Demosaicking: Color Filter Array Interpolation in Single-Chip Digital Cameras

B. K. Gunturk, J. Glotzbach, Y. Altunbasak, R. W. Schafer, and R. M. Mersereau

Center for Signal and Image Processing
Georgia Institute of Technology
Atlanta, Georgia, 30332-0250
{bahadir, glotz, yucel, rws, rmm }@ece.gatech.edu

Draft for the IEEE SPM Special Issue on Color Image Processing

1 Introduction

Digital cameras have become popular and many people are choosing to take their pictures with digital cameras instead of film cameras. When a digital image is recorded, the camera needs to perform a significant amount of processing to provide the user a viewable image. This processing includes white balance adjustment, gamma correction, compression and more (reference "Color Image Processing Pipeline in Digital Still Cameras" in this issue). A very important part of this image processing chain is color filter array interpolation or demosaicking.

A color image requires at least three color samples at each pixel location. Computer images often use red, green, and blue. A camera would need three separate sensors to make these measurements. To reduce size and cost, many cameras use a single sensor array with a color filter array. The color filter array allows only one part of the spectrum to pass to the sensor so that only one color is measured at each pixel. This means that the camera must estimate the missing two color values at each pixel. This process is known as demosaicking.

Several patterns exist for the filter array. The most common array is the Bayer color filter array, shown in Figure 1. The Bayer array measures the green image on a quincunx grid and the red and blue images on rectangular grids. The green image is measured at a higher

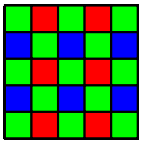


Figure 1: Bayer color filter array arrangement.

(a) (b)

Figure 2: Bicubic interpolation used for color filter array interpolation results in numerous artifacts. (a) Original image. (b) Bicubic interpolation.

sampling rate because the peak sensitivity of the human visual system lies in the medium wavelengths, corresponding to the green portion of the spectrum. Although other patterns exist (some using CMYG instead of RGB), this article discusses the demosaicking problem with reference to this color filter array.

If the measured image is divided by measured color into three separate images, this problem looks like a typical image interpolation problem. Therefore, one might try to apply standard image interpolation techniques. Bi-cubic interpolation is a common image interpolation technique that produces good interpolation results when applied to grayscale images. However, when bi-cubic interpolation is used for this problem, the resulting image shows many visible artifacts. This is illustrated in Figure 2.

This result motivates the need to find a specialized algorithm for the demosaicking problem. Bi-cubic interpolation and other standard interpolation techniques treat the color image as three independent images. However, the three images in a color image are generally highly correlated. Many algorithms have been published suggesting how to use this correlation. This article surveys many of these algorithms and discusses the results in terms of objective and subjective measures.

2 Demosaicking Methods

We examine demosaicking methods in three groups. The first group consists of heuristic methods; the second group uses restoration techniques; and the last group is based on image formation modeling.

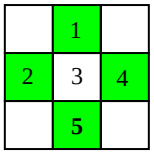
2.1 Group I: Heuristic Approaches

Heuristic approaches do not try to solve a mathematically defined problem, but they are based on reasonable assumptions about color images. Heuristic approaches are spatially adaptive, and they may exploit correlation among the color channels. We examine the heuristic approaches in six categories.

2.1.1 Edge-Directed Interpolation

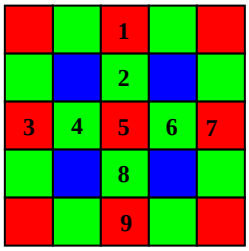
Although non-adaptive algorithms (e.g., bilinear interpolation, bi-cubic interpolation) can provide satisfactory results in smooth regions of an image, they usually fail in textured regions and edges. *Edge-directed interpolation* is an adaptive approach, where edge detection is performed for each pixel in question, and interpolation is done along the edges rather than across them.

In the demosaicking problem, edge-directed interpolation is usually applied to the green channel, which contains most of the spatial information and—therefore—is sampled more densely than the red and blue channels. A simple way of performing edge detection is to compare the absolute difference among the neighboring pixels [1]. Referring to Figure 3, horizontal and vertical gradients at a missing green location can be calculated from the horizontally and vertically adjacent green pixels. If the horizontal gradient is larger than the vertical gradient, suggesting a possible edge in the horizontal direction, interpolation is performed along the vertical direction. If the vertical gradient is larger than the horizontal gradient, interpolation is performed only in the horizontal direction. When the horizontal and vertical gradients are equal, the green value is obtained by averaging its four neighbors.



1. Calculate horizontal gradient $\Delta H = |G2 - G4|$
2. Calculate vertical gradient $\Delta V = |G1 - G5|$
3. If $\Delta H > \Delta V$,
   $G3 = (G1 + G5)/2$
   Else if $\Delta H < \Delta V$,
   $G3 = (G2 + G4)/2$
   Else
   $G3 = (G1 + G5 + G2 + G4)/4$

Figure 3: Edge-directed interpolation in [1] is illustrated. G1, G2, G4, and G5 are measured green values; G3 is the estimated green value at pixel 3.



1. Calculate horizontal gradient $\Delta H = |(R3 + R7)/2 - R5|$
2. Calculate vertical gradient $\Delta V = |(R1 + R9)/2 - R5|$
3. If $\Delta H > \Delta V$,
   $G5 = (G2 + G8)/2$
   Else if $\Delta H < \Delta V$,
   $G5 = (G4 + G6)/2$
   Else
   $G5 = (G2 + G8 + G4 + G6)/4$

Figure 4: Edge-directed interpolation in [2] is illustrated for estimating the green (G) value at pixel 5. The red (R) values are used to determine the edge direction. When the missing green pixel is at a blue pixel, the blue values are used to determine the edge direction.

It is also possible to compare the gradients against a predetermined threshold value [1].

The edge-directed interpolation approach in [1] can be modified by using larger regions (around the pixel in question) with more complex predictors and by exploiting the texture similarity in different color channels. In [2], the red and blue channels (in the 5×5 neighborhood of the missing pixel) are used instead of the green channel to determine the gradients. In order to determine the horizontal and vertical gradients at a blue (red) sample, second-order derivatives of blue (red) values are computed in the corresponding direction. This algorithm is illustrated in Figure 4.

Once the missing samples of the green channel are found, the red and blue channels are interpolated. A typical approach for the red/blue interpolation is constant-hue-based interpolation, which is explained in the next section.

2.1.2 Constant-Hue-Based Interpolation

One commonly used assumption in demosaicking is that the hue (color ratios) within an object in an image is constant. Although this is an oversimplification of image formation, it is a decent assumption within small neighborhoods of an image. This perfect inter-channel correlation assumption is sometimes formulated such that the color differences within small neighborhoods are constant. This constant color ratio (or difference) assumption prevents abrupt changes in color intensities, and has been extensively used for the interpolation of the chrominance (red and blue) channels [3, 4, 5, 2, 6, 7]. The demosaicking algorithms that are based on this assumption are called *constant-hue-based interpolation* or *smooth-hue transition* methods.

As a first step, these algorithms interpolate the luminance (green) channel, which is done using bilinear or edge-directed interpolation. The chrominance (red and blue) channels are then estimated from the interpolated "red hue" (red-to-green ratio) and "blue hue" (blue-to-green ratio). To be more explicit, the interpolated "red hue" and "blue hue" values are multiplied by the green value to determine the missing red and blue values at a particular pixel location. The hues can be interpolated with any method (bilinear, bi-cubic, edge-directed, etc.).

Instead of interpolating the color ratios, it is also possible to interpolate the color differences or the logarithm of the color ratios. The algorithm based on color difference interpolation is illustrated in Figure 5.

The constant-hue-based interpolation is sometimes used with median filtering to reduce color artifacts. For example, in [6], the green channel is interpolated bilinearly; and then the missing red and blue pixels are found from the median filtered red-green difference and blue-green difference.

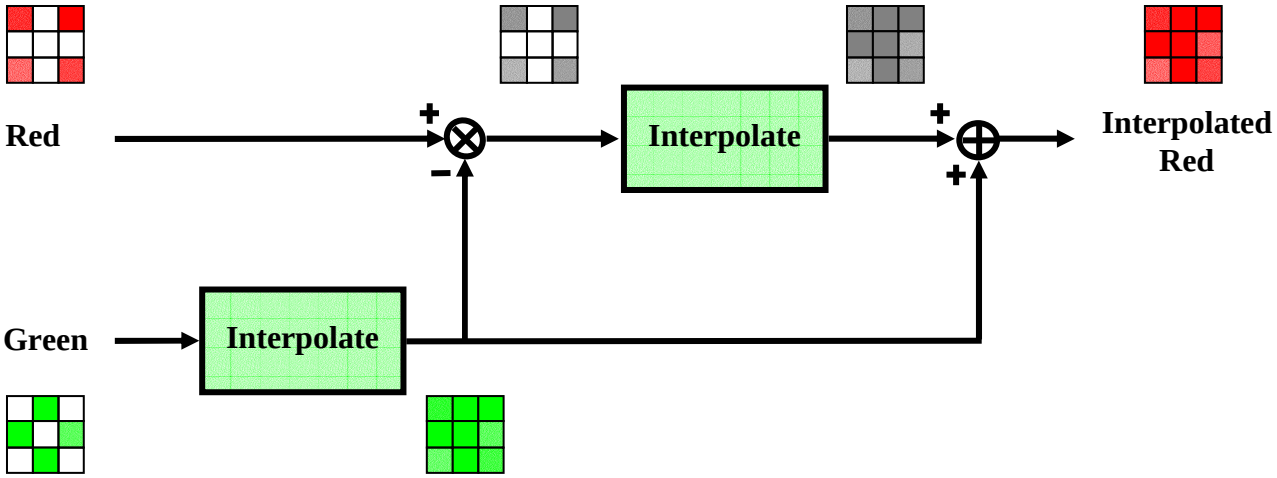
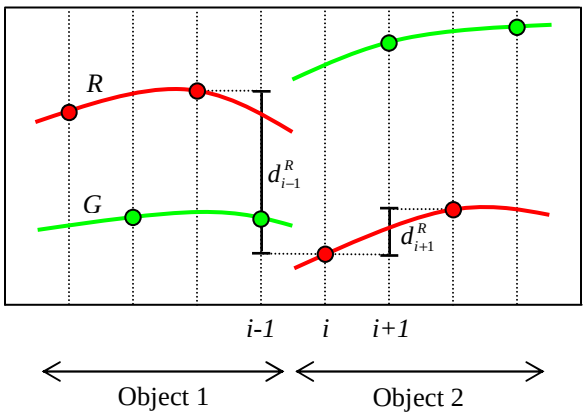


Figure 5: Constant-difference-based interpolation is illustrated.

2.1.3 Weighted Sum

In the edge-directed interpolation, the edge direction is found first, and then the missing sample is estimated by interpolating along the edge. This is a "hard" decision process. Instead, the likelihood of an edge in a certain direction can be found, and the interpolation can be done based on the edge likelihoods. Such an algorithm was proposed by Kimmel in [7]. The algorithm defines edge indicators in several directions as measures of edge likelihood in those directions, and determines a missing pixel intensity as a weighted sum of its neighbors. If the likelihood of an edge crossing in a particular direction is high, the edge indicator returns a small value, which results in less contribution from the neighboring pixel that direction. The algorithm for one-dimensional signals is illustrated in Figure 6. The green channel is interpolated first; the red and blue channels are interpolated from the red/green and blue/green ratios. The color channels are then updated iteratively to obey the color-ratio rule. The extension to two-dimensional images is straightforward [7].

A similar algorithm was proposed recently in [8], where edge indicators are determined in a 7×7 window for the green and a 5×5 window for the red/blue channels. In this case, the edge indicator function is based on the L_1 norm (absolute difference) as opposed to the L_2 norm of [7].



Define:

$$d_i^S \triangleq S(i+1) - S(i-1)$$

$$e_i^S \triangleq \frac{1}{\sqrt{1+\left(d_i^S\right)^2}}$$

- Interpolate the green at the missing locations:

$$G(i) = \frac{e_{i-1}^R G(i-1) + e_{i+1}^R G(i+1)}{e_{i-1}^R + e_{i+1}^R}$$

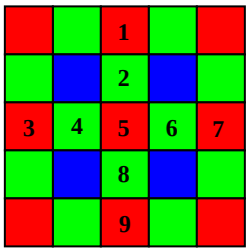
- Repeat for three times:
  - Interpolate the red using the ratio rule:

$$R(i) = G(i) \frac{e_{i-1}^G \frac{R(i-1)}{G(i-1)} + e_{i+1}^G \frac{R(i+1)}{G(i+1)}}{e_{i-1}^G + e_{i+1}^G}$$

  - Correct the green to fit the ratio rule:

$$G(i) = R(i) \frac{e_{i-1}^R \frac{G(i-1)}{R(i-1)} + e_{i+1}^R \frac{G(i+1)}{R(i+1)}}{e_{i-1}^R + e_{i+1}^R}$$

Figure 6: [7] is illustrated for a one-dimensional signal. S is a generic symbol for red (R) and green (G). $d_i{}^S$ is the gradient for channel S at location i; and $e_i{}^S$ is the corresponding edge indicator.



1. Calculate horizontal gradient $\Delta H = |G4 - G6| + |R5 - R3 + R5 - R7|$
2. Calculate vertical gradient $\Delta V = |G2 - G8| + |R5 - R1 + R5 - R9|$
3. If $\Delta H > \Delta V$,
   $G5 = (G2 + G8)/2 + (R5 - R1 + R5 - R9)/4$
   Else if $\Delta H < \Delta V$,
   $G5 = (G4 + G6)/2 + (R5 - R3 + R5 - R7)/4$
   Else
   $G5 = (G2 + G8 + G4 + G6)/4 + (R5 - R1 + R5 - R9 + R5 - R3 + R5 - R7)/8$

Figure 7: The Adams-Hamilton method [9] is illustrated for estimating the green (G) value at pixel 5. The red (R) and green values are used to determine the edge direction and to estimate the missing value. When the missing green pixel is at a blue pixel, the blue and green values are used.

2.1.4 Second-Order Gradients As Correction Terms

In [9], Adams and Hamilton used edge-directed interpolation for the green image. This result was improved by using correction terms from the red and blue samples. They computed the Laplacian for the red or blue samples along the interpolation row or column, and used this to correct the simple averaging interpolation. This correction term reduces aliasing and improves the mid-frequency response for the green image. Figure 7 illustrates this algorithm.

Assume that the horizontal direction has been chosen and consider the interpolation along a single row, simplifying our analysis to one dimension. The block diagram is shown in Figure 8. The output of this system is given by

$$\hat{G} = G_s(z)H_1(z) + R_s(z)H_2(z) \tag{1}$$

with the sampling equations given by

$$G_s(z) = 1/2\ G(z) + 1/2\ G(-z) \tag{2}$$

$$R_s(z) = 1/2\ R(z) - 1/2\ R(-z). \tag{3}$$

The aliasing terms have opposite signs because of the phase shift in the original sampling grid. The filter kernel of $H_1(z)$ is $[1/2 \quad 1 \quad 1/2]$ and the kernel of $H_2(z)$ is $[-1/4 \quad 0 \quad 1/2 \quad 0 \quad -1/4]$. The responses of these filters are shown in Figure 9. Because the response of H_1 has a wide transition band, components in the medium-frequency range will be attenuated and the spectral copies of these components (from the $G(-z)$ term) are passed into the output image causing aliasing distortions.

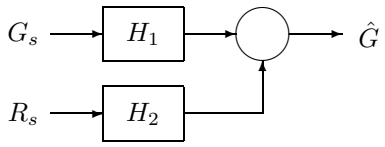


Figure 8: Block diagram for Adams-Hamilton green interpolation.

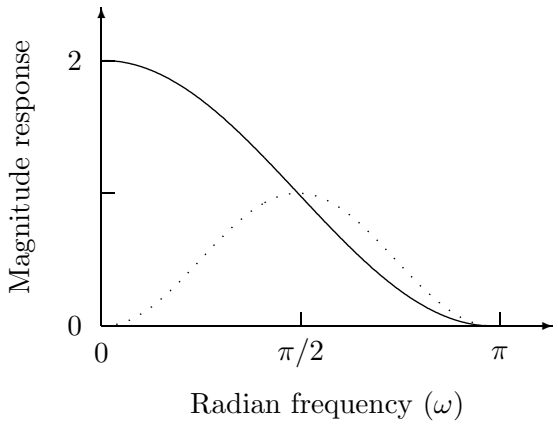


Figure 9: Filter responses for Adams-Hamilton method (solid $H_1(z)$, dotted $H_2(z)$).

The red filter, H_2, corrects this. Because of the high correlation between channels, it is assumed that the components that cause the distortion in the green image are also present in the red image. The bandpass filter, H_2, isolates components in the transition band of H_1. The expression for the aliasing terms, $1/2\ G(-z)H_1(z) - 1/2\ R(-z)H_2(z)$, shows that when the outputs of the filters are combined, the aliasing distortions can be cancelled if $H_1(z)$ and $H_2(z)$ have the same magnitude at frequencies greater than $\pi/2$.

2.1.5 Homogeneity-Directed Interpolation

Instead of choosing the interpolation direction based on edge indicators, it is possible to use different measures and to impose different constraints. In [10], local homogeneity is used as an indicator to choose between horizontally and vertically interpolated intensities. The homogeneity-directed interpolation imposes the similarity of the luminance and chrominance values within small neighborhoods. Referring to Figure 10, the RGB data is first interpolated horizontally and vertically. (The green channel is interpolated using red and blue data as correction terms, as in [9]. The red and blue channels are interpolated from the interpolated

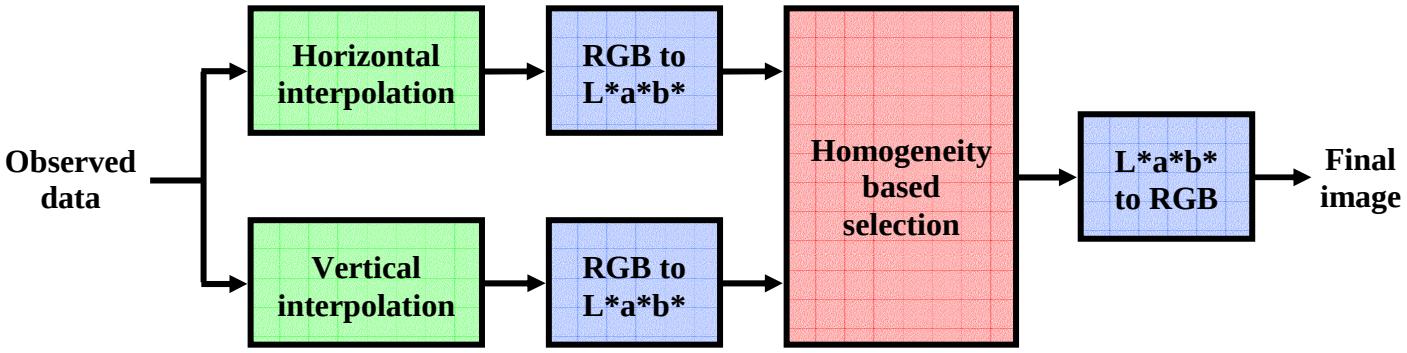


Figure 10: Block diagram of the homogeneity-directed interpolation in [10].

red-green difference and blue-green difference as shown in Figure 5.) The interpolated images are then converted to the CIELab space. In the CIELab space, horizontally or vertically interpolated intensities are chosen based on the local homogeneity. To measure the local homogeneity, the number of pixels that have similar luminance and chrominance values to the horizontal and vertical candidates are found; and the candidate that is more similar to its neighborhood is chosen.

2.1.6 Pattern Matching

Several algorithms attempt to find a pattern in the data or fit the data to one of several templates. A different interpolator is applied for each template. This allows a different method to be used for edges and smooth regions.

In [3], Cok describes a pattern matching algorithm to be used on the green image. Each missing green value is classified as a stripe, edge, or corner, corresponding to the features expected to be found in natural images. The human visual system is sensitive to these features so interpolating them correctly is important. After classifying the pixel, the appropriate interpolator is applied to estimate the missing value.

Wu *et al.* uses the green image to detect patterns in the image in [11]. The algorithm determines which directions have the least amount of change in the green image. A direction with a large amount of change indicates that an edge is present in that region. In this case, the two directions with the smallest change are used for the interpolation. This allows the

algorithm to detect edges in the horizontal, vertical, and diagonal directions.

In [12], Chang *et al.* introduced a method using directional information and added the ability to use multiple directions. This method uses eight possible horizontal, vertical, and diagonal interpolation directions. A gradient is computed for each direction and then a threshold is computed based on these gradients to determine which directions are used. The threshold is defined as $T = k_1 m + k_2(M + m)$, where m is the minimum gradient in the set, M is the maximum gradient of the the set and k_1 and k_2 are constants. The authors suggest $k_1 = 1.5$ and $k_2 = 0.5$. For each direction included in the interpolation, an average red, green, and blue value is computed. For each of the missing colors at the current pixel, the difference between the average of the missing color and the average of the color of the current pixel is calculated. This color difference is added to the value of the current pixel to estimate the missing color value.

2.2 Group II: Reconstruction Approaches

The second group of algorithms makes some assumptions about the inter-channel correlation or the prior image, and solves a mathematical problem based on those assumptions. In [13], Glotzbach *et al.* assumes that the high-frequency components of the red, green, and blue channels are identical, and replaces the aliased components of the red and blue color planes with the high-frequency components of the green plane, which are less likely to be aliased. In [14], Gunturk *et al.* adds a data-consistency constraint to the similar-high-frequency-components assumption of [13], and develops an iterative reconstruction algorithm. In [15], Mukherjee *et al.* proposes a Bayesian estimation algorithm that includes a spatial-smoothness assumption as a regularization term.

2.2.1 Alias Canceling Interpolation

There are two observations that are important for the demosaicking problem. The first is that for natural images, there is a high correlation among the red, green, and blue channels. All three channels are very likely to have the same texture and edge locations. (Because of

the similar edge content, we expect this inter-channel correlation to be even higher when it is measured between the high-frequency components of the channels.) The second observation is that digital cameras use a color filter array (CFA) in which the luminance (green) channel is sampled at a higher rate than the chrominance (red and blue) channels. Therefore, the green channel is less likely to be aliased, and details are preserved better in the green channel than in the red and blue channels. (The high-frequency components of the red and blue channels are affected the most in CFA sampling.) In demosaicking, it is the interpolation of the red and blue channels that is the limiting factor in performance. Color artifacts, which become severe in high-frequency regions such as edges, are caused primarily by aliasing in the red and blue channels. Although this fact is acknowledged by the authors of most demosaicking algorithms, inter-channel correlation has not been used effectively to retrieve the aliased high-frequency information in the red and blue channels.

In [13], the green image is used to add high-frequency information and reduce aliasing in the red and blue images. First, the red and blue images are interpolated with a rectangular lowpass filter according to the rectangular sampling grid. This fills in the missing values in the grid, but allows aliasing distortions into the red and blue output images. These output images are also missing the high-frequency components needed to produce a sharp image. However, because the green image is sampled at a higher rate, the high-frequency information can be taken from the green image to improve an initial interpolation of the red and blue images. A horizontal highpass filter and a vertical highpass filter are applied to the green image. This provides the high-frequency information that the low sampling rate of the red and blue images cannot preserve. Aliasing occurs when high-frequency components are shifted into the low-frequency portion of the spectrum, so if the outputs of the highpass filters are modulated into the low-frequency regions, an estimate of the aliasing in the red and blue images can be found. This estimate is used to reduce the aliasing in the red and blue images, as illustrated in Figure 11. This method relies on the assumption that the high-frequency information in the red, green, and blue images is identical. If this assumption does not hold, the addition of the green information into the red and blue images can add unwanted distortions. This method also makes the assumption that the input image is band-limited

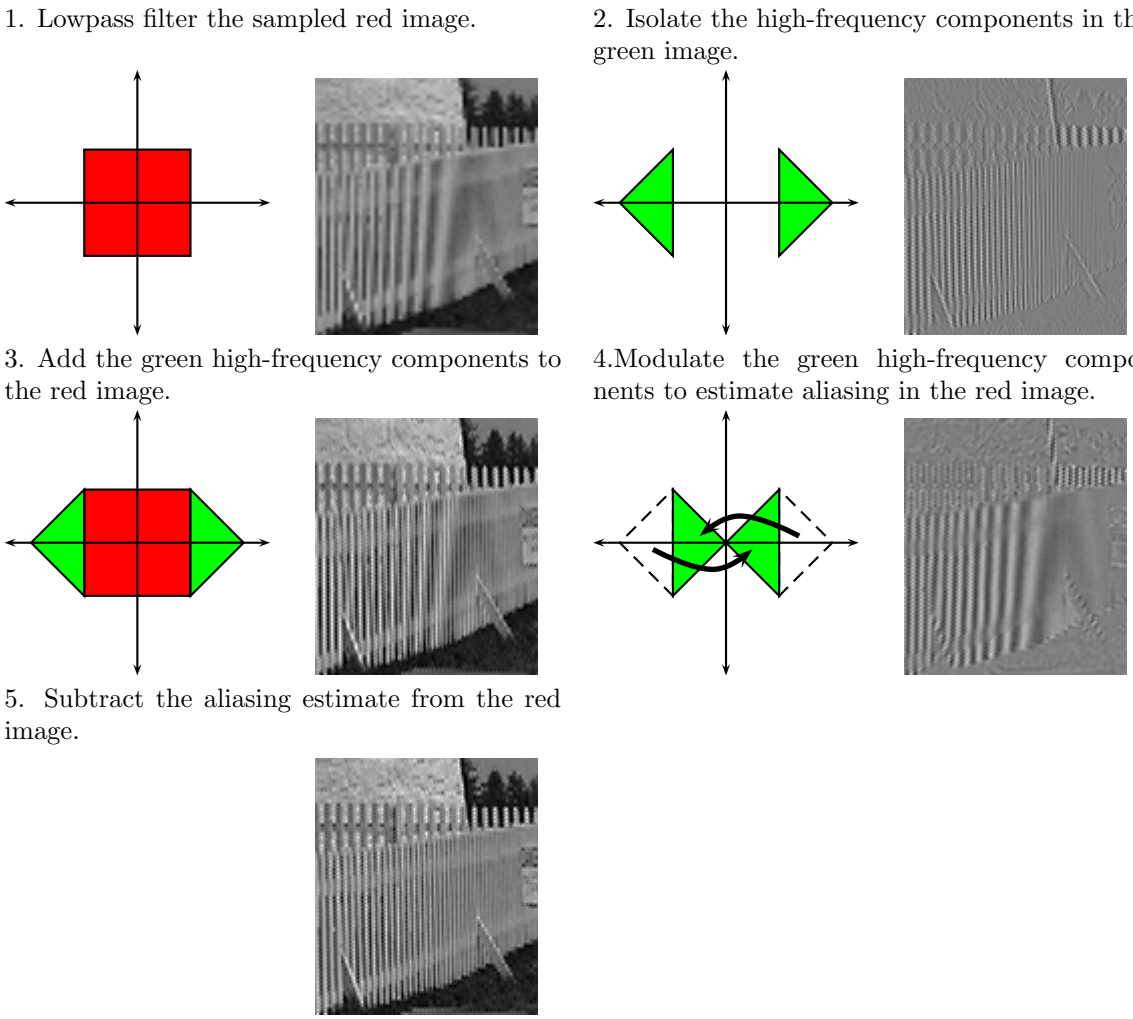


Figure 11: High-frequency information from the green image is modulated and used to cancel aliasing in the red image.

within the diamond-shaped Nyquist region of the green quincunx sampling grid. When this assumption fails, the aliasing artifacts are enhanced instead of reduced because the green image also contains aliasing. This system is composed entirely of linear filters, making it efficient to implement.

2.2.2 Projections onto Convex Sets (POCS) Approach

The algorithm presented in the previous section proposes to decompose the green channel into its frequency components and then add the high-frequency components of the green channel to the low-pass filtered red and blue channels. This is based on the observation that

the high-frequency components of the red, blue, and green channels are similar and the fact that the green channel is less likely to be aliased. There are a couple of problems with this approach. First, it is not possible to construct ideal low-pass and high-pass filters with finite spatial extent. Secondly, the high-frequency components of the red, green, and blue channels may not be identical. Therefore, replacement of the high-frequency components of the red and blue channels with those of the green channel may not work well.

In [14], Gunturk *et al.* proposes an algorithm that ensures data consistency at the cost of higher computational complexity. The algorithm defines two constraint sets, (one imposing similar high-frequency components in the color channels and the other ensuring data consistency,) and reconstructs the color channels using a projections onto convex sets (POCS) technique.

A. Constraint Sets

Set-theoretic reconstruction techniques produce solutions that are consistent with the information arising from the observed data or prior knowledge about the solution. Each piece of information is associated with a constraint set in the solution space, and the intersection of these sets represents the space of acceptable solutions [16]. In [14], two types of constraint sets are defined for the demosaicking problem. The first comes from the observed color samples. The interpolated color channels must be consistent with the color samples captured by the digital camera. We denote $O(n_1, n_2)$ as this observed data, which has red, green, and blue samples placed according to the CFA used. (n_1, n_2) are ordered pairs of integers denoting the pixel locations. By defining Λ_R, Λ_G, and Λ_B as the set of pixel locations, (n_1, n_2), that have the samples of red, green, and blue channels, respectively, the "observation" constraint set C_o is written as follows:

$$C_o = \{S(n_1, n_2) : S(n_1, n_2) = O(n_1, n_2) \quad \forall \quad (n_1, n_2) \in \Lambda_S \ , \quad S = R, G, B\} \, , \tag{4}$$

where S is a generic symbol for the interpolated color channels, which can be R for the red channel, G for the green channel, and B for the blue channel.

The second constraint set is a result of the high inter-channel correlation. In [14], it was

shown that color channels have very similar detail (high-frequency) subbands. This information would not be enough to define constraint sets if all channels lost the same amount of information in sampling. However, it was also pointed out that the red and blue channels lose more information (details) than the green channel when captured with a color filter array. Therefore, it is reasonable to define constraint sets on the red and blue channels that force their high-frequency components to be similar to the high-frequency components of the green channel.

In [14], Gunturk *et al.* proposed to decompose the signals into their subbands using a filterbank and impose the high frequency similarity in the detail subbands. Let $h_0(\cdot)$ and $h_1(\cdot)$ be the low-pass and high-pass analysis filters, which form a perfect reconstruction filterbank with synthesis filters $g_0(\cdot)$ and $g_1(\cdot)$. Then the four subbands of a two-dimensional signal $S(n_1, n_2)$ are

$$(\mathcal{W}_1 S)(n_1, n_2) = h_0(n_1) * [h_0(n_2) * S(n_1, n_2)] \tag{5}$$

$$(\mathcal{W}_2 S)(n_1, n_2) = h_1(n_1) * [h_0(n_2) * S(n_1, n_2)] \tag{6}$$

$$(\mathcal{W}_3 S)(n_1, n_2) = h_0(n_1) * [h_1(n_2) * S(n_1, n_2)] \tag{7}$$

$$(\mathcal{W}_4 S)(n_1, n_2) = h_1(n_1) * [h_1(n_2) * S(n_1, n_2)], \tag{8}$$

where $(\mathcal{W}_1 S)$ is the approximation subband, and $(\mathcal{W}_2 S)$, $(\mathcal{W}_3 S)$, $(\mathcal{W}_4 S)$ are the horizontal, vertical, and diagonal detail subbands, respectively.

We can now define the "detail" constraint set C_d that forces the details (high-frequency components) of the red and blue channels to be similar to the details of the green channel as follows:

$$C_d = \left\{ \begin{array}{c} S(n_1, n_2) : |(\mathcal{W}_k S)(n_1, n_2) - (\mathcal{W}_k G)(n_1, n_2)| \leq T(n_1, n_2) \\ \\ \forall\ (n_1, n_2) \in \Lambda_S, \quad \text{for } k = 2, 3, 4 \ \text{ and } \ S = R, B \end{array} \right\}, \tag{9}$$

where $T(n_1, n_2)$ is a positive threshold that quantifies the "closeness" of the detail subbands to each other. If the color channels are highly correlated, then this threshold should be small; if the correlation is not high, then the threshold should be larger. Although $T(n_1, n_2)$

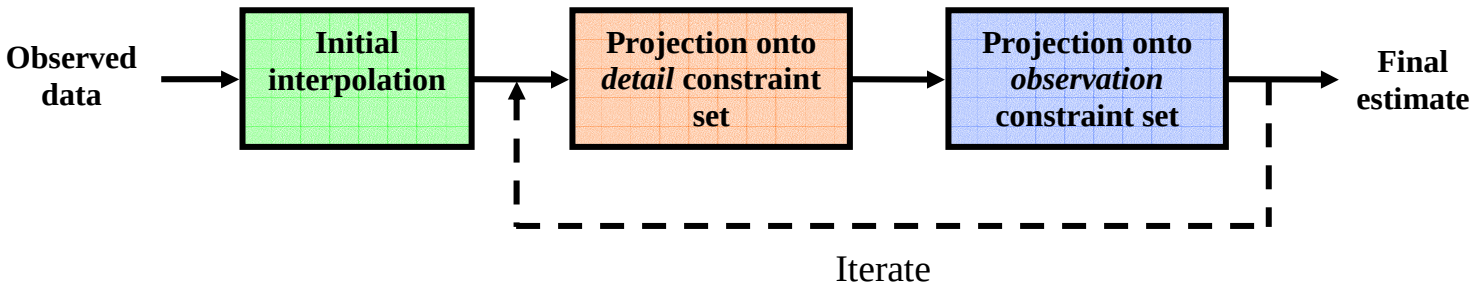


Figure 12: Alternating projections algorithm.

is a function of image coordinates in general, it is also possible to use a predetermined fixed value for it. One choice is to set $T(n_1, n_2)$ to zero for all (n_1, n_2), which is a result of the high-correlation assumption. In [14], it was also discussed how to choose a non-uniform threshold.

B. Alternating Projections Algorithm

The block diagram of the POCS algorithm is given in Figure 12. The first constraint set that is used in the reconstruction is the "observation" constraint set given in (4), which states that the reconstructed signal must be consistent with the observed data. Therefore, projection onto the observation constraint set is simply insertion of the observed data into their corresponding locations in the current image. This is illustrated in Figure 13.

The other constraint set is the "detail" constraint set given in (9), which states that the high frequency portions of the red/blue channels are similar to those of green channel. The projection operation onto the detail constraint set is illustrated in Figure 14.

The "observation" projection ensures that the interpolated channels are consistent with the observed data; the "detail" projection reconstructs the high-frequency information of the red and blue channels, and imposes edge consistency between the channels. By alternately applying these two projections onto the initial red and blue channel estimates, these channels are enhanced.

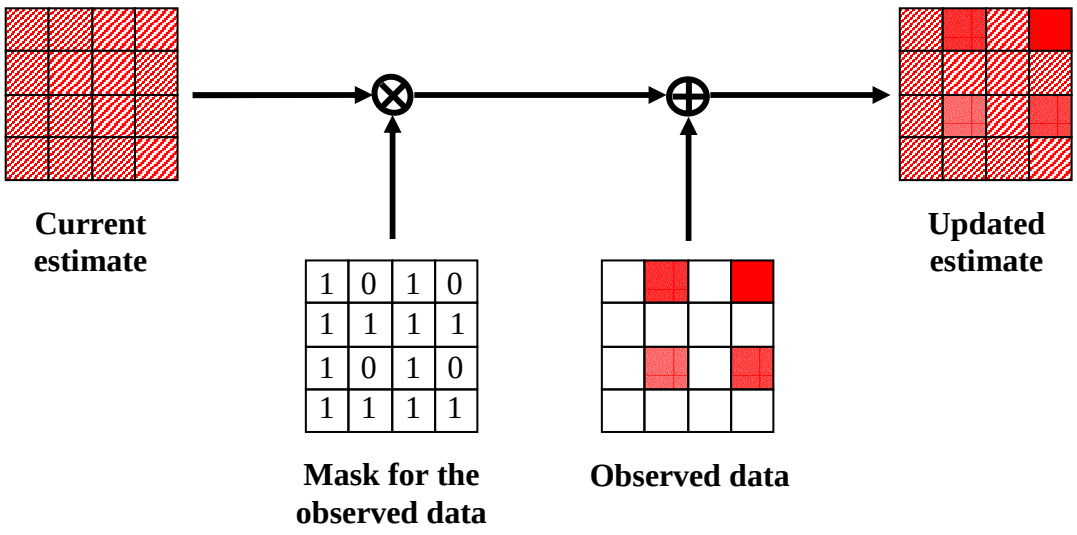


Figure 13: Projection onto observation constraint set.

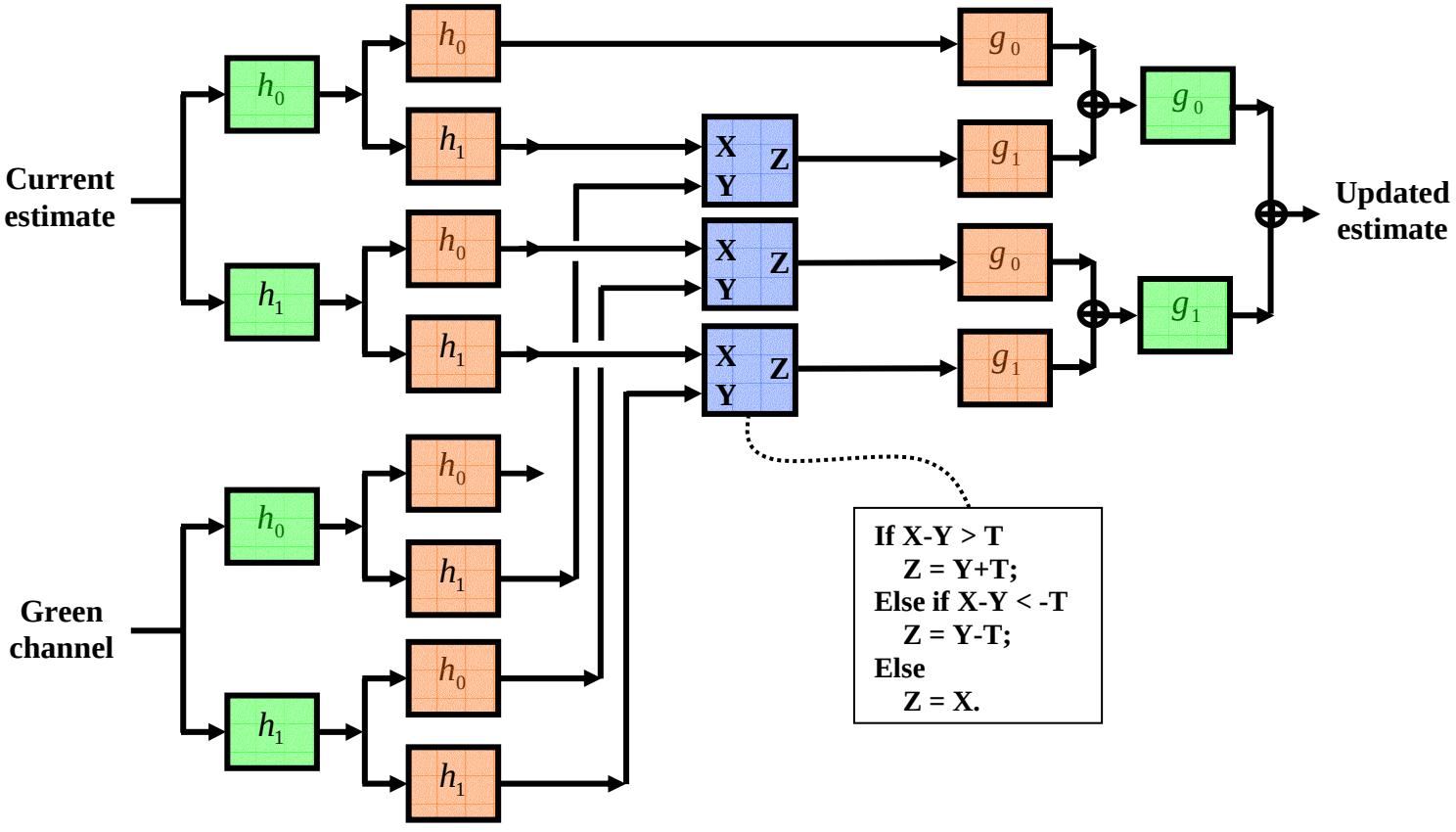


Figure 14: Projection onto detail constraint set.

2.2.3 Bayesian Approach

With the Bayesian estimation approach, it is possible to incorporate prior knowledge about the solution (such as spatial smoothness and constant color ratio) and the noise statistics into the solution. We start by defining a CFA mask function $f_s(n_1, n_2)$:

$$f_S(n_1, n_2) = \left\{ \begin{array}{ll} 1, & (n_1, n_2) \in \Lambda_S \\ 0, & otherwise \end{array} \right\}, \tag{10}$$

where S is R, G, B for the red, green, and blue channels. Denoting $v(n_1, n_2)$ as the additive sensor noise, the observation process can be formulated as

$$O(n_1, n_2) = \sum_{S=R,G,B} f_S(n_1, n_2) S(n_1, n_2) + v(n_1, n_2). \tag{11}$$

In the maximum *a posteriori* probability (MAP) formulation, the observed data $O(n_1, n_2)$, the full color channels $S(n_1, n_2)$, and the additive noise $v(n_1, n_2)$ are all assumed to be random processes. Denoting $p\left(S|O\right)$ as the conditional probability density function (PDF), the MAP estimate $\hat{S}$ is given by

$$\hat{S} = \arg\max_S \left\{ p\left(S|O\right) \right\} = \arg\max_S \left\{ p\left(O|S\right) p\left(S\right) \right\}, \tag{12}$$

where Bayes' rule and the fact that O is independent of S were used. To find the MAP estimate $\hat{S}$, the conditional PDF, $p\left(O|S\right)$, and the prior PDF, $p\left(S\right)$, need to be modeled. The conditional PDF $p\left(O|S\right)$ is derived from the noise statistics, which is usually assumed to be white Gaussian. As for the prior PDF, different models have been proposed.

In [15] and [17], Markov Random Field (MRF) models were used. In MRF processing, the conditional and prior PDFs can be modeled as Gibbs distributions. The Gibbs distribution has an exponential form, and it is characterized by an *energy function* and a *temperature* parameter. A PDF with Gibbs distribution can be written as

$$p(w) = \frac{1}{Z} e^{-U(w)/T}, \tag{13}$$

where $U(\cdot)$ is the energy function, T is the temperature parameter, and Z is the normalization constant. One feature of the MRF is that the total energy function U can be written as a

sum of local energy functions, which allows for localized reconstruction [18]:

$$U = \sum_{n_1,n_2} \sum_{k} \alpha_k U_k(n_1, n_2), \tag{14}$$

where $U_k(n_1, n_2)$ is kth type of energy function at location (n_1, n_2) and α_k is the corresponding weight of $U_k(n_1, n_2)$ among the other energy function types. In [15], three types of local energy functions are defined at each pixel location. The first energy function is associated with the additive noise:

$$U_1(n_1, n_2) = \sum_{S=R,G,B} \left\{ (1 - f_S(n_1, n_2)) \frac{\left(S(n_1, n_2) - S^{(0)}(n_1, n_2)\right)^2}{2\sigma^2(n_1, n_2)} \right\}, \tag{15}$$

where $S^{(0)}(n_1, n_2)$ is the initial estimate and $\sigma^2(n_1, n_2)$ is the noise variance. The second energy function imposes spatial smoothness, and is defined as follows:

$$U_2(n_1, n_2) = \sum_{m_1=-p}^{p} \sum_{m_s=-p}^{p} \sum_{S=R,G,B} \left\{ (1 - f_S(n_1, n_2)) \left(S(n_1, n_2) - S(n_1 + m_1, n_2 + m_1)\right)^2 \right\}. \tag{16}$$

The third energy function imposes constancy of cross color ratios:

$$U_3(n_1, n_2) = \sum_{m_1=-p}^{p} \sum_{m_s=-p}^{p} \sum_{S=R,G,B} \left\{ (1 - f_S(n_1, n_2)) \left(\frac{S(n_1, n_2)}{O(n_1, n_2)} - \frac{S(n_1 + m_1, n_2 + m_1)}{\sum_{T=R,G,B} f_T(n_1, n_2) T(n_1 + m_1, n_2 + m_1)} \right)^2 \right\}.$$

Once the local energy functions are defined, the solution minimizing the total energy can be found using a variety of techniques. In [15], *simulated annealing* technique was used.

As an alternative, [17] proposed a prior based on the steerable wavelet decomposition. With the steerable wavelet decomposition, images can be represented as a sum of bandpass components, each of which can be decomposed into a set of oriented bands using steerable filters. The idea is that instead of imposing an isotropic smoothness, such a directional decomposition allows edge-oriented smoothness. This means that across-the-edge averaging can be avoided. Let $B_k(n_1, n_2)$ be the kth scale in the Laplacian pyramid decomposition of the image $S(n_1, n_2)$. The image $S(n_1, n_2)$ can be constructed by summing of $B_k(n_1, n_2)$ over all scales:

$$S(n_1, n_2) = \sum_{k} B_k(n_1, n_2). \tag{17}$$

Then the energy function $U_k(n_1, n_2)$ at a particular scale k is

$$U_k(n_1, n_2) = \sum_{n_1, n_2} \sum_{m_1, m_2 \in N(n_1, n_2)} g\left(E_{k,\theta(\mathbf{n},\mathbf{m})}(n_1, n_2)\right) \left(B_k(n_1, n_2) - B_k(m_1, m_2)\right)^2, \qquad (18)$$

where $g(\cdot)$ is a monotonically increasing function, $E_{k,\theta(\mathbf{n},\mathbf{m})}$ is the directional energy at the kth scale and in the direction of (m_1, m_2) from (n_1, n_2), and $N(n_1, n_2)$ is a neighborhood of (n_1, n_2). The optimization algorithm tries to find $S(n_1, n_2)$ that minimizes the energy function $U_k(n_1, n_2)$ at all scales. The directional energy function $E_{k,\theta}$ is defined so that it returns high values along directions with similar intensities [17]. Therefore, $E_{k,\theta}$ is small in directions where an edge is crossed. Since the energy function $U_k(n_1, n_2)$ neglects the difference between neighboring pixels, the strength of the penalty is less in the cross-edge directions. A gradient descent procedure can be used to reconstruct $S(n_1, n_2)$ [17]. (See Figure 15.)

2.3 Group III: Image Formation Modeling

The last group of methods models the image formation process and formulates the demosaicking problem based on this model. To understand this approach, we first need to understand the imaging process. This is usually modeled as a linear process between the light radiance arriving at the camera and the pixel intensities produced by the sensors. Most digital cameras use charge-coupled device (CCD) sensors. In a CCD camera, there is a rectangular grid of electron-collection sites laid over a silicon wafer to record the amount of light energy reaching each of them. When photons strike these sensor sites, electron-hole pairs are generated; and the electrons generated at each site are collected over a certain period of time. The numbers of electrons are eventually converted to pixel values.

Each sensor type has a specific spectral response $L_S(\lambda)$, which is a function of the spectral wavelength λ, and a spatial response $h_S(x, y)$, which results from optical blur and the spatial integration at each sensor site. (Typical spectral sensor sensitivities are illustrated in Figure 16.) The imaging process can be formulated as

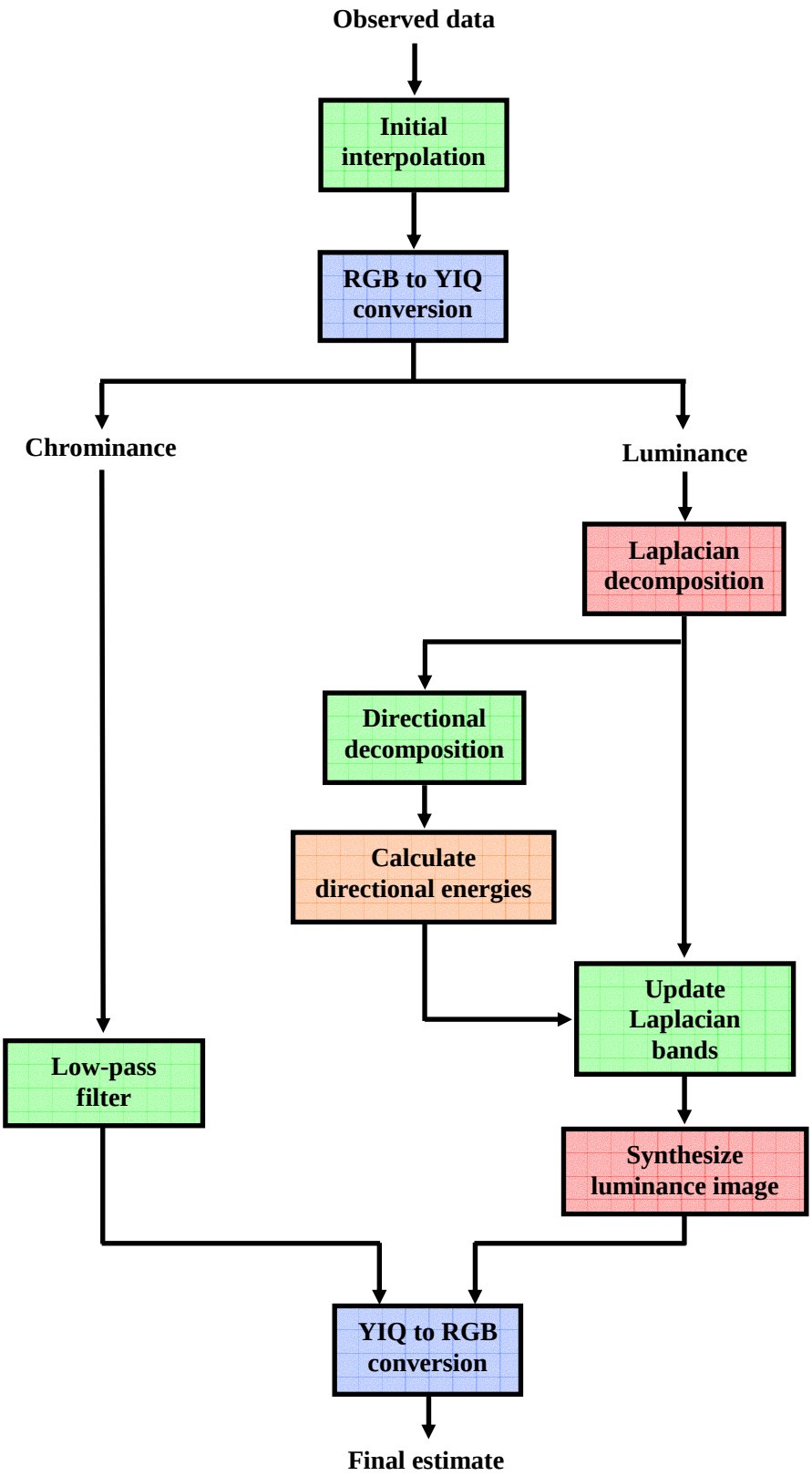


Figure 15: Block diagram of [17].

$$S(x,y) = \int\int\int L_S(\lambda) h_S(x-u, y-v) r(u,v,\lambda) du\ dv\ d\lambda + N_S(x,y) \tag{19}$$

where $S(x,y)$ is the pixel value at spatial location (x,y), $r(x,y,\lambda)$ is the incident radiance, and $N_S(x,y)$ is the additive noise that is a result of thermal/quantum effects and quantization. There are couple of assumptions in this formulation: (i) the input-output relation is assumed to be linear; (ii) the spatial blur $h_S(x,y)$ is assumed to be space-invariant and independent of wavelength; (iii) only the additive noise is considered. These assumptions are reasonable for practical purposes.

Since we are dealing with digital data, we need to have the discrete version of (19):

$$S(n_1,n_2) = \sum_{l} \sum_{m_1,m_2} L_S(l) h_S(n_1-m_1, n_2-m_2) r(m_1,m_2,l) + N_S(n_1,n_2). \tag{20}$$

The color filters sample the signal $S(n_1,n_2)$ to produce a mosaic plane $O(n_1,n_2)$, which has red, green, and blue pixels. Therefore, the observation model is a linear system, which can be written in the compact form

$$\mathbf{O} = \mathbf{H}\mathbf{r} + \mathbf{N} \tag{21}$$

where $\mathbf{r}$, $\mathbf{O}$, and $\mathbf{N}$ are the stacked forms of $r(m_1,m_2,l)$, $O(n_1,n_2)$, and $N_S(n_1,n_2)$, respectively; and $\mathbf{H}$ is the matrix that includes the combined effects of optical blur, sensor blur, spectral response, and CFA sampling. In [19], [20], and [21] the minimum mean square error (MMSE) solution of (21) is given:

$$\mathbf{r}^{MMSE} = E\left[\mathbf{r}\mathbf{O}^T\right]\left(E\left[\mathbf{O}\mathbf{O}^T\right]\right)^{-1}\mathbf{O}, \tag{22}$$

where $E[\cdot]$ is the expectation operation. In [21], the point spread function is taken as an impulse function; and $\mathbf{r}$ is represented as a weighted sum of spectral basis functions to reduce the dimensionality of the problem. (Later, [22] extended [21] to include the PSF in the reconstruction.) In [20], adaptive reconstruction and ways to reduce computational complexity are discussed.

An interesting approach to solve (22) is provided in [19]. The approach constructs a finite-support filter that is optimal under a set of assumptions. We now switch to frequency-domain formulation to explain this approach (as was done in [19]). We first define $\mathbf{r}(\mathbf{n})$ as the stacked form of $r(n_1, n_2, l)$, where $\mathbf{n}$ is a short notation for (n_1, n_2). (With the application of principal component analysis on surface reflectance, it has been found that the dimensionality of $\mathbf{r}(\mathbf{n})$ is typically less than 7.) Defining $\hat{\mathbf{r}}(w)$, $\hat{O}(w)$, and $\hat{N}(w)$ as the Discrete Space Fourier Transform (DSFT) of $\mathbf{r}(\mathbf{n})$, $O(\mathbf{n})$, and $N(\mathbf{n})$ respectively, the frequency domain equivalent of (21) can be expressed as

$$\hat{O}(w) = \hat{\mathbf{T}}(w)\hat{\mathbf{r}}(w) + \hat{N}(w) \tag{23}$$

where $\hat{\mathbf{T}}(w)$ is the DSFT of a linear shift-invariant matrix impulse response that includes the effects of the optical blur, sensor spectral response, and CFA sampling. With this frequency-domain observation formulation, the corresponding MMSE solution in (22) can be written as

$$\hat{\mathbf{r}}^{MMSE}(w) = \mathbf{\Gamma}_{\mathbf{r}O}(w)\mathbf{\Gamma}_O(w)^{-1}\hat{O}(w) \tag{24}$$

where $\mathbf{\Gamma}_{\mathbf{r}O}$ and $\mathbf{\Gamma}_O$ are the power spectral densities of $\mathbf{r}O^*$ and O, respectively. These two power spectral densities can be written in terms of the power spectral densities of the input signal and the additive noise:

$$\mathbf{\Gamma}_{\mathbf{r}O}(w) = \mathbf{\Gamma}_{\mathbf{r}}(w)\hat{\mathbf{T}}(w)^*, \tag{25}$$

$$\mathbf{\Gamma}_O(w) = \hat{\mathbf{T}}(w)\mathbf{\Gamma}_{\mathbf{r}}(w)\hat{\mathbf{T}}(w)^* + \mathbf{\Gamma}_N(w) \tag{26}$$

where $\mathbf{\Gamma}_{\mathbf{r}}$ is the power spectral density (PSD) of $\mathbf{r}$, and $\mathbf{\Gamma}_N$ is the PSD of N.

The additive noise is often assumed as white noise process; therefore, the PSD $\mathbf{\Gamma}_N$ has a diagonal form: $\mathbf{\Gamma}_N = \sigma^2\mathbf{I}$. On the other hand, $\mathbf{\Gamma}_{\mathbf{r}}$ is not obvious. Taubman showed that one prior that is independent of scale at the which the image is formed has the following form:

$$\mathbf{\Gamma}_{\mathbf{r}}(w_r, w_\theta) = \frac{1}{w_r^2} f(w_\theta)\mathbf{\Gamma}_0 \tag{27}$$

where $\mathbf{\Gamma}_{\mathbf{r}}(w_r, w_\theta)$ is the prior PSD expressed in polar coordinates, with w_r and w_θ being the radial and angular frequencies. The angular distribution $f(w_\theta)$ is usually taken as constant; and $\mathbf{\Gamma}_0$ is another constant arising from surface spectral reflectances. In practice, a finite

support filter whose DSFT is $\mathbf{\Gamma}_{\mathbf{r}O}(w)\mathbf{\Gamma}_O(w)^{-1}$ in (24) is estimated. To find that filter, $\mathbf{\Gamma}_{\mathbf{r}O}(w)$ and $\mathbf{\Gamma}_O(w)^{-1}$ must be evaluated numerically.

Classical Wiener filtering results give the optimal solution, $\mathbf{F} = \mathbf{Z}\mathbf{U}^{-1}$. $\mathbf{Z}$ is a matrix whose row vectors are given by $\mathbf{Z}[\mathbf{k}]^t$ in (28). $\mathbf{U}$ is a Toeplitz matrix with elements, $U[\mathbf{k}]$, defined in (29).

$$\begin{aligned}\mathbf{Z}[-\mathbf{k}] &= E\left[\mathbf{r}[\mathbf{n}]O[\mathbf{n}-\mathbf{k}]\right] \\ &= \frac{1}{(2\pi)^2}\int_{-\pi}^{\pi}\int_{-\pi}^{\pi}\mathbf{\Gamma}_{\mathbf{r}O}(\omega)e^{j\omega^t\mathbf{k}}d\omega; \quad \mathbf{k}\in[-\kappa,\kappa]^2 \end{aligned} \tag{28}$$

$$\begin{aligned}U[-\mathbf{k}] &= E\left[O[\mathbf{n}]O[\mathbf{n}-\mathbf{k}]\right] \\ &= \frac{1}{(2\pi)^2}\int_{-\pi}^{\pi}\int_{-\pi}^{\pi}\Gamma_O(\omega)e^{j\omega^t\mathbf{k}}d\omega; \quad \mathbf{k}\in[-\kappa,\kappa]^2. \end{aligned} \tag{29}$$

Both $\Gamma_O(\omega)$ and $\mathbf{\Gamma}_{\mathbf{r}O}$ are undefined at $\omega = \mathbf{0}$ because $\mathbf{\Gamma}_{\mathbf{r}}(\omega)$ (using the scale invariant prior) is undefined at $\omega = \mathbf{0}$. Thus, these integrals cannot be computed. Taubman found that numerical integration of these integrals using a dense grid of frequencies excluding the point $\omega = \mathbf{0}$ led to large numerical errors. He suggests a solution to this problem, numerically integrating the integrals in ((28)) and ((29)) on a coarse grid of frequencies to obtain estimates of the integrals. These rough estimates are computed for each $\mathbf{k}\in[-\kappa,\kappa]^2$ and assembled into matrices $\mathbf{Z}'$ and $\mathbf{U}'$. The solution is found by solving

$$\mathbf{F}' = \left(\mathbf{Z}' + s\mathbf{I}_L\Gamma_0\overline{\mathbf{H}}^*\right)\left(\mathbf{U}' + s\overline{\mathbf{H}}\Gamma_0\overline{\mathbf{H}}^*\right)^{-1} \tag{30}$$

where $\overline{\mathbf{H}}$ is a stacked matrix with $(2\kappa+1)^2$ copies of $\hat{H}(\mathbf{0})$, L is the dimension of $\mathbf{\Gamma}_0$, and s is a parameter chosen as large as possible without introducing numerical instability. $\hat{H}(\mathbf{0})$ describes the DC-response of the system, similar to $\hat{\mathbf{T}}(\omega)$, excluding the sampling aliases. This solution comes from solving the limit

$$\lim_{\varepsilon\to 0}\mathbf{Z}^{\varepsilon}(\mathbf{U}^{\varepsilon})^{-1} \tag{31}$$

where the ε-modified matrices address the singularity at $\omega = \mathbf{0}$. The details can be found in [19].

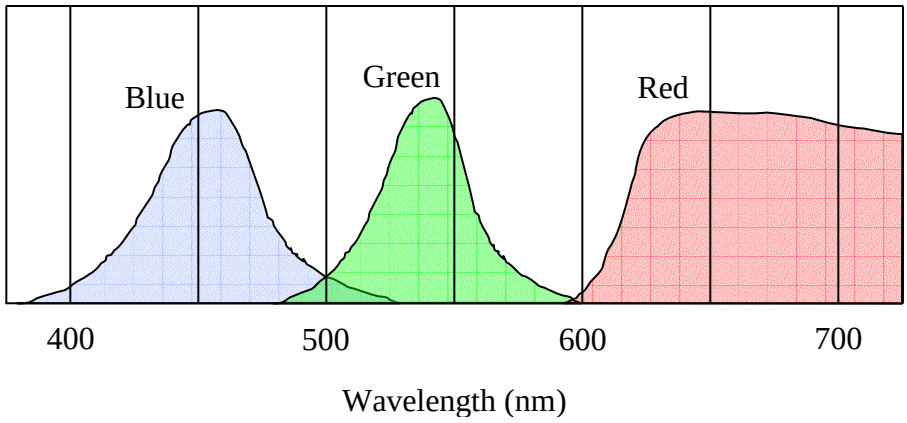


Figure 16: Typical spectral sensitivity of the sensors are shown. The sensitivities may go beyond $700nm$; therefore, digital cameras are equipped with infrared rejection filters.

The matrices $\mathbf{\Gamma}_{\mathbf{r}O}$ and $\mathbf{\Gamma}_O$ are different for each type of pixel (R, B, G_r, and G_b). Therefore, a different filter kernel will be computed for each type of pixel. Taubman combines this into a single problem by considering the overlying grid of superpixels, $[GR; BG]$, and treating these as the fundamental sampling unit. The filter kernel computed from this system generates an estimate of the superpixel instead of just one element of the group.

3 Comparison

In this section, the algorithms are compared with objective measures (mean square error) and subjective image quality. Image results from each of the algorithms are provided. For these experiments, simulated sampling was used where full-color images were sampled to match the CFA sampling process. A better comparison would be to use real data from a camera system, but we don't have the capability of performing this comparison at this time.

Twenty-four digital color images were used in the objective measure experiments. These images are part of the Kodak color image database and include various scenes. The images were sampled according to the Bayer CFA and reconstructed with each of the algorithms. The mean square error was measured for each color plane on each of the output images to determine the difference between the original image and the reconstructed image. The bar graph in Figure 17 shows the average MSE over the set of images, along with error bars

showing the 25%-75% range for the set of images. The graph shows that the POCS method performs best on average in terms of MSE and the small range shown in the graph shows that it is also robust and performs well for all of the images.

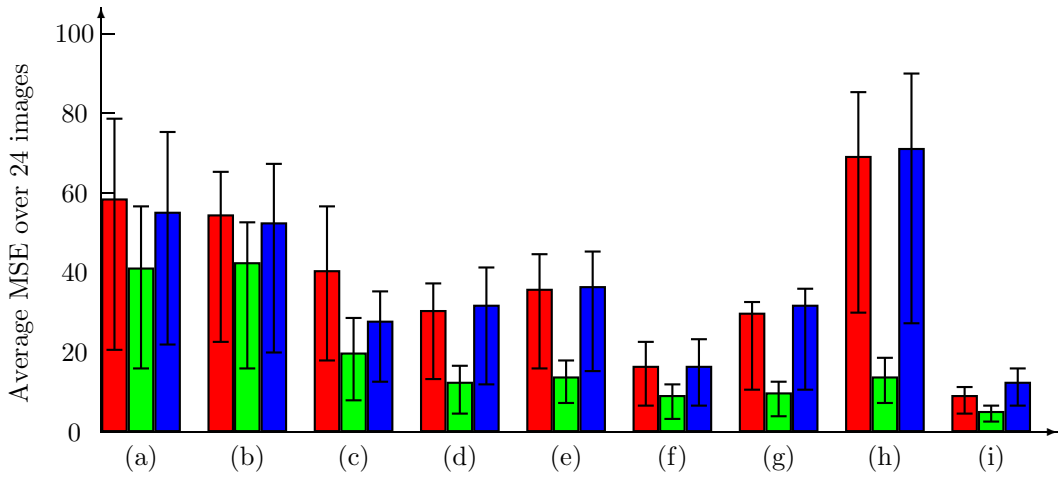


Figure 17: Average mean square error for different algorithms. (a) Edge-directed interpolation in [1] (b) Constant hue-based interpolation in [5] (c) Weighted sum in [7] (d) Second-order gradients as correction terms in [9] (e) Bayesian approach in [15] (f) Homogeneity-directed in [10] (g) Pattern matching (Chang) in [12] (h) Alias cancellation in [13] (i) POCS in [14]

The numbers can only provide part of the overall story. An important evaluation is the subjective appearance of the output images. For this, several example images are presented. The *Sail* image in Figure 18 shows how the algorithms respond to sharp transitions. An example of zipper effect artifacts can be seen in (d) and (e). Most of the algorithms do a decent job reconstructing this example. Figure 19 shows the *Lighthouse* image. This example includes a picket fence from a perspective that increases spatial frequency along the fence. Aliasing is a prominent artifact in this image. The homogeneity-directed interpolation algorithm reconstructs this image best. Very little aliasing is present in the output image. The *Boat* image in Figure 20 contains lines at various angles across the image. This is a good example to show how the algorithms respond to features at various orientations. The POCS algorithm and the homogeneity-directed interpolation algorithm show very few of the aliasing artifacts present in the other output images. This shows that these algorithms are fairly robust to the orientation of various features. According to the MSE measurements, POCS is the best algorithm, but the output images from the homogeneity-directed method

have fewer artifacts. This suggests the need to use subjective evaluations along with objective measures.

In [22], Longere et al. provide a perceptual assessment of demosaicking algorithms. They compared several algorithms with a subjective experiment. The results of their first experiment showed that the subjects favored sharpness and the algorithms providing a sharp image were highly favored. The experiment was repeated with the result images normalized for sharpness. After this adjustment, the results showed more variation and no one algorithm was highly favored.

4 Conclusions and Future Directions

The sensor size of digital cameras continues to decrease, providing sensor arrays with larger numbers of pixels. Today, five and six mega-pixel cameras are common. The increased sampling rate of these cameras reduces the probability of aliasing and other artifacts. Also, Foveon has invented an imaging sensor, the X3 sensor, that is able to capture red, green, and blue information at every pixel – eliminating the need for demosaicking in the digital camera pipeline.

However, research into the demosaicking problem is still an important problem. This research has provided an understanding of the image modelling process. The estimation methods discussed in this article describe the image formation process, describing how the natural scene is transformed into a digital image. The correlation between the three color planes has also been explored. This extends beyond three color planes into hyperspectral image processing.

Processing time is often an important measure for algorithms implemented in real-time systems. A photographer needs to be able to take pictures at a fast rate and the image processing can sometimes limit this. Several cameras, especially the more expensive digital SLR cameras, provide access to the raw image data captured by the sensor. With this data, the images can be processed at a later time on a computer. In this case, processing time is

not critically important. Therefore, algorithms that perform well, but are computationally complex, can still be considered in off-line processing applications.

References

[1] R. H. Hibbard, "Apparatus and method for adaptively interpolating a full color image utilizing luminance gradients," *U.S. Patent 5,382,976*, January 1995.

[2] C. A. Laroche and M. A. Prescott, "Apparatus and method for adaptively interpolating a full color image utilizing chrominance gradients," *U.S. Patent 5,373,322*, December 1994.

[3] D. R. Cok, "Signal processing method and apparatus for producing interpolated chrominance values in a sampled color image signal," *U.S. Patent 4,642,678*, February 1986.

[4] J. A. Weldy, "Optimized design for a single-sensor color electronic camera system," *SPIE*, vol. 1071, pp. 300–307, 1988.

[5] J. E. Adams, "Interactions between color plane interpolation and other image processing functions in electronic photography," *Proc. SPIE Cameras and Systems for Electronic Photography and Scientific Imaging*, vol. 2416, pp. 144–151, February 1995.

[6] W. T. Freeman, "Method and apparatus for reconstructing missing color samples," *U.S. Patent 4,774,565*, 1988.

[7] R. Kimmel, "Demosaicing: image reconstruction from ccd samples," *Proc. Trans. Image Processing*, vol. 8, pp. 1221–1228, 1999.

[8] W. Lu and Y.-P. Tan, "On new method and performance measures for color filter array," *to appear in IEEE Trans. Image Processing*, 2003.

[9] J. E. Adams and J. F. Hamilton, "Design of practical color filter array interpolation algorithms for digital cameras," *Proc. SPIE Real Time Imaging II*, vol. 3028, pp. 117–125, February 1997.

[10] K. Hirakawa and T. W. Parks, "Adaptive homogeneity-directed demosaicing algorithm," *to appear in Proc. IEEE Int. Conf. Image Processing*, 2003.

[11] X. Wu, W. K. Choi, and P. Bao, "Color restoration from digital camera data by pattern matching," *SPIE*, vol. 3018, pp. 12–17, 1997.

[12] E. Chang, S. Cheung, and D. Y. Pan, "Color filter array recovery using a threshold-based variable number of gradients," *SPIE*, vol. 3650, pp. 36–43, 1999.

[13] J. W. Glotzbach, R. W. Schafer, and K. Illgner, "A method of color filter array interpolation with alias cancellation properties," in *Proc. IEEE Int. Conf. Image Processing*, 2001, vol. 1, pp. 141–144.

[14] B. K. Gunturk, Y. Altunbasak, and R. M. Mersereau, "Color plane interpolation using alternating projections," *IEEE Trans. Image Processing*, vol. 11, no. 9, pp. 997–1013, September 2002.

[15] J. Mukherjee, R. Parthasarathi, and S. Goyal, "Markov random field processing for color demosaicing," *Pattern Recognition Letters*, vol. 22, pp. 339–351, 2001.

[16] P. L. Combettes, "The foundations of set theoretic estimation," *Proc. of the IEEE*, vol. 81, no. 2, pp. 182–208, February 1993.

[17] Y. Hel-Or and D. Keren, "Image demosaicing method utilizing directional smoothing," *U.S. Patent 6,404,918*, July 2002.

[18] S. Geman and D. Geman, "Stochastic relaxation, gibbs distributions and the bayesian distribution of images," *IEEE Trans. Pattern Analysis and Machine Intelligence*, , no. 6, pp. 721–741, 1984.

[19] D. Taubman, "Generalized wiener reconstruction of images from colour sensor data using a scale invariant prior," in *Proc. IEEE Int. Conf. Image Processing*, 2000, vol. 3, pp. 801–804.

[20] H. J. Trussell and R. E. Hartwig, "Mathematics for demosaicking," *IEEE Trans. Image Processing*, vol. 3, no. 11, pp. 485–492, April 2002.

[21] D. H. Brainard, "Bayesian method for reconstructing color images from trichromatic samples," *Proc. IS & T 47th Annual Meeting*, pp. 375–380, 1994.

[22] P. Longere, X. Zhang, P. B. Delahunt, and D. H. Brainard, "Perceptual assessment of demosaicing algorithm performance," *Proceedings of the IEEE*, vol. 90, no. 1, pp. 123–132, January 2002.

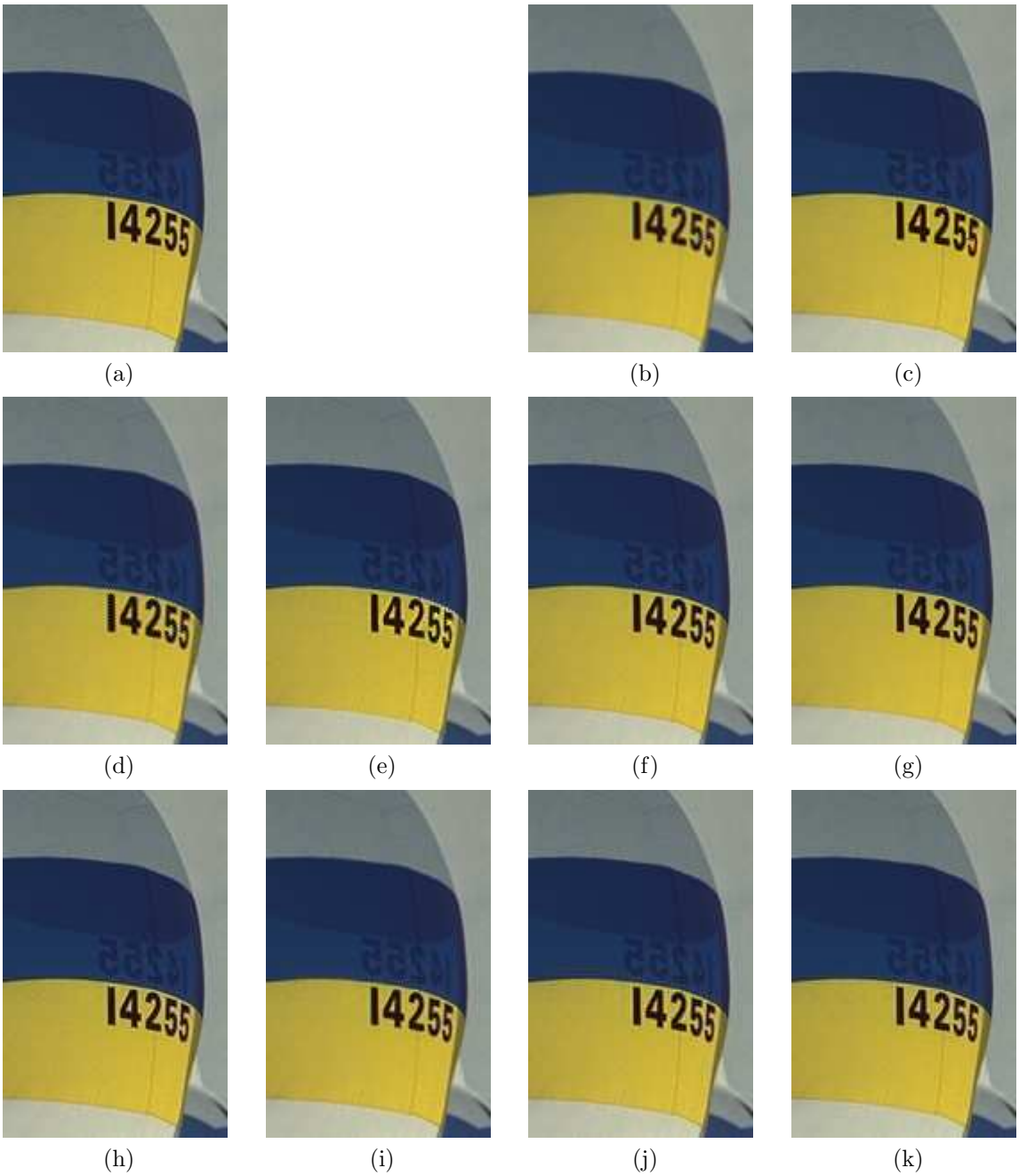


Figure 18: Result images for example sail image. (a) Original image (b) Bilinear interpolation (c) Edge-directed interpolation in [1] (d) Constant hue-based interpolation in [5] (e) Weighted sum in [7] (f) Second-order gradients as correction terms in [9] (g) Bayesian approach in [15] (h) Homogeneity-directed in [10] (i) Pattern matching (Chang) in [12] (j) Alias cancellation in [13] (k) POCS in [14]

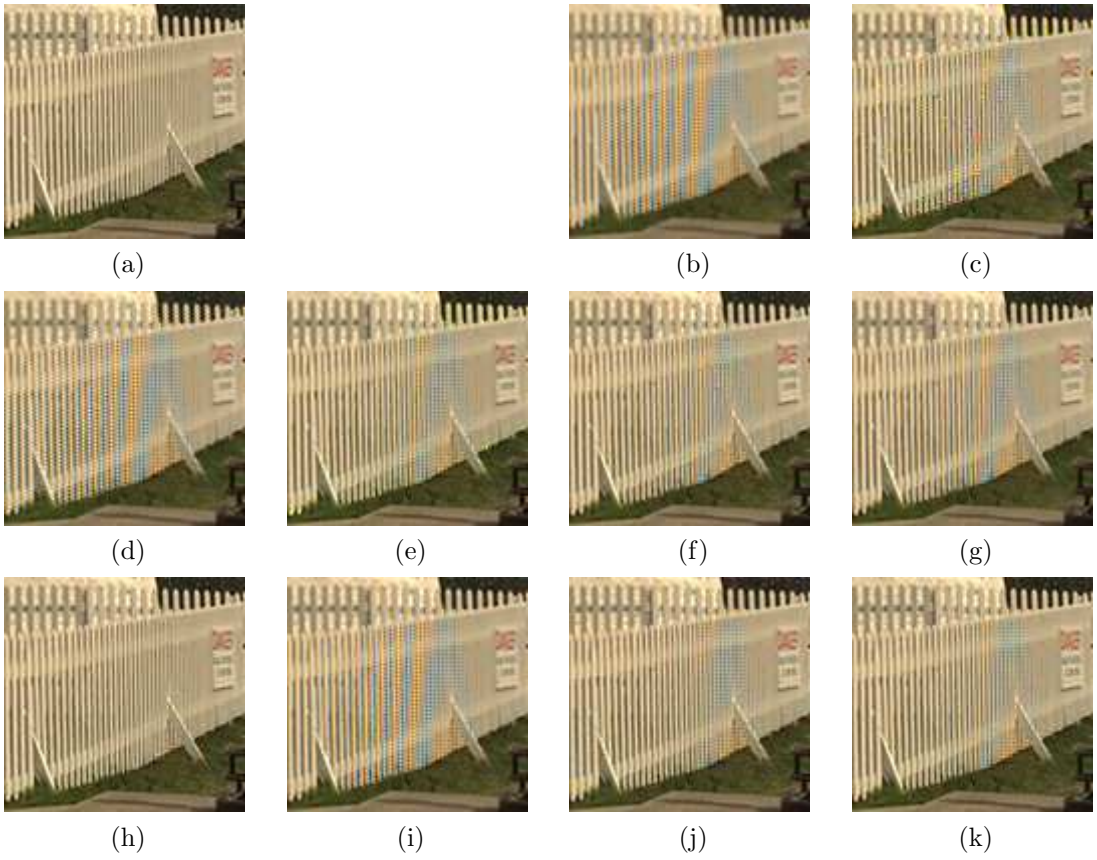
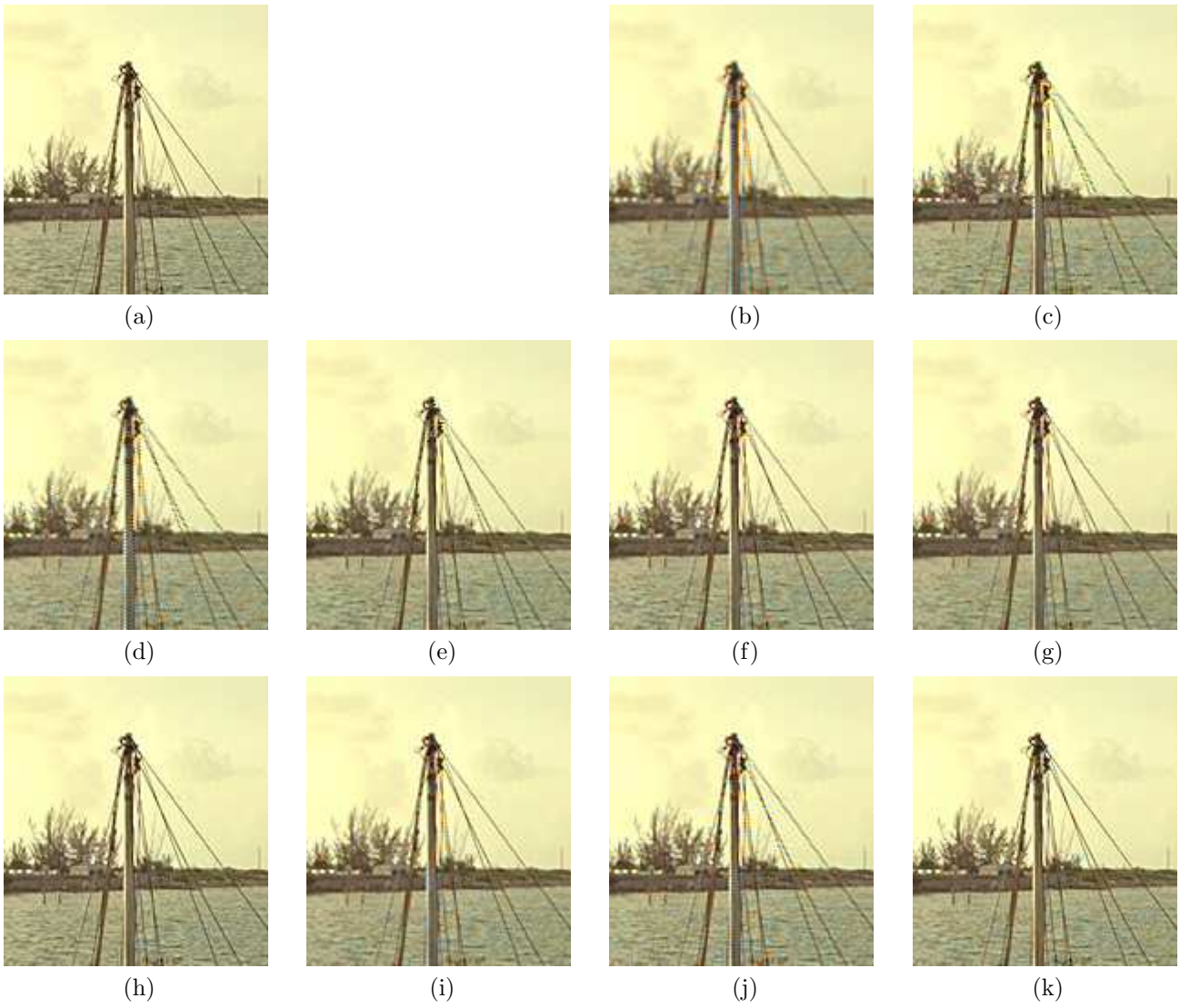


Figure 19: Result images for example lighthouse image. (a) Original image (b) Bilinear interpolation (c) Edge-directed interpolation in [1] (d) Constant hue-based interpolation in [5] (e) Weighted sum in [7] (f) Second-order gradients as correction terms in [9] (g) Bayesian approach in [15] (h) Homogeneity-directed in [10] (i) Pattern matching (Chang) in [12] (j) Alias cancellation in [13] (k) POCS in [14]



(a) (b) (c)

(d) (e) (f) (g)

(h) (i) (j) (k)

Figure 20: Result images for example boat image. (a) Original image (b) Bilinear interpolation (c) Edge-directed interpolation in [1] (d) Constant hue-based interpolation in [5] (e) Weighted sum in [7] (f) Second-order gradients as correction terms in [9] (g) Bayesian approach in [15] (h) Homogeneity-directed in [10] (i) Pattern matching (Chang) in [12] (j) Alias cancellation in [13] (k) POCS in [14]