Interactive Image Colorization and Recoloring based on Coupled Map Lattices

V. Konushin, V. Vezhnevets

Graphics and Media Lab., Lomonosov Moscow State University,
Moscow, Russia

kvad@inbox.ru, vvp@graphics.cs.msu.ru

Abstract

Colorization is a computer-assisted process of adding color to a monochrome image or movie. Most current colorization algorithms either require a significant user effort or have large computational time. In any case, colorization of real size images remains a time-consuming, tedious task.

In this paper we present a new colorization method, based on GrowCut image segmentation algorithm. In our approach a user just marks some pixels with desired colors, and the algorithm propagates these colors to the remainder of the image. After the initial colorization is computed, the user can interactively adjust and refine the colors of the image. The main advantage of the proposed method over the existing ones is that modification and refining of the image colors doesn't lead to full recomputation of the image colorization and is performed very fast, thanks to evolutionary nature of the coupled map lattices (CML). Our algorithm can also be successfully applied to image recoloring. We demonstrate our results on a variety of examples.

Keywords: *colorization, recoloring, segmentation, coupled map lattice, cellular automata, GrowCut*

1. INTRODUCTION.

Colorization is a computer-assisted process of adding color to a monochrome image or movie. Recoloring is a process of changing color in some part of an image or movie. Both processes are illustrated on figure 1.

Figure 1: Examples of image colorization and recoloring by our algorithm.

There are many old black-and-white movies now that are not profitable to be colorized manually. Many people have black-and-white photos of their grandparents or some other ancestors, but do not have enough experience to colorize the pictures by themselves. Computer-assisted algorithm could help in both cases. Recoloring is used in special effects and digital photography. Both tasks attract a lot of attention in image-editing community, according to many discussions and tutorials on the World Wide Web.

Usually to colorize an image an artist segments the image into regions and assigns colors to them. Thereby, we decided to use an algorithm, based on one of image segmentation algorithms. In particular, we chose GrowCut [1], because of its high degree of interactivity, which allows easy and intuitive correction. Other valuable advantages of GrowCut are simplicity, multi-label assignment, speed and easy extensibility. In our algorithm, user must only mark some pixels with desired colors, and the algorithm propagates these colors to the remaining image. Our primary aim was to create an algorithm, suitable for use by a simple user. So it must have intuitive interface, be fast enough, and supply methods for incremental fine-tuning the result.

The rest of the paper is organized as follows: Section 2 gives a brief description of previous algorithms. Section 3 describes the proposed algorithm. Section 4 gives the results. In section 5 future works are discussed.

2. PREVIOUS WORKS.

Many colorization algorithms, like the one in [2], work only with colorization of movies. They require a user to manually paint one frame and then use optical flow, motion detection and tracking to propagate color to other frames.

One of the commercial software packages, BlackMagic [3], which is designed for still image colorization, provides the user with a range of useful brushes and color palettes. The main drawback of it is that a segmentation task is done completely manually.

Another approach is to colorize a grayscale image by transferring color from a reference color image. In Welsh et al. [4] pixels in grayscale and reference color images are matched by comparing the luminance values in the neighborhood. Color is transferred from matched pixels.

Yu-Wing Tai et al. [5] described a general color transfer algorithm. At first they perform probabilistic segmentation of both images, using Gaussian Mixture Model (GMM), then map Gaussian components in one image to components in the second image, and then modify pixel's colors according to this mapping function. In case of a grayscale image, they use only luminance

component for grayscale image segmentation. All these algorithms fail, when differently colored regions have similar intensities. A colorized image can have only the same color spectrum, as the target image. In addition there is almost no user control over the process of colorizing.

Levin et al. [6] based on a simple assumption, that neighboring pixels in space-time with similar intensities should have similar colors. They formulate colorization as a quadratic optimization problem and solve it by standard techniques. This algorithms gives high quality colorization, but is rather time-consuming, and what is much more important, it requires the colorization to be fully recomputed after even slightest change of the initial marked pixels.

3. OUR APPROACH.

We propose a colorization method, based on coupled map lattices (CML) [8] - an extension of cellular automata [9]. Our main goal was to create a fully interactive colorization algorithm.

3.1 Colorization

As mentioned above, our algorithm is based on GrowCut [1] and we will use the same notation for description.

A bi-directional, deterministic cellular automaton (CA) is a triplet $A = (S, N, \delta)$, where S is a non-empty state set, N is the neighborhood system, and $\delta : S^N \to S$ is a local transition rule, which gives a cell's state at $t+1$ time step, given the states of it's neighborhood cells at time step t. CA operates on a discrete lattice of sites $p \in P \subseteq Z^n$, in our case Z^n is a bi-dimensional array of pixels. Coupled map lattice (CML) extends cellular automaton by replacing of the discrete state values of CA cells with continuous real values. Just as CAs, CMLs are discrete in space and time and provide an elegant technique for modeling dynamical systems with complex behavior.

A cell's state S_p is a pair $(\theta_p, \vec{C}_p)$ - $\theta_p \in [0,1]$ is a cell's 'strength' and $\vec{C}_p$ - a feature vector of the cell. We use Moore neighborhood system:

$$N(p) = \{q \in Z^n : \|p - q\|_\infty = \max_{i=1,n} |p_i - q_i| = 1\}.$$

Each pixel corresponds to a cell of the CML. We use the color vector of the corresponding pixel in YUV color space as a feature vector $\vec{C}_p$ of the cell. Initially, strengths of all cells are set to 0 and, as an image is a grayscale one, U and V components of feature vector are set to 0 ($\forall p \; \vec{C}_p(U) = \vec{C}_p(V) = 0$). When a user marks pixel p with color C^*, strength θ_p is set to 1 and $\vec{C}_p(U) = C^*(U)$, $\vec{C}_p(V) = C^*(V)$.

At iteration $t+1$ transition rule δ updates each cell's p state as follows:

1. $C_p^{t+1} = C_p^t$
2. $\theta_p^{t+1} = \theta_p^t$
3. for $\forall q \in N(p)$
4. if $g(\left\|\vec{C}_p^t(Y) - \vec{C}_q^t(Y)\right\|_1) * \theta_q^t > \theta_p^{t+1}$
5. $\theta_p^{t+1} = g(\left\|\vec{C}_p^t(Y) - \vec{C}_q^t(Y)\right\|_1) * \theta_q^t$
6. $\vec{C}_p^{t+1}(U) = \dfrac{(\vec{C}_p^t(U) * \theta_p^t + \vec{C}_q^t(U) * \theta_p^{t+1})}{\theta_p^t + \theta_p^{t+1}}$
7. $\vec{C}_p^{t+1}(V) = \dfrac{(\vec{C}_p^t(V) * \theta_p^t + \vec{C}_q^t(V) * \theta_p^{t+1})}{\theta_p^t + \theta_p^{t+1}}$
8. end if
9. end for

where $g(x) = \exp(-\alpha * x)$, but in the general case can be any monotonous decreasing function bounded to $[0,1]$.

As in GrowCut, we can use biological metaphor for this algorithm's explanation. We inject bacteria in every cell, corresponding to a marked pixel. Its properties are U and V components of provided color. The bacteria start to spread from marked pixels to the remainder of the image. During the spreading, bacteria's strength is decreasing, proportional to the difference between pixel's luminances along the bacteria path. This is controlled by $g(\left\|\vec{C}_p^t(Y) - \vec{C}_q^t(Y)\right\|_1)$ factor. While occupying free cells bacteria preserves its properties, but when it conquers already occupied cell, the bacteria mutates – its new properties is a mixture of properties of bacteria in a captured cell and of attacker's. This provides blending of colors on the borders of objects.

The computation continues until no bacteria can capture its neighbor. The convergence is guaranteed by a bounded and increasing sequence of each cell's strength.

3.2 Recoloring

Recoloring is provided by the same algorithm, but with a slightly different input. In order to recolor an object in an image, user must define a rough mask around it and mark some pixels inside the object with desired colors. Pixels outside the mask are treated just as the marked ones in colorization. Color from the marked pixels and from the exterior of the mask is propagated to the remainder of the mask. In recoloring a decrease of bacteria's strength between pixels is based on the similarity of all 3-color channels of the original image.

3.3 Refinement.

After the initial colorization (or recoloring) is computed - user can refine it by marking additional pixels with some colors. Strengths

of cells, corresponding to the marked pixels are set to 1, and their feature vectors are set to newly assigned colors. They start to spread trying to occupy neighboring pixels. This refinement produces only a local change in the CA state and is computed very fast. The user can observe and control the change of the colorization by adding more marked pixels with brush-like interface.

4. RESULTS.

Figure 2 shows one of the results of our colorization. It took 12 seconds on a Pentium-4 2.4 GHz with 512 RAM to colorize this image of size 632*1000.

On figure 3 images from Berkley Image database [7] are initially desaturated, and then are recolored. Original, ground truth images are also presented.

Figure 4 shows examples of recoloring by our algorithm. Despite the rough mask, color from inside the object, hasn't propagated outside it.

One can see on these examples the main drawback of our current algorithm – poor color blending. Color blending on the borders of objects depends on the position of marked pixels: blending occurs on the side, where marked pixels are positioned father from the border. Bacteria from the other side first reach the border, cross it, and spread further, capturing cells on this side. Then bacteria from this side recapture these cells, and colors of two bacteria are mixing. There is no blending at all, when marked pixels are positioned at the same distance from the border.

(a) (b)

Figure 2: (a) – original black-and-white photo with scribbled colors. (b) – result of colorization

5. SUMMARY AND FUTURE WORK.

Despite good results, shown by some of the exisintg image colorization algorithms, colorization of real size images (for example 2 mega pixel images) remains very time-consuming and tedious work. In this paper, we proposed a new algorithm, which enables a user to colorize a real size image with less effort. User just scribbles with desired colors in an image and the algorithm propagates the colors to the remainder of the image. Adding new scribbles or changing color of previous scribbles doesn't require recomputing the whole colorization, and is performed very fast. The main drawback of current algorithm is poor color blending.

In future we plan to consider several bacteria in each cell in order to mix colors on the borders properly. We also plan to implement the algorithm on GPU to increase its speed further.

6. REFERENCES

[1] V. Vezhnevets, V. Konouchine *"'GrowCut' - Interactive Multi-Label N-D Image Segmentation By Cellular Automata"*. Graphicon-2005, Novosibirsk Akademgorodok, Russia, 2005.

[2] W. Markle, B. Hunt "*Coloring a black and white signal using motion detection.*" Canadian patent no. 1291260, Dec. 1987

[3] Neuraltek, BlackMagic photo colorization software, version 2.8. http://www.timebrush.com/blackmagic, 2003

[4] T. Welsh, M. Ashikhmin, K. Mueller. "*Transferring color to greyscale images.*" *In Proceedings of the 29th Annual Conference on Computer Graphics and interactive Techniques*, pp 277–280, 2002.

[5] Y. Tai, J. Jia, C. Tang "*Local Color Transfer via Probabilistic Segmentation by Expectation-Maximization*", 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05), Volume 1, pp. 747-754, 2005

[6] A. Levin, D. Lischinski, Y. Weiss. "*Colorization using optimization.*" *ACM Transactions on Graphics*, Volume 23, Issue 3, pp.689–694, 2004.

[7] D. Martin, C. Fowlkes, D. Tal, J. Malik "*A Database of Human Segmented Natural Images and its Application to Evaluating Segmentation Algorithms and Measuring Ecological Statistics*" in Proc. 8th Int'l Conf. Computer Vision, vol. 2, pp 416-423, July 2001

[8] K. Kaneko (Ed.), "*Theory and Applications of Coupled Map Lattices*", Wiley, New York, 1993.

[9] J. von Neumann, "*Theory of Self-Reproducing Automata*", A. W. Burks, Ed. Urbana, IL: Univ. of Illinois Press, 1966.

About the author(s)

Vladimir Veznevets is a research fellow at Graphics and Media Laboratory of Moscow State Lomonosov University. He graduated with honors Faculty of Computational Mathematics and Cybernetics of Moscow State University in 1999. He received his PhD in 2002 in Moscow State University also. His research interests include image processing, pattern recognition, computer vision and adjacent fields. His email address is vvp@graphics.cs.msu.ru.

Vadim Konouchine is a forth year student of Moscow State Lomonosov University, Faculty of Computational Mathematics and Cybernetics. His research interests lie in the fields of statistics and image processing. His email address is kvad@inbox.ru.

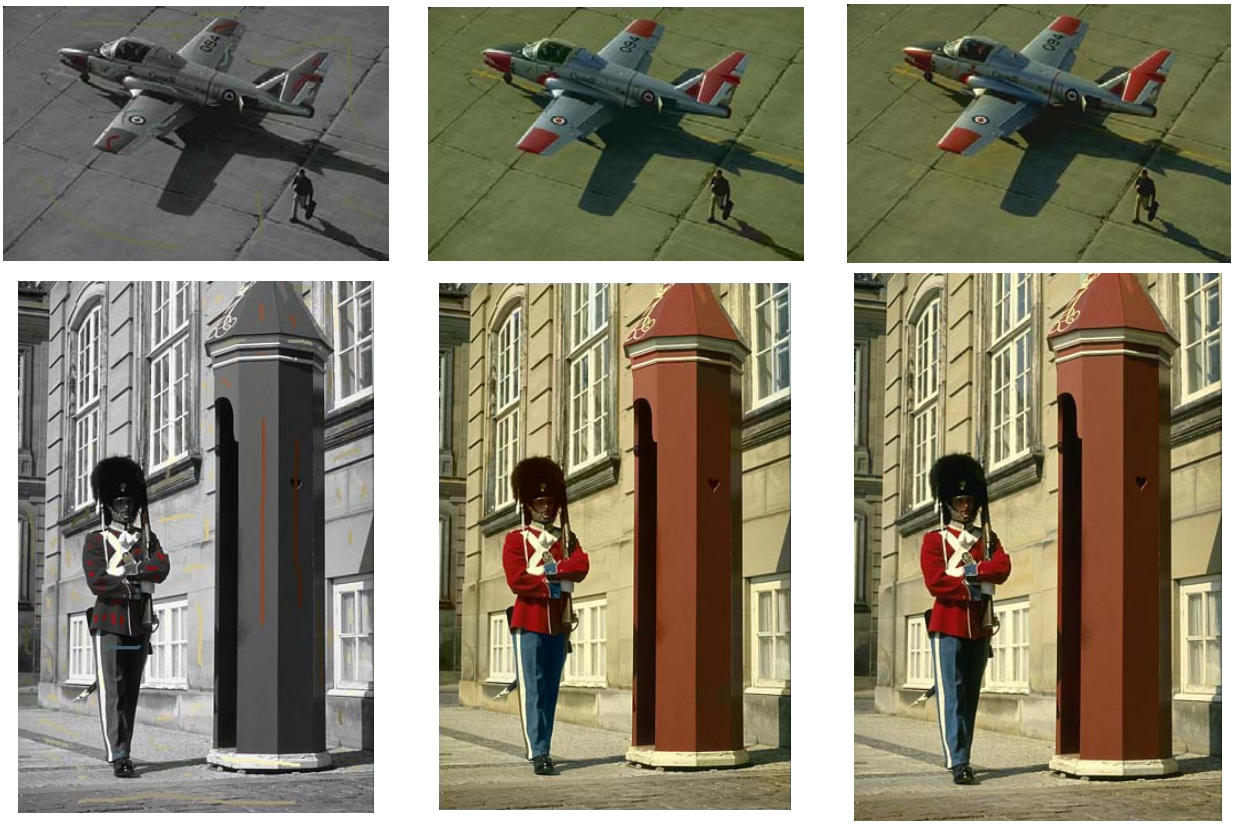

Figure 3: left column – desaturated images with scribbled colors, central column – resulting color images, right column – original images.

Figure 4: left column – original images, central column – white pixels are constrained to keep their original color, right column – resulting images.