Color Processing - A New Approach Based
On Discrete Dynamic Systems

Georgios-Tsampikos Michailidis and Ioannis Andreadis
Democritus University of Thrace, Greece

ABSTRACT: Nowadays, although an increased number of people tend to use high-quality images for their daily activities, there are some applications where it is useful to have a reduced number of colors in our images. In this article, a color reduction approach is presented based on Cellular Automata (CA). More specifically, we exploit the inherent properties of CA, in order to locally detect the dominant color values and replace the colors of a predetermined neighborhood with these values. This method is applicable to any type of color image and it can accommodate any type of color space. The final results indicate that there is enough space for future improvements in this field.

1 INTRODUCTION: The reduction of the number of colors in digital images is still an active research area in image processing. Color reduction is the process of reducing the number of colors in an image, in order to improve its display or file handling. Generally, a digital image is described by an array of pixels and, in color images, the color of every pixel is expressed by a vector of values of the three color components (Red, Green and Blue). True-type color images may consist of more than 16 million different colors, in a 24-bit RGB color space. Each primary color component represents an intensity which varies from 0 to a maximum value, which corresponds to full saturation of that color. However, in many applications, such as image segmentation, analysis, compression and transmission, it is preferable to have images with a limited number of colors. The main reason to reduce the number of colors is actually the need to decrease their storage requirements.

Several techniques have been proposed for color reduction, but the majority of them are based either on splitting algorithms [1-3], which divide the color space in disjoint regions, or on cluster analysis [4-6], where vector classifiers are used to find the optimal palette. The purpose of this article is to present a new approach to this issue and one of our research fields, where CA are used to color reduction, in order to find the “basic” colors in the image and eliminate the unnecessary color information, exploiting only the neighboring relationships between the pixels of the image. Generally, CA have not been used in color reduction applications, so we try to make a first step in this field using a fast and simple CA algorithm. This technique is appropriate for reducing the number of colors of images at a preprocessing stage for other image-processing applications, and our goal is to reveal the restrictions and difficulties that appear in this color reduction scheme.

In the next sections, there is a short reference on the general framework of CA, along with the description of the proposed algorithm and its experimental results.

2 THE GENERAL FRAMEWORK OF CELLULAR AUTOMATA: CA can be used widely for image processing due to their inherent local structure and simple parallel computing implementation. A CA is a collection of "colored" cells (in image processing, pixels) on a grid of predefined shape and it is evolved through a number of discrete time steps according to a set of transition rules. These rules are based on the state of a local neighborhood and they are applied iteratively for as many time steps as required.

One of the most important properties of CA is the type of grid on which it is computed. The simplest grid can be one-dimensional, while in two dimensions, triangular, square and hexagonal grids may be considered. The number of colors (or distinct states) $k$ of CA should also be specified. This number is typically an integer and in our case it will be the number of different colors (or intensities) of a color channel, i.e. $k=0..255$. In addition to the grid on which a CA
lives and the colors of its cells, the neighborhood over which cells affect one another must also be specified. One easy choice is to use the nearest neighbors, where only the cells directly adjacent to a given (central) cell may be affected at each time step. In normal 2-D CA grids, two types of neighborhood are considered: Von Neumann neighborhood and Moore neighborhood. The Von Neumann neighborhood is a 4-neighborhood CA, consisting of the cell along with its four immediate non-diagonal neighbors. The Moore neighborhood is a 8-neighborhood CA, consisting of the cell along with its eight surrounding neighbors. It is also possible to have an extended Von Neumann or Moore neighborhoods, which keep the same basic pattern as above, but their neighborhoods reach over the distance of the next \( r \) adjacent cells.

In addition, in the proposed technique, the CA rules are based on a particular type of CA systems, the non-local CA [7]. The non-local CA have mainly the following characteristics: (a) the space, time and state values of each component (i.e. pixel) are discrete and their state value is finite; (b) the rule applied to one component is the same as those applied to all other components; (c) the updating of the state value of every component is synchronized. In image processing, this means that, although non-local CA are the same as that of usual CA, they are allowed to have non-local connections between pixels of the image. Thus, they provide further flexibility to use better the local color information and improve the final results.

3 PROPOSED METHOD: Generally, true-color digital images use red, green, and blue channels (each with 8-bit resolution), to specify the color of each pixel on the image. As a result, images usually are composed of a large number of distinguishable colors, but, actually, the human eye can only distinguish less than a thousand of colors, therefore a smaller number of colors is sufficient for human perception. In addition, 24-bit digital images can have many different colors, so it is impractical for an algorithm to exhaustively search the best solution for the color reduction problem.

Therefore, our algorithm is proposed to provide a trade off between acceptable quality and running time, as we emphasize on the execution time. It exploits the inherent properties of CA, in order to locally select the representative colors of a predetermined neighborhood on the image and express the rest of the local colors with that values. The basic idea behind the CA rules is to simplify the color selection procedure and reduce the time consumption as much as possible. In addition, due to the nature of CA, the color reduction problem can be expressed in terms of smaller, local sub-problems, which leads to the creation of a recursive algorithm.

Since there are numerous rules that can be applied to reduce the colors in the initial image, a considerable effort was devoted to explore the effects of different rules and use only those that have proven to be good in estimating the “basic” colors of it. Before getting into details about the CA transition rules, we provide, for convenience, a definition to characterize the neighboring area that surrounds the central pixel of the CA. The Neighboring Area \( N_{AH} \) is defined as the “extended” neighborhood of the central pixel of the transition rule and its indices indicate the horizontal(\( H \)) and vertical(\( V \)) number of pixels that comprise the extended neighboring area. For instance, \( N_{AH}^3 \) with \( l=[-1..-3] \) and \( k=[1,2] \), represents the extended neighborhood of three left (minus sign) and two down (plus sign) pixels of the central one. Thus, based on CA, we consider the following transition rules.

a) In the Von Neumann neighborhood,
if \((C_{i,j} = C_{i+1,j})\) \(\text{OR} (C_{i,j} = C_{i,j+1}),\) then 
\((C_{i,j} = C_{i,j+1})\) \(\text{OR} (C_{i,j} = C_{i,j-1})\)

In addition, 
if \((C_{i,j} \neq C_{i,j-1})\) \(\text{AND} (C_{i,j} \neq C_{i,j+1}),\) 
while \((NA_{0}^{0} = NA_{m}^{0})\) \(\text{OR} (NA_{0}^{l} = NA_{0}^{m}),\) 
then \((C_{i,j} = C_{i,j+1})\) \(\text{OR} (C_{i,j} = C_{i,j-1})\) 
with \(l=[-1..-3], n=[1..3], \) \(k=[-1..-3] \) \(\text{and} \) \(m=[1..3]\)

Else 
if \(((C_{i,j} \neq C_{i,j-1})\) \(\text{AND}(C_{i,j} = C_{i,j+1}))\) \(\text{OR} \) 
\(((C_{i,j} \neq C_{i,j+1})\) \(\text{AND}(C_{i,j} = C_{i,j-1}))\),

Typically, the overall operation of the CA is such that at each pixel, the transition rule is tested to check if it matches the pixel neighborhood pattern. If so, the central pixel is replaced by the proper color value, otherwise it remains unaltered. We should mention here that the initial RGB image was first separated into its three color components \((R(i,j), G(i,j) \text{ and } B(i,j))\) and each transition rule is applied to each color channel separately.

The first CA rule is actually a rule set and the last two rules of this rule set are also applied to the vertical direction at the same time step. This rule set states that if there are one, two or three pixels in the image with different colors between uniform areas with same color values, then replace them with that value. This rule is based on the observation that objects on color images, although they appear uniform for human perception, are mainly noisy with unwanted color variations. Noise in color images may result from many sources, such as the imaging sensor itself or sensor malfunction. For this reason, digital images contain thousands of colors and make color reduction a difficult task. Our goal is to make the noisy pixels to be absorbed by the uniform areas by finding the most suitable local color values to replace them.

In the second CA rule, we check in the extended Moore’s neighborhood with radius \(r=2\), if there is a dominant color value and replace the center pixel with this value. More specifically, the second rule states,

\[
\text{if } |C_{i+\mu,j+\nu}-c_{\text{val}}| = 0 \text{ then } n = n + 1, \\
\forall (\mu, \nu \in [-r,r]) \text{ while } (\mu \neq 0 \wedge \nu \neq 0) \text{ and } n \in \mathbb{Z}
\]

\[
\text{if } (n > \text{dom_value}) \text{ then } C_{(i,j)} = c_{\text{val}}, \\
\forall (c_{\text{val}} \in [0, 255]) \text{ while } \text{dom_value} = 13
\]

Furthermore, if the difference between the minimum and the maximum color value of that neighbourhood is more than a threshold \((\text{Thr})\) value and the variance is large, then it replaces all the color values of that neighbourhood with the dominant value. The threshold value is manually set, according to the desired quality of the final results, and it is related with the variation of colors in the local neighbourhood. If it is set to a high value, the number of colors in the image will be reduced, but the deformation of objects will be increased, and vice versa.

In the third step, a local color histogram is created for each channel in the extended Moore’s neighbourhood with radius \(r=5\) and if the neighboring frequency values for each tonal value are less than the half of it, then their color values are replaced by the central one. Actually, this is a kind of local adaptive color quantization, where a palette is created adaptively to the image contents, for each local neighbourhood, using only CA. This rule is also iterated for 5
times, in order all colors in local neighbourhoods to be classified. In addition, we noticed that it is not necessary to construct the local histogram for every pixel in the image, but it is sufficient to sample it and create the histograms only for the sampled pixels. The (central) pixels of the Moore’s neighbourhood will be chosen by the sampling equation \( C(i,j) = C(Ti, Tj) \), where \( C(i,j) \) are the pixels in the original image, \( C(Ti, Tj) \) are the central pixels and \( T \) is the sampling interval, which is in our case the same as the radius of the extended Moore’s neighborhood.

The final CA rule is identical to the first one, in order to correct any distortions in object boundaries, caused by the previous CA rules.

Although it is difficult to reduce accurately the color values in all areas of the original image due to the nature of CA, experimental results in next section show that CA systems are very promising in this field.

4 EXPERIMENTAL RESULTS: The evaluation of the proposed technique was based on a set of many 24-bit images and here we present four of them with different sizes.

![Fig 1. The original (a) and the final (b) images of the presented CA color reduction technique, for \( Thr=60 \).](image)

DISCUSSION AND CONCLUSIONS: The final results in Fig. 1 are 10-bit color images, in order to balance the image quality and the fast execution time. The 10 bits are adaptively distributed in the three color components, according to the effect of CA rules in each color channel. From the extracted results, it is clear that the colors are uniformly spread in the final images but due to the nature of CA, the details of some object boundaries are deformed. Color information is adequate in the majority of areas, but the square neighborhoods of CA do not help to maintain smoothness on convex objects. In our future research, we will focus on refining the CA rules, in order to reduce further the number of colors in the final images and on finding appropriate neighborhoods that will not deform the object boundaries.

REFERENCES: