# Image Compression Methodology Based on Fuzzy Transform

Petr Hurtik and Irina Perfilieva

Centre of Excellence IT4Innovations division of the University of Ostrava Institute for Research and Applications of Fuzzy Modeling
30. dubna 22, 701 00, Ostrava, Czech Republic
{petr.hurtik,irina.perilieva}@osu.cz

**Abstract.** The main objective of our work is to develop an effective algorithm for image compression. We use both lossy and non-lossy compression to achieve best result. Our compression technique is based on the direct and inverse fuzzy transform (F-transform), which is modified to work with dynamical fuzzy partition. The essential features of the proposed algorithm are: extracting edges, automatic thresholding, histogram adjustment. The article provides a comparison of our algorithm with the image compression algorithm (JPEG) and other existing algorithms [1, 7] based on fuzzy transform.

## 1 Introduction

By image compression we mean a reduction in size of the image with the purpose to save space and by this, a transmission time. Digital images are usually identified with their intensity functions which, being measured in the interval $[0,1]$, can be represented by fuzzy relations. Therefore, in the literature on fuzzy sets and their applications, a continuously growing interest to the problems of image compression was expected. However, this was not the case. Below, we will give a short overview of main ideas which influenced a progress in image compression on the basis of fuzzy sets.

A pioneering publication of Lotfi A. Zadeh [10] discussed the issue of data summarization and information granularity. It has been noticed that a max – min - composition with a fuzzy relation works as a summarization/compression tool. Then in a series of papers (see [2, 3]), the idea to associate image compression with the theory of fuzzy relation equations was intensively investigated. The correspondence between a quality of reconstruction and a $t$-norm in a generalized max $-t$ - composition with a fuzzy relation was analyzed in [3, 4]. A new idea which influences a further progress in fuzzy based image compression came with the notion of F-transform [5]. In [1], it has been shown that the F-transform based image compression is better than the best possible fuzzy relation based one. However, the former was still worse than JPEG technique. A certain improvement of the F-transform based image compression was announced in [6].

A new wave of interest to the discussed problem came with more sophisticated applications of the F-transform to image processing, especially to the problem of edge detection [9]. It has been noticed that the quality of reconstructed image strongly depends on the quality of reconstructed edges. This idea is elaborated in details in the proposed contribution.

Á. Herrero et al. (Eds.): Int. JointConf. CISIS'12-ICEUTE'12-SOCO'12, AISC 189, pp. 525–532.
springerlink.com 

## 2 Compression

Image compression means a reduction in size of the image. By image we mean a discrete function $f$ with two variables which is defined on the domain $[1,N] \times [1,M]$ and takes values from [0, 255]. The value $f(x,y)$ characterizes intensity of the gray level of the pixel whose coordinates are $(x,y)$. Below, we will refer to $f$ as to intensity function or image. By compression we mean a certain transformation of $f$ which results in a new image function $f'$ defined on $[1,N'] \times [1,M']$ where $N' < N, M' < M$. A compression is characterized by its ratio $CR$ which is equal to $N'M'/NM$. We have to solve two problems: reduce size of compressed image and obtain decompressed image most similar to original one.

We propose a compression alorithm which is based on the discrete F-transform in combination with saving sharp edges. This algorithm consists of the following steps: find and store information about gradient (section 2.1); compute range of intensity over an image block and make a desicion regarding further partition of this block (section 2.2); compress by the F-transform (section 2.3) and store histogram of the original image (section 2.4).

Let us make a short overview of some contemporary techniques used for compression. The idea to partition an image area into blocks according to respective ranges of the intensity function is taken from png graphics format. Representation of a compressed image by the result of a certain transform is usual for the JPEG format. Modern compression algorithms use several transforms: discrete wavelet transform, discrete cosine transform, Burrows-Wheeler transform and many others.

In our approach, we combine both lossy and nonlossy compression - gradient pixels are stored by nonlossy format, areas by lossy F-transform. We propose decompression of an image after compression. Decompression is the inverse transformation with respect to compression, it means that we transform $N' \times M'$ back into $N \times M$.

### 2.1 Image Gradient Separation

Gradient separation (or edge detection) is the first step of the image compression algorithm. The notion of edge is informally characterized as an area where a significant change of intensity occurs. In practice, this characterization connects edges with ares where first or second derivative of intensity function $f$ attains its extremal value. In our approach, we take the above given characterization literally and propose to classify an edge area on the basis of a difference $g$ between maximal and minimal values of intensity function $f$ over it:

$$g(x,y) = max(f(x',y')) - min(f(x',y')) \tag{1.1}$$

$$x' \in \{x-1, x, x+1\}; y' \in \{y-1, x, y+1\}.$$

The area with high values of the difference $g$ is not a subject of compression. Due to this fact, a sharpness of a reconstructed image is as good as in the original one. The proposed approach is sensitive to noise, more than if partial derivatives are computed by e.g., Sobel operators. The result of the gradient separation algorithm is shown in fig 1. We propose fixed mask matrix $3 \times 3$ pixels. In order to reduce that kind of sensitivity,

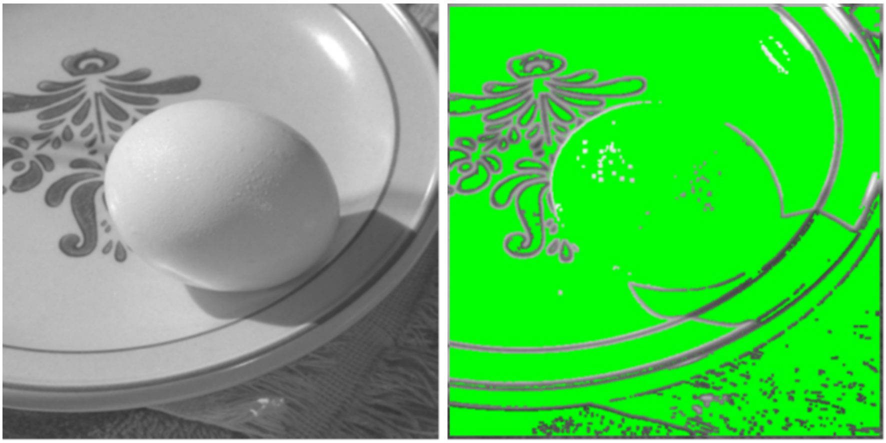

**Fig. 1.** Left: original picture. Right: pixels with hight difference value

we propose to use a dynamic threshold $T$ for selecting high values of the difference $g$. Due to a space limitation, we will skip a detailed description of choosing $T$.

### 2.2 F-Transform

Below, we shortly recall the basic facts about one-dimensional the F-transform [5]. For simplicity, we apply it to a function $f$ of one variable defined on $[a,b]$: let $x_1 < ... < x_n$ be fixed nodes within $[a,b]$. We say that fuzzy sets $A_1,...,A_n$, identified with ther memership functions $A_1(x),...,A_n(x)$ defined on$[a,b]$ form a fuzzy partition of $[a,b]$ if they fulfil the following conditios for $i = 1,...,n$ are fulfilled:

1. $A_i : [a,b] \to [0,1], A_i(x_i) = 1$;
2. $A_i(x) = 0$ if $x \notin (x_{i-1}, x_{i+1})$, where we assume $x_0 = x_1 = a$ and $x_{n+1} = x_n = b$;
3. $A_i(x)$ is a continuous function on $[a,b]$;
4. $A_i(x)$ stricly increases on $[x_{i-1}, x_i]$ for i= $2,...,n$ and stricly decreases on $[x_i, x_{i+1}]$ for $i = 1,...,n-1$.
5. $A_i(x)$ stricly increases on $[x_{i-1}, x_i]$ for i= $2,...,n$ and stricly decreases on $[x_i, x_{i+1}]$ for $i = 1,...,n-1$.

$$F_i = \frac{\sum_{j=1}^{m} f(p_j) A_i(p_j)}{\sum_{j=1}^{m} A_i(p_j)} \tag{1.2}$$

for $i = 1,...,n$. Shapes of basic functions are not prederminited, so that we use triangular membership functions due to simplicity of coding. Following inverse F-transform is then defined by:

$$f_{F,n}(p_j) = \frac{\sum_{i=1}^{n} F_i A_i(p_j)}{\sum_{i=1}^{n} A_i(p_j)} \tag{1.3}$$

In our case, we are using two-dimensional F-transform described in [9]. Let us remark that in the above given characterization of a fuzzy partition, we did not use the Ruspini condition. By this, we obtain a certain flexibility in choosing a partition.

### 2.3 Evaluation of Intensity Range in Area

Image compression algorithm is usually applied to smaller subareas. The main problem is finding size of those areas. For instance, if we have large area of one color, but with some small detail of different color, we have two options: we can compress it as one area, but the detail will be lost. Or we can divide the are with the detail into smaller areas in order tol keep that small detail. In the last case, we have to memorize many small areas of one color. We propose to solve this problem by using the F-transform with a non-uniform partition which is chosen on the basis of the following procedure. Each area $E$ is characterized by its width $E_w$ and its height $E_h$; At the beginning of algorithm we set $E_w = N; E_h = M$. The values of function $g$ are computed on the basis of (1) where $x' \in [max(0, x - E_w), x]; y' \in [max(0, y - E_h), y]$. If $g(x,y) \leq D$, $D \in [0,255]$ and means user defined threshold for control algorithm power we choose the respective area $E$ as an element of the partition of the F-transform. Owerthise, we divide area $E$ into four symmetrical subareas and continue recursively. Dividing is terminated if the condition of minimal difference $D$ is true, or the condition of minimal area is true:

$$(E_w \leq S \vee E_h \leq S) \vee g(x,y) \leq D. \tag{1.4}$$

In (4) $S$ means minimal size (of width, or height) of basic funcions and $D$ means threshold of minimal intensity difference. These two values $S$ and $D$ are defined by a user, and both of them influence power the of compression algorithm. The result of the divide algorithm with red colored borders of an areas is shown in fig 2.

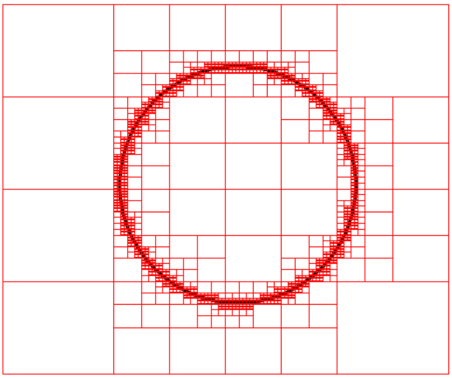

**Fig. 2.** Example of dividing the area

### 2.4 Image Histogram

The F-transform based compression is lossy and therefore the histogram of an original image is changed after decompression. We propose to store the histogram of the original image and apply it to obtain a better reconstruction of compressed image. We compute the cumulative distribution function, say $C$ as a characteristic of the histogram and store the respective vector of values of $C$. This additional vector is added to the stored information about the compressed image. As a results, the quantity of the stored information increases so that the proposed earlier compression ratio does not fully characterize a size of the stored information. To take into account the actual a size of the stored information we increase the compression ratio by the respective quantity. For example: if an image has the dimension $512 \times 512$ px then we increase compress ratio value by 0.0009.

## 3 Decompression

The decompression is a transform from $M'N'$ space back to $MN$ space. We propose the decompression algorithm based on the inverse F-transform (3). Because an application of the direct and inverse F-transform leads to the lossy decompression, our goal is to minimize data loss. We propose to minimize the loss by decompression of the stored gradient pixels (chap. 3.1) and histogram restore (chap. 3.2).

### 3.1 Decompression of Gradient Pixels

The area with high values of the difference $g$ (see Section 2.1) is added to the image reconstructed by the inverse F-transform. We have to put pixels from this area into their own layer above the currently decompressed layer. After that we can merge layers hierarchicly.

### 3.2 Histogram Restore

After applying the F-transform and its inverse the range of intensity changes. In order to restore the range of intensity of an original image in the reconstruction, we use the

**Fig. 3.** Left: without histogram restore, PSNR = 29dB. Right: with histogram restore, PSNR = 30dB.

**Fig. 4.** Left: original; right: proposed reconstruction, CR=0.08, PSNR=29dB

**Fig. 5.** Left: proposed reconstruction, CR=0.25, PSNR=37dB; right: proposed reconstruction, CR=0.44, PSNR=43dB

**Fig. 6.** Left: JPEG, CR=0.25, PSNR=39dB; right: JPEG, CR=0.43, PSNR=46dB

stored information about the cumulative distribution function $C$. This step allows to increase the quality of reconstruction. For example, there is figure 3 for comparison between image with and without histogram restore.

## 4 Estimation of a Quality of Reconstruction

The following criterion is used for estimation of a quality of a reconstructed image. *PSNR* (Peak Signal to Noise Ratio) measures a similarity between an original image and its reconstruction after. Higher value of *PSNR* means better quality of result.

$$PSNR = 20log\left(\frac{max(f)}{\sqrt{MSE}}\right)[dB] \tag{1.5}$$

$$MSE = \frac{1}{M \cdot N}\sum_{x=0}^{M-1}\sum_{y=0}^{N-1}(f(x,y) - q(x,y))^2$$

By $max(f)$ we mean the is maximum value of the intensity of the original image $f$. By $q$ we mean the means intensity value of the decompressed image.

## 5 Experiments

In Figures 4 and 5, we demonstrate the results of the proposed technique. In Figure 6, we show the results of the JPEG algorithm with the same compression ratio (*CR*). For the chosen benchmark ”Cameraman” from the Coral Gallery, the JPEG is slightly better. However, for the created by us picture in Fig. 2, the proposed algorithm shows better results that JPEG algorithm. In order to [1] you can see, that the results of the proposed algorithm are slightly better that the previous one.

## 6 Conclusion

We have proposed a new compression method on the basis of the F-transform. In comparison with the previous one [7], the newly proposed compression uses the following improvements: edge extraction, dynamic area division described by non-uniform partition and histogram adjustment. Our next research will be focusing on large-size images, color images, estimation of a speed of our algorithm and detailed comparison with the JPEG technique.

**Acknowledgement.** This work was parially supported by the European Regional Development Fund in the IT4Innovations Centre of Excellence project (CZ.1.05/1.1.00/02.0070) and the project SGS12/PrF/2012.

## References

1. Di Martino, F., Loia, V., Perfilieva, I., Sessa, S.: An image coding/decoding method based on direct and inverse fuzzy transforms. International Journal of Approximate Reasoning 48, 110–131 (2008)
2. Hirota, K., Pedrycz, W.: Fuzzy relational compression. IEEE Trans. Syst., Man, Cybern. 29(pt. B), 407–415 (1999)
3. Nobuhara, H., Pedrycz, W., Hirota, K.: Fast Solving Method of Fuzzy Relational Equation and Its Application to Lossy Image Compression/Reconstruction. IEEE Trans. Fuzzy Syst. 8(3), 325–334 (2000)
4. Pedrycz, W., Hirota, K., Sessa, S.: A Decomposition of Fuzzy Relations. IEEE Trans. Syst., Man, Cybern. 31(pt. B), 657–663 (2001)
5. Perfilieva, I.: Fuzzy transforms. Fuzzy Sets and Systems 157, 993–1023 (2006)
6. Perfilieva, I., De Baets, B.: Fuzzy Transform of Monotonous Functions with Applications to Image Processing. Information Sciences 180, 3304–3315 (2010)
7. Perfilieva, I., Pavliska, V., Vajgl, M., De Baets, B.: Advanced image compression on the basis of fuzzy transforms. In: Proceedings of the Conference on IPMU 2008, Torremolinos, Malaga, Spain (2008)
8. Perfilieva, I.: Fuzzy transforms of monotone functions with application to image compression. Information Sciences 180, 3304–3315 (2010)
9. Perfilieva, I., Hodakova, P., Hurtik, P.: F1-transform Edge Detector Inspired by Cannys algorithm. IPMU, Italy (accepted, 2012)
10. Zadeh, L.: Fuzzy sets and information granularity. In: Gupta, M., Ragade, R., Yager, R.R. (eds.) Advances in Fuzzy Set Theory and Applications, pp. 3–18. North Holland, Amsterdam (1979)
11. Ziv, J., Lempel, A.: A Universal Algorithm for Sequential Data Compression. IEEE Transactions on Information Theory 23(3), 337–343 (1977)