Wavelet-Based Image Compression with Polygon-Shaped Region of Interest

Yao-Tien Chen[1], Din-Chang Tseng[1], and Pao-Chi Chang[2]

[1] Department of Computer Science and Information Engineering,
National Central University, Chung-li, Taiwan
{ytchen, tsengdc}@ip.csie.ncu.edu.tw
[2] Department of Communication Engineering, National Central University, Chung-li, Taiwan
pcchang@ce.ncu.edu.tw

Abstract. A wavelet-based lossy-to-lossless image compression technique with polygon-shaped *ROI* function is proposed. Firstly, split and mergence algorithms are proposed to separate concave *ROIs* into smaller *convex ROIs*. Secondly, row-order scan and an adaptive arithmetic coding are used to encode the pixels in *ROIs*. Thirdly, a lifting integer wavelet transform is used to decompose the original image in which the pixels in the *ROIs* have been replaced by zeros. Fourthly, a wavelet-based compression scheme with adaptive prediction method (*WCAP*) is used to obtain predicted coefficients for difference encoding. Finally, the adaptive arithmetic coding is also adopted to encode the differences between the original and corresponding predicted coefficients. The proposed method only needs less shape information to record the shape of *ROI* and provides a lossy-to-lossless coding function; thus the approach is suitable for achieving the variety of *ROI* requirements including polygon-shaped *ROI* and multiple *ROIs*. Experimental results show that the proposed lossy-to-lossless coding with *ROI* function reduces bit rate as comparing with the *MAXSHIFT* method in *JPEG2000*; moreover, when the image without *ROI* is compressed by the proposed lossless coding, the proposed approach can also achieve a high compression ratio.

Keywords: Image compression, region of interest (*ROI*), lossy-to-lossless coding, *ROI* coding, difference encoding.

1 Introduction

Image compression is used to reduce the image data size as small as possible under a tolerance limit of errors. In general, the techniques of image compression can be classified into two major categories: loss and lossless. Lossy compression requires not only less storage space, but also less transmission time or bandwidth, while lossless compression can completely reconstruct the original data. In addition to offering high-quality compression, an effective approach to image compression should further incorporate value-adding functions, such as *ROI* coding and lossy-to-lossless coding. A *ROI* refers to a special region in an image that is of particular interest or imperative importance to the user who can free to identify the *ROI* based on ones needs. In

L.-W. Chang, W.-N. Lie, and R. Chiang (Eds.): PSIVT 2006, LNCS 4319, pp. 878 – 887, 2006.

general, an image can be separated into important and non-important parts for a particular purpose; the important part represented by the *ROI* is often compressed by a lossless style while the non-important part can be compressed by a lossy-to-lossless style to achieve a tradeoff between the fidelity and the coding efficiency. That is, the *ROI* undergoes lossless compression first and then finer details of the remaining part of the image are gradually added at a later stage to achieve lossy-to-lossless coding.

A lot of different techniques of *ROI* coding have been proposed recently. Li and Li [1] proposed shape adaptive discrete wavelet transform (*SA-DWT*) for arbitrarily shaped object coding. With the use of the transform, the spatial correlation and wavelet transform properties, such as locality property and self-similarity across subbands, are preserved. Tasdoken and Cuhadar [2] proposed region-based integer wavelet transform (*RB-IWT*) as an alternative to *SA-DWT*. The *RB-IWT* enables lossless coding of image regions which can not be achieved by *SA-DWT* due to the fixed-precision representation of wavelet coefficients. Fukuma *et al*. [3] introduced a switching wavelet transform by using shorter-length basis for *ROI* and longer-basis for non-*ROI*. The bases with different lengths provide better compression quality than a fixed-length wavelet transforms. Liu *et al*. [4] proposed a method for chromosome image compression which combines lossless compression of chromosome *ROIs* with lossy-to-lossless coding for the remaining image parts. The method performs a differential operation on chromosome *ROIs* for decorrelation, and is followed by integer wavelet transforms on *ROIs* and the remaining image parts. The boundary of chromosome *ROIs* are then traced by chain code method.

The model of *ROI* coding supported in *JPEG2000* is based on scaling the wavelet coefficients. The technique can be further classified into two different methods: general scaling-based and *MAXSHIFT* method [5]. For a general scaling-based method, a shape encoder/decoder is required to encode/decode the shape information (*i.e.*, the shape of *ROI*). This makes both encoder and decoder more complicated and increases the bit rate; moreover, the method needs a *ROI* mask indicating which wavelet coefficients have to be transmitted exactly in order for the receiver to reconstruct the desired region perfectly. In contrast, the *MAXSHIFT* method does not need the shape information. However, if there are multiple *ROIs* with different degrees of interest, the *MAXSHIFT* method has to handle more difficult problems than a general scaling method, since the dynamic range has to be increased significantly.

To solve the mentioned problems, we propose a wavelet-based image compression technique with polygon-shaped *ROI* and lossy-to-lossless coding. Firstly, split and mergence algorithms are proposed to separate concave *ROIs* into smaller *convex ROIs*. Secondly, row-order scan and an adaptive arithmetic coding are used to encode the pixels in *ROIs*. Thirdly, a lifting integer wavelet transform is used to decompose the original image in which the pixels in the *ROIs* have been replaced by zeros. Fourthly, a wavelet-based compression scheme with adaptive prediction method (*WCAP*) is used to obtain predicted coefficients for difference encoding. Finally, the adaptive arithmetic coding is also adopted to encode the differences between the original and corresponding predicted coefficients. We only need less shape information to achieve the polygon-shaped *ROI* and multiple *ROIs* with different degrees of interest. Furthermore, the proposed approach does not need to generate the *ROI* mask.

The remaining sections of this paper are organized as follows. In Section 2, we present the proposed approaches: split and mergence algorithms, lifting integer wavelet transform, and the *WCAP* method. Experiments are reported in Section 3. Conclusions are given in Section 4.

2 The Proposed Approach

The block diagram of the proposed approach shown in Fig. 1 is composed of seven processes: *ROI* selection, *Graham*'s scan algorithm [6], split and mergence algorithms, row-order scan, lifting integer wavelet transform, *WCAP* method [7], and adaptive arithmetic coding.

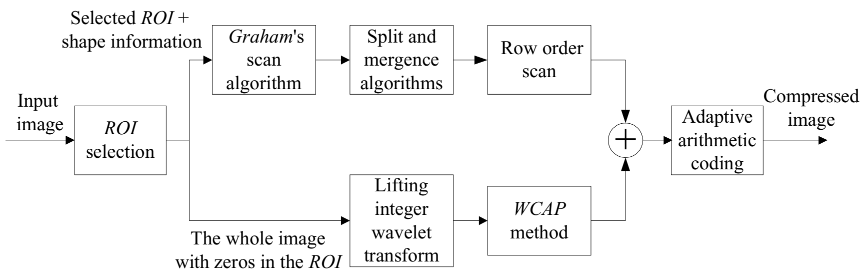

Fig. 1. The block diagram of the proposed approach

ROIs are always formed by polygons or circles. When a *ROI* is formed by a polygon, the *ROI* is represented by the *ordered vertices* of the polygon and the *ordered vertices* just constitute the shape information. A polygon R in an image is called convex if line segment $\overline{ab}$ for any pair of pixels a, b in R is completely in R. The *convex hull* of a polygon R is the smallest *convex polygon* containing R and represented by its *convex vertices*. Examples of polygon-shaped *ROI* and *convex polygon* are given in Fig. 2. In Fig. 2 (a), a polygon-shaped *ROI* is represented by *ordered vertices*: P_1, P_2, P_3, P_4, P_5, P_6, P_7, and P_8, where P_i, $1 \le i \le 8$, are denoted by coordinates in the image. In Fig. 2 (b), a *convex polygon* is represented by corresponding *convex vertices*: P_1, P_2, P_3, P_4, and P_5. On the other hand, when the *ROI* is formed by a circle, the shape of *ROI* is represented by the center point and radius of the circle. Similarly, the center point and radius just constitute the shape information.

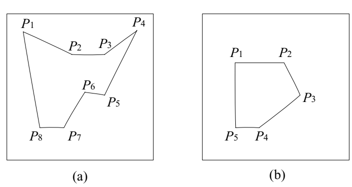

Fig. 2. Examples of polygon-shaped *ROIs* and *convex polygons*. (a) A polygon-shaped *ROI* and the *ordered vertices*. (b) *A convex polygon* and the *convex vertices*.

2.1 Split and Mergence Algorithms

Given the shape information, both the encoder and the decoder use split and mergence algorithms to split the concave *ROI* into multiple *convex ROIs*. Then we can simply use row-order scan to extract all coefficients in all shaped *ROIs*. Hence, the encoder just needs to transmit very little shape information to the decoder, and the decoder can perfectly reconstruct the *ROIs*. The split and mergence algorithms consist of three steps: (*i*) judging whether a given *ROI* is convex, (*ii*) exploiting split algorithm to separate the *ROI* into multiple non-overlapped *convex ROIs* if necessary, and (*iii*) merging two or more *ROIs* to constitute a larger *convex ROI* if the mergence still satisfies the condition of *convex hull*.

A given *ROI* is convex if each *ordered vertex* is exactly scanned once by *Graham*'s scan algorithm, and these vertices must form a *convex ROI*; otherwise, split algorithm will be used to achieve the necessary condition of *convex hull*. Split algorithm is a recursive approach dividing a *ROI* into multiple non-overlapped smaller *convex ROIs* step by step. The algorithm is divided into the following four steps:

Step 1. Identify the *convex ROI* by tracing vertex one after one in the given *ordered vertices*. If any current vertex violates the condition of *convex hull*, a smaller *convex ROI* will be successfully split. Meanwhile, the algorithm pushes a vertex prior to the current vertex into a *temporary queue*.
Step 2. The split is processing until the given *ordered vertices* is empty, and one or more *convex ROIs* are obtained upon the completion of this step.
Step 3. All vertices in the *temporality queue* are ejected to compose new *ordered vertices* for subsequent steps.
Step 4. Repeat from *Step*1 until no *ROI* to be split.

After all *non-convex ROIs* are split, mergence algorithm is then performed to merge adjacent *convex ROIs* into larger convex *ROIs* by the following two steps:

Step 1. Any two adjacent *convex ROIs* (*i.e.*, there exists a common boundary between the two *ROIs*) is merged to generate a larger *convex ROI* if they satisfy the condition of *convex hull*.
Step 2. The result of *Step* 1 is regarded as the input of mergence algorithm for the next step until no two adjacent *ROIs* can be further merged.

Illustration of the split and mergence algorithms is shown in Fig. 3. An original *ROI* composed of *ordered vertices*: P_1, P_2, P_3, P_4, P_5, P_6, P_7, P_8, P_9, P_{10}, and P_{11} is given in Fig. 3 (a). The first splitting *convex ROI* (P_2, P_3, P_4) is shown in Fig. 3 (b); then four *convex ROIs*, (P_2, P_3, P_4), (P_4, P_5, P_6), (P_6, P_7, P_8), and (P_8, P_9, P_{10}, P_{11}), are successively split as shown in Fig. 3 (c). Meanwhile, the *ordered vertices* (P_1, P_2, P_4, P_6, P_8, P_{11}) is pushed into the *temporary queue* and regarded as an input *ROI* in the second step as shown in Fig. 3 (d). Two *convex ROIs*, (P_1, P_2, P_4) and (P_4, P_6, P_8, P_{11}), are split as shown in Fig. 3 (e). The *ordered vertices* (P_1, P_4, P_{11}) forms a *convex ROI* in this step; thus the split algorithm stops and mergence algorithm starts as shown in Fig. 3 (f). Two *convex ROIs*, (P_1, P_2, P_4) and (P_1, P_4, P_{11}), are merged to constitute (P_1, P_2, P_4, P_{11}) as shown in Fig. 3 (g). Two *convex ROIs*, (P_4, P_5, P_6) and (P_4, P_6, P_8, P_{11}), are merged to constitute (P_4, P_5, P_6, P_8, P_{11}) as shown in Fig. 3 (h).

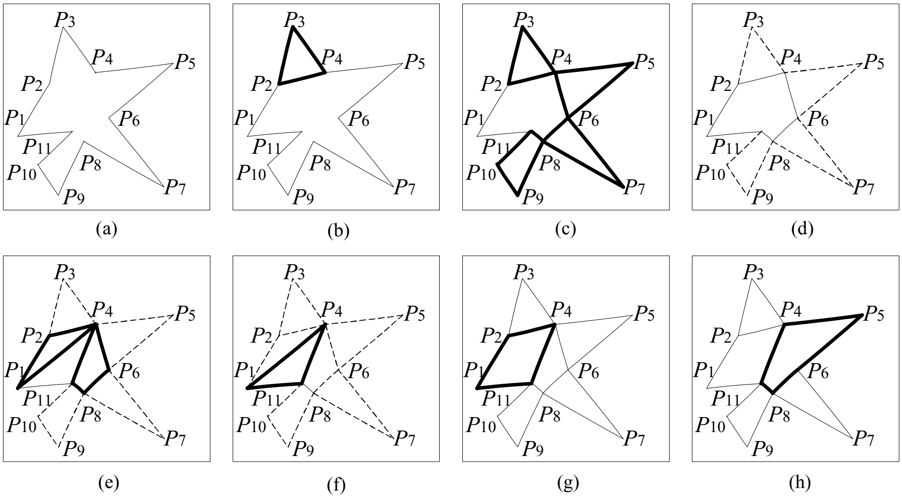

Fig. 3. Illustration of the split and mergence algorithms. (a) Original *ROI*. (b) The first *convex ROI* is split. (c) Four *convex ROIs* are obtained after the first step. (d) (P_1, P_2, P_4, P_6, P_8, P_{11}) forms an input *ROI* in the second step. (e) Two *convex ROIs*, (P_1, P_2, P_4) and (P_4, P_6, P_8, P_{11}), are obtained after the second step. (f) (P_1, P_4, P_{11}) forms a *convex ROI*. (g) (P_1, P_2, P_4) and (P_1, P_4, P_{11}) are merged to generate (P_1, P_2, P_4, P_{11}). (h) (P_4, P_5, P_6) and (P_4, P_6, P_8, P_{11}) are merged to constitute (P_4, P_5, P_6, P_8, P_{11}).

For each non-overlapped *convex ROI*, *convex vertices* are linked to form the boundary of the *convex ROI* and then row-order scan is adopted to scan all coefficients in the multiple *convex ROIs*. Thus we can precisely identify all coefficients in the non-overlapped *convex ROI*. For circular-shaped *ROI*, shape information is directly used to identify all coefficients in the *ROI* without using split and mergence algorithms. Thus we demonstrates that the proposed approach indeed only needs less shape information to achieve the *ROI* coding as compared with the conventional shape coding methods.

The advantage of the *MAXSHIFT* method surpassed to the general scaling-based method is that the *ROI* coding does not need shape information at the decoder. However, generating the *ROI* mask still remains at the encoder, and the *ROI* mask calculation is complicated. Thus the computational complexity of these two methods is relatively higher than that of split and mergence algorithms. Since the proposed approach does not need to generate the *ROI* mask and is easy to implement, it is more efficient for lossless compression with *ROI* selection.

2.2 Lifting Wavelet Transform

Our wavelet transform was implemented by lifting scheme performed with the following three steps: (*i*) *split* step for sorting the input into the even and the odd entries, (*ii*) *prediction* step for giving the value at the even entries and predicting the value at the odd entries, and (*iii*) *update* step for updating even entries up to date to reflect knowledge of the input. Lifting integer wavelet transform means that the wavelet transform can transform integers to integer coefficients and perfectly reconstruct the original integers from the integer coefficients. Lifting integer wavelet transform is capable of accomplishing fast in-place computation and especially appropriate for lossless data compression. A variety of transforms can be applied for lossless data compression. Nevertheless, according to the suggestions of the previous work [7], *S+P* transform is generally considered to be the best one. The *S+P* transform is described as

$$d^{(1)}[n] = x[2n+1] - x[2n],$$
$$s[n] = x[2n] + \left\lfloor \frac{d^{(1)}[n]}{2} \right\rfloor, \tag{1}$$

and

$$d[n] = d^{(1)}[n] + \left\lfloor \frac{2}{8}(s[n-1] - s[n]) + \frac{3}{8}(s[n] - s[n+1]) + \frac{2}{8}d^{(1)}[n+1] + \frac{1}{2} \right\rfloor,$$

where [.] is a notation of signal, $d[n]$ and $s[n]$ are the highpass and lowpass coefficients respectively after the transform, $x[.]$ denotes the original signal, and $\lfloor . \rfloor$ is a truncation operator.

2.3 WCAP Method

The *WCAP* method was proposed by Chen *et al.* [7]. Initially, the method analyzes the higher-correlation coefficients, where wavelet coefficients are regarded as the predictor (independent) and response (dependent) variables of a prediction equation. Then based on the higher-correlation coefficients, the method launched the selection of predictor variables using a conditional statistical test to determine which relative predictor variables should be included in the prediction equation. The generated prediction equations are then applied to predict most wavelet coefficients except the lowest-resolution coefficients.

In most previous studies, the predictions were generally conducted with a fixed number of predictor variables at fixed locations. Actually, every kind of images not only has its own statistical distribution but also demonstrates different properties in different wavelet subbands. To achieve a more accurate prediction for compression, the number of predictor variables must be adaptively adjusted based on the image's properties. Thus instead of relying on a fixed number of predictors on fixed locations, the *WCAP* method uses adaptive prediction approach to overcome the *multicollinearity* problem and takes the wavelet interscale persistence and intrascale clustering properties to achieve high compression ratio.

In general, the probability distribution of the symbols to be encoded is unknown. Thus a method called adaptive arithmetic coding [8] which is combined from an adaptive probability estimation and an arithmetic coding is pursued to increase compression ratio. Adaptive arithmetic coding uses a real number to represent a sequence of symbols and updates the probability of symbol based on distribution of input symbol, whenever getting one input symbol. Finally, compression of images is achieved via this adaptive arithmetic coding.

3 Experiments

In our experiments, all test images are 512×512 gray-level images as shown in Fig. 4. The *ROIs* are manually selected. If there are *ROIs* selected, the *ROIs* will be encoded by lossless style and the remaining parts are encoded by lossy-to-lossless style; otherwise, the entire image will be encoded by lossless style. At first, the polygon-shaped *ROIs* and progressive lossy-to-lossless coding were examined to demonstrate the abilities of the proposed approach as shown in Fig. 5 (a). In Fig. 5 (b), a *ROI* was selected on the *lena*'s face and partial hat for lossless encoding. The remaining part of the image was gradually added to reconstruct the original image. As indicated from Figs. 5 (c) to (i), the remaining part was divided into eight bitplanes and starts bitplane encoding from most significant bit (*MSB*) to least significant bit (*LSB*) to achieve progressive lossy-to-lossless coding.

The comparison of bit rates among adaptive arithmetic coding, *MAXSHIFT* method, and the proposed approach is shown in Table 1. From the table, we find that the proposed approach has the best compression rate for all six standard images. To understand the improved degree of the proposed approach over other methods in compression rate, we here define an improvement ratio (*IR*) to evaluate the improvement of compression performance for method *B* over method *A* as

$$IR = \frac{\text{bitrate of method } A - \text{bitrate of method } B}{\text{bitrate of method } A} \times 100\% . \tag{2}$$

Here, the *MAXSHIFT* method was adopted to evaluate the improvement ratio of the proposed approach. One polygon-shaped *ROI* was selected on each of the six test images; the *ROI* approximately covers 15 - 25% of the images. The improvement ratios are given in Table 1. From the table, we find that the improvement ratios of the proposed method over the *MAXSHIFT* method are approximately 2.01 - 6.02%.

(a) *Lena* (b) *Goldhill* (c) *Boat*

(d) *Barbara* (e) *Baboon* (f) *Airplane*

Fig. 4. Six test gray-level images

(a) Original image (b) *ROI* (c) Bitplane 1~2

(d) Bitplane 1~3 (e) Bitplane 1~4 (f) Bitplane 1~5

(g) Bitplane 1~6 (h) Bitplane 1~7 (i) Bitplane 1~8

Fig. 5. Progressive lossy-to-lossless coding with polygon-shaped *ROI*

Table 1. The comparison of bit rates among adaptive arithmetic coding, *MAXSHIFT* method, and the proposed approach

Method / Image	*Adaptive arithmetic coding*	*MAXSHIFT method*	*The proposed approach*	Improvement ratio
Lena	4.86	4.65	4.49	3.44%
Goldhill	5.44	5.23	5.01	4.21%
Boat	5.02	4.65	4.37	6.02%
Barbara	5.71	5.19	4.95	4.62%
Baboon	6.62	6.48	6.35	2.01%
Airplane	4.62	4.04	3.81	5.69%

The compressions without *ROI* selection were also examined. The comparison of the proposed method with other lossless coding techniques: *CALIC* [9] and *JPEG2000* are given in Table 2. From the table, we can find that the proposed approach offers a better performance than the other two methods. The improvement ratios of the proposed method over the *JPEG2000* and *CALIC* methods are from 7.36 to 10.7% and from 1.69 to 5.46%, respectively.

As indicated by the above experiments, the main contributions of the proposed approach are to offer the high-performance polygon-shaped *ROI* coding and progressive lossy-to-lossless coding. Moreover, the proposed approach is also superior to *JPEG2000* and *CALIC* methods for lossless compression without *ROI* selection.

Table 2. Comparison of lossless compression for *JPEG2000*, *CALIC*, and the proposed approach in bits/pixel

Image \ Method	*JPEG2000*	*CALIC*	*The proposed approach*
Lena	4.33	4.10	4.01
Goldhill	4.85	4.58	4.33
Boat	4.42	4.15	4.08
Barbara	4.81	4.54	4.42
Baboon	5.98	5.66	5.54
Airplane	3.82	3.55	3.44

4 Conclusions

In this paper, a wavelet-based image compression technique with polygon-shaped *ROI* function and lossy-to-lossless coding was proposed. Firstly, split and mergence algorithms were proposed to separate concave *ROIs* into smaller *convex ROIs*. Secondly, row-order scan and an adaptive arithmetic coding were used to encode the pixels in *ROIs*. Thirdly, a lifting integer wavelet transform was used to decompose the original image in which the pixels in the *ROIs* had been replaced by zeros. Fourthly, a wavelet-based compression scheme with adaptive prediction method (*WCAP*) was used to obtain predicted coefficients for difference encoding. Finally, the adaptive arithmetic coding was also adopted to encode the differences between the original and corresponding predicted coefficients.

The proposed approach possesses the following advantages: (*i*) only needing less shape information to reconstruct the *ROIs*, (*ii*) providing a progressive lossy-to-lossless coding, achieving polygon-shaped *ROIs*, and supporting multiple *ROI*s. Thus the proposed approach is suitable for achieving the variety of *ROI* requirements. Experimental results show that the proposed lossy-to-lossless coding with *ROI* is superior to the *MAXSHIFT* method in *JPEG2000*; moreover, for lossless compression without *ROI* selection, the proposed approach has also obtained the best performance.

Now, the *ROIs* are manually selected; further study on automatic determination of *ROIs* will be achieved by integrating the level set methods [10].

References

1. Li, S., Li, W.: Shape-adaptive discrete wavelet transforms for arbitrarily shaped visual object coding. IEEE Trans. Circuits and Systems for Video Technology, vol.10 (2000) 725-743
2. Tasdoken, S., Cuhadar, A.: ROI coding with integer wavelet transforms and unbalanced spatial orientation trees. In Proc. of the 25th Annual Int. Conf. of the IEEE Eng. in Medicine and Biology Society, vol.1 (2003) 841-844
3. Fukuma, S., Tanaka, T., Nawate, M.: Switching wavelet transform for ROI image coding. IEICE Trans. Fund. Electron. Commun. Comput. Sci., vol.E88, (2005) 1995-2005
4. Liu, Z., Xiong, Z., Wu, Q., Wang, Y., Castleman, K.: Cascaded differential and wavelet compression of chromosome images. IEEE Trans. on Biomedical Engineering, vol.49 (2002) 372-383
5. Christopoulos, C., Askelof, J., Larsson, M.: Efficient methods for encoding regions of interest in the upcoming JPEG2000 still image coding standard. IEEE Signal Processing Letters, vol.7 (2000) 247-249
6. Berg, M. de, Kreveld, M. van, Overmars, M., Schwarzkopf, O.: Computational Geometry: Algorithms and Applications. Springer, Berlin (2000)
7. Chen, Y.-T., Tseng, D.-C., Chang, P.-C.: Wavelet-based medical image compression with adaptive prediction. In Proc. of Int. Sym. on Intelligent Signal Processing and Communication Systems, Hong Kong (2005) 825-828
8. Moffat, A.,: Linear time adaptive arithmetic coding. IEEE Trans. Inform. Theory, vol.36 (1900) 401-406
9. Wu, X., Memon, N.,: Context-based lossless interband compression-extending CALIC. IEEE Trans. Image Processing, vol.9, (2000) 994-1001
10. Chan, T. F., Vese, L. A.,: Active contours without edges. IEEE Trans. Image Processing, vol.10 (2001) 266-277