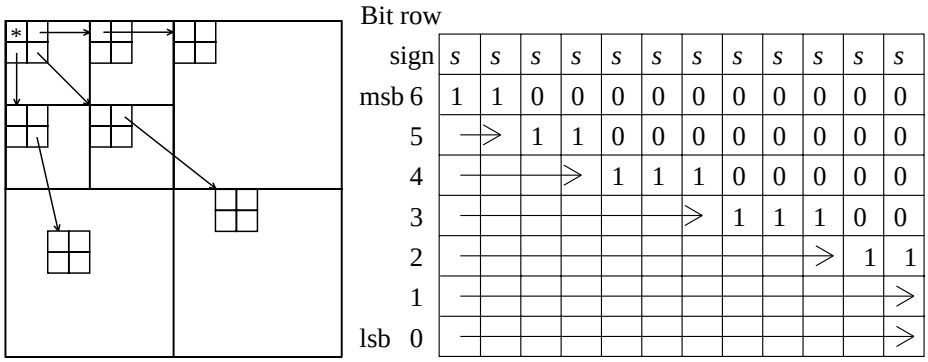

Figure 3: Left: hierarchical structure. Right: binary representation of the magnitude-ordered coefficients

Figure 1 illustrates the algorithm for level-1 SVD MRA on a 32×32 matrix.

Figure 2 illustrates the difference between SVD and 9/7 wavelet multiresolution at level 1 for the octagon figure. One notices that the four isolated diagonal segments appear in the lower-left and lower-right detail parts of the SVD and wavelet multiresolution, respectively. The singular values and left singular vectors for the level-1 SVD MRA of the octagon image are in the vector S and the columns of U, respectively,

```
S =  4554.4    U = 0.5000   -0.0000   -0.7071   -0.5000
     3524.2        0.5000    0.7071   -0.0000    0.5000
     3524.2        0.5000   -0.7071    0.0000    0.5000
     2024.0        0.5000    0.0000    0.7071   -0.5000
```

One sees that the first column of U is a lowpass filter.

The norm of the nth row of A is equal to the nth singular value of T because the columns of the matrix V are orthonormal. The $b^2 \times b^2$ orthogonal matrix U and the singular values are needed for the inverse transform.

3 SPIHT

The SPIHT [7] algorithm is based on the following two observations:

Observation 1. *The pixels of the analyzed image having large absolute values are concentrated in the upper-left corner.*

Observation 2. *SPIHT encodes zerotrees based on the principle that when a wavelet coefficient has small absolute value, then points at other levels corresponding to this coefficient also have small absolute values.*

SPIHT has three ordered lists:

- the list of significant pixels (LSP),
- the list of insignificant pixels (LIP),
- the list of insignificant sets (LIS).

LIP and LIS are searching areas. LSP lists the pixels whose absolute values are greater than 2^N, thus requiring more than N bits. Each pixel of LIP is tested whether its absolute value is less than 2^N or not. Each pixel of LIS is tested whether all absolute values of its descendants are less than 2^N. At the first step, all the pixels of LIS are type 'A'. Some pixels of LIS will be changed from type 'A' to type 'B' in the following **SP procedure**:

(SP1) LSP is taken as an empty list and LIP is the set of top level coefficients. LIS is the set of top level wavelet coefficients and all the pixels of LIS are type 'A'. N is set to the most significant bit of all coefficients.

(SP2) Check each pixel of LSP and output 0 if its Nth bit is 0, and output 1 otherwise.

(SP3) Check each pixel of LIP and output 0 if its absolute value is less than 2^N. Otherwise, output 1 and, moreover, output 0 when the value of this pixel is negative and 1 if positive, and move this pixel to LSP.

(SP4) Check each pixel of LIS.

(a) When the pixel is of type 'A', output 0 if the absolute values of all its descendants are less than 2^N. Otherwise, output 1 and do the following:
   (i) Check all four children.
   (ii) When the absolute value of a child is greater than or equal to 2^N, output 1 and, moreover, output 0 or 1 according to the sign of this child and add this child to LSP.
   (iii) When the absolute value of this child is less than 2^N, add this child to the end of LIP.
   (iv) When this pixel has grandchildren, move it to the end of LIS as a pixel of type 'B'.

(b) When a pixel is of type 'B', output 0 if the absolute values of all descendants, apart from the children, are less than 2^N. Otherwise, output 1 and add each child to the end of LIS as type 'A' and delete this pixel from LIS.

(SP5) Set N to $N-1$ and go to step (SP2).

(SP6) When the number of output bits exceeds the threshold (which is decided by user's bpp), then stop this procedure.

The SPIHT algorithm is very efficient for high compression rate when N is large but does not minimize memory nor bandwidth and is not designed to look at regions of interest, as opposed to JPEG 2000.

The run-length and Huffmann coding can quantize the analyzed image economically by the following **HU procedure**:

(HU1) Divide each block of the analyzed image by some integer which depends on the image and the block location. Each pixel of this divided image is rounded to an integer. This quantized analyzed image has many 0 entries.

(HU2) Reshape this image into a long row vector. In this step, use the following two methods:

(a) Reshape each block into a vector and stack these vectors together.

(b) Use the hierarchical tree (0 tree) algorithm.

(HU3) Compress the 0 entries of this long row vector by the run-length coding.

(HU4) Compress the run-length coded image by `gzip`.

4 Hybrid Wavelet-SVD Method

We propose a hybrid method which combines wavelet and singular value decompositions. The **analysis procedure** consists in the following three steps:

(AN1) Transform the $m \times n$ image X into the analyzed image X_1 by the level-two DWT using the 9/7 biorthogonal wavelets.

(AN2) Decompose X_1 into 2×2-block SVD MRA up to level six to get X_2.

(AN3) Compress X_2 by SPIHT and compress the resulting image with `gzip`.

The **synthesis procedure** consists in the following three steps:

(SY1) Uncompress the `gzip` image with `gunzip` and decode the compressed code to $\hat{X}_2$.

(SY2) Obtain the synthesized image $\hat{X}_1$ by the inverse 2×2-block SVD transform.

(SY3) Obtain the reconstructed image $\hat{x}$ from $\hat{X}_1$ by the inverse DWT.

We have the following observation:

OBSERVATION 3. *Our hybrid wavelet-SVD method is better than SVD alone, It is better than biorthogonal wavelet for the `fp1` and `barb` images.*

This observation leads to the following **conclusions**:

(C1) The SVD decomposition depends on the data and cannot deal with data in time-frequency domain. Because our hybrid method contains wavelet analysis, which is a kind of time-frequency analysis, our hybrid method performs better.

(C2) The blocking effect in our hybrid method is weaker than with SVD, because we use long-filter wavelets in the last synthesis step.

5 Numerical Experiments

Eight bit-per-pixel (bpp) images have been compressed by the following methods.

- `bior4.4` is the biorthogonal wavelet filter with 9/7 taps of [2].
- `db2` is Daubechies' compactly supported wavelet filter with $N = 2$.
- `2by2SVDMR` and `4by4SVDMR` are the SVD multiresolution with block size 2 and 4, respectively.
- `JPEG` is MATLAB's `imwrite` function.
- `2by2KLTMR` and `4by4KLTMR` are the KLT multiresolutions with block size 2 and 4, respectively.
- `bior4.4+SVD` consists of the following two steps. In the first step, the image is transformed by `bior4.4` wavelet to level 2. In second step, the transformed image is decomposed by `2by2SVDMR` to level 6.

The SPIHT algorithm [7] is used for coding the MRA methods.

Six well-known images, 512×512 Lena, Boats, Barb, and Yogi, 512×640 Goldhill, and 768×768 fp1, shown in Fig. 4 have been tested. The fp1 image is a sample of the FBI WSQ FINGERPRINT COMPRESSION DEMOS 4.2.5.

Four objective measures, PSNR, MSE, MaxErr, and SNR, defined below, were applied to $m \times n$ original and reconstructed images, X and $\hat{x}$.

DEFINITION 1 ***Peak Signal to Noise Ratio*** (PSNR) *and* ***Signal to Noise Ratio*** (SNR) *are:*

$$\mathrm{PSNR} = 10 \log_{10} \left(\frac{255^2 mn}{\|X - \hat{x}\|_F^2} \right), \quad \mathrm{SNR} = 10 \log_{10} \left(\frac{\|X\|_F^2}{\|X - \hat{x}\|_F^2} \right),$$

where the square of the Frobenius norm of an $m \times n$ matrix A is

$$\|A\|_F^2 = \sum_{i=1}^{m} \sum_{j=1}^{n} A_{i,j}^2.$$

The ***mean square error*** *and the* ***maximum error*** *are*

$$\mathrm{MSE} = \frac{1}{mn} \|X - \hat{x}\|_F^2, \quad \mathrm{MaxError} = \|X - \hat{x}\|_\infty.$$

In this work, bpp is the number of bits in the `gzip` image divided by the number of bits in the original image.

Peak Signal to Noise Ratio (PSNR) with the bior4.4 method is generally higher except for the Yogi image at 1 and 0.5 bpp where 2by2SVDMR and 2by2KLTMR are superior.

The numerical results listed in Tables 1 and 2 lead to the following **conclusions**:

(C3) At high compression ratio, that is, low bpp, block effects appeared for SVD, KLT, and JPEG, especially remarkable for `SVD2by2` and `KLT2by2`. On the other hand, in case of wavelet with long filters, images were out of focus. Our hybrid method using 9/7 wavelet with SVD lies between these two opposite cases.

(C4) For the fingerprint, our hybrid method using 9/7 wavelet with SVD was superior to the other methods.

(C5) Better performance was obtained with short-filter `SVD2by2` and `KLT2by2` for Yogi as it uses fewer grey levels,

(C6) For other images, our hybrid method performed a little bit inferior to wavelet `bior4.4`, but superior to SVD, KLT, and JPEG.

Every experiment was run four times successively under the same conditions, and the cputime, measured with the MATLAB `profile` function, was taken to be the mean value of the last three runs. The computations were done on a portable PC with the following specifications: Pentium III 866 Mhz, 512 MB memory, Microsoft Windows 2000 and MATLAB R13. Partial results are listed in Table 1 for the first five figures. Fuller results are in [3].

Figure 4: The six original figures.

Table 1: Results for Lena, Boats and Goldhill at 0.25 bpp, Barb at 1.5 bpp, and Yogi at 0.5 bpp. The bpp used by JPEG is indicated in the first column.

Image	Method	Level	PSNR	MSE	MaxErr	SNR	CPU
Lena	bior4.4	6	33.4193	29.5901	41.9485	27.7383	4.96
	db2	6	32.0355	40.6943	44.8188	26.3544	4.64
	2by2SVDMR	6	30.3235	60.3568	64.4865	24.6425	3.14
	4by4SVDMR	4	30.8061	54.0094	73.974	25.1251	3.24
0.26	JPEG		30.7576	54.6158	82	25.0766	0.97
	2by2KLTMR	6	30.2218	61.7871	61.0016	24.5408	34.74
	4by4KLTMR	4	30.7977	54.1135	63.7171	25.1167	9.17
	bior4.4+SVD	2+6	32.2857	38.416	52.0023	26.6046	5.55
Boats	bior4.4	6	29.4905	73.1191	80.172	24.1479	4.98
	db2	6	28.6923	87.8713	75.3258	23.3497	4.53
	2by2SVDMR	6	27.5786	113.5583	81.147	22.236	3.27
	4by4SVDMR	4	27.8278	107.2269	87.8259	22.4852	3.23
0.25	JPEG		27.3174	120.5969	109	21.9748	0.99
	2by2KLTMR	6	27.658	111.5017	82.5188	22.3154	34.10
	4by4KLTMR	4	27.9562	104.1012	87.8324	22.6136	9.17
	bior4.4+SVD	2+6	28.5882	90.0041	78.3846	23.2456	5.43
Gold	bior4.4	6	30.5292	57.5658	51.2446	23.5659	6.33
	db2	6	29.77	68.5611	64.5146	22.8068	5.34
	2by2SVDMR	6	29.3633	75.2917	73.4238	22.4001	3.74
	4by4SVDMR	3	29.6741	70.0923	66.1833	22.7108	3.65
0.26	JPEG		29.6083	71.1619	70	22.6451	1.15
	2by2KLTMR	6	29.5023	72.9213	74.7445	22.539	43.05
	4by4KLTMR	3	29.8132	67.8827	60.2749	22.8499	11.07
	bior4.4+SVD	2+6	29.9528	65.7348	68.479	22.9896	6.24
Barb	bior4.4	6	30.3506	59.9822	34.2253	24.0747	5.94
	db2	6	30.0205	64.7188	37.389	23.7446	5.56
	2by2SVDMR	6	29.4611	73.6165	39.915	23.1852	4.14
	4by4SVDMR	4	29.8336	67.565	37.9863	23.5577	4.11
1.51	JPEG		28.2041	98.3269	51	21.9282	1.01
	2by2KLTMR	6	29.4167	74.3729	40.5208	23.1408	34.47
	4by4KLTMR	4	29.9307	66.0707	39.6838	23.6548	9.76
	bior4.4+SVD	2+6	30.4038	59.252	38.8405	24.1279	6.51
Yogi	bior4.4	6	31.431	46.7712	66.2005	24.8921	5.06
	db2	6	30.3525	59.9555	71.4808	23.8136	4.63
	2by2SVDMR	6	34.1772	24.8517	61.563	27.6384	3.37
	4by4SVDMR	4	28.8366	85.0009	127.6891	22.2977	3.35
0.51	JPEG		28.8926	83.9116	112	22.3537	0.96
	2by2KLTMR	6	34.4421	23.3812	62.9133	27.9033	34.08
	4by4KLTMR	4	29.2121	77.9592	126.2277	22.6732	9.07
	bior4.4+SVD	2+6	28.8919	83.9258	108.8086	22.353	5.65

6 Visual Inspection of the Fingerprint Image at 0.15 bpp.

The six compression methods, `bior4.4`, `db2`, `2by2SVDMR`, `4by4SVDMR`, `4by4KLTMR`, and `bior4.4+SVD`, applied to the 768×768 `fp1` image produce very similar synthesized images at 0.15 bpp on the screen and in 40%-reduced print form. However, at high compression ratio, that is, low bit per pixel, visual inspection is necessary to ascertain the quality of synthesized images.

It is seen in Table 2 for `fp1` that PSNR is below 30 db at bpp = 0.15 so that some visual deterioration of the synthesized images may be expected. Blocking effects (BE) and blurring of the fingerprint image at 0.15 bpp can be observed at 200% and 300% magnification with Adobe Illustrator. The following list goes from low to high performance.

- `jpeg`: strong BE at 200%
- `2by2SVDMR`: weak BE at 200%, strong at 300%
- `2by2KLTMR`: weak BE at 200%, moderate at 300%
- `4by4SVDMR` weak BE at 200%, slightly strong at 300%
- `4by4KLTMR`: weak BE at 200%, moderate at 300%
- `db2`: weak BE at 200% with a little blurring
- `bior4.4+SVD`: weak BE at 600% with some blurring in parts
- `bior4.4`: weak BE at 600% with some blurring in parts

Visual inspection corroborates the PSNR.

Figures 5 and 6 show a magnified part of the fingerprint image at 0.15 bpp. Again magnification is by Adobe Illustrator. It is seen that apart from `bior4.4+SVD` and `bior4.4`, the other methods introduce blocking effects.

The curves in Figs. 7 show that the new hybrid method, `bior4.4+SVD`, has higher PSNR against bpp than other methods for the fingerprint image.

References

[1] Aase, S. O., Husøy, J. H., Waldemar, P. (1999) *A critique of SVD-based image coding systems*, Proceedings of the 1999 IEEE International Symposium on Circuits and Systems VLSL, **4** pp. 13–16, IEEE Press, Piscataway, NJ.

[2] Antonini, M., Barlaud, M., Mathieu, P., Daubechies, I. (April 1992) *Image coding using wavelet transform*, IEEE Trans. Image Processing, **1** 205–220.

[3] Ashino, R., Morimoto, A., Nagase, M., Vaillancourt, R. *Image compression with multiresolution singular value decomposition and other methods*, Mathematical and Computer Modelling, to appear

[4] Chen, J. (2000) *Image compression with SVD*, ECS 289K Scientific Computation, Dec. 13, 2000 13 pages. `http://graphics.cs.ucdavis.edu/~jchen007/UCD/ECS289K/Project.html`

[5] Gerbrands, J. J. (1981) *On the relationships between SVD, KLT, and PCA*, Pattern Recognition, **14** 375–381

[6] Kakarala, R., Ogunbona, P. O. (May 2001) *Signal analysis using a multiresolution form of the singular value decomposition*, IEEE Trans. on Image Processing, **10**, No. 5, 724–735.

[7] Said, A., Pearlman, W. A. (June 1996) *A new fast and efficient image codec based on set partitioning in hierarchical trees*, IEEE Trans. on Circuits and Systems for Video Technology, **6**, 243–250.

[8] Unser, M. (1993) *An extension of the Karhunen–Loève transform for wavelets and perfect reconstruction filterbanks* SPIE, **2034** Mathematical Imaging, 45–56.

[9] Waldemar P., Ramstad, T. A. (1997) *Hybrid KLT-SVD image compression*, in 1997 IEEE International Conference on Acoustics, Speech, and Signal Processing, **4** pp. 2713–2716, IEEE Comput. Soc. Press, Los Alamitos, CA.

Table 2: Numerical results for the 768×768 `fp1` image

bpp	Method	Level	PSNR	MSE	MaxErr	SNR	cpu
1	bior4.4	6	39.684	6.994	14.727	37.221	10.3
	db2	6	38.130	10.002	22.632	35.667	9.7
	2by2SVDMR	6	36.569	14.329	21.849	34.106	6.3
	4by4SVDMR	4	38.0430	10.204	18.85	35.581	6.3
	JPEG		37.825	10.730	22	35.363	1.7
	2by2KLTMR	6	35.793	17.132	22.308	33.331	76.9
	4by4KLTMR	4	38.222	9.793	17.41	35.759	20.2
	bior4.4+SVD	2+6	40.221	6.180	14.96	37.759	11.9
0.5	bior4.4	6	35.724	17.404	32.551	33.262	9.8
	db2	6	33.834	26.89	35.070	31.372	9.0
	2by2SVDMR	6	31.483	46.219	44.070	29.020	5.5
	4by4SVDMR	4	33.633	28.171	37.668	31.170	5.6
	JPEG		34.000	25.887	37	31.538	1.68
	2by2KLTMR	6	31.165	49.721	43.419	28.70	76.2
	4by4KLTMR	4	33.766	27.321	38.342	31.304	19.5
	bior4.4+SVD	2+6	35.954	16.509	28.77	33.491	11.0
0.25	bior4.4	6	32.533	36.287	44.579	30.071	9.3
	db2	6	30.525	57.622	53.073	28.067	8.7
	2by2SVDMR	6	28.197	98.497	67.898	25.734	5.1
	4by4SVDMR	4	29.513	72.743	65.411	27.051	5.2
	JPEG		28.990	82.056	84	26.527	1.68
	2by2KLTMR	6	28.052	101.838	70.428	25.589	75.8
	4by4KLTMR	4	29.666	70.222	73.483	27.204	19.2
	bior4.4+SVD	2+6	32.436	37.106	42.748	29.97	10.6
0.15	bior4.4	6	29.877	66.893	60.519	27.415	11.0
	db2	6	28.418	93.592	69.640	25.956	10.1
	2by2SVDMR	6	26.517	145.013	94.399	24.054	5.9
	4by4SVDMR	4	26.784	136.353	94.678	24.322	6.0
	JPEG		24.248	244.495	129	21.786	1.91
	2by2KLTMR	6	26.417	148.400	92.085	23.954	76.8
	4by4KLTMR	4	27.007	129.534	92.582	24.545	20.0
	bior4.4+SVD	2+6	29.906	66.441	62.348	27.444	12.1

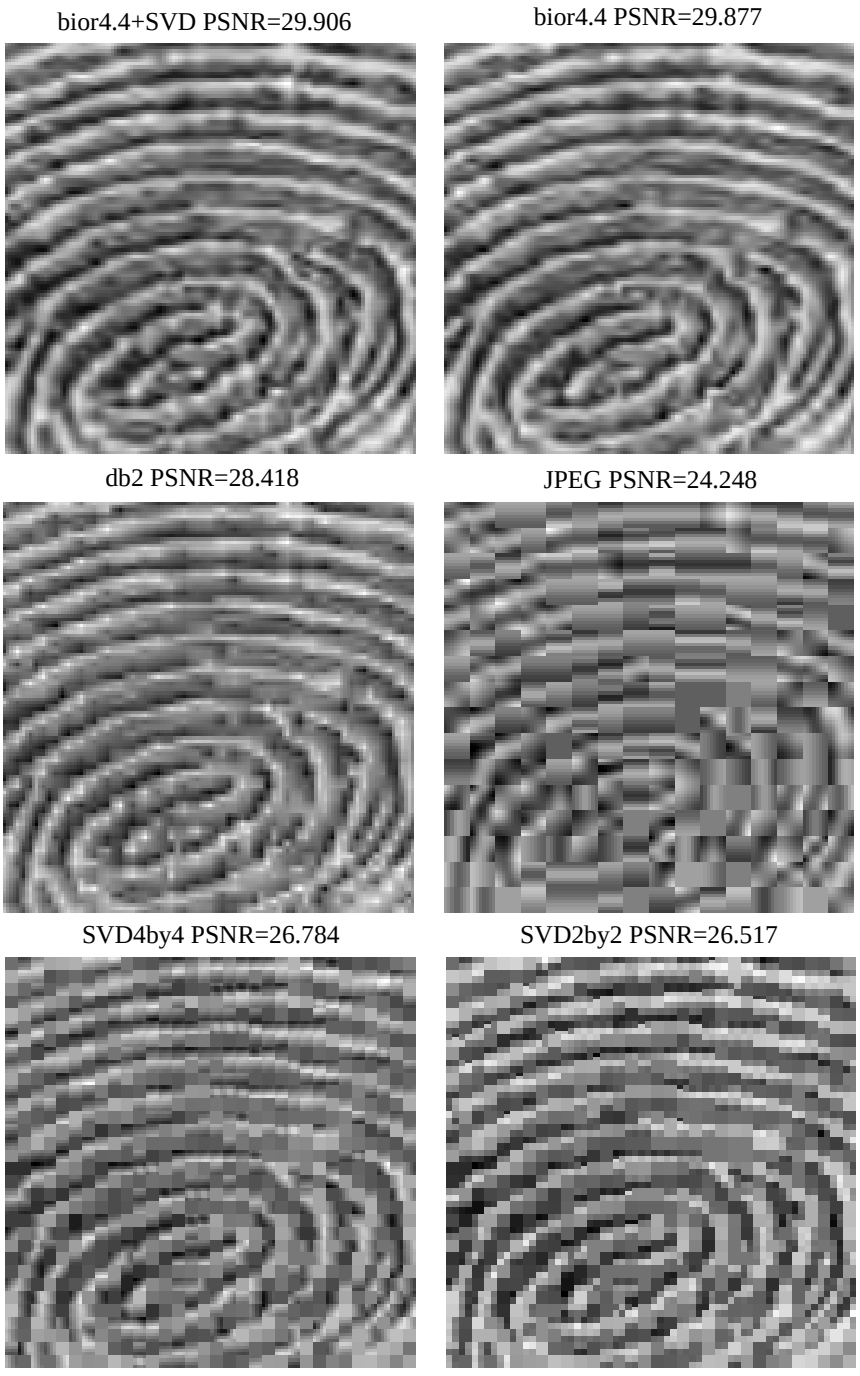

Figure 5: Compressed fingerprint image at 0.15 bpp for `bior4.4+SVD`, `bior4.4`, `db2`, `jpeg`, `4by4SVDMR` and `2by2SVDMR`.

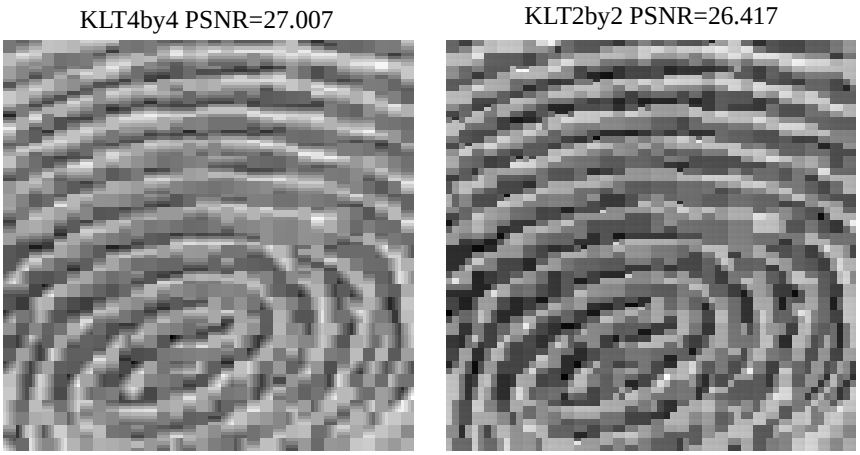

Figure 6: Compressed fingerprint image at 0.15 bpp for `4by4KLTMR` and `2by2KLTMR`.

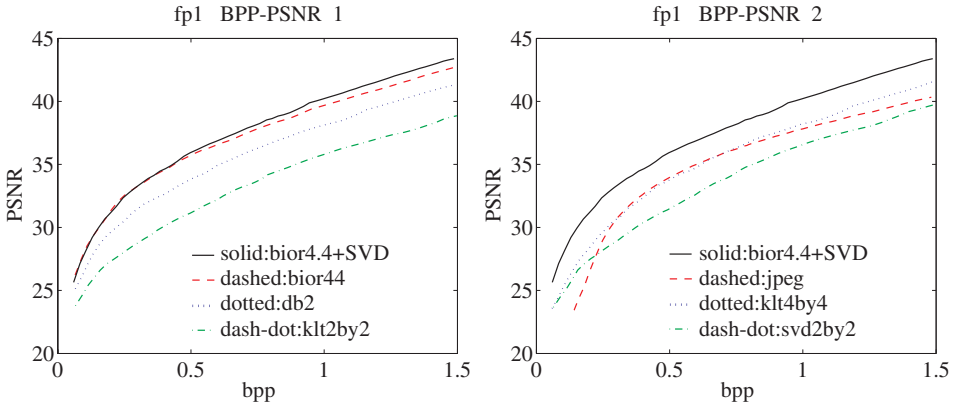

Figure 7: PSNR curve against bpp. Top, for `bior4.4+SVD`, `bior4.4`, `db2`, `klt2by2`; bottom `bior4.4+SVD`, `jpeg`, `klt4by4`, and `svd2by2`.