IMPLEMENTING THE FREQUENCY FILTERING INTO A DICTIONARY IMAGE

Gabriela Dudek[†], Przemyslaw Borys

Department of Physical Chemistry and Technology of Polymers
Silesian University of Technology
Ks. M. Strzody 9, 44-100 Gliwice, Poland

(Received June 10, 2011)

In our previous research we have proposed new method for image compression which was based on LZ77 dictionary algorithm. We introduced two modifications such as inaccurate color matching and noise acceptance. Experimental results presented in that paper proved that the new method of image compression gives promising results as compared with original LZ77 dictionary algorithm. In this paper, we propose to supplement the previous algorithm with frequency content reduction. To this end we propose an interpolation based scaling and a compression along the vertical axes. The obtained results are compared to the previously proposed method.

DOI:10.5506/APhysPolB.42.1837
PACS numbers: 07.05.Pj

1. Introduction

Data compression [1, 2] has always been important and it becomes even more popular and important nowadays. A simple characterization of data compression is that it transforms a string of characters in some representation into a new string (of bits, for example) which contains the same information but whose length is as small as possible. Data compression has important applications in the areas of data transmission [3] and data storage [4]. Many data processing applications require storage of large volumes of data, and the number of such applications is constantly increasing as the use of computers extends to new disciplines. At the same time, the proliferation of computer networks is resulting in massive data transfer along the communication lines. Compressing data to be stored or transmitted reduces storage and/or communication costs.

[†] gmdudek@.polsl.pl

There are "lossless" and "lossy" forms of data compression. Lossless compression algorithms [5, 6, 7] are used to squeeze down images and then restore them again for viewing completely unchanged. The classic lossless compressed image file scheme was the Compuserve Graphics Interchange Format (GIF) [8], which used a variation of the lossless LZW compression algorithm [9]. GIF had a few limitations. One of them was that it could not compress images with color resolutions greater than 8 bits. This was not a great problem with monochrome images or with simple non-realistic block color images like diagrams, but it was troublesome with color photographs, which are normally 24 bits and had to be reduced in color. To overcome the mentioned drawbacks, another lossless compression scheme was proposed under the name PNG (Portable Network Graphics). It is based on the deflate compression algorithm and image filtering for the prediction of pixel values [10]. It can be applied to images with 24-bit or 48-bit color resolutions, and provide many other handy features. In average, PNG provides about 15% better compression ratios than GIF.

Lossy compression [11, 12, 13], in contrast, is based upon the assumption that the data does not have to be stored exactly. Much information can be simply thrown away from images, video data, and audio data, and when uncompressed such data can appear indistinguishable, from the original, for a human user. Compression ratios can be an order of magnitude greater than those available from lossless methods.

The Joint Photographic Experts Group (JPEG) [14, 15, 16] specification has become the practical standard for storing realistic still images using lossy compression. JPEG can generally give much higher compression ratios than GIF while retaining good image quality. Useful JPEG compression ratios are typically in the range of about 10:1 to 20:1. JPEG allows to obtain higher compression ratios on images with high pixel resolutions, because the degradation is not as noticeable on a big image.

2. Method

The method of image compression presented in the previous paper is based on the LZ77 compression algorithm with two modifications: imperfect color matching and noise acceptance [17]. In the present paper, we introduce the compression along vertical axes and the concept of scaling. JPEG algorithm works so well because of reduction in the high frequency coefficients in a FFT expansion. This corresponds to a low pass filtering of the image, which can (at least qualitatively) be seen as a decrease in the sampling rate. In our approach, scaling at the compression stage and interpolation at decompression were designed to mimic this behavior in a dictionary algorithm (where we cannot apply the Fourier transform directly). The quality of such approach is assessed by the PSNR measurements.

Comparison of the results was done by using the Lena image shown in Fig. 1, which was chosen as a reference point.

Fig. 1. The image of Lena.

2.1. LZ77 image compression with imperfect color matching and noise acceptance

Our previous compression method [17] was based on LZ77 dictionary algorithm extended with imperfect color matching and noise acceptance, procedures.

The imperfect color matching is a process in which we attribute the encountered pixel value to a given sample in the dictionary. We do not reduce the histogram of an image to a smaller spectrum of colors but merely set a condition for color similarity $\hat{x}$ and x reads [18, 19]

$$\text{quantization} \geq |\hat{x} - x|, \tag{1}$$

where $\hat{x}$ — stored value and x — encountered value.

The quantization parameter has great impact on the quality of restored images, as well as on the degree of their compression. Of course, when compression ratio is high due to the quantization (PSNR is small), the decompressed Lena image looks worse than in cases of low compression ratio (the level of quantization is smaller then).

The idea of noise acceptance is related to an observation that the repeating sequences of boxes can be loaded with some percentage of errors. In such case, a classical algorithm would encode them as two separate strings. In our approach, the sequences of boxes which do not differ much in the number of incorrect levels, are recognized the same, and recorded as callback to the sequence stored in the dictionary.

If $\lambda(a,b)$ is the function which measures the number of differences in the blocks and $l(a)$ is the length of the block "a", an acceptance condition can be formulated as follows

$$\frac{\lambda(a,b)}{l(a)} < \text{ noise ratio}\,. \tag{2}$$

In such a way noise ratio is introduced to image which enlarges the compression of image considerably, but also influences its quality. For small values of noise ratio there is no influence on PSNR and compression ratio. There is a specific point of which the degree of image compression starts to grow.

2.2. Algorithm with scaling factor and compression along both the horizontal and vertical axes

The algorithm considered in our previous paper [17] contributed to the classical LZ77 compression algorithm with quantization and noise levels. In this method the compression is performed by looking up repeating sequences in a string made of concatenated horizontal image lines. The dictionary sequences were aligned horizontally within the image. In the present method, we compress the image both horizontally and vertically. Moreover, we introduce scaling of the image. Four main classes of image scaling are possible: grayscale–grayscale, binary–binary, binary–grayscale (typically downscaling), grayscale–binary (typically upscaling) [20].

We model the original set of pixels as a continuous interpolated surface (bicubic interpolation) which we re-sample according to the assumed scaling factor.

3. Results

The goal of this paper was to check whether the proposed alterations to our previous method of image compression [17] improve the compression and retain the image's quality. In the compression procedure we first added only the imperfect color matching and noise level to the LZ77 dictionary algorithm. Next, we introduced the vertical compression, and finally — the image scaling. We inspected the influence of these parameters on the compression level and on the visual quality of decompressed Lena image.

3.1. The influence of vertical axes on the compression level

The compression along vertical axes has positive impact on compression. Comparing the level of compression for image of Lena when we only compress along horizontal axes with compression along both horizontally and vertically we can increase compression ratio about 3 percent. We present this influence for the image of Lena in Fig. 2. As can be seen the decompressed image of Lena is looking alike in both cases for the same initial conditions (noise ratio:2, quant accuracy:5). In this case the value of PSNR are the same (about 34.5 dB).

(a) (b)

Fig. 2. The decompressed image of Lena for (a) previous method [17] — compression ratio:3.2; PSNR:34.5 dB, (b) new method — compression ratio:3.3; PSNR: 34.6 dB (quant accuracy:5, noise ratio:2).

3.2. The influence of image scaling on the level of compression

For the Lena image the values of PSNR and compression ratio in relation to the scaling factor are presented in Tables I (a) and II (b). As can be seen from these tables the image scaling increases the compression ratio compared to the previous algorithm. This happens, of course, at cost of quality, as measured by PSNR.

The plot of relationship between image scaling and compression ratio is shown in Fig. 3. By increasing the scaling factor we can obtain a difference in the compression ratio between the previous [17] and new compression algorithms (Table I). Supplying the LZ77 compression algorithm with the image scaling and noise levels, keeping quantization level equals to 0 (no quantization) the compression gain becomes a quadratic function of scaling coefficient. After introducing inaccurate color matching the relation changes to linear. The gain grows slower than quadratic. Quantization seems thus to improve the compression to larger extent at high resolution than at lower resolution. This may be a result of color averaging with increased scaling.

TABLE I

PSNR and compression ratio (a) *versus* quantization (noise ratio is 2.0), (b) noise ratio (quantization is 2.0) — for previous [17] and new method.

(a)

Interpolation	Quant accuracy	PSNR [dB]		Compression ratio		Comparing the compression
		previous method	new method	previous method	new method	
1.2	0	-	27.7	0	1.4	1.4
	5	34.5	27.5	3.2	4.4	1.4
	10	30.1	26.6	6.8	8.6	1.3
	15	27.6	25.6	10.3	13.5	1.3
	20	25.4	24.1	15.1	18.3	1.2
	25	24.0	23.4	18.4	23.4	1.3
	30	22.9	21.6	21.4	28.6	1.3
	35	21.7	20.3	28.6	36.7	1.3
1.4	0	-	27.0	0	2.0	2.0
	5	34.5	26.8	3.2	5.6	1.7
	10	30.1	26.0	6.8	10.7	1.6
	15	27.6	25.2	10.3	16.1	1.6
	20	25.4	23.8	15.1	21.4	1.4
	25	24.0	22.7	18.4	28.6	1.5
	30	22.9	21.7	21.4	32.1	1.5
	35	21.7	20.2	28.6	42.8	1.5
1.6	0	-	26.6	0	2.6	2.6
	5	34.5	26.4	3.2	6.8	2.1
	10	30.1	25.8	6.8	12.8	1.9
	15	27.6	24.9	10.3	19.8	1.9
	20	25.4	23.9	15.1	25.7	1.7
	25	24.0	22.3	18.4	36.7	2.0
	30	22.9	21.8	21.4	42.8	2.0
	35	21.7	20.6	28.6	51.4	1.8
1.8	0	-	26.3	0	3.3	3.3
	5	34.5	26.1	3.2	8.3	2.6
	10	30.1	25.5	6.8	15.1	2.2
	15	27.6	24.6	10.3	23.4	2.3
	20	25.4	23.3	15.1	32.1	2.1
	25	24.0	22.5	18.4	36.7	2.0
	30	22.9	21.4	21.4	42.8	2.0
	35	21.7	19.9	28.6	64.2	2.3
2.0	0	-	26.0	0	4.1	4.1
	5	34.5	25.8	3.2	9.5	2.9
	10	30.1	25.2	6.8	18.3	2.7
	15	27.6	24.1	10.3	25.7	2.5
	20	25.4	23.2	15.1	36.7	2.4
	25	24.0	22.2	18.4	42.8	2.3
	30	22.9	21.2	21.4	51.4	2.4
	35	21.7	19.9	28.6	64.2	2.3

(b)

Interpolation	Noise ratio	PSNR [dB]		Compression ratio		Comparing the compression
		previous method	new method	previous method	new method	
1.2	0	40.3	27.7	1.5	2.1	1.4
	2	40.3	27.7	1.5	2.1	1.4
	7	40.3	27.7	1.5	2.1	1.4
	10	40.3	27.6	1.5	2.2	1.5
	15	36.1	27.6	1.6	2.3	1.4
	20	31.9	26.6	1.9	2.6	1.4
	25	30.2	25.9	2.0	2.8	1.4
	30	28.8	25.8	2.2	3.0	1.4
	35	27.4	24.7	2.4	3.3	1.4
1.4	0	40.3	26.9	1.5	2.8	1.9
	2	40.3	26.9	1.5	2.8	1.9
	7	40.3	26.9	1.5	2.8	1.9
	10	40.3	26.8	1.5	2.9	1.9
	15	36.1	26.6	1.6	3.0	1.9
	20	31.9	25.8	1.9	3.4	1.8
	25	30.2	25.4	2.0	3.6	1.8
	30	28.8	24.2	2.2	4.0	1.8
	35	27.4	24.2	2.4	4.4	1.8
1.6	0	40.3	26.6	1.5	3.5	2.3
	2	40.3	26.6	1.5	3.5	2.3
	7	40.3	26.6	1.5	3.5	2.3
	10	40.3	26.5	1.5	3.6	2.4
	15	36.1	26.3	1.6	3.7	2.3
	20	31.9	25.6	1.9	4.2	2.2
	25	30.2	25.3	2.0	4.5	2.3
	30	28.8	24.8	2.2	4.8	2.2
	35	27.4	24.1	2.4	5.4	2.3
1.8	0	40.3	26.2	1.5	4.4	2.9
	2	40.3	26.2	1.5	4.4	2.9
	7	40.3	26.2	1.5	4.4	2.9
	10	40.3	26.2	1.5	4.4	2.9
	15	36.1	25.9	1.6	4.6	2.9
	20	31.9	25.2	1.9	5.1	2.7
	25	30.2	24.8	2.0	5.5	2.8
	30	28.8	24.2	2.2	5.8	2.6
	35	27.4	23.4	2.4	6.4	2.7
2.0	0	40.3	25.9	1.5	5.2	3.5
	2	40.3	25.9	1.5	5.2	3.5
	7	40.3	25.9	1.5	5.2	3.5
	10	40.3	25.9	1.5	5.2	3.5
	15	36.1	25.7	1.6	5.5	3.4
	20	31.9	24.8	1.9	6.1	3.2
	25	30.2	24.6	2.0	6.4	3.2
	30	28.8	24.2	2.2	6.8	3.1
	35	27.4	23.4	2.4	7.8	3.3

3.3. The visual inspection of the decompressed Lena image dependence on image scaling and compression along both axes

The compression ratio can be increased by introducing the compression along both axes and scaling. To inspect this, we compress the Lena image to different values of the scaling factor with horizontal and vertical buffer look ups. The PSNR values and the compression ratio are presented in Tables II (a) and II (b). The compression along vertical axes allows to improve the compression ratio by about 3 percent. The additional profit is that the PSNR values stay the same (the quality of the encoded images are the same).

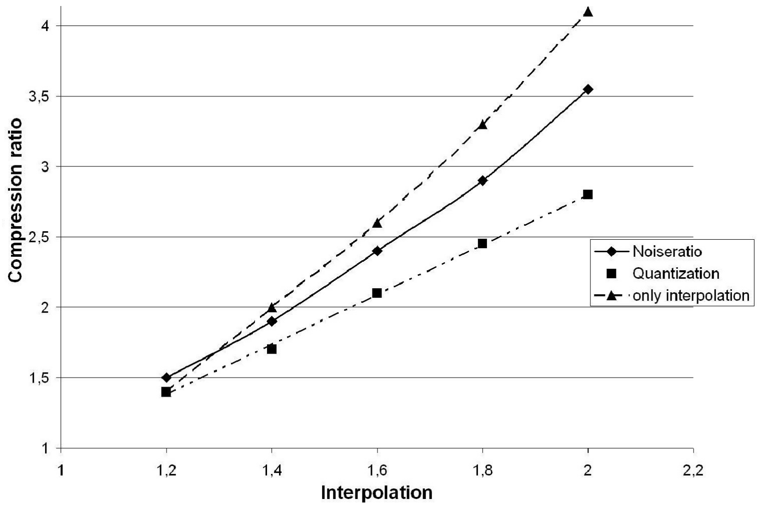

Fig. 3. Scaling factor *versus* compression ratio and influence of noise ratio and quantization to this relation.

For visual inspection we present the decompressed images of Lena in Fig. 4. In all cases the quantization accuracy equals 10 and noise ratio is 2.

As can be seen from these figures, the larger scaling coefficients slightly influence the visual effect of the decompressed Lena image. This effect is not as pronounced as in the case of large quantization levels because the quantization changes the decompressed image very much.

3.4. Comparison with JPEG2000 compression method

In order to check the quality of our new compression method, we compared this one with JPEG2000 chosen as a reference point. During compression the quality was varied from 90 to 2. For each quality PSNR was calculated to check how does the JPEG2000 method influence the decompressed image. Next, we compared the JPEG2000 compression effect against our method. For comparison we have chosen three decompressed images in which the compression ratios were similar (and equal about 10). The result of comparison between JPEG2000 method, our previous and new method is presented respectively in Fig. 5.

From the results of this study we conclude that our new modified LZ77 Image Compression (with imperfect color matching, noise acceptance, compression along both axes and interpolation) offers comparable (as measured by PSNR) image quality to our previous method (with imperfect color matching and noise acceptance only) but improves the visual impression

TABLE II

PSNR and compression ratio *versus* (a) quantization (b) noise ratio for previous [17] and new method — interpolation and both axes.

(a)

Interpolation	Quant accuracy	PSNR [dB]		Compression ratio		Comparing the compression
		previous method	new method	previous method	new method	
1.2	0	-	27.7	0	1.4	1.4
	5	34.5	27.5	3.2	4.4	1.4
	10	30.1	26.6	6.8	9.2	1.4
	15	27.6	25.6	10.3	14.3	1.4
	20	25.4	24.1	15.1	19.8	1.3
	25	24.0	23.4	18.4	25.7	1.4
	30	22.9	21.6	21.4	32.1	1.5
	35	21.7	20.3	28.6	36.7	1.3
1.4	0	-	27.0	0	2.0	2.0
	5	34.5	26.8	3.2	5.7	1.8
	10	30.1	26.0	6.8	10.7	1.6
	15	27.6	25.2	10.3	17.1	1.7
	20	25.4	23.8	15.1	23.4	1.6
	25	24.0	22.7	18.4	32.1	1.8
	30	22.9	21.7	21.4	36.7	1.7
	35	21.7	20.2	28.6	42.8	1.5
1.6	0	-	26.6	0	2.6	2.6
	5	34.5	26.4	3.2	6.9	2.1
	10	30.1	25.8	6.8	13.5	2.0
	15	27.6	24.9	10.3	21.4	2.1
	20	25.4	23.9	15.1	28.6	1.9
	25	24.0	22.3	18.4	36.7	2.0
	30	22.9	21.8	21.4	42.8	2.0
	35	21.7	20.6	28.6	51.4	1.8
1.8	0	-	26.3	0	3.3	3.3
	5	34.5	26.1	3.2	8.6	2.6
	10	30.1	25.5	6.8	16.1	2.4
	15	27.6	24.6	10.3	23.4	2.3
	20	25.4	23.3	15.1	32.1	2.1
	25	24.0	22.5	18.4	42.8	2.3
	30	22.9	21.4	21.4	51.4	2.4
	35	21.7	19.9	28.6	64.2	2.2
2.0	0	-	26.0	0	4.1	4.1
	5	34.5	25.8	3.2	9.9	3.0
	10	30.1	25.2	6.8	18.3	2.7
	15	27.6	24.1	10.3	25.7	2.5
	20	25.4	23.2	15.1	32.1	2.1
	25	24.0	22.2	18.4	51.4	2.8
	30	22.9	21.2	21.4	64.2	2.3
	35	21.7	19.9	28.6	64.2	2.3

(b)

Interpolation	Noise ratio	PSNR [dB]		Compression ratio		Comparing the compression
		previous method	new method	previous method	new method	
1.2	0	40.3	27.7	1.5	2.1	1.4
	2	40.3	27.7	1.5	2.1	1.4
	7	40.3	27.7	1.5	2.1	1.4
	10	40.3	27.6	1.5	2.2	1.5
	15	36.1	27.6	1.6	2.4	1.5
	20	31.9	26.6	1.9	2.7	1.5
	25	30.2	25.9	2.0	3.0	1.5
	30	28.8	25.8	2.2	3.3	1.5
	35	27.4	24.7	2.4	3.7	1.5
1.4	0	40.3	26.9	1.5	2.8	1.9
	2	40.3	26.9	1.5	2.8	1.9
	7	40.3	26.9	1.5	2.8	1.9
	10	40.3	26.8	1.5	2.9	1.9
	15	36.1	26.6	1.6	3.1	1.9
	20	31.9	25.8	1.9	3.6	1.9
	25	30.2	25.4	2.0	3.9	1.9
	30	28.8	24.2	2.2	4.3	1.9
	35	27.4	24.2	2.4	4.8	2.0
1.6	0	40.3	26.6	1.5	3.6	2.4
	2	40.3	26.6	1.5	3.6	2.4
	7	40.3	26.6	1.5	3.6	2.4
	10	40.3	26.5	1.5	3.6	2.4
	15	36.1	26.3	1.6	3.9	2.4
	20	31.9	25.6	1.9	4.5	2.4
	25	30.2	25.3	2.0	4.8	2.4
	30	28.8	24.8	2.2	5.2	2.4
	35	27.4	24.1	2.4	5.2	2.4
1.8	0	40.3	26.2	1.5	4.4	2.9
	2	40.3	26.2	1.5	4.4	2.9
	7	40.3	26.2	1.5	4.4	2.9
	10	40.3	26.2	1.5	4.5	3.0
	15	36.1	25.9	1.6	4.7	2.9
	20	31.9	25.2	1.9	5.5	2.9
	25	30.2	24.8	2.0	6.0	3.0
	30	28.8	24.2	2.2	6.4	2.9
	35	27.4	23.4	2.4	7.3	3.0
2.0	0	40.3	25.9	1.5	5.4	3.6
	2	40.3	25.9	1.5	5.4	3.6
	7	40.3	25.9	1.5	5.4	3.6
	10	40.3	25.9	1.5	5.4	3.6
	15	36.1	25.7	1.6	5.7	3.6
	20	31.9	24.8	1.9	6.4	3.4
	25	30.2	24.6	2.0	6.9	3.4
	30	28.8	24.2	2.2	7.6	3.4
	35	27.4	23.4	2.4	8.6	3.6

perceived by the operator. In spite of the great visual improvement of the image in our new method, JPEG2000 compression method still looks better.

4. Concluding remarks

In this paper we compared the previously proposed method of image compression in which we added imperfect color matching and noise acceptance to the new method in which we introduced compression along vertical axes and scaling.

The previous method allowed to obtain compression ratios up to 90. Introducing compression along vertical axes, we increased the compression ratio by about 3 percent. The second modification, the image scaling is an obvious way to enlarge the compression. For bigger scaling it is possible to obtain a greater compression. For the scaling factor equal to 2 we could

(a)

(b)

(c)

Fig. 4. The decompressed image of Lena when the scaling factor equals (a) 1.2 — compression ratio:9.2; PSNR:26.6 dB, (b) 1.6 — compression ratio:10.7; PSNR: 26.0 dB, (c) 2.0 — compression ratio:18.3; PSNR:25.2 dB (guant accuracy:10, noise ratio:2).

increase compression by over 4 times (Table II). However, the visual effect of decompressed images was worse for higher compression ratios (Fig. 4).

The scaling factor increases the difference between previous [17] and new compression method. For LZ77 compression algorithm with the image scaling and noise levels the compression ratio *versus* scaling coefficient is a quadratic function (Fig. 3). Adding inaccurate color matching to other modifications changes the kind of graph from quadratic to linear. This effect could be caused by an interference between interpolation and compression with inaccurate color matching. Quantization seems thus to improve the

(a) (b) (c)

Fig. 5. The decompressed image of Lena when compression ratio is about 10 for (a) JPEG2000 compression method PSNR:40.5 dB, (b) our previous method of compression PSNR:27.6 dB, (c) our new compression method PSNR:26.6 dB.

compression to larger extent at high resolution than at lower resolution. This may be a result of color averaging with increased scaling.

After introduction of compression along vertical axes and scaling, we could obtain a large compression improvement compared to the previous method. These compression ratios were obtained at cost of the visual effect of the decompressed image (Fig. 4). This adverse impact was more pronounced in the previous compression method than in a new one. Comparing the visual image quality in our new method with JPEG2000 compression method, JPEG still gave the best results, but the PSNR difference (32 percent) is not enormous and an improvement is possible with more sophisticated approaches to scaling and interpolation.

The authors would like to thank the Silesian University of Technology for providing financial support under the project No GG-105/RW2/2010.

REFERENCES

[1] D. Salomon, *Data Compression: The Complete Reference*, third ed., Springer, 2004.

[2] K. Sayood, *Introduction to Data Compression*, second ed., Academic Press, Morgan, Kaufmann Publishers, San Francisco 2000.

[3] S. Benedetto, E. Biglieri, *Principles of Digital Transmission: With Wireless Applications*, Springer, 2008.

[4] EMC, *Information Storage and Management: Storing, Managing, and Protecting Digital Information*, Wiley, 2009.

[5] Lihong Zhao, Yanan Tian, Yonggang Sha, Jinghua Li, *Frontiers of Electrical and Electronic Engineering in China* **4**, 1 (2009).

[6] C. Luigi, D.A. Sergio, L. Luca, *Math. Comput. Sci.* **3**, 391 (2010).

[7] D. Bing, Y. ZhongFu, *Exp. Astron.* **27**, 19 (2009).

[8] CompuServe Incorporated, *Graphics Interchange Format*, Columbus, Ohio, 1990.

[9] J. Ziv, A. Lempel, *IEEE Trans. Inf. Theory* **23**, 337 (1977).

[10] G. Roelofs, *PNG: The Definitive Guide*, O'Reilly Media, 1999.

[11] F. Marcelloni, M. Vecchio, *Evolutionary Intelligence* **3**, 137 (2010).

[12] H. Grailu, M. Lotfizad, H. Sadoghi-Yazdi, *Internationa Journal on Document Analysis and Recognition (IJDAR)* **12**, 227 (2009).

[13] J. Rasmusson, J. Ström, T. Akenine-Möller, *The Visual Computer* **26**, 17 (2010).

[14] Lifang Yu, Yao Zhao, Rongrong Ni, Zhenfeng Zhu, *Soft Computing* **13**, 393 (2009).

[15] L. Weihai, Y. Yuan, Y. Nenghai, *Signal Processing* **89**, 1821 (2009).

[16] L. Qingzhong *et al.*, *Information Sciences* **180**, 1643 (2010).

[17] G. Dudek, P. Borys, Z.J. Grzywna, *Image and Vision Computing* **25**, 883 (2007).

[18] M.N. Nasrabadi, R.A. King, *IEEE Trans. Commun.* **36**, 957 (1988).

[19] N. Akrout, R. Prost, R. Goutte, *Image and Vision Computing* **12**, 627 (1994).

[20] J.D. Foley, *Computer Graphics: Principles and Practice*, Addison-Wesley, second edition, 1990.