Improving the Performance of Colorization Algorithms Using Multiscale Image Analysis

Matthew Sills
May 4, 2010

Abstract

An important problem in Image Analysis is, given a monochrome image with a small amount of color, to generate a fully colorized image. Ha-Quang, Kang and Le [2] propose a regularization using a reproducing kernel Hilbert space. They also suggest, but do not implement, utilizing the multiscale structure of an image to generate the kernel. The multiscale structure is likely to be helpful in constructing the kernel, as color is coherent within scales, but usually not across them. We use the $\delta$-relatively-stable scales of an image [1] to automatically select a few important scales, and incorporate these scales into the construction of the kernel. This procedure almost always results in better performance of the colorization, both in quantifiable terms, such as PSNR, and qualitatively, notably, by reducing color bleeding.

1 Motivation and Introduction

Our work is motivated by the following two applications: First, a number of frescoes painted by Antonio Mantegna were photographed in monochrome before World War II, during which they were very badly damaged [2]. A very small amount of color remains. Image colorization methods might be helpful in restoring these works of art, and others. Second, given a monochrome image, we might be interested in adding small amounts of color to the monochrome image, and having it automatically and coherently propagate to the rest of the image. For example, given a monochrome image of a car, we might be interested in how it would look in different colors. Broadly speaking, these are both instances of the following extension problem.

Suppose that we are given an image domain $\Omega \subset \mathbb{R}^2$ on which a complete grayscale image is given, say $g : \Omega \to \mathbb{R}$, and a subset $D \subset \Omega$ on which the color is given, say $f : D \to \mathbb{R}^n$ (for trichromatic color, $n = 3$). So, $\text{brightness}(f) = g|_D$. Then colorizing the image consists of finding a $F : \Omega \to \mathbb{R}^n$ such that $F|_D \approx f$. We can cast this problem in variational terms, as finding the minimizer of the energy functional:

$$\inf_{F \in X_1(\Omega)} \gamma \mathcal{F}_1(F) + \mathcal{F}_2(F - f)$$

(1)

$\mathcal{F}_1$ and $\mathcal{F}_2$ are respectively functionals on Banach spaces $X_1(\Omega)$ and $X_2(\Omega)$ [2]. The first term is often called the regularization term, and the second the fidelity term. The fidelity term enforces $F \approx f$, while the regularization term imposes certain constraints on the behavior of $F$. Ha-Quang, Kang and Le propose using reproducing kernel Hilbert spaces (RKHS) to model $X_1(\Omega)$, and achieve very realistic results [2]. This paper
is concerned primarily with extending their methods to incorporate the multiscale structure of an image. Section 2 gives a brief overview of the method we are interested in extending, and follows mostly from [2]. Section 3 describes how to implement our approach given a suitable multiscale representation of the image. With that in mind, section 4 describes a method for automatically selecting a few important scales for an image. Section 5 gives the explicit algorithm and a few details of its implementation. Finally, section 6 summarizes the results of incorporating the multiscale structure into the colorization.

2 Method of Ha-Quang, Kang and Le

2.1 RKHS

The theory of Reproducing Kernel Hilbert Spaces is very rich. For background on it, and its applications to the problem at hand, see [2]. For completeness, we include the definition of a RKHS. Suppose that \( K: D \times D \to \mathbb{R} \) is a positive definite kernel on \( D \). The Reproducing Kernel Hilbert Space with kernel \( K \), denoted by \( \mathcal{H}_K \), is the unique Hilbert space of functions \( f: D \to \mathbb{R} \) such that:

1. \( K(x, \cdot) \in \mathcal{H}_K \) for all \( x \in D \)
2. \( \text{span}\{K(x, \cdot)\}_{x \in D} \) is dense in \( \mathcal{H}_K \)
3. \( f(x) = \langle f, K(x, \cdot) \rangle_{\mathcal{H}_K} \)

The last property is known as the reproducing property [2].

2.2 The Method

If \( D \) is a discrete set, say \( D = \{x_i\}_{i=1}^m \), and we are given \( f(x_i) \) for each \( i \). Then the energy minimization problem becomes [2]:

\[
F_\gamma = \arg\min_{F \in \mathcal{H}_K(\Omega)} \frac{1}{m} \sum_{i=1}^m \|F(x_i) - f(x_i)\|_W^2 + \gamma \|F\|_{\mathcal{H}_K(\Omega)}^2
\]  

(2)

Here, \((W)\) is some Hilbert space. This is just the discrete version of the problem as given in 1. For our purposes, the following is the key result.

**Proposition 1.** The unique solution \( F_\gamma \) of 2 is given by:

\[
F_\gamma(x) = \sum_{i=1}^m K(x, x_i) a_i
\]  

(3)

where \( a_i \in W \) satisfy:

\[
\sum_{j=1}^m K(x_i, x_j) a(j) + m\gamma a_i = f(x_i)
\]  

(4)

We include this for completeness, and do not give a proof here. Interested readers can view Proposition 1 of [2]. To reconstruct an image, we can compute the kernel \( K \), solve the linear system given in 4, and then compute the value at each pixel following 3.
Following Ha-Quang, Kang and Le, the kernel we will consider from here is given by:

\[ k(x, y) = \exp \left( -\frac{g(\vec{x}) - g(\vec{y})}{2\sigma_1(2r + 1)p} \right) \exp \left( -\frac{f - f'}{\sigma_2(\sqrt{N^2 + M^2})^p} \right) \] (5)

Here, the image is of size \( N \times M \), and \( \vec{x} \) and \( \vec{y} \) are \( r \times r \) patches around \( x \) and \( y \) respectively. The first term enforces closeness of colors, while the second term enforces spatial closeness regularity, and the parameters \( \sigma_1 \) and \( \sigma_2 \) balance their respective contributions to the kernel. If the color given is highly localized, then we will want \( \sigma_2 >> \sigma_1 \). For implementation efficiency reasons, we will almost always use \( p = 1 \) and \( r = 0 \) (that choice allows us to vectorize the algorithms, gaining a speedup of approximately 100 times). Finally, we use the chrominance-brightness model for representing color, as it tends to give better results than RGB [2]. We may now discuss how to incorporate the multiscale structure of the image into the construction of \( K \).

### 3 Multiscale Approach

We propose to modify the approach of Ha-Quang, Kang and Le by incorporating the multiscale structure of the image into the construction of the kernel \( K \). Suppose that \( \{g_1, \ldots, g_r\} \subset \{g_i\} \) is a multiscale representation of the monochrome image \( g \). At each scale \( i \) and corresponding \( g_i \) and parameters \( p_i, r_i, \sigma_{i,1}, \sigma_{i,2}, \) compute the kernel \( K_i \) as usual.

\[ K_i(x, y) = \exp \left( -\frac{g_i(\vec{x}) - g_i(\vec{y})}{2\sigma_{i,1}(2r + 1)p_i} \right) \exp \left( -\frac{f - f'}{\sigma_{i,2}(\sqrt{N^2 + M^2})^p_i} \right) \] (6)

Then take:

\[ K = K_1K_2\ldots K_r \] (7)

\[ F = \inf u \{ \gamma \|u\|^2_{H^s(\Omega)} + \|u - f\|^2_{L^2(D)} \} \] (8)

The motivation for considering the colorization problem at multiple scales is that color is more coherent within scales than across them. Given close objects of similar brightness, the single-scale approach will lead to color bleeding, while the multiscale approach will maintain the color differences (as they will be different at some scale). Furthermore, while the human eye does not distinguish very well between colors of the same brightness, it can determine changes in color rather well. Hence, a colorized image that is mostly correct, but with some color bleeding will appear noticeably different. Consider the colorization in figure 1. The color bleeds noticeably from the red object in the center to the turquoise texture around it. However, if we select a few important local scales and perform the multiscale colorization, the result is sharp, as in figure 2. The second scale contributes almost nothing to the kernel as its corresponding kernel is very close to 1. However, notice that at the first scale, the regions are far more distinct than in the grayscale version of the original. This is what leads to the sharper color in the colorization.

Of course, this “toy” example is synthetic in two ways: the brightness of the background texture and the foreground object are very close (so color bleeding will occur), and the image has two readily identifiable scales. Indeed, for more complicated images, it is not immediately clear how to choose the important scales. The theory of \( \delta \)-regular scales [1] will allow us to choose important scales algorithmically.
Figure 1: From left to right: The original image; the given color; the result using single-scaled RKHS approach. All results were generated using $r = 0$, $p = 1$, $\sigma_1 = 1$, $\sigma_2 = 5$.

Figure 2: Two important local scales and the colorization using the multiscale kernel (using the same $D$ as above).

4 Automatically Selecting Scales

We will use the theory of $\delta$-relatively-stable scales [1] to select a few important local scales of the image. Following Le [1] and Jones and Le [3], denote by $Sf(t, \theta)$, the family:

$$Sf(t, \theta) = t^{-\theta} \inf_{(u,v) \in BV \times L^1} \{ t \|u\|_{BV} + \|v\|_{L^1} : f = u + v \}$$

(9)

Here, $\|u\|_{BV}$ may be thought of as a regularization term, while $\|v\|_{L^1}$ may be thought of as a fidelity term, and $t$ may be thought of as a tuning parameter that balances the two. We consider this particular family because the term $\inf_{(u,v) \in BV \times L^1} \{ t \|u\|_{BV} + \|v\|_{L^1} : f = u + v \}$ exhibits discontinuities with respect to $t$. That
is, the features of the image are diffused over very short intervals. This regularization also has the important property that it admits very fast and efficient computation of minimizers (using graph cuts) [4].

![Image of fidelity term vs. t](image)

**Figure 3**: The fidelity term $\|u_t - g\|_{L^1}$ vs. $t$ for the image above. Note the sharp increases as the texture and object are removed. See Chan and Esedoglu [5] for the theory of this regularization.

### 4.1 $\delta$-relatively-stable scales

Now denote by $T_{f,\theta}$ the set of local maxima of $S_f(t, \theta)$.

![Image of Sf for a given image](image)

**Figure 4**: An example of $S_f$ for a given image

Following Le [1], also define the following:

- $T_f = \cup_{\theta \in (0,1)} T_{f,\theta}$. These are the **global interpolating scales** of $f$.
- Call $t \in T_f$ a **stable scale** if $t \in T_{f,\theta}$ for all $\theta \in I$ for some $I \subset (0,1)$, $I > 0$
- For $t \in T_f$, let $I_{f,t}$ be that largest $I$ such that the above property holds. If $|I_{f,t}| > \delta$, say that $t$ is **$\delta$-absolutely-stable**.
- Let $\delta^* = \sup \{|I_{f,t}| : t \in T_f\}$. Say that $t \in T_f$ is **$\delta$-relatively-stable** if $|I_{f,t}|/\delta^* > \delta$

For example, for $\delta = .5$, the $\delta$-relatively-stable scales are those such that no scale is a local maximum for twice as many values of $\theta \in (0,1)$. 5
As the definition of $\delta$-relatively-stable scales (and figure 5(b)) makes clear, there are only a few $\delta$-relatively-stable scales for a given image, and $\delta$ sufficiently large. This makes them very well suited to act as important scales for an image. We are now in a position to automatically colorize an image using the multiscale approach.
Figure 5: An example of $\delta$-relatively-stable scales for $\delta = .5$

Figure 6: An example of automatically generated scales for girl.jpg, $\delta = .5$
5 Colorization Algorithm

5.1 The Algorithm

We present the following algorithm to colorize an image.

Multiscale Colorization

Input: The image \( f \), parameter \( \delta \), mask \( D \)

Output: The colorized Image

begin

\textbf{comment}: Initialize the values

\( g := \text{grayscale}(f) \)
\( n := \text{height}(g), m := \text{width}(g) \)
\( t := [-50, \ldots, 250] \)
\( \tau := 1.02 \cdot t \)

\textbf{comment}: Get the multiscale decomposition

\textbf{for} \( i := 1 \) to \( 301 \) \textbf{step} 1 \textbf{do}

\( \text{scales}[i] := \inf_u \left\{ \frac{1}{\tau[i]} \| u \|_{BV} + \| g - u \|_{L_1} \right\} \)

\textbf{end}

\textbf{comment}: Select a few important scales

\( \delta\text{scales} := \delta\text{-relatively-stable scales of } f \)
\( \text{lows} := -50 :: \delta\text{scales} \)
\( \text{highs} := \delta\text{scales :: 250} \)
\( \text{midpoints} := (\text{highs} + \text{lows})/2 \)

\textbf{comment}: Generate the kernel

\( \text{multikernel} := 1_{n \times m} \)

\textbf{for} \( t \in \delta\text{scales} \)

\( \text{kernel} := \text{compute the kernel for scales}[t+51] \)

\( \text{multikernel} := \text{kernel} .* \text{multikernel} \)

\textbf{end}

\textbf{comment}: Finish

Following section 2, restore the image using the normal method on \text{multikernel}

end
5.2 Implementation Details and Optimizations

5.2.1 Logarithmic Discretization

In the above algorithm, the scales are discretized logarithmically (i.e. the tuning parameter of the energy functional is given by $1/1.02^t$). The reason for this is that $S_f(t, \theta)$ satisfies the following regularity property:

**Proposition 2.** For every $\theta \in (0, 1)$, $S_f(t, \theta)$ is continuous and differentiable almost everywhere. Furthermore, suppose that the derivative exists. Let $\tau = \log_a(t)$, $a > 1$. Then:

$$\frac{\partial}{\partial \tau} S_f(\tau, \cdot) \leq C_1 \|f\|_{(BV, L^1)_{\theta, \infty}}, \quad C_1 = \frac{2}{\ln(a)}$$

(10)

For a proof of this regularity property, see for example [1] or [3]. We will not reproduce the proof here, and will only note that as a result of the theorem, a logarithmic discretization is the right way to go about searching for the local maxima. The maxima may be very densely packed with respect to a uniform discretization. However, using a logarithmic discretization, they satisfy the regularity condition in equation 10.

![Figure 7: $S_f(t, \theta)$, but uniformly discretized. Notice the high levels of oscillation and compare with figure 4(b)](image)

5.2.2 Taking midpoints

The $\delta$-relatively-stable scales represent the scales at which the image is changing fastest, while the areas between them represent scales at which the image is stable. Hence, the areas between the $\delta$-relatively-stable scales can be thought of as the image with certain features removed. For that reason, we compute the kernel at the midpoints between the $\delta$-relatively-stable scales. This is a purely empirical result; the results seem to

---

Here, $\| \cdot \|_{(BV, L^1)_{\theta, \infty}}$ is the norm on the Banach interpolating space $(BV, L^1)_{\theta, \infty}$, which is given by:

$$\|f\|_{(BV, L^1)_{\theta, \infty}} = \sup_{0 < t < \infty} |S_f(t, \theta)|$$

For more information, see Le [1]. Jones and Le [3] consider a homogeneous Triebel-Lizorkin function space, $F^{\alpha}_{p, \infty}(\mathbb{R}^2)$, and arrive at a similar result.
be better using the midpoints. Whenever $\delta$-relatively-stable scales are referenced, it will mean the midpoints between the scales.

### 5.2.3 Optimizations

The algorithm given is unoptimized. If implemented directly, it would be prohibitively slow. To implement the algorithm, note that the system of equations in equation 4 requires the kernel $K(x, y)$ only for values $(x, y) \in D \times D$, while the reconstruction in equation 3 requires only the kernel $K(x, y)$ for values $(x, y) \in \Omega \times D$. Thus, we need only calculate $K_{\Omega \times D}$ in order to colorize the image. Typically, $\frac{\Omega}{D} < .005$, so only computing $K_{\Omega \times D}$ entails a speedup of over 200 times.

We can also speed up the algorithm by looking at fewer scales, for example by beginning with a coarser discretization and then progressively refining it.

### 6 Discussion and Results

#### 6.1 Future Directions

Before we discuss the performance of the algorithm, a few remarks on possible future directions of research. First, only one kernel was considered here, and it would be worthwhile to investigate whether other kernels exhibit similar improvements when incorporating the multiscale structure of the image. Moreover, although the results are encouraging, the method of automatically selecting scales requires more research. From a performance standpoint, generating a multiscale representation for an image is very time-consuming, and optimizing it would remove a significant bottleneck in the process. Finally, the methods were only tested for small sets of random points. An appealing application of the colorization methods is to transfer color from one image to another, or to draw “squiggles” of color on a monochrome image, and to have the color propagate coherently to the rest of the image ([2] and [6]). Because of time constraints, these applications were not explored, but we suspect that they would be well-suited to a multi-scale approach.

#### 6.2 General Performance

We compare the performance of the multiscale approach with the single scale approach. The multiscale colorization outperforms the usual method almost universally, and often by very significant margins, for example on persian_rug1.jpg (on a very small number of domains, the usual method performs better on carpet2). For each image, progressively larger sets of random pixels were computed, and both algorithms ran. To show the robustness of the results, the value of $\delta$ was changed between images. It is worthwhile to keep in mind that the only parameters specified were the mask and the value $\delta$. All scales were selected automatically.
Figure 8: RMSE and PSNR for carpet2.jpg vs. ID/$Ω$.

Figure 9: RMSE and PSNR for ferrari2.jpg vs. ID/$Ω$. 

11
6.3 Color Coherence and Color Bleeding

Because color tends to be coherent within scales but usually not across them, the multiscale approach prevents color bleeding significantly better than the single-scaled approach. This is important, because given two images with similar peak-signal-noise-ratios (or RMSEs), the presence of color bleeding in one can make it appear less like the original than the other. In all of the following examples, the parameters given were: $p = 1, r = 0, \sigma_i,1 = 1, \sigma_i,2 = 5 \forall i$, though the results hold for other choices as well.
Figure 12: Clockwise from top: The original image, the given mask (keeping only .1% of the color, the colorization using the multi-scale method, the colorization using the original method. The multiscale colorization is much sharper.
Figure 13: Images are respectively the same as in figure 12. Notice that although the RMSEs are very close (21.6 vs. 20.1), the colorization using the multiscale approach is noticeably more realistic.
Figure 14: The same image, with different masks. Notice the difference between the colorization near the edge.
Figure 15: The same image, with different masks. Notice the difference between the colorization near the edge.
Figure 16: Colorization for persian_rug1.jpg. The multiscale colorization is markedly better.
Figure 17: The multiscale colorization is reasonable (especially given only .01% of the color) while the normal colorization is not.
Figure 18: The colors in the fur bleed in the single-scale colorization, but not in the multiscale one.
7 Acknowledgements

I would like to thank Professors Triet M. Le and Daniel Spielman for their generous help throughout this process.

References


