



Grayscale Image Colorization using Seeded Cellular Automaton

Nherie Lynn U. Celeste

Division of Natural Sciences and Mathematics
University of the Philippines Visayas Tacloban College
Magsaysay Boulevard, Tacloban City, Leyte, Philippines

John Paul T. Yusiong*

Division of Natural Sciences and Mathematics
University of the Philippines Visayas Tacloban College
Magsaysay Boulevard, Tacloban City, Leyte, Philippines

Abstract: Image Colorization, or simply colorization, refers to the computer-assisted process of assigning colors to pixels of a grayscale image in order to convert monochromatic images into visually plausible colored images. Furthermore, adding color to a grayscale image is a very important technique to improve its visual quality. While the traditional method of colorization dealt with segmenting the images into different regions before flooding the segmented areas with colors, another approach in image colorization usually involves drawing colored scribbles on grayscale images. These scribbles will then act as markers or “seeds” that determine the color of a pixel in the image. Nevertheless, colorization is an expensive, tedious and time-consuming process even with current technologies. Thus, several researchers presented different approaches to solve these problems. This research work proposes a scribble-based method of image colorization with the use of the Seeded Cellular Automaton. Experiment results revealed that the Seeded Cellular Automaton successfully produced colored images from grayscale images with acceptable and satisfactory visual qualities.

Keywords: Colorization, Image Segmentation, Ford-Bellman Algorithm, Cellular Automaton, Seeded Cellular Automaton

I. INTRODUCTION

Image colorization, or simply colorization, refers to the computer-assisted process or technique, which is either automatic or user-assisted approach of adding color to a monochromatic image or a grayscale image. Usually, colorization is the process of assigning colors to pixels in an image that enables users to convert grayscale images into visually plausible colored images. This transformation enhances the visual attractiveness of a grayscale image [1,2].

The term was originally introduced by Wilson Markle back in 1970 to describe the process he invented, colorization of black and white movies with the use of a computer [3]. Nevertheless it is presently a generic term used to refer to any technique of adding colors to monochromatic images or films.

In general, image colorization involves segmenting a grayscale image into regions with similar attributes and then applying colors to the segmented regions. However, image colorization is an expensive, tedious and time-consuming process even with current technologies. Thus, several researchers presented different approaches to address these problems.

For instance, Levin, et.al [1] proposed a scribble-based method to convert the colorization process into an optimization problem with a quadratic cost function while Chen and Wang [3] improved the research of Levin, et. al [1] by applying sub-sampling and edge detection, and introducing a new weighting function. The edge detection method reduced color bleeding while the new weighting function minimized computational complexity. Similarly, Yatziv and Sapiro [4] calculated the geodesic distance of the pixels from the colored scribble to determine the future color of those pixels, while Koo and Cho [5] solved the colorization problem by using a soft image segmentation technique based on Random Walks.

II. IMAGE COLORIZATION

Colors influence a person’s perception of visual information. For this reason, adding color to a grayscale image is necessary to enhance the image. This is accomplished by

assigning three-dimensional (RGB) pixel values in a one-dimensional image [5, 6,7].

Also, adding color to a grayscale image is necessary as it improves the visual quality of the image. This in turn increases its visual appeal by providing a much richer semantic meaning. Therefore, the image attains a remarkable visual appearance that elicits stronger emotions [8].

However, several color spaces, also known as color models, exist to represent colors for different applications. A color model is a technique of specifying a color mathematically or numerically. Examples of color models are RGB, HSI, YUV, CIE L*a*b* [9, 10, 11].

As stated in [3,8, 12], existing image colorization researches can be classified into two broad categories: interaction-based method or scribble-based method and example-based method. These methods are also referred to as color propagation method and color transfer method, respectively.

In the scribble-based method, the user draws several colored scribbles on the grayscale image and the colors are automatically propagated to the different pixels, that is, colorization is performed based on a set of manually drawn scribbles on a grayscale image [2, 13]. On the other hand, the example-based method automates the colorization process by providing colored images as reference images. A reference image provides the color information used to colorize a given grayscale image. Nonetheless, propagating colors to the different pixels entails the use of different algorithms [7, 8,12].

In this paper, the researchers implemented a scribble-based technique to convert grayscale images into visually plausible colored images using the Seeded Cellular Automaton segmentation algorithm[14].

III. SEEDED CELLULAR AUTOMATA SEGMENTATION ALGORITHM

A cellular automaton (CA) is a discrete dynamical system wherein its fundamental characteristic is that space and time are discrete. Moreover, based on the transition rules, CA iteratively

performs local computations on a grid of cells or a lattice of sites.

The simple structure of the cellular automaton fascinated a number of researchers from various disciplines. As a result, researchers used CA to model various dynamical systems in several application domains. It is one of the oldest models of natural computing, originally introduced by von Neumann and Ulam. As such, the simplest model of a CA has the following attributes; cell and cellular space (grid), neighborhood for local interaction, cell state and transition rule. The cells influence neighboring cells based on the transition rules and as a result the cells change states. For a two-dimensional CA, the two common neighborhoods used are the von Neumann neighborhood, a diamond-shaped neighborhood, and the Moore neighborhood, a square neighborhood [15, 16, 17].

Furthermore, to become acquainted with researches on CA, the works of Ganguly et al. and Kari [16, 18] are excellent references since they presented a comprehensive assessment of the different researches involving cellular automata.

In image processing, the grid of cells is an image and the cells are the pixels, where a cell's state contains the pixel's value, label and distance. The transition rule defines the type of image processing task.

For instance, in seeded image segmentation techniques, segmenting an image may depend on the weighted distance methods like the Ford-Bellman's and Dijkstra's shortest path algorithms. When using CA in image segmentation, the CA transition rule computes the shortest path between pixels in an image using a certain weighted distance method. In this approach, a graph represents an image where the pixels are vertices and edges connect neighboring pixels. Hence, to determine the likelihood of assigning a particular user-provided label (seed) to each unlabeled pixel it is necessary to evaluate the weighted distance between vertices [19, 20].

Researchers implemented the FBA in the form of a CA where the CA segments an image by implementing the Ford-Bellman's shortest path algorithm (FBA) in such a way that the CA transition rule computes for the FBA shortest path. CA applies this Ford-Bellman's CA transition rule simultaneously for all cells and iterates until no cell changes its state. However, this method is referred to as Seeded Cellular Automaton (SCA) since it requires *a priori* label or seed [14, 19, 20].

Kauffman and Piche [19, 20] implemented the SCA algorithm, where the CA transition rule computes for the FBA shortest path using (1).

$$\lambda^{t+1}(p) = \min \left[\lambda^t(p), \min_{q \in N_p} \{ \lambda^t(q) + w_{pq} \} \right] \quad (1)$$

where $\lambda^t(p)$ is the distance of a cell at time step t , N_p is the neighborhood of cell p and w_{pq} is the edge weight between p and q .

The pseudo-code of the SCA for the FBA with k -seeds is as follows [19, 20]:

Begin

[01] $s_l \in V$ where $l \in [1, K]$ with K the total number of labels

[02] $\lambda(s_l) \leftarrow 0$ and for all $p \neq s_l$, $\lambda(p) \leftarrow \infty$

[03] $\text{label}(s_l) \leftarrow l$ and for all $p \neq s_l$, $\text{label}(p) \leftarrow 0$

[04] for $\forall p \in V$

[05] $\text{label}^{k+1}(p) = \text{label} \left\{ \min \left[\lambda^t(p), \min_{q \in N_p} \{ \lambda^t(q) + w_{pq} \} \right] \right\}$

[06] $\lambda^{k+1}(p) = \min \left[\lambda^t(p), \min_{q \in N_p} \{ \lambda^t(q) + w_{pq} \} \right]$

[07] End for

End

The algorithm starts by assigning color labels and weighted distances (cell states) to each cell (lines 2-3), where the initial color label for labeled cells is the user-provided color and null or zero for non-labeled cells while the initial distance for labeled cells s_l , is zero and infinity for non-labeled cells. For each iteration $k + 1$ (lines 4-7), the cell label, label^{k+1} , and the cell state, λ^{k+1} are updated based on the neighbors of the cell, N_p . CA iterates as long as any cell changes its state.

IV. SCHROMATICA: AN SCA-BASED IMAGE COLORIZATION METHOD

As shown in Fig. 1, the proposed SCA-based image colorization method consists of four basic steps.

The input image is an $N \times M$ grayscale image and modeled as a two-dimensional CA. The image colorization process starts by asking the user to draw colored scribbles on the image. These manually drawn scribbles affect how colors are assigned to the different regions in the image using SCA. The Seeded Cellular Automata Segmentation algorithm propagates these colors to the different regions with the objective of producing a visually plausible colored image [14, 19, 20].

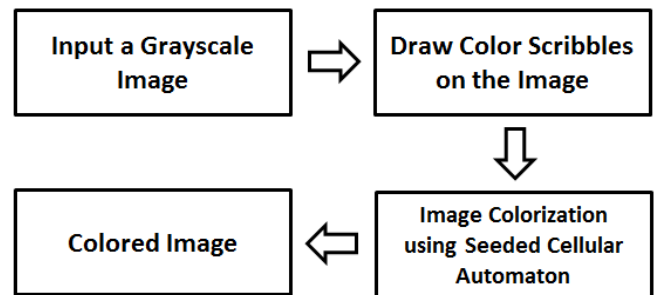


Figure 1. An SCA-based Image Colorization Method

The image is regarded as a discrete object and it is represented as a graph, $G = (V, E)$, where each pixel is a vertex v , such that $v \in V$. These vertices are connected by a set of edges, E , such that each edge e connects two adjacent vertices v_i and v_j where $e \in E$. As shown in Fig. 2, each vertex has three attributes; the pixel's intensity or luminance value, its distance, and a color label. Moreover, seed vertices are the pixels that match the locations of the colored scribbles. Also, vertices are referred to as cells during the cellular automaton phase.

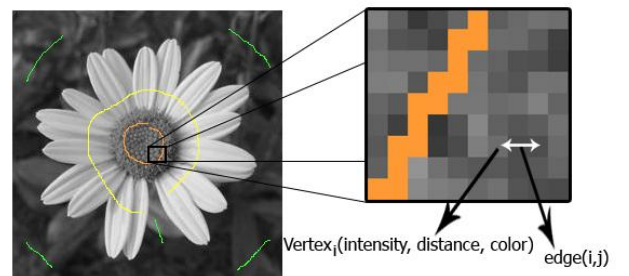


Figure 2. Attributes of a Vertex

A. Initialization of Vertices

For all seed vertices, the initial distance is zero, $\lambda(v) = 0$, while the label corresponds to the color of the scribble of a certain pixel, $\text{label}(v) = c_k$, where c is a color in the scribble set, S^k and k is the cardinality of S^k . On the other hand, all non-seed vertices have an initial distance of infinity, $\lambda(v) = \infty$ and an initial label of null, $\text{label}(v) = \text{null}$.

Using the YUV color model, the intensity or luminance value of a vertex is calculated using (2) where the Red, Green

and Blue values are derived from the original grayscale value of the pixel. To calculate for the U and V chrominance components use (3) and (4). Fig. 3 illustrates the initial values of the seed and non-seed vertices.

$$Y = [(299 * R) + (587 * G) + (114 * B)] / 1000 \quad (2)$$

$$U = [(-147 * R) - (289 * G) + (436 * B)] / 1000 \quad (3)$$

$$V = [(615 * R) - (515 * G) - (100 * B)] / 1000 \quad (4)$$

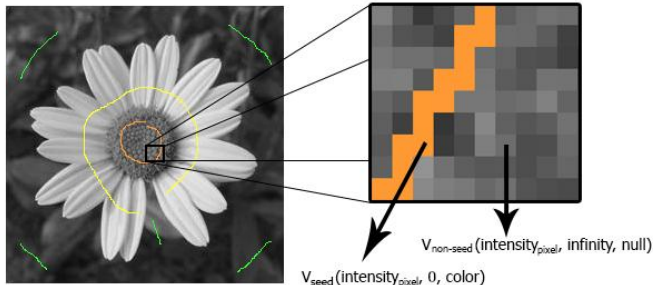


Figure 3. Initialization of Vertices

B. Image Colorization using SCA for the FBA

Using the implementation of SCA as proposed by Kauffman and Piche [19, 20], the cells in a two-dimensional CA are the vertices of a graph (pixels in an image). This study employed the von Neumann neighborhood with a range of 1 as shown in Fig. 4.

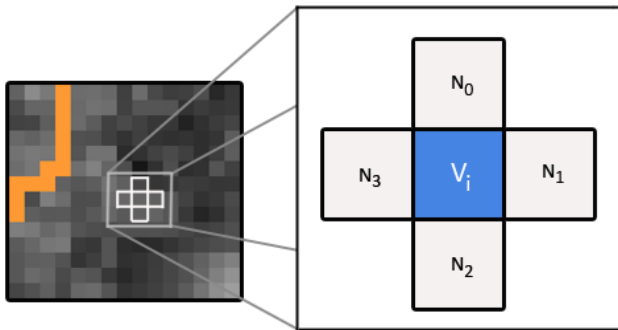


Figure 4. Von Neumann neighborhood method with range=1

Also, the cost or weight of an edge, w_{ij} , is determined using (5), where the function $g(\cdot)$ is the intensity of a grayscale pixel.

$$w_{ij} = \| g(v_i) - g(v_j) \| \quad (5)$$

Fig. 5 illustrates how to calculate for the edge weight between two vertices.

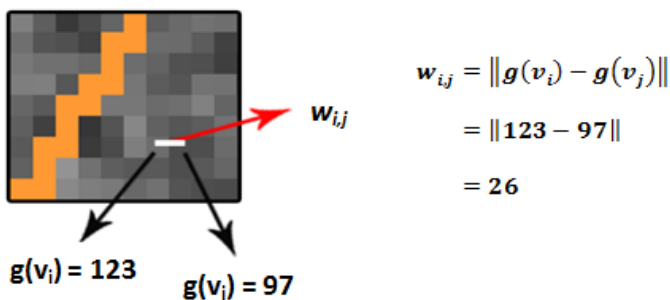


Figure 5. Edge Cost Computation

The SCA computes for the weighted distances between vertices and determines the new value of each cell's distance and label in each iteration. A cell's new state, its new distance,

is determined according to (1) while the new label of a cell is computed using (6).

$$\text{label}^{k+1}(p) = \text{label} \left\{ \min \left[\lambda^t(p), \min_{q \in N_p} \{ \lambda^t(q) + w_{pq} \} \right] \right\} \quad (6)$$

Each cell in the two-dimensional cellular automaton is evaluated using (1) and (6). This evaluation is performed as long as there is a cell whose state can still be updated, that is, as long as there is a cell whose distance can still be minimized.

For instance, Fig. 6 shows a sample target cell that has to be updated. Using the von Neumann neighborhood with range of 1, the target cell will have four neighbor cells. In this example, the edge weights of the neighbors were calculated using (5). Using (1) and (6), the new distance of the target cell is 103 and the new label of the target cell is blue, respectively.

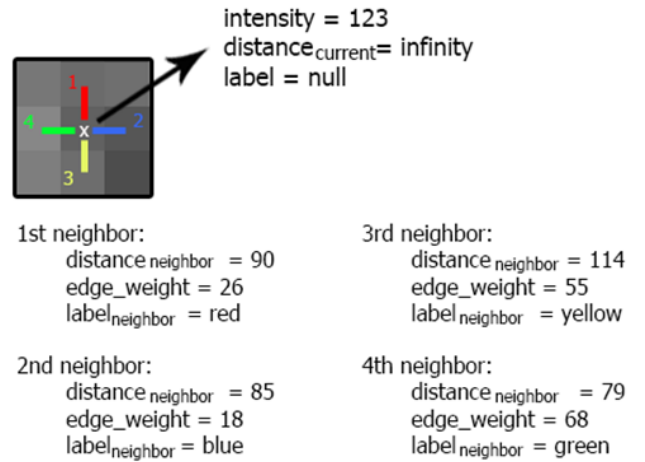


Figure 6. Sample target vertex for updating

C. Embedding of Colors to the Grayscale Image

When the cell updating step is completed, the new YUV values of each pixel in the grayscale image are determined using (7), (8) and (9) where $U_{grayscale}$ and $V_{grayscale}$ denote the U and V of the original values of the grayscale pixel while U_{color} and V_{color} denote the U and V of the color labels.

$$Y_{final} = \text{pixel intensity of the grayscale image} \quad (7)$$

$$U_{final} = (U_{grayscale} + U_{color})/2 \quad (8)$$

$$V_{final} = (V_{grayscale} + V_{color})/2 \quad (9)$$

Using the derived YUV values, the final RGB values for all the pixels are calculated using (10), (11) and (12) as shown in the paper by Ibraheem, et al. [11], and these new RGB values are embedded in the grayscale image to produce a colored image.

$$R = Y + (1.140 * V) \quad (10)$$

$$G = Y - (0.396 * U) - (0.581 * V) \quad (11)$$

$$B = Y + (2.032 * U) \quad (12)$$

V. EXPERIMENTS AND RESULTS

To evaluate the effectiveness of Seeded Cellular Automaton in Image Colorization, experiments were performed in a computer running on Intel® Atom CPU Z520 processor and a RAM of 2048 MB. The images used in these experiments are shown in Fig. 7.



Figure 7. Test images

In the first experiment, the effectiveness of the SCA-based image colorization method using the von Neumann neighborhood technique with a range of 1 was assessed using the images in Fig. 7 and the colored images produced are shown in Fig. 8 to Fig. 10 including the grayscale image and the image with scribbles.

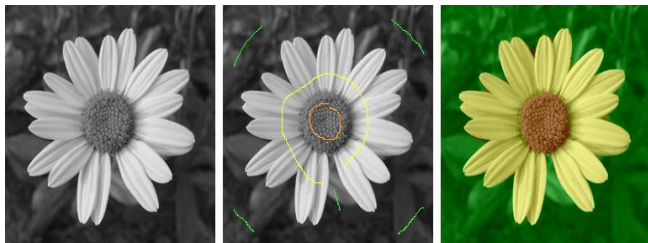


Figure 8. Flower test image

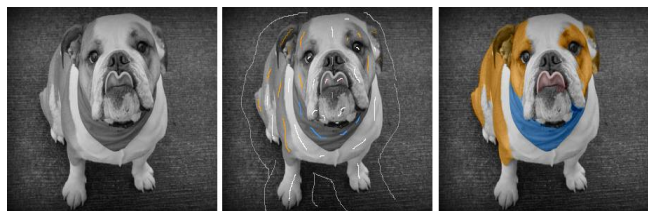


Figure 9. Dog test image



Figure 10. "Shirley" test image

As seen in Fig. 8 to Fig. 10, the SCA-based image colorization method produced visually plausible colored images, that is, the colored images have an acceptable visual quality. However, one crucial factor in the scribble-based method is selecting sensible colors so that the SCA produce visually appealing colored images.

Another experiment was performed to further evaluate the quality of the colored images produced by the SCA. The resulting colored images were compared to the colored images produced in a previous research [4]. Fig. 11 to Fig. 13 show a comparison of colored images produced by these two different approaches.

Although the colors used in this experiment were carefully selected to match the colors used in [4] and results show that the SCA-based image colorization method produced colored images with comparable visual quality, still some problems were observed. First, the colored image appears dull and this may be attributed to clipping of the pixel values to the range of values of the color models used. That is, some values are truncated when converting RGB to YUV and YUV to RGB.

It was also difficult to map more than one color in the same region. For instance, in Fig. 13(b), the face of the child lacks the pink tone. This arose due to the difficulty of finding the right shade of the skin and the limitation of the proposed method to blend more than one color in a single homogeneous region.

Nonetheless, despite these limitations, the SCA-based method successfully converted a grayscale image to a colored image.



Figure 11. Goldy test image



Figure 12. Casino test image

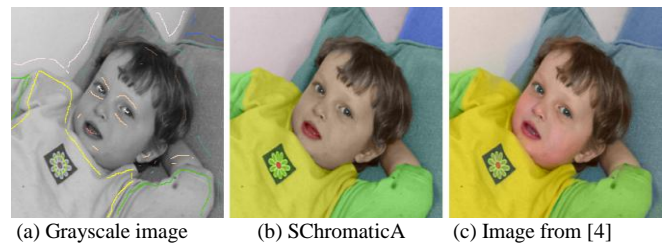


Figure 13. "Gili" test image

A third experiment was conducted to evaluate how the choice of the neighborhood method affects the performance of the SCA-based image colorization technique. The flower image and the Garfield image were used to assess the neighborhood methods. The two neighborhood methods used in this experiment are the von Neumann neighborhood and the Moore neighborhood.

Fig. 14 shows the result when using the von Neumann neighborhood while Fig. 15 shows the result when using the Moore neighborhood. Fig. 14 and Fig. 15 show the grayscale flower image with colored scribbles followed by colored images representing the output of the algorithm after every tenth iteration.



Figure 14. SCA image colorization using the von Neumann neighborhood method with a range of 1 on the flower image



Figure 15. SCA image colorization using the Moore neighborhood method with a range of 1 on the flower image

Results using the flower image show that the von Neumann neighborhood of range 1 produced a colored image on the 67th iteration with a running time of 18.46 seconds. On the other hand, the Moore neighborhood of range 1 produced a colored image after 64 iterations with a running time of 19.40 seconds.

Similarly, for the Garfield image as shown in Fig. 16, the SCA produced a colored image after 191 iterations in 34.16 seconds using the von Neumann neighborhood ($r=1$) while it produced a colored image after 176 iterations in 42.66 seconds using the Moore neighborhood ($r=1$).

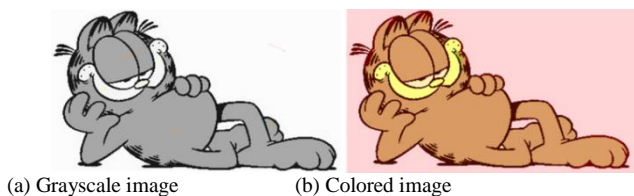


Figure 16. Garfield image

Based on the results, it can be observed that when the SCA used the Moore neighborhood, the colored image is produced after a small number of iterations while it required a larger number of iterations when using the von Neumann neighborhood. However, in terms of running time, the Moore neighborhood required more time to complete the process as compared to the von Neumann neighborhood.

This behavior can be attributed to the properties of the neighborhood methods. For instance, when using the Moore neighborhood with a range of 1, each cell is evaluated against

eight neighbor cells and when using the von Neumann neighborhood method with a range of 1, each cell is evaluated against four neighbor cells.

VI. CONCLUSIONS

In this research, the Seeded Cellular Automaton algorithm is used in image colorization. Results show that the SCA-based image colorization method produced visually plausible images given a grayscale image with user-drawn colored scribbles.

Although, certain problems are observed such as the inability to blend two or more colors in a certain region and the slight loss of brightness in the image, still the colored images have a remarkable visual appearance.

Even though the neighborhood technique affects the image colorization process in terms of the number of iterations and running time, still the colored images produced are of the similar visual quality.

Hence, the Seeded Cellular Automaton algorithm is an effective algorithm for image colorization.

VII. REFERENCES

- [1] A. Levin, D. Lischinski and Y. Weiss, "Colorization using optimization," ACM Transactions on Graphics-Proceedings of ACM SIGGRAPH 2004, vol. 23, no. 3, 2004, pp. 689-694.
- [2] M. Kawulok and B. Smolka, "Image colorization with competitive propagation paths and chrominance blending," Studia Informatica, vol. 31, no. 2A, 2010, pp. 321-333.
- [3] Y. Chen and S. Wang, "Image colorization with edge detection and sub-sampling," Proceedings of the 2010 3rd International Congress on Image and Signal Processing (CISP), vol. 3, 2010, pp. 1129-1132.
- [4] L. Yatziv and G. Sapiro, "Fast image and video colorization using chrominance blending," IEEE Transactions on Image Processing, vol. 15, no. 5, 2006, pp. 1120-1129.
- [5] H. Koo and N. Cho, "Colorization based on soft segmentation," Journal of Electronic Imaging, vol. 20, no.1, 2011, pp. 010502-010502-3.
- [6] A. Kalia and B. Singh, "Colorization of grayscale images: an overview," Journal of Global Research in Computer Science, vol 2, no. 8, 2011, pp. 34-37.
- [7] T. Welsh, M. Ashikhmin and K. Mueller, "Transferring color to grayscale images," ACM Transactions on Graphics-Proceedings of ACM SIGGRAPH 2002, vol. 21, no. 3, 2002, pp. 277-280.
- [8] X. Wang, J. Jia, H. Liao and L. Cai, "Affective image colorization," Journal of Computer Science and Technology, vol. 27, no. 6, 2012, pp. 1119-1128.
- [9] T. Acharya and A. Ray, Image processing: principles and applications, John Wiley & Sons, Inc., Hoboken, New Jersey, 2005, pp. 37-59.
- [10] C. Chien and D. Tseng, "Color image enhancement with exact HSI color model," International Journal of Innovative Computing, Information and Control, vol. 7, no. 12, 2011, pp. 6691-6710.
- [11] N. Ibraheem, M. Hasan, R. Khan and P. Mishra, "Understanding Color Models: A Review," ARPN Journal of Science and Technology, vol. 2, no. 3, 2012, pp. 265-275.
- [12] C. Gauge and S. Sasi, "Automated colorization of grayscale images using texture descriptors," ACEEE Int. J. on Information Technology, vol. 1, no. 1, 2011, pp. 46-48.
- [13] V. Konushin and V. Vezhnevets, "Interactive image colorization and recoloring based on coupled map lattices," In Proceedings of Graphicon 2006, 2006, pp. 231-234.
- [14] R. Beasley, "Semiautonomous medical image segmentation using seeded cellular automaton plus edge detector," ISRN Signal Processing, vol. 2012, 2012.
- [15] A. Popovici and D. Popovici, "Cellular automata in image processing," In Proceedings of the 15th International

Symposium on the Mathematical Theory of Networks and Systems, 2002.

- [16] J. Kari, "Theory of cellular automata: A survey," *Theoretical Computer Science*, vol. 334, no. 1-3, 2005, pp. 3-33.
- [17] E. Kim, T. Shen and X. Huang, "A parallel cellular automata with label priors for interactive brain tumor segmentation," 2010 IEEE 23rd International Symposium on Computer-Based Medical Systems (CBMS), 2010, pp. 232-237.
- [18] N. Ganguly, B. Sikdar, A. Deutsch, G. Canright, and P. Chaudhuri, A Survey on Cellular Automata, Technical Report, Centre for High Performance Computing, Dresden University of Technology, 2003.
- [19] C. Kauffmann and N. Piche, "Seeded ND image segmentation by cellular automaton on GPU," *International Journal of Computer Assisted Radiology and Surgery*, vol. 5, no. 3, 2010, pp. 251-262.
- [20] C. Kaufmann and N. Piche, "A cellular automaton framework for image processing on GPU," *Pattern Recognition*, 2009.