Enhancing Demosaicking Algorithms using Loopy Propagation

Kevin Grant, David Mould, Michael Horsch, Eric Neufeld
Department of Computer Science
University of Saskatchewan
57 Campus Drive
Saskatoon, SK
S7N 5A9
{kjg658, mould, horsch, eric}@cs.usask.ca

Keywords: Demosaicking, iterative relaxation, demosaicking algorithm, digital photography

Abstract

Consumer-level digital cameras observe a single value at each pixel. The remaining two channels of a three-channel image are reconstructed through a process called demosaicking. This paper describes a methodology for enhancing current demosaicking methods. Using an iterative relaxation approach from probabilistic AI literature, our empirical results show that we can improve the results of the standard algorithms using monitored successive application of those algorithms. We apply the new technique to several algorithms: hue-based interpolation, gradient-based interpolation, and adaptive colour plan interpolation; and we show a significant improvement in mean-squared error over both RGB and CIE colour spaces using each of these algorithms.

1 Introduction

A digital still camera employs a sensor called a charge-coupled device (CCD) that converts incoming light image to a digital image. The voltage at each photodiode (pixel) is directly proportional to the light intensity. Hence, the image coming through this CCD is grayscale. In order to invoke colour in an image, a colour filter must be placed over the diodes.

Professional level digital cameras make use of a mirror which directs the light through three sensors, each overlain with a specified colour filter. The resulting digital images are then recom-

bined, creating an image with all three colour channels. Although this yields the highest quality images, the cameras are both expensive and bulky due to the use of three CCD photodiodes per pixel.

Another way of capturing the colour is to rotate three filters in front of the same image. This reduces the cost of the camera considerably, but does little to reduce its size. The camera is ineffective for high-speed photography, as three different images taken at different moments must be recombined.

In order to create an inexpensive all-purpose camera, each photodiode is overlain with one colour filter. The two unknown channel values for a particular diode are estimated using interpolation methods. The most common filter pattern is the Bayer colour array [1], shown in Figure (1). The dominant green (luminance) channel is sampled as much as the red and blue (chrominance) channels combined. The human eye is most responsive to changes in green; that is, the efficiency of human observation peaks very close to the wavelength of green light [14] (see Figure 2). The interpolation of the chrominance channels in most algorithms is directly affected by the quality of the luminance estimation. Therefore, effective interpolation of luminance is of utmost importance. That being said, the green channel is sampled at 50%, so standard demosaicking techniques work very well. In fact, the luminance channel is generally unresponsive to the iterative techniques of this paper. In some cases, the progress is negative.

We focus on improving the chrominance channel interpolation given an effective luminance estimate. Given observed information (the known colours), and hypothesized colours

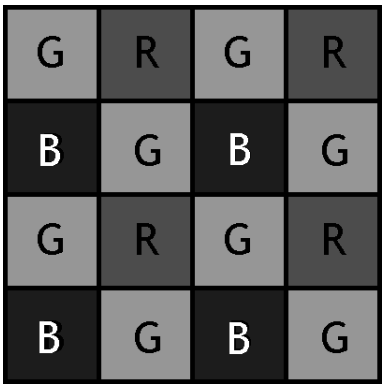

Figure 1: The Bayer colour array (US Patent 3,971,065).

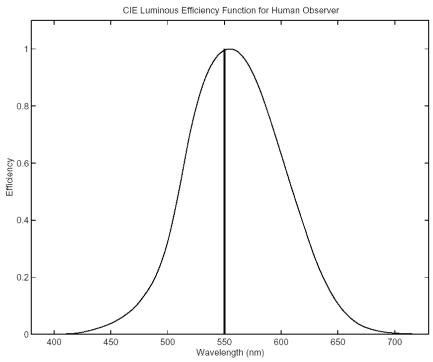

Figure 2: Luminous efficiency of human observer (from Ramanath et al.). Note that the wavelength of green light is denoted by the vertical line.

given these observed values (the unknown values), the problem is quite naturally modelled by probability theory $P(Hypothesis|Evidence)$. Naively applying a Bayesian approach yields an intractable result, even using approximation methods. However, by utilizing the underlying mechanics of a well known approximation scheme, Loopy Belief Propagation [12], we empirically demonstrate that the major algorithms discussed by Ramanath [14] can be consistently improved.

The structure of the paper is as follows. Section 2 discusses some preliminaries, both in Bayesian calculation and demosaicking strategies. Section 3 outlines the previous and current research in the area of image reconstruction, with an emphasis on Bayesian techniques and demosaicking. Section 4 outlines the parameters of the new algorithm. Section 5 shows our results and gives a comparison between previous algorithms. Finally, section 6 outlines the conclusions of the paper, and presents some further research ideas.

2 Preliminaries

2.1 Imaging

Demosaicking an image involves interpolating missing colour information given some observed values. Let **V** represent the set of missing colours, and let **O** represent the set of observed values. We want to develop a function **F** that takes the observed values as a parameter and returns the unobserved values:

$$\mathbf{V} = \mathbf{F}(\mathbf{O}). \tag{1}$$

In most cases, each unknown value is estimated independently of other colours. Let V_i denote the ith value of **V**. Then:

$$V_i = \mathbf{F}(\mathbf{O}) \tag{2}$$

for all i. In general, the function applied will be dependent upon the particular value from **V** that we are deriving. Hence, let F_i denote the function applied to **O**, $F_i \in \mathbf{F}$. Then:

$$V_i = F_i(\mathbf{O}). \tag{3}$$

The *neighbourhood* of an unknown value V_i, denoted as $N_i \subset \mathbf{O}$, is a subset of the observed values with the property that an instantiation of values for N_i renders the estimation of V_i independent of the rest of **V**. More formally:

$$V_i = F_i(\mathbf{V}) = F_i(N_i) \tag{4}$$

The size of the neighbourhood directly influences the complexity and quality of the algorithm [7, 15]. Hence, choosing a proper neighbourhood is critical. In this work, we tested a variety of neighbourhood sizes.

2.2 Loopy Propagation

Belief Propagation is a polynomial-time inference algorithm for Bayesian polytrees [13], based on message passing between nodes in the graph. This algorithm can be generalized to undirected graphs (for example, to Markov random fields); in this presentation, we consider undirected graphs.

Let X denote a node, and let $U_1 ... U_k$ denote its neighbours. Let $M_X(U_i)$ denote the message passed from neighbour i, which is a distribution

over U_i. The formula for computing posterior probabilities for a node X is as follows:

$$Bel(X) = \alpha \sum_{\mathbf{U}} P(X|\mathbf{U}) \prod_{i=1}^{k} M_X(U_i), \quad (5)$$

where α is a *normalizing* constant.

The message passing algorithm sends messages from a node to its neighbours. For correctness, the outgoing message $_{U_i}(X)$ to a node U_i must disregard any incoming message received from U_i:

$$M_{U_i}(X) = \sum_{\mathbf{U} \cup X - U_i} P(X|\mathbf{U}) \prod_{j=1, j \neq i}^{k} M(U_j). \quad (6)$$

When a node has received all messages from all neighbours except U_i, it can send a message to U_i.

For trees, the belief propagation algorithm is complete and correct. However, it has been shown [12] that applying the algorithm to general graphs can lead to rapid convergence to a reasonable approximation of the correct distributions (hence the name Loopy Belief Propagation). In general graphs, the algorithm starts by having all nodes send initial, uninformative messages.

Loopy Belief Propagation is the inspiration of the approach of this paper; we do not, however, compute probabilities, as this is computationally infeasible. Rather, we use the key ideas to estimate colour directly: The incoming messages are used for local computation, and outgoing messages summarize all of the information received, *omitting* any information received from the neighbour to which the message is going.

3 Previous Work

3.1 Demosaicking

Most demosaicking algorithms use the Bayer colour array [1] (see Figure (1)). The four algorithms compared in this paper were summarized by Ramanath [14]. They include bilinear interpolation, hue-based interpolation [4], gradient-based interpolation [11], and adaptive color plan interpolation [8]. As well, we consider the effects of median filtering [6], also discussed by Ramanath. Details of these algorithms will be discussed in subsequent sections.

Ramanath notes two major types of errors in reconstructed images:

1. The *zipper effect*, which causes blurring across the border of an intensity step.

2. *Confetti error*, which produces a "speckled" effect around either a dark spot in a bright neighbourhood or a bright spot in a dark neighbourhood.

The conclusions were that gradient-based and adaptive color plan interpolation produced very little zipper effect, while median filtering successfully reduced confetti error in most cases.

Demosaicking can be modelled as a uncertainty problem in probability. Bayesian approaches to image reconstruction began almost 20 years ago;Geman and Geman [7] used a Markov Random field approach to reconstruct images that had suffered various forms of deformation. Similar to the message-passing approach, they used Gibbs sampling and local computation to calculate probabilistic distributions over the gray-scale colours, with excellent results.

More recently, Brainard [3] took a more biological approach, studying closely the mechanics of the human optical system and trying to mimic them in engineering application. The human eye operates similarly to a digital camera, in that we sample individual color channels at individual locations and interpolate the missing colours. Experiment shows that the human visual system uses all three colour channels simultaneously when interpolating the missing colours, rather than independently estimating the three colour channels. It would be otherwise impossible for humans to interpolate colours so flawlessly (since the frequency is well above the Nyquist limit). This explains the effectiveness of the gradient-based and adaptive interpolation algorithms, which use the three colour channels in conjunction with one another. Brainard also developed a Bayesian algorithm for reconstruction based on these results [2].

Kimmel [10] introduces an algorithm that in some respects is similar to the one presented here. The algorithm is based on Cok's template matching algorithm [5]. Then the colours are adjusted across edges in the photograph using a gradient function. Finally, an enhancement is applied using inverse diffusion.

For the purposes of the paper, only the algorithms from Ramanath et al. [14] were used. The gradient and adaptive interpolation algorithms seemed to produce better results than Kimmel's method [10]; hence, we chose to focus on these algorithms instead.

4 The Algorithm

We chose to directly calculate colour based on the incoming messages, rather than computing a probability for each possible colour. This reduced the complexity of the algorithm to $\mathbf{O}(cnk)$, where n is the number of variables, k is the size of the largest family, and c is the overhead of combining messages from neighbours.

The application of loopy propagation was straightforward. The messages of a node consisted of the neighbourhood values (defined earlier as N_i). Hence, given neighbours N_i, the following represents the update function for value V_i:

$$V_i = F_i(\mathbf{M}_{V_i}(N_i)). \tag{7}$$

The combination functions that were tested were simply the functions from Ramanath et al. [14]. In most cases, the luminance functions differed from the chrominance functions.

The second parameter is the definition of the outgoing messages from each node. However, each function defines the combination of each individual message separately. Hence, for neighbour $U_j \in N_i$, the following represents the message $M_{U_j}(V_i)$:

$$M_{U_j}(V_i) = F_i(\mathbf{M}_{V_i}(N_i) - M_{V_i}(U_j)). \tag{8}$$

The third parameter is neighbourhood considerations. Geman and Geman addressed this problem. Although larger neighbourhoods logically encode more data, they contribute exponentially to the complexity of the algorithm. In this paper, two different neighbourhood types are considered:

1. 4-Neighbourhood: the pixels immediately

above, below, and to the left and right of the pixel in question.

2. 8-Neighbourhood: the 4-neighbourhood plus the adjacent corners.

Using the standard approach, the effective neighbourhood size is limited to wherever colours are observed. For example, interpolating red on a green square limits interpolation to two neighbouring values. The iterative nature of our algorithm allows the neighbourhood to be extended. That is, after the first iteration, all colour channels at all pixel locations have been filled in, and can be used in future calculations as an "observed" value.

The final parameter is the choice of dummy message, that is, the message sent when there is not one to be calculated (i.e., initially). We follow the lead of Murphy et al. [12], who achieved good results using an identity message (a message that does not affect the resulting computation). We also employed a special identity message, telling a receiving node to disregard this message it in its computation. Hence, in some cases, depending on the neighbourhood, the algorithm required at least 2-3 passes before finishing.

4.1 Details

The following describes the finer details of each algorithm tested. It presents the original algorithm, an interpretation of its design, the combination function for the algorithm, and the structure of the messages sent from each pixel.

For the following, let X represent the current pixel, and let Y_X represent the unknown colour value (which can be any of the three colour channels). Let G_X represent the green value at the current pixel location, let Z_X represent the opposite chrominance value as the one being interpolated, and let O_X represent the observed value at pixel X. Let the neighbours of the current pixel be represented by $U_1..U_k$, and by subscripting Y, G, or Z with U_i or simply i, we represent the corresponding colour channels at each neighbour. Let $M_{Y_X}(U_i)$ represent the message to X from U_i representing the colour Y.

4.1.1 Bilinear Interpolation

Bilinear Interpolation is the simplest approach to demosaicking. It is simply the arithmetic average of the surrounding observed colour values. That is, for each colour channel, the interpolated value of Y is:

$$Y_X = (1/k) \sum_{j=1}^{k} Y_j. \tag{9}$$

In the iterative algorithm, Y_i is simply represented by the messages sent by U_i:

$$Y_X = (1/k) \sum_{j=1}^{k} M_{YsX}(U_j). \tag{10}$$

The message sent to neighbour U_i from Y is:

$$M_{YU_i}(X) = (1/(k-1)) \sum_{j=1, j\neq i}^{k} M_{YX}(U_j). \tag{11}$$

This represents the interpolated colour value minus the input from U_i. In the following, we will assume that whenever the algorithm calls for a neighbouring value, it will use the message sent from the corresponding neighbour.

4.1.2 Hue-based Interpolation

Hue-based interpolation was first proposed by Cok [4], and is based on the observation that the red:green and blue:green ratios remain consistent amongst objects in a picture. In the first pass, the green colour channel is completed using bilinear interpolation. For each chrominance channel Y, let the surrounding observed colors be represented by Y_i for each neighbour U_i, $1 \leq i \leq k$. Then, each unknown value is interpolated using the following:

$$Y_X = G_X(\frac{1}{k}) \sum_{j=1}^{k} M_{YX} \frac{U_j}{G_j}. \quad (12)$$

The message sent to neighbour U_i from Y is:

$$M_{YU_i}(X) = G_X(\frac{1}{k-1}) \sum_{j=1, j\neq i}^{k} M_{YX} \frac{U_j}{G_j}. \quad (13)$$

4.1.3 Gradient-based Interpolation

Gradient-based interpolation was introduced by Laroche and Prescott [11]. Like hue-based interpolation, it also has two separate algorithms for the luminance and chrominance channels. However, the interpolation of the green channel involves a decision. Let $p_i = \langle L, R \rangle, L, R \in X$ represent two pixels that are on geometric opposite sides of X, and let P represent the set of all $p_i, 1 \leq i \leq h$. For each X_i neighbouring X, let W_{X_i} represent the neighbour of X_i on the geometrically opposite side of X_i from X. Then, we create classifiers α_i corresponding to each $p_i \in P$:

$$\alpha_i = abs((O_{W_L} + O_{W_R})/2 - O_X). \quad (14)$$

Each α_i represents the second derivative for the colour channel O_X running in the direction from L to R. We choose the smallest gradient (i.e. the highest probability of belonging to the same object) to interpolate the green value from the surrounding pixels. Let α_{min} represent the smallest classifier, and let $p_{min} = \langle L_{min}, R_{min} \rangle$ be the corresponding pair. Then:

$$G_X = (G_{L_{min}} + G_{R_{min}})/2. \quad (15)$$

If there is more than one minimum α value, then the result for each is calculated, and G_X becomes the arithmetic average of all minima.

Calculating the message involves the classifiers as well. The message sent to L_{min} is $G_{R_{min}}$, and $G_{L_{min}}$ is sent to R_{min}. The other neighbours simply receive G_X.

The chrominance channel calculations are not based on classifiers. The combination function is given by:

$$Y_X = G_X + \frac{1}{k} \sum_{j=1}^{k} (M_{Y_X}(U_j) - G_j). \quad (16)$$

and the message passed to neighbour i is:

$$M_{YU_i}(X) = G_X + \frac{1}{k-1} \sum_{j=1, j\neq i}^{k} (M_{YX}(U_j) - G_j). \quad (17)$$

4.1.4 Adaptive Color Plan Interpolation

This is a slight variation of gradient-based interpolation. It uses classifiers for both luminance and chrominance interpolation.

The classifier for the green channel is similar to that of gradient-based interpolation, only

slightly more sophisticated. Using the same terminology as the previous section, the classifiers are defined by:

$$\alpha_i = | - O_{W_L} + 2O_X - O_{W_R}| + |G_L - G_R|. \quad (18)$$

Letting α_{min} represent the smallest classifier, and $p_{min} = \langle L_{min}, R_{min} \rangle$ be the corresponding pair, we have:

$$\begin{aligned} G_X &= \frac{G_{L_{min}} + G_{R_{min}}}{2} \\ &+ \frac{-O_{W_{L_{min}}} + 2O_X - O_{W_{R_{min}}}}{4} \end{aligned}$$

Again, if there is more than one minimum α value, then the results for each is calculated, and G_X becomes the arithmetic average of all of them.

The interpolation of a chrominance value is also based on classifiers:

$$\alpha_i = | - G_L + 2G_X - G_R| + |Y_L - Y_R|. \quad (19)$$

Then, the value of Y_X becomes:

$$\begin{aligned} Y_X &= \frac{Y_{L_{min}} + Y_{R_{min}}}{2} \\ &+ \frac{-G_{L_{min}} + 2G_X - G_{R_{min}}}{4}. \end{aligned}$$

with the same rule about multiple minimal classifiers.

During testing, we observed that the adaptive color plan iterative algorithm would not converge, but would instead oscillate erratically; the oscillation was caused by a flip in the minimum alpha values. We resolved the issue by using the classifiers in a different manner. Rather than have the classifiers represent a boolean value (either use their corresponding colour values or disregard them), we used them as inverse weights. The smaller the classifier, the larger the effect that the corresponding colour values contributed to the final result. Let T_i represent the result that would be obtained if α_i were the lowest classifier. The weights were normalized, converted such that they sum to 1.0 while maintaining their relative proportionality. Let β_i represent the normalized result of α_i. The resulting value is as follows:

$$Y_X = \sum_{i=1}^{h} \beta_i T_i. \quad (20)$$

With this modification, the adaptive color plan algorithm converged similarly to the other algorithms.

5 Results

When testing the algorithms, two error metrics were used. The first was Mean Squared Error (MSE), which is the Euclidean distance of the real colour value vs. the interpolated colour value in RGB. The second is the ΔE^*_{ab} colour error, which is the MSE in the CIE colour space. The second metric is more relevant to the case of digital photography, as it is more closely aligned with human perception [14].

A number of test photos were used for comparison. In particular, four representative photos were chosen (see Figure 3). We included the parrot photograph from Ramanath et. al that displays high spatial variance [14]. We chose the sailboat and lighthouse pictures from Kimmel, as they produce interesting artifacts. Finally, we chose the statue picture from Kimmel, to test the sharpness of our algorithm. It is easy enough

Figure 3: Four example images: the Parrot image from Ramanath et.al, Sailboat image from Kimmel; the Lighthouse image from Kimmel; the Statue image from Kimmel

to show using the conventional algorithms that the quality of the chrominance interpolation is directly dependent on the quality of the green interpolation (with the exception of bilinear interpolation). We have only three methods of green interpolation (as hue-based interpolation uses bilinear interpolation to calculate its luminance channel).

In all of our testing, the error in the luminance monotonically increases as the iterations increase, regardless of the picture or neighbourhoods chosen. We conclude that the iterative approach is ineffective for the green channel, and in all future tests, the green channel is interpolated using conventional methods. [1]

Next, we analyzed the convergence effect of each algorithm. The test was done using our four test pictures. Note that only one iteration is performed on the green channel. Only ΔE^*_{ab} is shown; the results for MSE are similar. Figures 4, 5, 6, and 7 show the convergence effect for each case. All of the algorithms benefited from this iterative relaxation approach, with the exception of bilinear. The quality of the bilinear interpolation consistently decreased with iterations. Hence, it was disregarded in further analysis.

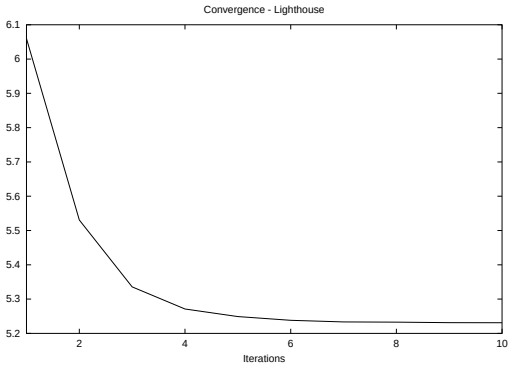

Figure 4: ΔE^*_{ab} rates on *lighthouse*

[1] It is worth noting at this point that a full neighbourhood and one iteration corresponds exactly to the corresponding conventional method in all of the interpolation algorithms.

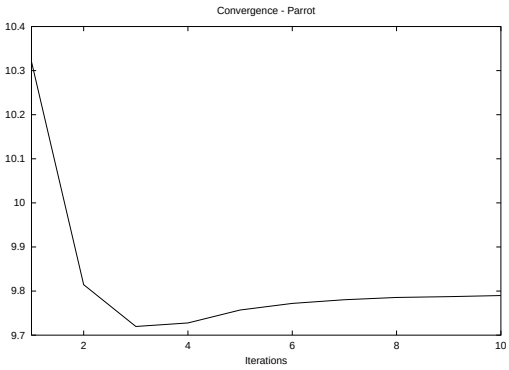

Figure 5: ΔE^*_{ab} rates on *parrot*

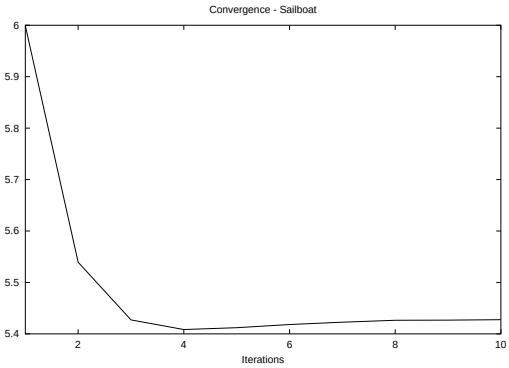

Figure 6: ΔE^*_{ab} on *statue*

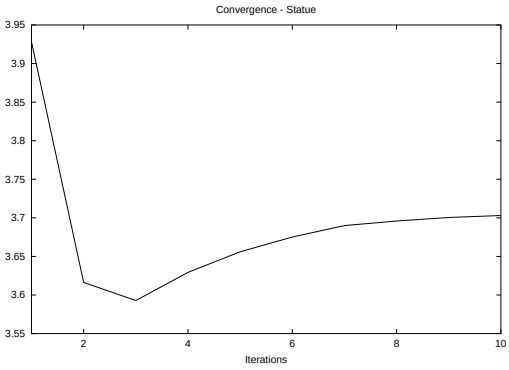

Figure 7: ΔE^*_{ab} on *sailboat*

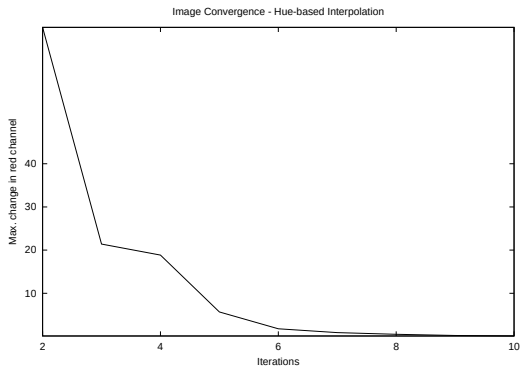

Figure 8: The maximum change in the red channel using hue-based interpolation.

In the initial iterations, there is a significant decrease in the error. Unfortunately, in most cases, the error derivative changes sign very early on, and the error begins to increase. We attribute this behaviour to the locality of objects in an image: during the first few iterations, a pixel receives information from nearby pixels, which are part of the same object, and this increases the quality of the interpolation. However, after a number of iterations, the pixel receives information from outside its own object, contaminating the original information and causing a loss of interpolation quality.

Even though the error does not monotonically decrease, the algorithm still converges to a solution. Figures 8, 9, and 10 show the change in the red channel per iteration. The same results are seen in the blue channel. Hence, as the iterations progress, the changes in the colour channels decrease. From these results, we can specify a termination for the algorithm, when the change in the picture falls below a specific threshold.

Based on these results, the algorithm is as fol-

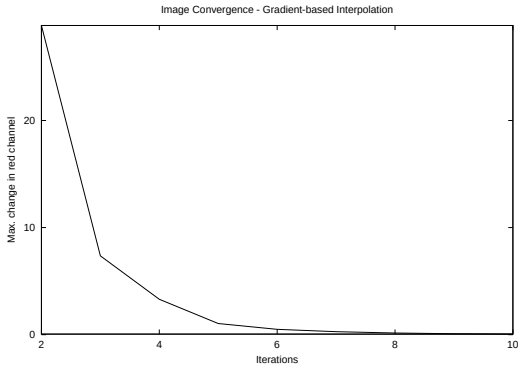

Figure 9: The maximum change in the red channel using gradient-based interpolation.

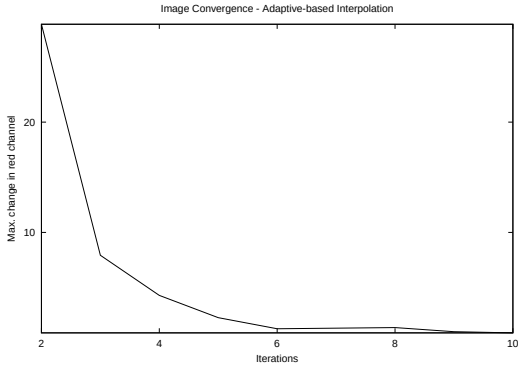

Figure 10: The maximum change in the red channel using adaptive color plan interpolation.

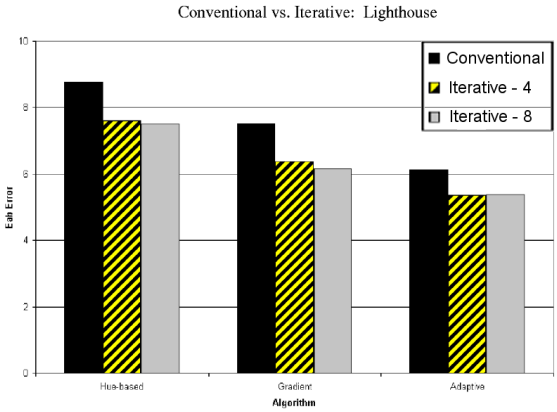

Figure 11: Algorithmic comparison using *Lighthouse* image

lows:

1. Interpolate green channel using one of the three algorithms

2. Interpolate red channel using one of the four algorithms, until either the threshold is reached, or the maximum number of iterations has been reached.

3. Interpolate blue channel using one of the four algorithms, until either the threshold is reached, or the maximum number of iterations has been reached.

4. Recombine the four channels into an image; return this image.

Figures 11, 12, 13, and 14 show comparisons between the conventional methods and the iterative methods. Iterations varied between 3 and 12, depending on the method.

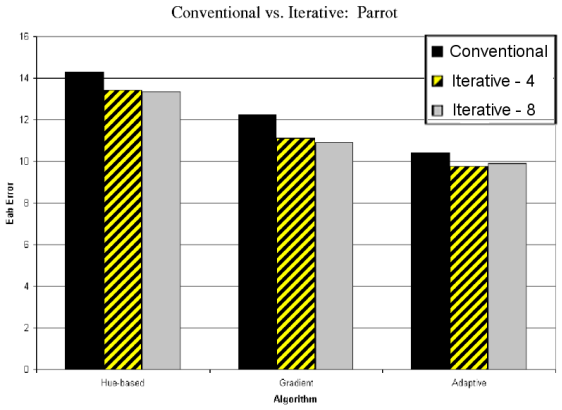

Figure 12: Algorithmic comparison using *Parrot* image

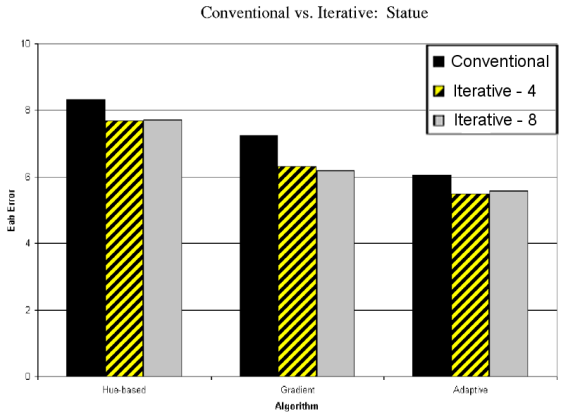

Figure 13: Algorithmic comparison using *Statue* image

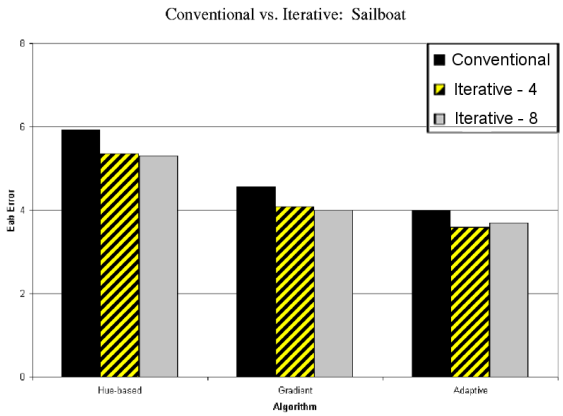

Figure 14: Algorithmic comparison using *Sailboat* image

5.1 Discussion

The iterative approach improved on the original results for all combinations of image and algorithm. In most cases, the improvement was significant, providing up to 8% improvement over the standard approach. The optimal neighbourhood varied with the algorithm; the hue-based and gradient based algorithms generally saw their best results with a full 8-neighbourhood, whereas the adaptive algorithm worked best with a 4-neighbourhood. However, the effect of the neighbourhood was marginal, and similar performance was observed for both neighbourhoods.

The iterative approach inherently affords certain advantages. It requires minimal additional resources beyond that necessary to run the underlying conventional algorithm, and in consequence is suitable for deployment on a digital camera. Iterative algorithms have desirable characteristics in a real-time context: partial results are available immediately but more pro-

cessing can still be done when time permits. Finally, the iterative message-passing approach is an instance of a methodology which can potentially be applied to other algorithms beyond those for which we report results in this paper.

Our algorithm has higher cost than does the the original, which had complexity $\mathbf{O}(nk)$ for n image pixels and neighbourhood size k. Our iterative method has cost $a \times n \times (k + k^2)$ = $\mathbf{O}(ank^2)$ where a is the number of iterations. The increase, a factor of ak, is significant but manageable; in an onboard context, the algorithm would be run in parallel by dedicated hardware. Also, the incremental nature of the algorithm means that partial results are available at lower cost.

6 Conclusions and Future Work

The iterative approach described above can be used to generate better pictures, that is, pictures that more closely resemble the original scene. In all but a few cases, the iterative approach had error lower than that of the corresponding conventional algorithm. The cost of the iterative approach is polynomial, a factor of n higher than its conventional counterpart; however, this cost is customizable, in that we can halt and report the results at any time. In the limiting case, we perform one iteration, exactly equivalent to running the conventional algorithm.

The overhead required to deploy the iterative algorithm onboard a digital still camera is minimal, as the method enhances other algorithms already so deployed. The only major addition would be a source of RAM for storing the intermediate pixel values and message values, and the logic for computing message values.

We reported that the error reduction is not monotonic, and proposed the explanation that increase in error is caused by messages from outside the appropriate local neighbourhood. Thus, extensions to the algorithm may involve modelling boundaries of objects, and blocking information from crossing these boundaries. The size of the objects in an image will give us a clue as to the optimal number of iterations. Alternatively, we might approach the issue of deciding optimality by treating the computation as an action under uncertainty [9].

Finally, the weighted approach used in the amended version of adaptive color plan presents a possible new method of demosaicking that will be explored further. The weighted approach more naturally lends itself to the message-passing algorithm, hence, some modifications to our current tactics will perhaps lead to an overall better solution.

7 Acknowledgments

This work was supported in part by NSERC PGS A-255374-2002.

References

[1] B. Bayer. Color imaging array. *United States Patent*, 3,971,065, 1976.

[2] D. H. Brainard. Bayesian method for reconstructing color images from trichromatic sam-

ples. *Proceedings of IS&T's 47th Annual Meeting*, pages 375–380, 1994.

[3] D. H. Brainard and D. Sherman. Reconstructing images from trichromatic samples: from basic research to practical applications. *Proceedings of the 3rd IS&T/SID Color Imaging Conference*, pages 4–10, 1995.

[4] D. Cok. Signal processing method and apparatus for producing interpolated chrominance values in a sampled color image signal. *United States Patent*, 4,642,678, 1987.

[5] D. Cok. Reconstruction of CCD images using template matching. *Proceedings of IS&T's Annual Conference*, pages 380–385, 1994.

[6] W. Freeman. Median filtering for reconstructing missing color samples. *United States Patent*, 4,724,395, 1988.

[7] D. Geman and S. Geman. Stochastic relaxation, Gibbs distributions, and the Bayesian restoration of Images. *IEEE Trans. Pattern Anal. Machine Intell.*, PAMI-6:721–741, 1984.

[8] J. Hamilton and J. Adams. Adaptive color plan interpolation in single sensor color electronic camera. *United States Patent*, 5,629,734, 1997.

[9] E. Horvitz, Y. Ruan, C. Gomes, H. Kautz, B. Selman, and M. Chickering. A Bayesian Approach to Tackling Hard Computational Problems. *Proceedings of the Seventeenth Conference on Uncertainty and Artificial Intelligence*, pages 235–244, 2001.

[10] R. Kimmel. Demosaicing: image reconstruction from color CCD samples. *IEEE Trans. Image Processing*, 8:1221–1228, 1999.

[11] C. Laroche and M. Prescott. Apparatus and method for adaptively interpolating a full color image utilizing chrominance gradients. *United States Patent*, 5,373,322, 1994.

[12] K. Murphy, Y. Weiss, and M. Jordan. Loopy Belief Propagation for Approximate Inference: An Empirical Study. *Proceedings of the Fifteenth Conference of Uncertainty in Artificial Intelligence*, pages 467–475, 1999.

[13] J. Pearl. *Probabilistic Reasoning in Intelligent Systems*. Morgan Kaufman, 1988.

[14] G. L. B. R. Ramanath, W.E. Snyder and W. Sander. Demosaicking methods for Bayer Color Arrays. *Journal of Electronic Imaging*, 11:306–315, 2002.

[15] R. Ramanath. Interpolation Methods for the Bayer Color Array. *M.Sc. Thesis*, 2000.