An Edge Preserving Image Resizing Method Based on Cellular Automata

Konstantinos Ioannidis, Ioannis Andreadis, and Georgios Ch. Sirakoulis

Department of Electrical and Computer Engineering,
Democritus University of Thrace, Xanthi, Greece
{kioannid,iandread,gsirak}@ee.duth.gr

Abstract. This paper introduces a novel image resizing method for both color and grayscale images. The method could be beneficial in applications where time and quality of the processed images are crucial. The basic idea of the proposed method relies on preserving the edges by partitioning the digital images into homogenous and edge areas during the enlargement process. In addition, the basic fundamentals of Cellular Automata were adopted in order to achieve better performance both in terms of processing time as well as in image quality. By creating appropriate transition rules, the direction of the edges is considered so that every unknown pixel is processed based on its neighbors in order to preserve the quality of the edges. Results demonstrate that the proposed method improves the subjective quality of the enlarged images over conventional resizing methods while keeping the required processing time in low levels.

Keywords: Image resizing, Color/Grayscale image enlargement, Edge-oriented method, Cellular automata.

1 Introduction

The main objective of the image resizing methods is to generate a high resolution image from its lower resolution version. Digital images and video sequences result in large amount of image data. Efficient manipulation of these types of data in systems with limited technical specifications is a significant issue in their overall performance. Image interpolation techniques are the most commonly adopted methods for image enlargement. Nevertheless, conventional linear interpolation schemes based on space-invariant models fail to preserve the quality of the edges and consequently produce resized images with blurred edges or annoying zigzag artifacts.

Several commonly used interpolation methods have been suggested for image resizing, such as nearest neighbor interpolation [1], bilinear interpolation [1], bicubic interpolation [2] and spline interpolation [3]. Linear approaches are the most frequently applied for the resizing process due to their low computational burden. However, those methods produces image artifacts like blurring on edges since no information related to abrupt changes of pixel values is considered. On the contrary, nonlinear methods produce better results; nevertheless, they appear larger computational burden and involve blurring, as well. Various generic approaches have been proposed to

G.C. Sirakoulis and S. Bandini (Eds.): ACRI 2012, LNCS 7495, pp. 375–384, 2012.

improve the subjective quality of the interpolated images and overcome such deficiencies. In addition, the method in [4] is based on variation models with smoothing and orientation constraints. The nonlinear Partial Differential Equation (PDE) problem is simplified into a series of problems with explicit solutions. Furthermore, the area based interpolation scheme in [5] computes each interpolated pixel by proportional area coverage of a filtering window which is applied to the input image. A quadratic image interpolation method [6] has been proposed with adequate visual results nonetheless its computational cost remains in high levels. Finally, a method to estimate the model parameters piecewisely is proposed in [7] using an autoregressive image model. The method utilizes the covariance matrix of the high resolution image itself, with missing pixels properly initialized.

An alternative type of approaches has been introduced, namely edge-directed interpolation methods, in order to preserve the edges of the low resolution image and produce crisper results. Edge-directed interpolation methods apply a variety of operators according to the edge directions [8]. A fuzzy interpolation approach was proposed in [9] for two dimensional signal resampling, however, additional processing for edge identification is required. In addition, a neural network approach has been proposed in [10] to approximate the computational rules of interpolation algorithms for learning statistical inter-pixel correlation of interpolated images. The method in [11] comprises a hybrid artificial intelligence system. A fuzzy decision system was proposed to classify all the pixels of the input image into human perception nonsensitive class and sensitive class. The bilinear interpolation is applied to the nonsensitive regions while a neural network was used to interpolate the sensitive regions along the edges. Furthermore, the method proposed in [12] initially estimates local covariance coefficients from a low resolution image. These covariance estimates are used to adapt the interpolation at a higher resolution based on the geometric duality between the low-resolution and the high-resolution covariance. Despite the visually accurate resulted images, the above edge directed approaches show high levels of computational cost and thus, their application in real-time systems is restricted. In order to achieve frame rates close to real time limits while enhancing the quality of the edges, an edge-oriented method was proposed in [13]. The main idea is to discriminate the image into homogenous areas and edge areas, which are processed using different interpolation methods. The method achieves real-time image enlargement, nevertheless, the classification of the areas depends on a predefined threshold as well as two stages of process are required. Finally, Shi et al. in [14] initially expand the low resolution image using a bilinear interpolation method and a Canny edge detector [15] is applied to identify the edges of the upscaled image. The final pixel values are calculated by applying some refinement functions. Despite the satisfactory visual results and the high frame rates, the inaccuracies inserted by the initial bilinear enlargement lead to blurred edges and thus, the Canny edge detector is unable to detect significant edges.

In this paper, we introduce an edge-directed method which exploits the simplicity and the inherent parallelism of Cellular Automata [16]. Cellular automata are decentralized space-time systems where interactions are local and can be used to model physical systems. In addition, they consist of identical rectangular cells which results to a regular grid in one or more dimensions. Each cell can be marked with a finite number of states at each evolution step, which are updated synchronously according to a specified transition rule set and the states of its adjacent cells. Due to its finite nature, the cells belonging to the borders of the grid are updated based on the defined boundary conditions, i.e. periodic, fixed or reflection. Cellular automata are extensively used in a variety of applications including numerous image processing methods. In the past, they were exploited to perform pattern reconstruction tasks [17], border detection [18] and noise filtering [19]. In the proposed method, a Cellular automaton (CA) was applied to increase the resolution of digital images. The edges of a low resolution image are initially determined by applying the Canny edge detector leading to a bitwise edge map. This map is then considered as a cell grid along with a cell state which corresponds to the undefined pixel values. The CA evolves its state by applying the appropriate transition rules, which were constructed based on the orientation of the edges. Finally, a simple remapping of each cell state to pixel value is applied, which is based on the weighted summation of the adjacent pixel values. The method manages to preserve the initial edges adequately while achieving high frame rates in both color and in grayscale images. In order to evaluate the performance of the proposed method, a quantitative comparison with other related methods was applied proving its effectiveness.

The rest of the paper is organized as follows. In Section 2, the proposed method is analyzed while in Section 3, the experimental results are provided. Conclusions are drawn in Section 4.

2 Proposed Method

The proposed method aims at calculating the unknown pixel values, which are produced by the resizing process. The basic concept of the method is to initially classify the pixels of the initial image into two categories: homogenous areas and edge areas. The method exploits the capability of the Canny edge detector to accurately determine the edges of the image. Since the bitwise edge map is produced, the logical array is enlarged and is considered as a CA lattice. The unknown cells update their state according to the proposed transition rules. Finally, a simple transformation is applied to calculate the unknown pixel values based on the state of each CA cell. For color images, the above procedure is applied to each of the RGB vector separately. The overall process of the method is illustrated in Fig. 1.

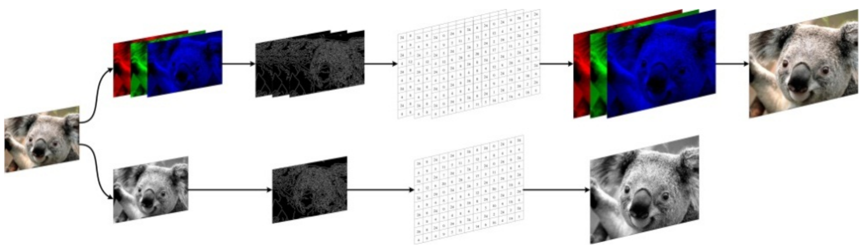

Fig. 1. Proposed CA based image resizing method

2.1 Edge Detection

The first stage of the method is the application of an edge detection technique in order to discriminate the homogeneous areas from the edge areas. Thus, any method reported in the literature could be applied. However, most of the methods which produce more accurate edges display higher computational burden. Therefore, their application in real-time systems is not permissible. In order to produce accurate edge maps with high frame rates, the Canny edge detector was incorporated. The most significant criteria of this selection are as follows:

- Correct detection: edges are detected with high probability when these exist in the real images.
- Accurate localization: marked edges are accurately close to the edges in the real images.
- Minimal response: a defined edge is detected only once, and where possible, noise should not create false edges.

The Canny edge detector is actually an optimal technique of edge detection and creation and its application relies on the criteria above: correct detection, accurate localization and minimal response. To satisfy these requirements, a technique which finds the function, which optimizes a given functional was used, namely the calculus of variations. The method produces binary edge maps by applying sequentially the following processing stages:

Stage 1: Filtering the image
The image is initially filtered using a 2D Gaussian filter of zero mean value and a predefined standard deviation σ in order to eliminate possible noise. The result is a slightly blurred version of the original image.

Stage 2: Defining the intensity gradient of the filtered image
At this stage, elementary edge detection operators, like Sobel, are used in order to define the first derivative both in the horizontal and the vertical direction. Thus, the gradient and direction of each pixel are defined by the following equations:

$$E_g = \sqrt{I_{Gx}^2 + I_{Gy}^2}, \quad E_d = \arctan\left(\frac{I_{Gy}}{I_{Gx}}\right) \tag{1}$$

Stage 3: Non maximum suppression
Given estimates of the image gradients, a search is then applied to determine if the gradient magnitude assumes a local maximum in the gradient direction. A pixel is defined as an edge pixel if its direction is larger than the average direction of its area.

Stage 4: Hysteresis thresholding
The last stage intends to further reduce the number of edge pixels that resulted during the above stages. For this purpose, two thresholds are used. The process starts by applying a high threshold and by using the directional information; edges can be

traced through the image. While an edge is traced, the lower threshold is applied in order to trace faint sections of edges. The most frequent value for the high threshold is considered to be related to the highest value of the gradient magnitude of the image while the low threshold is usually equal to *0.4*(high threshold)*.

Once the process is completed, a binary image is produced where each pixel is marked as either an edge pixel or a non-edge pixel. Essentially, a new image with the same dimensions is produced representing the edges of the initial image.

2.2 CA Resizing

Motivated by the binary nature of the calculated edge maps, a CA is proposed to define the state of the additional cells that resulted after the resizing process and eventually the pixel values Without loss of generality, it is assumed that the high resolution edge map $Y_{i,j}$ of size $2(M \times N)$ directly comes from size $M \times N$. Thus, it yields $Y_{2i,2j} = X_{i,j}$. Fig. 2 gives a schematic illustration of the resulted enlarged edge map.

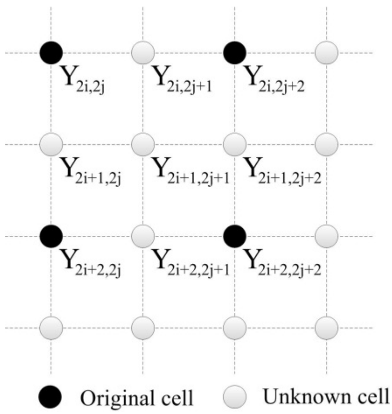

Fig. 2. Edge map after the enlargement

The enlarged edge map is considered to be a *2D* lattice of cells where every binary pixel is represented by a cell. Thus, the proposed CA grid has the same dimensions with the enlarged image. Moreover, in order to update the state of each cell, Moore neighborhood is applied. In addition, the set of states must be defined. Since the enlarged edge map includes non-edge cells (stated as "0"), edge cells (stated as "1") and undefined cells, we assume that the undefined cells are marked with the state "2". In concluding, the cells of the CA before its evolution can be marked with three states: "0" (non-edge cell), "1" (edge cell) and "2" (undefined cell). Taking advantage of the cellular automata flexibility, the transition rules as well as the states of the cells after the evolution are created in order to preserve the edges. The basic motive is to create states that eventually will produce crisper transitions of pixel values from non-edge pixels to edge pixels during the remapping process. By using such states, the orientation of each edge is considered. For example, based on the lattice of Fig. 2, let us assume that the cell $Y_{2i,2j}$ is marked as non-edge while the cell $Y_{2i,2j+2}$ is an edge cell. The intermediate cell $Y_{2i,2j+1}$ must be evolved to a cell that will produce a pixel value, which is closer to the value of the edge pixel and less close to the non-edge pixel. In addition, if no cell in its immediate neighbor is marked as a non-edge cell, its next state must correspond to a pixel value which leads to a homogenous area. It must be mentioned that all non-edge and edge cells of the enlarged map are surrounded by eight unknown cells and, after the evolution, they are marked with states which leads to equal pixel values with the corresponding pixels of the low resolution image. In addition, every cell stated as "2" appears either two (horizontal or vertical cells indexed as $Y_{2i,2j+1}$ and $Y_{2i+1,2j}$, respectively) or four cells (central cell indexed as $Y_{2i+1,2j+1}$) with known states in its Moore neighborhood, which will eventually designate the next state. The total number of the used states and the constructed transition rules is 24 therefore, each cell is marked with one discrete number between the range of [0,23] after the evolution of the CA. Finally, null boundary conditions are used in order to evolve the state of the frontier cells. Fig. 3 represents simple cases of the above rationality.

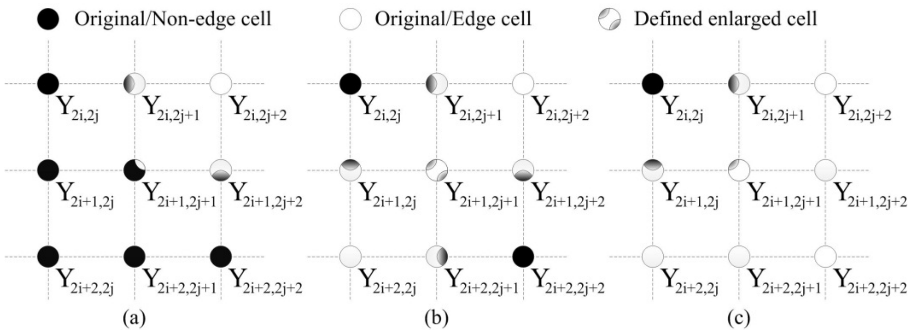

Fig. 3. Examples of the applied CA transition rules

2.3 Remapping Process

At this stage of the process, every pixel value of the resized image is defined based on the state of the corresponding cell. Let us consider that $f_{i,j}$ and $F_{2i,2j}$ corresponds to the low resolution image and the high resolution image, respectively. At the previous stage, cells indexed by ($2i$,$2j$) were marked with states that simply apply the following: $F_{2i,2j} = f_{i,j}$. On the contrary, in order to keep computational cost in low levels, pixels with the indices ($2i$,$2j+1$), ($2i+1$,$2j$) and ($2i+1$,$2j+1$) are expressed as a weighted summation of their adjacent pixel values of the low resolution image. Thus, for the remapping process of these pixels, the following expressions are introduced:

$$F_{2i,2j+1} = a_1 * f_{i,j} + a_2 * f_{i,j+1} \tag{2}$$

$$F_{2i+1,2j} = b_1 * f_{i,j} + b_2 * f_{i+1,j} \tag{3}$$

$$F_{2i+1,2j+1} = c_1 * f_{i,j} + c_2 * f_{i,j+1} + c_3 * f_{i+1,j} + c_4 * f_{i+1,j+1} \tag{4}$$

where a_k and b_k , k = 1,2, corresponds to the applied weights for the $F_{2i,2j+1}$ (horizontal) and $F_{2i+1,2j}$ (vertical) pixel values of the high resolution image, respectively, while c_m , m = 1,..4, depicts the weights applied for the central pixel $F_{2i+1,2j+1}$.

Each of the above weights is defined based on the state of the corresponding cell of the CA grid. In addition, the summation of each of the factors a_k,b_k and c_k must be equal to one. For example, assuming that cell (*2i*,*2j*) is defined as a non-edge cell and cell (*2i*,*2j+2*) as an edge cell (Fig. 3(a)), weight α_2 must be greater than α_1 in order to produce a crisper value transition between the non-edge and the edge pixel. In addition, if both pixels are denoted as non-edge or edge pixels, the weights are equal in order to produce an expanded homogenous or edged area. Based on the case that Fig. 3(c) presents, the cell $Y_{2i,2j+1}$ is surrounded by the non-edge cell $Y_{2i,2j}$ and the edge cell $Y_{2i,2j+2}$. Thus, the cell $Y_{2i,2j+1}$ has been depicted during the CA evolution with a state, which results to the following α_k weights: $\alpha_1 = 0.25$ and $\alpha_2 = 0.75$. In addition, the cell $Y_{2i+1,2j}$ is surrounded by an edge cell $Y_{2i+2,2j}$, and by a non-edge cell, $Y_{2i,2j}$, thus, its state after the CA evolution will correspond to a state where the following values will be used during the mapping process, $b_1 = 0.25$ and $b_2 = 0.75$. Finally, for the central cell $Y_{2i+1,2j+1}$, values $c_1 = 0.1$ and $c_2 = c_3 = c_4 = 0.3$ are applied since cells $Y_{2i,2j}$ and $Y_{2i,2j+1}$, $Y_{2i+1,2j}$, $Y_{2i+1,2j+1}$ are non-edge and edge cells, respectively.

3 Experimental Results

To evaluate the performance of the tested resizing methods including the proposed method, several tested were performed. Zero – order, bilinear, bicubic, the Nedi [12], the edge-oriented [13] and the proposed method were tested on both color and grayscale images of various resolutions. All original images were initially down sample and then up sampled by the same algorithm to meet the initial dimensions. The results shown in Fig. 4 are a comparison of all the applied algorithms for the *Cameraman* image. In order to quantify the effectiveness of every method, the Peak Signal-to-Noise Ratio (PSNR) metric was calculated, as given by:

$$PSNR = 20 * \log_{10}\left(\frac{255}{\sqrt{MSE}}\right) \tag{5}$$

where MSE stands for the Mean-Squared-Error, calculated by:

$$MSE = \frac{1}{M \times N}\sum_{i=0}^{M-1}\sum_{j=0}^{N-1}\left[I(i,j) - F(i,j)\right]^2 \tag{6}$$

where *I(i,j)* is the original image, *F(i,j)* is the approximated version of the image and *M*, *N* are the dimensions of the image respectively. For color images, the definition of PSNR is exact the same, except the MSE is the sum over all squared value differences divided by image size and by three.

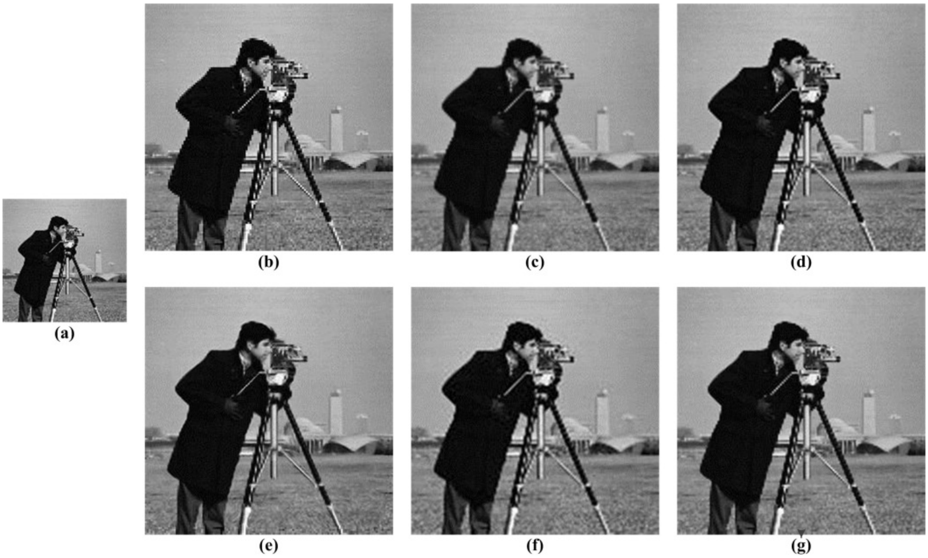

Fig. 4. Resized Cameraman Images: **(a)** Original image, **(b)** Nearest neighbor, **(c)** Bilinear, **(d)** Bicubic, **(e)** Nedi, **(f)** Edge-oriented and **(g)** CA based proposed method

All resulted PSNR values for every tested image are provided in Table 1, while in Table 2, the resulted execution time for each method is demonstrated. As it is depicted in Table 1, the proposed method produces sufficiently high PSNR values. Despite their low processing times, the Nearest-neighbor method produces zigzag artifacts over the edge areas while the bilinear method results blurred edges. In addition, the Bicubic method also produces blurred edges requiring more processing time. The highest PSNR values are produced by the Nedi algorithm nevertheless; the required execution time prohibits its use for real-time applications. In addition, the edge-oriented method produces adequate PSNR values with low processing time, however, it is a threshold dependent method since it is required to determine the edge areas. On the contrary, the proposed method produces sufficiently high PSNR values preserving the edges of the initial image. Moreover, exploiting the inherit parallelism of the CA, the method displays low execution time, making it appropriate for real time systems.

Table 1. Resulted PSNR(dB) values. NN: Nearest; BL: Bilinear; BC: Bicubic; ND: Nedi, EO: Edge – oriented; CA-R: Proposed method

Image	Method					
	NN	**BL**	**BC**	**ND**	**EO**	**CA-R**
Koala (Rgb 256×192)	23.70	25.37	25.14	32.45	29.30	30.32
Cam-man (Gr 128×128)	22.37	23.96	23.70	30.77	25.80	26.54
Lena (Rgb 150×150)	26.93	28.88	28.77	34.57	30.90	31.57
Box (Gr 320×240)	28.68	30.19	29.99	32.1	29.13	30.02
Building (Rgb 640×480)	29.92	31.71	31.73	33.91	30.62	31.58

Table 2. Execution time (msec). NN: Nearest; BL: Bilinear; BC: Bicubic; ND: Nedi, EO: Edge – oriented; CA-R: Proposed method

Image	Method					
	NN	**BL**	**BC**	**ND**	**EO**	**CA-R**
Koala (Rgb 256×192)	3.6	19	21	44860	91.2	84.4
Cam-man (Gr 128×128)	3.1	8.7	9.7	4500	38.2	36.9
Lena (Rgb 150×150)	3.9	13.2	15.9	18600	41.2	39.3
Box (Gr 320×240)	3.6	9.4	10.6	5000	63.2	58.6
Building (Rgb 640×480)	5	26.2	29.5	65290	102.4	99.4

4 Conclusions

In this paper, a new image resizing method based on Cellular Automata and the Canny edge detector was introduced. The Canny edge detector is initially applied in order to descriminate the edge areas from the homogenous areas. The resulted binary edge map is then upscaled and processed as a CA grid. Appropriate CA states and transition rules were constructed to evolve the CA, which eventually attempt to enhance the quality of the edged areas. The orientation of the edge cells is considered in order to preserve effectively the edges of the initial image. Finally, a simple linear transformation is applied to reevaluate the value of each pixel for the final resized image. In terms of quantative comparison based on the PSNR values, the method demonstrates sufficient performance while the required processing time is kept in low levels due to the parallel nature of the CA. The method could be considered as appropriate for systems with low specifications, i.e. low resolution cameras, when further image processing is required.

Rerefences

1. Jain, A.K.: Fundamentals of Digital Image Processing. Prentice-Hall, Upper Saddle River (1978)
2. Keys, R.G.: Cubic convolution interpolation for digital image processing. IEEE Trans. Acoust., Speech, Signal Process. 29, 1153–1160 (1981)
3. Hwang, J.W., Lee, H.S.: Adaptive image interpolation based on local gradient features. IEEE Signal Process. Lett. 29, 359–362 (2004)
4. Jiang, H., Moloney, C.: A new direction adaptive scheme for image interpolation. In: International Conference on Image Processing, Rochester, New York, USA, pp. 369–372 (2002)
5. Amanatiadis, A., Andreadis, I., Gasteratos, A.: A Log-Polar interpolation applied to image scaling. In: IEEE International Workshop on Imaging Systems and Techniques, Cracovia, Poland, pp. 1–5 (2007)
6. Muresan, D., Parks, T.: Adaptively quadratic (Aqua) image interpolation. IEEE Trans. Image Process. 13, 690–698 (2004)
7. Xiong, R., Ding, W., Ma, S., Gao, W.: Improved autoregressive image model estimation for directional image interpolation. In: 28th Picture Coding Symposium, Nagoya, Japan, pp. 442–445 (2010)

8. Cha, Y., Kim, S.: The error-amended sharp edge (EASE) scheme for imaging zooming. IEEE Trans. Image Process. 16, 1496–1505 (2007)
9. Chen, J.L., Chang, J.Y., Shieh, K.L.: 2D discrete signal interpolation and its image resampling application using fuzzy rule-based inference. Fuzzy Sets Syst. 114, 225–238 (2000)
10. Huang, Y., Fan, H.: Learning from interpolated images using neural networks for digital forensics. In: IEEE Conference on Computer Vision and Pattern Recognition, San Francisco, CA, pp. 177–182 (2010)
11. Lin, C.T., Fan, K.W., Pu, H.C., Lu, S.M., Liang, S.F.: An HVS-directed neural network based image resolution enhancement scheme for image resizing. IEEE Trans. Fuzzy Syst. 15, 605–615 (2007)
12. Li, X., Orchard, M.T.: New edge-directed interpolation. IEEE Trans. Image Process. 10, 1521–1527 (2001)
13. Chen, M.J., Huang, C.H., Lee, W.L.: A fast edge-oriented algorithm for image interpolation. Image and Vision Computing 23, 791–798 (2005)
14. Shi, H., Ward, R.: Canny edge based image expansion. In: IEEE International Symposium on Circuits and Systems, Scottsdale, Arizona, USA, pp. 785–788 (2002)
15. Canny, J.: A computational approach to edge-detection. IEEE Trans. Pattern Anal. Mach. Intell. 8, 679–700 (1986)
16. Wolfram, S.: Theory and applications of Cellular Automata. World Scientific, Singapore (1986)
17. Piwonska, A., Seredynski, F.: Discovery by genetic algorithm of Cellular Automata rules for pattern reconstruction task. In: 9th International Conference on Cellular Automata for Research and Industry, Ascoli Piceno, Italy, pp. 198–208 (2010)
18. Popovici, A., Popovici, D.: Cellular Automata in image processing. In: 15th International Symposium on Mathematical Theory of Networks and Systems, Notre Dame, Indiana, pp. 1–6 (2002)
19. Selvapeter, P.J., Hordijk, W.: Cellular Automata for image noise filtering. In: World Congress on Nature & Biologically Inspired Computing, Coimbatore, India, pp. 193–197 (2009)