

# IMAGE INTERPOLATION USING SPATIAL AND SPECTRAL CORRELATIONS

*Wenmiao Lu and Yap-Peng Tan*  
Division of Information Engineering  
School of Electrical and Electronic Engineering  
Nanyang Technological University  
Email: {ewmlu, eyptan}@ntu.edu.sg

## ABSTRACT

This paper presents a new method to interpolate digital color images by exploiting the spatial and spectral correlations inherent in the images. The proposed method involves two main steps: interpolating green image plane solely based on the local spatial correlation within the color plane, and interpolating red and blue image planes based on the spectral correlation between different color image planes. The performance of the proposed algorithm is evaluated by interpolating down-sampled color images back to their original image sizes and comparing the results with the original images as well as those generated by other existing image interpolation schemes. Experimental results show that the proposed method can better preserve valuable image attributes: edge sharpness and color consistency, which have great influence on the perceived visual quality of the interpolated images.

## 1. INTRODUCTION

Image interpolation is often required in many imaging applications for enlarging images or restoring low-resolution images to their original sizes. These include image zooming in digital imaging devices (such as digital camera, printer and scanner) and image sharing across heterogeneous, distributed multimedia systems [1]. As the reverse process of image down-sampling, a common means to reduce the spatial resolution of images for display, storage and transmission purposes, image interpolation aims to produce high-resolution images from their down-sampled versions.

We present in this paper a new and improved method for interpolating color images to higher spatial resolution. The proposed method manages to generate interpolated images that can better preserve the sharpness of image edges and contain less interpolation artifacts, such as color bleeding. The method involves two main steps: interpolating the green image plane solely based on the local spatial correlation and interpolating the red and blue color planes based on the spectral (inter-color plane) correlation. Although we present our method in this paper using RGB (red-green-blue) color images, each of which consists of red, blue, and green image planes, it can be readily applied to color images recorded in different color

spaces or color components, such as  $YCbCr$  color space or Yellow-Cyan-Magenta color components.

The rest of the paper is organized as follows. Section 2 provides a review of background theories and prior work on image interpolation. The overview of the proposed image interpolation algorithm is presented in Section 3. Section 4 describes the local spatial correlation based method for interpolating green image plane. Section 5 addresses the inter-color plane correlation-based method for interpolating red and blue image planes. The experimental results and their comparison with images produced by other interpolation methods are reported in Section 6. In Section 7, we conclude the paper.

## 2. REVIEW OF BACKGROUND THEORIES AND PREVIOUS WORK

Image interpolation, also known as image enlargement or image magnification, is normally used to increase the size or spatial resolution of images. Owing to its wide uses in many imaging applications, many linear and nonlinear image interpolation techniques have been proposed in the literature. Roughly, these techniques can be grouped into the following five main categories:

1. Single kernel based techniques that involve a convolution between low-resolution images and a 2D interpolation kernel, for example [1].
2. Multiple kernel based techniques that work with a set of pre-defined interpolation kernels, for example [2].
3. Transform based techniques that interpolate images in a transform domain using transform such as discrete cosine transform (DCT) or wavelet transform, for example [3].
4. Optimization based techniques that formulate image interpolation as a constrained optimization problem minimizing a specific cost function, for example [4].
5. Edge adaptive techniques that examine local image edge content and interpolate images in low frequency direction (along the edge), rather than in high frequency direction (across the edge), for example [5][6].

Brief summaries of these five categories of image interpolation techniques are provided in the following to serve as the literature review of our work.

## 2.1 Single kernel based image interpolation

This type of interpolation involves a convolution between the observed image samples and a 2D kernel. The three most commonly used convolution kernels for image interpolation are the zero-order-hold, bilinear and bicubic kernels respectively.

Zero-order-hold kernel is the simplest. Interpolation with this kernel is commonly referred to as nearest neighbor interpolation or pixel replication. Using this kernel, each unknown pixel is filled with the existing image pixel closest to its location. Due to this pixel replication, the zero-order-hold kernel usually produces images with a very ‘blocky’ image appearance. However, it is computationally simple and easy to be implemented.

Bilinear interpolation kernel is composed of two 1D linear interpolation kernels. The kernel involves the four image pixels nearest to the pixel to be interpolated. It gives smoother image appearance compared to that of zero-order interpolation and demands moderate computational cost.

Bicubic interpolation kernel locally fits a separable third-degree polynomial through sixteen adjacent image pixels. The fitting constraints tend to limit the amount of oscillation across an image edge. It gives smooth image appearance and maintains some sharp details. However, it is the most computational expensive among these three common kernels.

## 2.2 Multiple kernels based image interpolation

Multiple kernels based schemes examine local image structures in order to select an appropriate interpolation kernel for the pixel to be interpolated [2]. One set of kernels is usually pre-defined for the image interpolation, and the intensity structure of a local image neighborhood around the unknown pixel under consideration is analyzed to determine the most suitable interpolation kernel to be used.

## 2.3 Orthogonal transform based image interpolation

Orthogonal transforms, such as discrete cosine transform (DCT) and wavelet transform, have often been used for image interpolation. Image interpolation using wavelet transform exploits the fact that the signal evolution across different resolution scales can be used to predict the image detail at finer scales. For DCT based interpolation, previous work has implemented digital filters to perform the required spatial domain filtering in the DCT domain. For example, Hong [3] uses the local DCT coefficients to identify five different types of image edges. Once the edge type is determined, a single linear kernel corresponding to the detected edge type is used to interpolate the image.

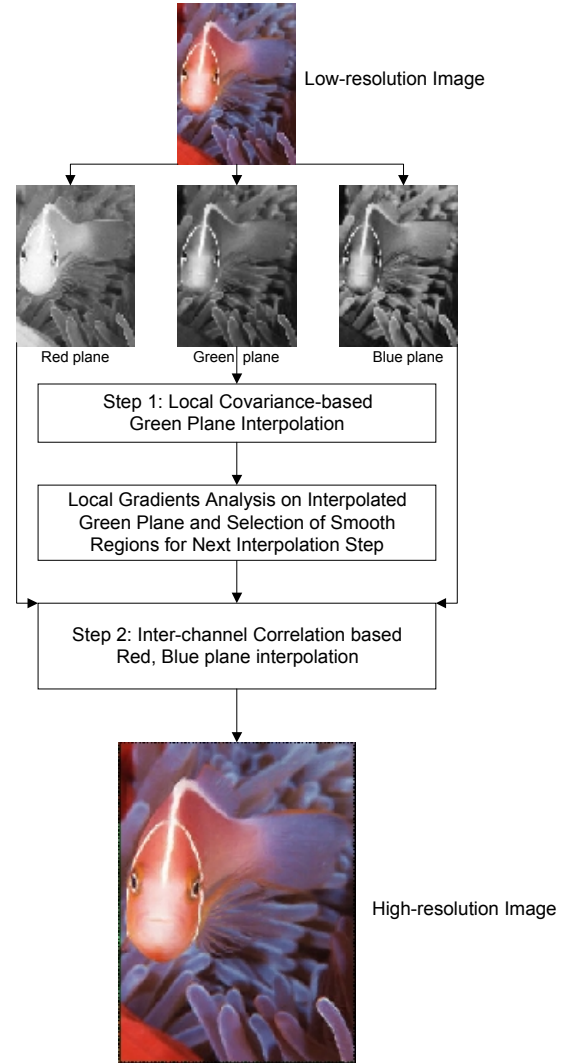


Figure 1: Overview of the proposed interpolation scheme.

## 2.4 Optimization based image interpolation

Optimization techniques formulate image interpolation as a constrained optimization problem minimizing a specific cost function. The choice of the cost function to be minimized is based on a combination of the restrictions imposed by certain considerations corresponding to the physical problem.

One representative work done by Ruderman and Bialek [4] shows that if the statistical property of a signal is known a priori, the observed signal samples can be used to estimate information beyond the Nyquist limit. They consider the interpolation a process of finding the “best” estimate of the original signal given the original signal statistics and the observed signal samples. They show that the optimal interpolation procedure for a Gaussian signal results in a convolution of the signal samples with an

interpolation kernel that is a function of the noise variance, the power spectrum, and the sampling rate of the original signal.

### 2.5 Edge adaptive image interpolation

Edge adaptive interpolation techniques recognize the fact that it is desirable to interpolate image along the edge instead of across it. Jensen and Anastassiou [5] propose a method that detects the presence of high contrast edges and then interpolate the image along the edges. Bayrakeri and Mersereau propose a weighted directional interpolation scheme [6], where the values interpolated along various directions are combined with appropriate weightings based on the variations in different directions.

### 2.6 Remarks

Due to their computational simplicity, image interpolation techniques with the use of a single linear interpolation kernel have been widely engaged in many imaging applications. However, natural images consist of multifold objects. The intensity of an image often changes abruptly across objects' boundaries. Thus, applying convolution with a linear interpolation kernel indiscriminately throughout an image irrevocably blurs the image edges and introduces artifacts around them. To resolve this problem, many nonlinear interpolation techniques that perform interpolation adapted to local image structures have been proposed. Although some of them are well established for grayscale images, a direct extension of these techniques to color images may lead to undesirable interpolation artifacts, such as color bleeding. Besides, this also incurs unnecessary high computational cost, as the inherent correlation between different color image planes is not exploited [7]. With these considerations in mind, we have developed a new method for color image interpolation.

## 3. ALGORITHM OVERVIEW

Figure 1 shows the overview of the proposed algorithm for interpolating low-resolution color images. It consists of two steps. First, the green image plane of a low-resolution color image is interpolated to the desired spatial resolution by exploiting the local spatial correlation within the green image plane. The fully interpolated green image plane is subject to gradient analysis in order to identify local image regions that are suitable for interpolating the red and blue pixel values. Second, within the selected image regions, the unknown red and blue values are interpolated based on the spectral correlation between different color image planes. The details of the proposed algorithm are presented in the following two sections.

## 4. GREEN IMAGE PLANE INTERPOLATION USING SPATIAL CORRELATION

The first step of the proposed algorithm is to fully interpolate a high-resolution green plane. As the green plane of a RGB image contains most of the spatial details of the captured scene, it should be interpolated with closer attention in order to reconstruct as many image details as possible. This step is an extension of Li's method [8], which is able to produce interpolated images with attractive perceptual quality.

The step is based on the classical Wiener filtering that is well known to be capable of obtaining the optimal estimate of unknown data from a Gaussian random process [9]. Although it is computationally easier to assume that image intensities of a local spatial neighborhood tend to be stationary and can be well modeled with Gaussian distribution, this is often not the case for natural images. Natural images are usually populated with edges that can be better characterized by abrupt changes of local statistics. In other words, the classical Wiener filtering may not be able to produce optimal results around or across image edges. To cope with this problem, the proposed algorithm interpolates the high-resolution pixel from a group of pre-selected neighboring pixels, which and the pixel to be interpolated (also referred to as the unknown pixel) can be well modeled with a stationary Gaussian process. By performing edge detection within a local neighborhood, only the pixels residing at the same side of the edge as the pixel to be interpolated are engaged in the interpolation process. This is the main difference from Li's method, which directly applies a covariance-based interpolator using all the available pixels around the pixels to be interpolated.

Let us illustrate the idea with an example. In Figure 2, the low-resolution green image plane  $G_{i,j}$  and the interpolated green image plane  $\tilde{G}_{i,j}$  of the image under consideration are living on an  $H \times W$  lattice and an  $mH \times nW$  lattice respectively, where  $m$  and  $n$  are the up-scaling factors in the vertical and the horizontal image dimensions, respectively. Although our algorithm can be used to up-sample images by arbitrary integer factors, we shall describe the algorithm using the case of  $m = n = 2$  and  $\tilde{G}_{2i,2j} = G_{i,j}$ . The low-resolution green pixels, denoted by filled circles, are separated by an edge as illustrated in Figure 2. By examining the locations of the unknown pixel and the neighboring low-resolution pixels around the image edge detected, our proposed algorithm determines the side of the edge at which the unknown pixel lies. (Note that the diameters of the filled circles are altered to represent the values of the green pixels.)

The unknown green pixel value at  $\tilde{G}_{2i+1,2j+1}$  is linearly estimated from its four nearest neighboring pixels (Figure 3: case 1) with equation

$$\tilde{G}_{2i+1,2j+1} = \sum_{k=0}^1 \sum_{l=0}^1 \alpha_{2k+l} G_{2(i+k),2(j+l)} \quad (1)$$

where  $\bar{\alpha} = [\alpha_1, \alpha_2, \alpha_3, \alpha_4]^T$  are proper weightings for the neighboring pixels. By using Wiener filter, the weightings  $\bar{\alpha}$  can be optimally estimated, in the sense of minimum mean-square error. It is equivalent to solving the following over-constrained linear equations:

$$\bar{g} = C^T \times \bar{\alpha} \quad (2)$$

$$\bar{\alpha} = R^{-1} \times \bar{r} \quad (3)$$

where  $R = CC^T$  is the local covariance matrix and  $\bar{r} = C\bar{g}$  is the covariance vector for the low-resolution pixels selected. The vector  $\bar{g} = [\dots g_k \dots]^T$  contains the green values of the low-resolution pixels, which locate at the same side of the edge as the high-resolution pixel  $\tilde{G}_{2i+1,2j+1}$  to be interpolated. From this, the matrix  $C$  whose  $k^{\text{th}}$  column consists of the green values of four nearest low-resolution interpolating neighbors of  $g_k$  can hence be constructed.

To obtain the complete high-resolution green plane, we apply the same algorithm to construct pixel  $\tilde{G}_{2i+1,2j}$  of the partially filled lattice. The only difference is that a diamond-shaped local window, as shown in Figure 3: case 2, is used to estimate the unknown pixel value. When no edge is found within a local image neighborhood, we revert to bilinear interpolation for the sake of simplicity.

In comparison to Li's method, the proposed technique can better preserve the fine details of image, such as short edges and isolated dots. Li applied a template of fixed size throughout the whole image and used all the low-resolution pixels inside the local windows to reconstruct the high-resolution images. It works fine for the long curved edges by smoothing along them. However, Li's method will over-smooth the fine image details. Inside the local windows around these fine image details, most of the neighboring pixels, which dominate the interpolation process, possess the statistical properties different from that of the pixel to be interpolated. As a result, those fine image details are lost in the reconstructed images. On the contrary, our proposed method avoids this problem by pre-selecting similar neighboring pixels. Not only is the smoothness of the long curved edges ensured, but also the sharpness of the short edges is preserved. This can be clearly seen from Figure 4, where interpolation result generated by Li's

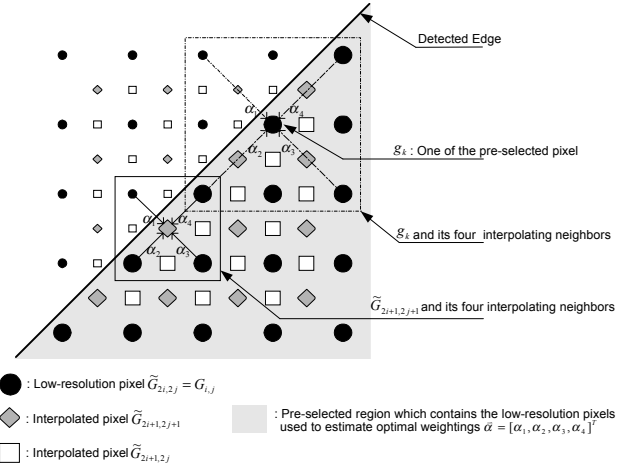


Figure 2: Lattices for image interpolation.

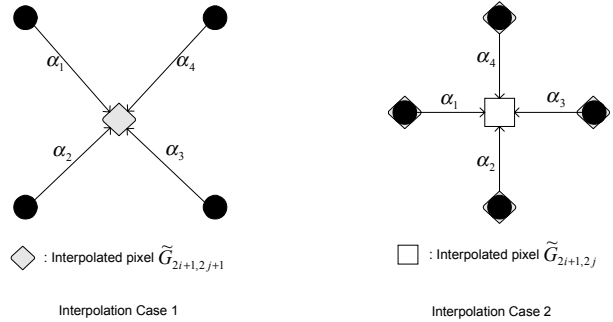


Figure 3: Pixel to be interpolated and its four nearest neighboring pixels.

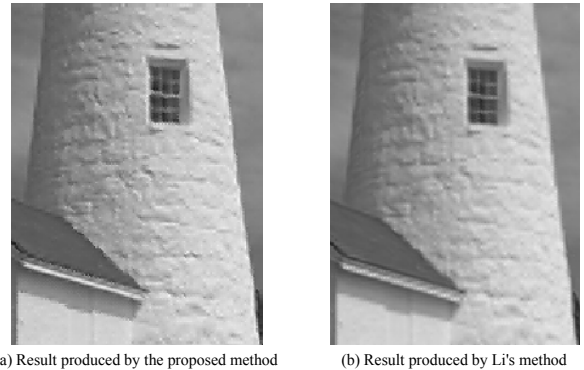


Figure 4: The images interpolated by (a) the proposed method and (b) Li's method.

method (posted on the web site of signal processing laboratory at Cornell University [11]) and the result generated by our proposed method are presented for visual comparison.

Although the adaptive selection of neighboring pixels and linear estimation based on Wiener filtering incur relatively expensive computational cost, the interpolation results give superior edge preservation, which is critical to the following interpolation of red and blue image planes.

## 5. RED AND BLUE IMAGE PLANE INTERPOLATION USING SPECTRAL CORRELATION

The red and blue image planes are interpolated by analyzing the fully interpolated green image plane. If the spatial correlation-based interpolator for the green plane is applied to up-sample the red and blue planes separately, very expensive computational cost will be incurred. Inspired by Chang's color interpolation scheme for the Bayer's CFA (Color Filter Array) pattern [10], we contrive an interpolator that exploits the spectral correlation between different color planes to construct the high-resolution red and blue image planes with less computation requirement and better color consistency. There are two underlying assumptions. First, the interpolated green plane contains sufficient gradient information for the complete color image. Second, within a given local image region, the differences between the red and green values of the neighboring image pixels are highly similar; this is also the case for the differences between the blue and green pixel values. The first assumption holds for most natural images, since the green image plane normally records the scene's spectral energy in a way similar to the response of the human visual system. The second assumption is decent for most image regions and only fails with sharp image edges.

To handle the exception due to sharp edges, only pixels in smooth image regions should be engaged in estimating the unknown red and blue pixel values. Specifically, the smooth image regions refer to those regions whose pixels have color values similar to the unknown pixel. These regions can be reasonably identified by using the local gradient information obtained from the interpolated green plane. Low gradient values indicate that the pixels have the similar color values, whereas high gradient values suggest that the pixels are from image regions with many details or sharp edges.

The computation of local image gradients is performed within a 5-by-5 local window of the interpolated green image plane, as shown in Figure 4. The window is divided into eight regions corresponding to the eight directions (N, E, S, W, NE, SE, SW, NW).

$G_1$	$G_6$	$G_{11}$	$G_{16}$	$G_{21}$
$G_2$	$G_7$	$G_{12}$	$G_{17}$	$G_{22}$
$G_3$	$G_8$	$G_{13}$	$G_{18}$	$G_{23}$
$G_4$	$G_9$	$G_{14}$	$G_{19}$	$G_{24}$
$G_5$	$G_{10}$	$G_{15}$	$G_{20}$	$G_{25}$

Figure 4: A 5-by-5 local window around pixel  $G_{13}$  of the interpolated green image plane.

$G_1$	$G_6$	$G_{11}$	$G_{16}$	$G_{21}$
$G_2, R_2, B_2$	$G_7$	$G_{12}, R_{12}, B_{12}$	$G_{17}$	$G_{22}, R_{22}, B_{22}$
$G_3$	$G_8$	$G_{13}, R_{13}, B_{13}$	$G_{18}$	$G_{23}$
$G_4, R_4, B_4$	$G_9$	$G_{14}, R_{14}, B_{14}$	$G_{19}$	$G_{24}, R_{24}, B_{24}$
$G_5$	$G_{10}$	$G_{15}$	$G_{20}$	$G_{25}$

(1) Type A Location

$G_1$	$G_6$	$G_{11}$	$G_{16}$	$G_{21}$
$G_2$	$G_7, R_7, B_7$	$G_{12}$	$G_{17}, R_{17}, B_{17}$	$G_{22}$
$G_3$	$G_8$	$G_{13}, R_{13}, B_{13}$	$G_{18}$	$G_{23}$
$G_4$	$G_9, R_9, B_9$	$G_{14}$	$G_{19}, R_{19}, B_{19}$	$G_{24}$
$G_5$	$G_{10}$	$G_{15}$	$G_{20}$	$G_{25}$

(2) Type B Location

$G_1$	$G_6, R_6, B_6$	$G_{11}$	$G_{16}, R_{16}, B_{16}$	$G_{21}$
$G_2$	$G_7$	$G_{12}$	$G_{17}$	$G_{22}$
$G_3$	$G_8, R_8, B_8$	$G_{13}, R_{13}, B_{13}$	$G_{18}, R_{18}, B_{18}$	$G_{23}$
$G_4$	$G_9$	$G_{14}$	$G_{19}$	$G_{24}$
$G_5$	$G_{10}, R_{10}, B_{10}$	$G_{15}$	$G_{20}, R_{20}, B_{20}$	$G_{25}$

(3) Type C Location

Figure 5: Three types of locations where the pixel to be interpolated can reside.

The local image gradient in each direction is computed by equations:

$$Gradient\_N = |G_{13} - G_{11}| + |G_{14} - G_{12}| + |G_8 - G_6| / 2 + |G_{18} - G_{16}| / 2;$$

$$Gradient\_E = |G_{13} - G_{23}| + |G_8 - G_{18}| + |G_{12} - G_{22}| / 2 + |G_{14} - G_{24}| / 2;$$

$$Gradient\_S = |G_{13} - G_{15}| + |G_{12} - G_{14}| + |G_{18} - G_{20}| / 2 + |G_8 - G_{10}| / 2;$$

$$Gradient\_W = |G_{13} - G_3| + |G_{18} - G_8| + |G_{14} - G_4| / 2 + |G_{12} - G_2| / 2;$$

$$Gradient\_NE = |G_{13} - G_{21}| + |G_9 - G_{17}| + |G_8 - G_{16}| / 2 + |G_{14} - G_{22}| / 2;$$

$$Gradient\_NW = |G_{13} - G_1| + |G_{19} - G_7| + |G_{18} - G_6| / 2 + |G_{14} - G_2| / 2;$$

$$Gradient\_SE = |G_{13} - G_{25}| + |G_7 - G_{19}| + |G_8 - G_{20}| / 2 + |G_{12} - G_{24}| / 2;$$

$$Gradient\_SW = |G_{13} - G_5| + |G_{17} - G_9| + |G_{12} - G_4| / 2 + |G_{18} - G_{10}| / 2;$$

$$Local\_gradients = \{Gradient\_N, Gradient\_E, Gradient\_S, Gradient\_W, Gradient\_NE, Gradient\_NW, Gradient\_SE, Gradient\_SW\}$$

(4)

where the image gradient in each region is computed by accumulating the absolute differences of green pixel values in the respective region. Instead of using two adjacent pixels, the difference between two alternate neighboring green pixels is computed to better capture the scene variation. The absolute differences are also properly weighted with weighting 1 or 0.5 according to their distances from the pixel under consideration.

Once we have obtained the local image gradient in each direction, we identify the smooth image regions by comparing each gradient magnitude to a threshold defined as  $T = \text{median}\{\text{local\_gradients}\}$ , where *local\_gradients* is the set of gradient magnitudes in the eight directions. Those regions with gradient magnitudes below this threshold are considered smooth. The pixels inside these smooth regions, which have complete RGB color values, are used to estimate the differences between the green pixel value and the unknown red/blue pixel value.

Since the red and blue image planes are to be interpolated by a factor of two in each image dimension, there are three different cases, which depend on the location of the pixel to be interpolated, for estimating the differences between color values. The three types of locations are shown in Figure 5, where the red and blue color values of  $G_{13}$  are to be estimated.

Let us use an example to illustrate how to compute the unknown color values  $R_{13}$  and  $B_{13}$ . For instance, if the image local gradients computed from regions {S, W, SE, SW} are less than the threshold  $T = \text{median}\{\text{local\_gradients}\}$ , the pixels which have complete RGB color values and locate inside the regions {S, W, SE, SW} will be used to estimate  $R_{13}$  and  $B_{13}$ .

For Type A location shown in Figure 5, the following table indicates the pixels used for estimating  $R_{13}$  and  $B_{13}$ .

	<i>Red</i>	<i>Green</i>	<i>Blue</i>
<b>S</b>	$R_{14}$	$G_{14}$	$B_{14}$
<b>W</b>	$(R_2 + R_4 + R_{12} + R_{14})/4$	$(G_2 + G_4 + G_{12} + G_{14})/4$	$(B_2 + B_4 + B_{12} + B_{14})/4$
<b>SE</b>	$(R_{14} + R_{24})/2$	$(G_{14} + G_{24})/2$	$(B_{14} + B_{24})/2$
<b>SW</b>	$(R_4 + R_{14})/2$	$(G_4 + G_{14})/2$	$(B_4 + B_{14})/2$

The average color values for pixels in these four regions are computed as follows:

$$\begin{aligned}
 R_{ave} &= [R_4 + (R_2 + R_4 + R_{12} + R_{14})/4 + (R_{14} + R_{24})/2 + (R_4 + R_{14})/2]/4; \\
 G_{ave} &= [G_4 + (G_2 + G_4 + G_{12} + G_{14})/4 + (G_{14} + G_{24})/2 + (G_4 + G_{14})/2]/4; \\
 B_{ave} &= [B_4 + (B_2 + B_4 + B_{12} + B_{14})/4 + (B_{14} + B_{24})/2 + (B_4 + B_{14})/2]/4;
 \end{aligned} \tag{5}$$

For Type B location shown in Figure 5, the following table indicates the pixels used for estimating  $R_{13}$  and  $B_{13}$ .

	<i>Red</i>	<i>Green</i>	<i>Blue</i>
<b>S</b>	$(R_9 + R_{19})/2$	$(G_9 + G_{19})/2$	$(B_9 + B_{19})/2$
<b>W</b>	$(R_7 + R_9)/2$	$(G_7 + G_9)/2$	$(B_7 + B_9)/2$
<b>SE</b>	$R_9$	$G_9$	$B_9$
<b>SW</b>	$R_{19}$	$G_{19}$	$B_{19}$

The average color values for pixels in these four regions are similarly computed as:

$$\begin{aligned}
 R_{ave} &= [(R_9 + R_{19})/2 + (R_7 + R_9)/2 + R_9 + R_{19}]/4; \\
 G_{ave} &= [(G_9 + G_{19})/2 + (G_7 + G_9)/2 + G_9 + G_{19}]/4; \\
 B_{ave} &= [(B_9 + B_{19})/2 + (B_7 + B_9)/2 + B_9 + B_{19}]/4;
 \end{aligned} \tag{6}$$

For Type C location shown in Figure 5, the following table indicates the pixels used for estimating  $R_{13}$  and  $B_{13}$ .

	<i>Red</i>	<i>Green</i>	<i>Blue</i>
<b>S</b>	$(R_8 + R_{10} + R_{18} + R_{20})/4$	$(G_8 + G_{10} + G_{18} + G_{20})/4$	$(B_8 + B_{10} + B_{18} + B_{20})/4$
<b>W</b>	$R_8$	$G_8$	$B_8$
<b>SE</b>	$(R_{18} + R_{20})/2$	$(G_{18} + G_{20})/2$	$(B_{18} + B_{20})/2$
<b>SW</b>	$(R_8 + R_{10})/2$	$(G_8 + G_{10})/2$	$(B_8 + B_{10})/2$

The average color values for pixels in these four regions can be similarly computed as:

$$\begin{aligned}
 R_{ave} &= [(R_8 + R_{10} + R_{18} + R_{20})/4 + R_8 + (R_8 + R_{10})/2 + (R_{18} + R_{20})/2]/4; \\
 G_{ave} &= [(G_8 + G_{10} + G_{18} + G_{20})/4 + G_8 + (G_8 + G_{10})/2 + (G_{18} + G_{20})/2]/4; \\
 B_{ave} &= [(B_8 + B_{10} + B_{18} + B_{20})/4 + B_8 + (B_8 + B_{10})/2 + (B_{18} + B_{20})/2]/4;
 \end{aligned} \tag{7}$$

The unknown red and blue color values are estimated as:

$$\begin{aligned}
 R_{13} &= G_{13} + (R_{ave} - G_{ave}); \\
 B_{13} &= G_{13} + (B_{ave} - G_{ave});
 \end{aligned} \tag{8}$$

Using different spatial information to interpolate each color value of the same pixel is one major source of color bleeding artifacts. By exploiting the spectral correlation between different color planes, we not only evade from the unnecessary heavy computations, but also ensure color consistency and hence reduce color-bleeding artifacts.

## 6. RESULTS

To gauge the efficacy of the proposed method, several color images were filtered and sub-sampled to one-quarter spatial resolution and then interpolated back to their original resolutions. The particular strengths and weakness of each interpolation scheme can be judged by comparing the up-sampled result to the original full-resolution image. Here we compare the results obtained from the proposed method with that of the two most widely used interpolation schemes -- bilinear and bicubic interpolation schemes.



Figure 6 shows a flower test image and its interpolated images produced with different interpolation schemes. To show the differences in detail, Figure 7 presents the enlarged small portions from the original and the interpolated images. A natural scene image containing many high-frequency spatial details and its reconstructed images are shown in Figure 8. Although some high-frequency details are lost in each of the interpolated images, the images produced by the proposed method can better preserve the edge sharpness and color consistency without introducing unpleasant interpolation artifacts.

## 7. CONCLUSION

We have presented a new and improved image interpolation method that can adapt to local image structure for constructing high-resolution color images from their down-sampled versions. Our main contributions include using image spatial correlation to form an edge directed interpolator for constructing a high-resolution green image plane and exploiting spectral correlation between different color planes to up-sample the red and blue color planes. Both steps involve with the searching for neighboring pixels that possess similar statistical characteristics as the pixel to be interpolated. Simulation results show that the proposed method can generate more visually pleasing images compared to that produced by other commonly used interpolation schemes.

## 8. REFERENCES

1. R.G. Keys, "Cubic convolution interpolation for digital image processing", IEEE Trans. Acoust., Speech, Signal Processing, vol. ASSP-29, no. 6, pp. 1153-1160, 1981.
2. Wang, Y. and Mitra, S.K., "Edge Preserved Image Zooming", Fourth European Signal Processing Conf., Grenoble, France. September 5-8, pp. 1445-1448, 1988.
3. Hong, K.P., Paik, J.K., Kim, H.J. and Lee, C.H., "An Edge-Preserving Image Interpolation System for a Digital Camcorder", IEEE Trans. Consumer Electronics, Vol. 42, No. 3, pp. 279-284, 1996.
4. Ruderman, D.L. and Bialek, W., "Seeing Beyond the Nyquist Limit", Neural Computation, Vol. 4, pp. 682-690, 1992.
5. Jensen, K. and Anastassiou, D., "Subpixel Edge Localization and the Interpolation of Still Images", IEEE Trans. Image Processing, Vol. 4, No. 3, pp. 285-295, 1995.
6. Bayrakeri, S.D. and Mersereau, R.M., "A New Method for Directional Image Interpolation", Proc. Int. Conf. Acoustics, Speech, Sig. Processing, Vol. 4, pp. 2383-2386, 1995.
7. H.Nicos and N.Anastasios, "Color Image Interpolation for high resolution acquisition and display devices", IEEE Trans. On Consumer Electronics, Nov 1995.
8. Xin Li and Michael Orchard, "New Edge Directed Interpolation", Int. Conf. On Image Processing, 2000.

9. N.S Jayant and P. Noll, "Digital coding of waveforms: principles and applications to speech and video", Prentice-Hall, 1984.
10. Chang, Ed., "Color Filter Array Recovery Using a Threshold-based Variable Number of Gradients", Proceedings of SPIE, January 1999.
11. Image interpolation web site at Cornell University, <http://www.ee.cornell.edu/~splab/interpolation/>



Figure 6: Original flower image and its interpolated results.



Figure 7: Portions of original and the interpolated images.



Original



Bilinear



Proposed  
Algorithm

Figure 8: Original scene image and its interpolated results.