Image Colorization

EE368/CS232 Spring-1314 Project Proposal

Masood Shaikh (masood@stanford.edu)

Department of Electrical Engineering, Stanford University

Illustration of colorization process: original, scribbled and colorized

Introduction

Colorization represents a computer-assisted process of adding color to grayscale still images or monochrome movies. Since the mapping from grey-scale pixel to color pixel is not unique, this process generally requires some user interactions to help confine the color selection. The process typically involves segmenting images into regions and tracking these regions across image sequences. Since image segmentation up to the desired level is computationally expensive and iterative; consequently, colorization requires considerable user intervention and remains a tedious, time-consuming, and expensive task. Simple colorization methods have evolved over the years that requires neither precise image segmentation, nor accurate region tracking. Traditional methods of colorization include a straightforward approach called luminance keying [1], that transforms grayscale values to color using a user-defined lookup table. This idea was further extended by Welsh et al [2] and Reinhard et al [3] to transfer the entire color “mood” of the source image to the target image by matching luminance and texture information between the images.

Description

Recently two approaches have been proposed for efficient image and video colorization that demonstrate high quality colorizations of stills and video clips from a relatively modest amount of user input. In both the algorithms, an artist only needs to annotate the image with a few color scribbles on the grayscale image, and the indicated colors are automatically propagated in both space and time to produce a fully colorized image or video. The first algorithm [4] is based on the premise that neighboring pixels in space-time that have similar intensities should have similar colors. This basic premise is further formalised using a quadratic cost function and an optimization problem is obtained that can be solved using standard methods such as least squares technique. This is a robust and accurate colorization algorithm, but it is computationally intensive.

The other algorithm proposed in [5] is based on the concept of luminance weighted chrominance blending and fast intrinsic distance computations. The intrinsic distance is introduced to define the contiguity between two pixels of the grayscale image. Each pixel with user specified color propagates its
chrominance values to all the adjacent pixels, and each pixel maintains a list of top 3 chrominance values with the minimum intrinsic distances. The resultant color of the given pixel is then the weighted (blended) sum of these 3 top chrominance values. This algorithm is computationally simple due to some distance calculation relaxations but yet an effective approach for image and video colorization.

The goal of this project is to study these two image colorization algorithms and implement one of them. For this quarter we might just concentrate on colorization of still images. The idea can be extended further to video colorization by combining motion estimation algorithm and local color propogation. The implementation and colorization demo will be done in Matlab and no Android Device will be used in this project.

**Schedule**

We intend to complete the project with the following milestones:

<table>
<thead>
<tr>
<th>Task</th>
<th>Week</th>
</tr>
</thead>
<tbody>
<tr>
<td>Implementation of test framework and scribbled gray-scale image input</td>
<td>1 Week (May 1- May 8)</td>
</tr>
<tr>
<td>Implementation of chrominance blending colorization OR colorization using optimization algorithm</td>
<td>2 Weeks (May 9 – May 23)</td>
</tr>
<tr>
<td>Debugging, testing and result generation</td>
<td>1 Week (May 24 – May 30)</td>
</tr>
<tr>
<td>Presentation and Final Report documentation</td>
<td>1 Week (May 31 – Jun 6)</td>
</tr>
</tbody>
</table>

**References**


