Colorization using Optimization

Xinling Chen

A Levin, D Lischinski, and Y Weiss. Colorization using optimization. In *ACM SIGGRAPH 2004 Papers*, pages 689–694, 2004

Levin et al.'s Colorization method [1] requires neither precise image segmentation, nor accurate region tracking, which is in fact based on a simple premise: neighboring pixels in space-time that have similar intensities should have similar colors. The authors accomplish this by formalizing this idea using a quadratic cost function, which leads to an optimization problem that can be solved efficiently using standard techniques. In their approach, the artist colors a small number of pixels in selected frames and the algorithm propagates these colors in a manner that respects intensity boundaries. They also demonstrate that their technique can be applied to still images, movie sequences and recoloring.

The algorithm is based on YUV color space, where Y is referred to as intensity while U and V represent the color. The input of the algorithm is an intensity volume $Y(x, y, t)$ and the outputs are color volumes $U(x, y, t)$ and $V(x, y, t)$. The algorithm is pretty straightforward, which is to minimize the difference between the color $U(r)$ at pixel r and the weighted average of the colors at neighboring pixels, since two neighboring pixels, r, s should have similar colors if their intensities are similar. The authors further give the object function to be optimized:

$$J(U) = \sum_r (U(r) - \sum_{s \in N(r)} w_{rs} U(s))^2 \tag{1}$$

We can see that there are three issues to deal with to minimize this function. First issue is the choice of weighting function wrs. In the paper, the authors experimented with two weight functions. One is based on the squared difference between the two intensities, commonly used by image segmentation. The other one is based on the normalized correction between the two intensities, where uses the mean and variance of the intensities in a window around r. Second, as one may notice, the variance in the object function is U while the input is Y, the intensity. The author further state that other works justified the assumption that the color at pixel $U(r)$ is a linear function of the intensity $Y(r) : U(r) = a_i * Y(r) + b_i$ and the linear coefficients a_i and b_i are the same for all the pixels in a small neighborhood around r. Happily, a simple elimination of the a_i, b_i variable yields an equation equivalent to the equation above. Last issue involves the definition of r/s neighboring pixels. In a single frame, they define two pixels as neighbors if their image locations are nearby while between successive frames, they define two pixels as neighbors if their image locations, after accounting for motion, are nearby.

In implementation, for still images, they used Matlabfs built in least squares solver for sparse linear system and used a multigrid solver for the movie sequences. And the correlation based window is used as weight function for the final results. They showed that their technique can be applied to recoloring and colorize still grayscale images and movie clips. By comparing their methods to two alternative methods, they further argued that their method is better than doing automatic segmentation first and then filling each segment.

It is clever that the author minimized the same cost function in the classical segmentation algorithm but under different constraints and introduced this new simple yet effective technique to do colorization. Although it drastically reduced the amount of input required from the user, scribbling can be still tedious for image with complex details and requires some skill to obtain natural-looking results. In this sense, it is desirable to generate such scribbles automatically from an example image provided by the user and then apply this effective colorization technique. Whatfs more, since it relies on continuity of gray levels to grow the affective regions, this technique fails when the image doesn't preserve grey-level continuity to facilitate the segmentation. Lastly, the paper itself also suggests some possible improvement, such as exploring alternative color spaces and propagation schemes that treat hue and saturation differently. In summary, the algorithm proposed in this paper is simple yet is state-of-art in terms of user-assisted methods for colorizing intensity-continuous images.