A FILTERING APPROACH TO EDGE PRESERVING MAP ESTIMATION OF IMAGES

David Humphrey and David Taubman

University of New South Wales,
Sydney, Australia
david.humphrey@student.unsw.edu.au, d.taubman@unsw.edu.au

ABSTRACT

We present a computationally efficient technique for MAP estimation of images in the presence of both blur and noise. The image is divided into statistically independent regions. Each region is modelled with a WSS Gaussian prior. Classical Wiener filter theory is used to generate a set of convex sets in the solution space, with the solution to the MAP estimation problem lying at the intersection of these sets. The proposed algorithm uses an underlying segmentation of the image, and a means of determining the segmentation and refining it are described. The algorithm is suitable for demosaicking of digital camera images and other image restoration problems.

1. INTRODUCTION

We consider the image formation model

$$\boldsymbol{\zeta} = A(\mathbf{g}) + \boldsymbol{\eta} \tag{1}$$

where the original image $\mathbf{g}$ is blurred by the linear shift invariant filter A and has additive Gaussian noise $\boldsymbol{\eta}$. Given this blurred, noisy image $\boldsymbol{\zeta}$ we wish to recover the original image $\mathbf{g}$. Among the many possible approaches to this problem, perhaps the most conceptually appealing is that of maximum a posteriori (MAP) estimation, that is, finding the most probable image $\mathbf{g}$ given the data $\boldsymbol{\zeta}$. MAP estimation has received considerable attention in the literature, with some of the many contributions being [1], [2], [3], [4]. A major difficulty in implementing MAP estimation is that of determining an appropriate prior for the image. Wide sense stationary Gaussian priors lead to computationally simple solutions via the Wiener filter, but are not capable of accurately reproducing image edges. Gibbsian priors have been suggested [2] in order to overcome this difficulty, but then the solution becomes computationally intensive, typically requiring hundreds of iterations of a stochastic relaxation procedure [2].

Although the Wiener filter solution generates artifacts in the vicinity of high dynamic range boundaries, it actually performs very well over the majority of most images. In fact, the Wiener filter solution has been implemented in commercial digital cameras to perform demosaicking [5],[6]. In this paper we address the principal difficulties with Wiener filtering, while retaining the computational simplicity of the filtering solution. We segment the image and model each region with a wide sense stationary Gaussian prior. The segmentation of the image is done based on edge detection applied to the pure Wiener filter result. Each region is assumed to be statistically independent of the other regions. We extend each region using linear prediction. Based on the current estimate of a region, the surrounds of the region, and the extension of the region, we generate the data which would have been observed had this region extended over the whole image. The Wiener filter is then applied to this compensated data. The resulting system, although huge, is linear, and can be solved by a number of iterative methods. Using the Successive Over-Relaxation (SOR) method [7],[8], it is found that the solution converges in just a few iterations. Moreover, the iterative procedure only need be carried out in the vicinity of the region boundaries.

Our approach aims to solve a similar problem to that described in [9]. The key distinction is that we do not require the use of adaptive filter kernels. Also, our method finds the exact solution in the presence of any arbitrary set of edges.

In Section 2 we formulate the problem and present the theoretical foundation of the solution. In Section 3 we discuss the implementation of the solution using a number of different iterative methods. In Section 4 we discuss issues involved in ensuring the solution converges. In Section 5 we present results in the one and two dimensional cases. In Section 6 we discuss some of the issues involved in determining and refining the segmentation.

2. THEORY

The distribution of the image $\mathbf{g}$ given the observed data for the image formation model of equation (1) can be shown to be [2]

$$P(\mathbf{g}|\boldsymbol{\zeta}) \propto \exp\{-(H(\mathbf{g}) + \|\boldsymbol{\zeta} - A(\mathbf{g})\|^2)\} \tag{2}$$

where A is the linear shift invariant operator from equation (1) and $\boldsymbol{\zeta}$ is the observed data. Assuming a Gaussian prior, the function $H()$ is quadratic and is given by

$$H(\mathbf{g}) = \sigma^2.\mathbf{g}.\Gamma^{-1}.\mathbf{g}' \tag{3}$$

where Γ is the autocorrelation matrix for the prior and σ^2 is the noise variance. Instead of choosing a stationary prior, which would lead to the Wiener filter solution, we split the image into a series of N regions, each of which is statistically independent of the others. The prior can then be written

$$H(\mathbf{g}) = \sum_{k=1}^{N} F_k(\mathbf{g}_k) \tag{4}$$

where each of the $F_k()$ is given by

$$F_k(\mathbf{g}_k) = \underset{\tilde{\mathbf{g}}_k}{\operatorname{argmin}}\{E_k(\mathbf{g}_k, \tilde{\mathbf{g}}_k)\} \tag{5}$$

Here, each of the $\mathbf{g}_k$ refers to one of the N regions in the image, the $\tilde{\mathbf{g}}_k$ are the minimum energy extension of each region, and the functions $E_k()$ are stationary Gaussian priors. We consider $\mathbf{g}_k$ to be a section of texture cut out from a complete image of that texture. The definition of $F_k()$ in equation (5) as the energy function resulting from the minimum energy, and hence most likely, extension of $\mathbf{g}_k$ allows us to solve our problem by minimizing across $E_k()$ rather than $F_k()$. Since $E_k()$ is stationary, it admits solution via the linear shift invariant Wiener filter. Our method allows for the prior $E_k()$ to be chosen separately for each region, but in the absence of any knowledge in excess of $\boldsymbol{\zeta}$ the same model may be used over all k. An appropriate model is the scale invariant prior proposed in [5].

We wish to find the most probable value of $\mathbf{g}$ in equation (2) using the energy function of equation (4). Thus the problem to be solved is

$$(\mathbf{g}, \tilde{\mathbf{g}}) = \underset{\mathbf{g}_k, \tilde{\mathbf{g}}_k}{\operatorname{argmin}}\left(\sum_{k=1}^{N} E_k(\mathbf{g}_k, \tilde{\mathbf{g}}_k) + \|\boldsymbol{\zeta} - A(\mathbf{g}_1, ..., \mathbf{g}_N)\|^2\right) \quad (6)$$

To develop the solution we expand the solution space to $(\mathbf{g}_k, \tilde{\mathbf{g}}_k, \tilde{\mathbf{g}}'_k)$, where the $\tilde{\mathbf{g}}_k$ and $\tilde{\mathbf{g}}'_k$ quantities are the extension of each region. At the end this section we will replace the $\tilde{\mathbf{g}}'_k$ with the $\tilde{\mathbf{g}}_k$. In the meantime, $\tilde{\mathbf{g}}_k$ and $\tilde{\mathbf{g}}'_k$ may be considered current and previous states of the extension of the k^{th} region in the iterative method used to solve equation (6). For simplicity of notation, in the rest of this section we will restrict our attention to two regions. The extension to any number of regions is straightforward.

In the expanded space of $(\mathbf{g}_k, \tilde{\mathbf{g}}_k, \tilde{\mathbf{g}}'_k)$ the problem (6) requires that

$$\tilde{\mathbf{g}}_k = \underset{\mathbf{h}}{\operatorname{argmin}}\, E_k(\mathbf{g}_k, \mathbf{h}), \quad k = 1, 2 \quad (7)$$

and in introducing $\tilde{\mathbf{g}}'_k$ we require that

$$\tilde{\mathbf{g}}_k = \tilde{\mathbf{g}}'_k, \quad k = 1, 2 \quad (8)$$

Using equation (8) and the fact that A is linear, we can write

$$A(\mathbf{g}_1, \mathbf{g}_2) = A(0, \mathbf{g}_2 - \tilde{\mathbf{g}}'_1) + A(\mathbf{g}_1, \tilde{\mathbf{g}}_1) \quad (9)$$

and

$$A(\mathbf{g}_1, \mathbf{g}_2) = A(\mathbf{g}_1 - \tilde{\mathbf{g}}'_2, 0) + A(\tilde{\mathbf{g}}_2, \mathbf{g}_2) \quad (10)$$

Minimising separately across $(\mathbf{g}_1, \tilde{\mathbf{g}}_1)$ and $(\mathbf{g}_2, \tilde{\mathbf{g}}_2)$, and applying equations (9) and (10), equation (6) can be rewritten as two minimisation procedures :

$$(\mathbf{g}_1, \tilde{\mathbf{g}}_1) = \underset{\mathbf{g}_1, \tilde{\mathbf{g}}_1}{\operatorname{argmin}}\, E_1(\mathbf{g}_1, \tilde{\mathbf{g}}_1) + \|(\boldsymbol{\zeta} - A(0, \mathbf{g}_2 - \tilde{\mathbf{g}}'_1)) - A(\mathbf{g}_1, \tilde{\mathbf{g}}_1)\|^2 \quad (11)$$

$$(\mathbf{g}_2, \tilde{\mathbf{g}}_2) = \underset{\mathbf{g}_2, \tilde{\mathbf{g}}_2}{\operatorname{argmin}}\, E_2(\mathbf{g}_2, \tilde{\mathbf{g}}_2) + \|(\boldsymbol{\zeta} - A(\mathbf{g}_1 - \tilde{\mathbf{g}}'_2, 0)) - A(\tilde{\mathbf{g}}_2, \mathbf{g}_2)\|^2 \quad (12)$$

Equations (11) and (12) can be recognised as the solution obtained by applying the Wiener filter to the observed data compensated for the existence of the other region, that is to $(\boldsymbol{\zeta} - A(0, \mathbf{g}_2 - \tilde{\mathbf{g}}'_1))$ and $(\boldsymbol{\zeta} - A(\mathbf{g}_1 - \tilde{\mathbf{g}}'_2, 0))$ respectively. Applying the constrained Wiener filter in this manner to region 1, and using matrix notation, we get

$$(\mathbf{g}_1, \tilde{\mathbf{g}}_1)^t = W(\boldsymbol{\zeta} - A(0, \mathbf{g}_2 - \tilde{\mathbf{g}}'_1)) \quad (13)$$

which provides one equation for each point in $(\mathbf{g}_1, \tilde{\mathbf{g}}_1)$.

Equation (7) can be solved by applying a recursive linear prediction filter to extend each region. For each point $\tilde{g}_1[\mathbf{i}]$ in $\tilde{\mathbf{g}}_1$ we have

$$\tilde{g}_1[\mathbf{i}] = \sum_{\mathbf{x} \in R} p[\mathbf{x}] h[\mathbf{i} - \mathbf{x}], \quad \mathbf{h} = (\mathbf{g}_1, \tilde{\mathbf{g}}_1) \quad (14)$$

where $p[\mathbf{x}]$ are the linear predictor coefficients and R is the region of support of the linear predictor. This provides an equation for each point in $\tilde{\mathbf{g}}_1$. Considering equations (13) and (14), it is apparent that we have an overconstrained system. Since both sets of equations were developed from the single minimisation problem of equation (6) they must be consistent and we choose to discard the equations resulting from the application of the Wiener filter in the extension region. Since after applying this change to equation (13) it no longer contains $\tilde{\mathbf{g}}_1$on the left hand side, we can replace $\tilde{\mathbf{g}}'_1$on the right hand side with $\tilde{\mathbf{g}}_1$without disturbing the filtering structure of the solution. With these changes, and using $W_{\mathbf{g}_1}$ to represent the Wiener filter matrix applied only to points in $\mathbf{g}_1$, equation (13) becomes

$$\mathbf{g}_1 = W_{\mathbf{g}_1}(\boldsymbol{\zeta} - A(0, \mathbf{g}_2 - \tilde{\mathbf{g}}_1)) \quad (15)$$

3. IMPLEMENTATION

Rearranging equation (15) gives

$$\mathbf{g}_1 + W_{\mathbf{g}_1} A(0, \mathbf{g}_2 - \tilde{\mathbf{g}}_1) = W_{\mathbf{g}_1}(\boldsymbol{\zeta}) \quad (16)$$

In vector notation the single equation resulting from applying the Wiener filter at the point $\mathbf{i}$ is

$$\begin{pmatrix} 1 & \mathbf{wa}' \end{pmatrix} \begin{pmatrix} g_1[\mathbf{i}] \\ \mathbf{g}_2 - \tilde{\mathbf{g}}_1 \end{pmatrix} = w(\boldsymbol{\zeta})[\mathbf{i}] \quad (17)$$

where the row vector $\mathbf{wa}'$ holds the samples of the blur filter convolved with the Wiener filter which act on the extension region when the WA filter is applied at the point $\mathbf{i}$, and $w(\boldsymbol{\zeta})[\mathbf{i}]$ is the result of applying the Wiener filter to the original data at the point $\mathbf{i}$. Likewise, each point in the extension $\tilde{\mathbf{g}}_1$ is associated with a linear prediction equation (14), which is of the form

$$\begin{pmatrix} -1 & \mathbf{p}_{\tilde{\mathbf{g}}_1}[\mathbf{i}] & \mathbf{p}_{\mathbf{g}_1}[\mathbf{i}] \end{pmatrix} \begin{pmatrix} \tilde{g}_1[\mathbf{i}] \\ \tilde{\mathbf{g}}_1 \\ \mathbf{g}_1 \end{pmatrix} = 0 \quad (18)$$

where $\mathbf{p}_{\mathbf{g}_1}[\mathbf{i}]$ and $\mathbf{p}_{\tilde{\mathbf{g}}_1}[\mathbf{i}]$ are the linear predictor coefficents for predicting the point $\mathbf{i}$, applying to points in $\mathbf{g}_1$ and $\tilde{\mathbf{g}}_1$ respectively. Since the shape of a region is arbitrary, a small number of different linear predictor kernels is required to cope with differing local edge orientations and curvature. In our experience about 10 different kernels is satisfactory. Note that an exhaustive list of linear predictor kernels is not required as the kernel only need match a subset of the pixels available to predict from. Taking one equation for each point in the image from (17) and one equation for each point in the extension from (18) gives a well defined linear system. The actual size of the extension required can be determined by observing that, if the WA filter has a $k \times k$ support, then equation (17) will require points a distance $k/2$ into the extension. Also, note that if equation (17) is applied a distance greater than $k/2$ from any boundary, it reduces to $g_1(\mathbf{i}) = w(\boldsymbol{\zeta})[\mathbf{i}]$, the Wiener filter solution. Hence, only those extension or image points a distance $k/2$ or less from a boundary need be considered further.

The linear system thus created may be written, for two regions,

$$\underbrace{\begin{bmatrix} \mathbf{I} & W_{\mathbf{g}_1}A & -W_{\mathbf{g}_1}A & \mathbf{0} \\ W_{\mathbf{g}_2}A & \mathbf{I} & \mathbf{0} & -W_{\mathbf{g}_2}A \\ P_{\mathbf{g}_1} & \mathbf{0} & P_{\tilde{\mathbf{g}}_1} & \mathbf{0} \\ \mathbf{0} & P_{\mathbf{g}_2} & \mathbf{0} & P_{\tilde{\mathbf{g}}_2} \end{bmatrix}}_{\mathbf{M}} \underbrace{\begin{bmatrix} \mathbf{g}_1 \\ \mathbf{g}_2 \\ \tilde{\mathbf{g}}_1 \\ \tilde{\mathbf{g}}_2 \end{bmatrix}}_{\mathbf{g}} = \underbrace{\begin{bmatrix} W_{\mathbf{g}_1}(\zeta) \\ W_{\mathbf{g}_2}(\zeta) \\ \mathbf{0} \\ \mathbf{0} \end{bmatrix}}_{\mathbf{b}} \quad (19)$$

Here the matrices of linear predictor coefficients $P_{\mathbf{g}}$ and $P_{\tilde{\mathbf{g}}}$ are formed by stacking the coefficients from equation (18), with the -1 term incorporated on the diagonal of $P_{\tilde{\mathbf{g}}}$. The details of the system (19) depend on the segmentation, the Wiener filter, the blur filter, and the linear predictor. As it is impractical to generate the actual matrix system for anything except very small images, we look to iterative methods for the solution. The most straightforward and intuitive iterative method is to solve the equation associated with each point $\mathbf{i}$ (either equation (17) for points in the image or equation (18) for points in the extension) assuming that the image and extension at points other than $\mathbf{i}$ are known. For image points in $\mathbf{g}_1$, this is simply applying the WA filter to $(\mathbf{g}_2 - \tilde{\mathbf{g}}_1)$ where $\tilde{\mathbf{g}}_1$ exists, while for extension points it corresponds to generating the extension via recursive application of the linear predictor. This is also known as Gauss-Seidel iteration [7],[8]. The rate of convergence of the Gauss-Seidel method can be greatly increased by simultaneously solving blocks of equations. In our case this occurs naturally (with each block being either a region $\mathbf{g}_k$ or an extension $\tilde{\mathbf{g}}_k$) since the solution for points in a given region is independent of other points in that region, while recursive application of the linear predictor means that the only extension points used at each step are those that have already been calculated in the current iteration.

The SOR method [8] is a generalization of the Gauss-Seidel method, which introduces a parameter ω describing the amount by which the value at each point is changed. If the new value assigned to a pixel g by the Gauss-Seidel method is g^{gs}_{new}, then the value assigned by the SOR method is $g^{SOR}_{new} = g + \omega\left(g^{gs}_{new} - g\right)$. The implementation uses the same simple filtering structure as the Gauss-Seidel method and converges more quickly for appropriate ω. This is the method found to be most appropriate for our problem.

Projection methods form another major class of iterative methods which are appropriate to the problem. The simplest applicable projection method which preserves the filtering structure of the solution and is guaranteed to converge, is projection onto convex sets (POCS), in which each equation in (19) is treated as defining a hyperplane. We project the current estimate of the solution onto each of these hyperplanes in turn. To project onto the hyperplane defined by equation (17) we note that the normal direction to this hyperplane is given by $\left(1 \;\; \mathbf{wa}'\right)$. Given current values $(g_{1,c}[\mathbf{i}], \mathbf{g}_{2,c}, \tilde{\mathbf{g}}_{1,c})$ of the solution, the values after projection are given by

$$\begin{pmatrix} g_1[\mathbf{i}] \\ \mathbf{g}_2 - \tilde{\mathbf{g}}_1 \end{pmatrix} = \frac{(w(\zeta)[\mathbf{i}] - c)}{(1 + \|wa'\|^2)} \begin{pmatrix} 1 \\ \mathbf{wa} \end{pmatrix} + \begin{pmatrix} g_{1,c}[\mathbf{i}] \\ \mathbf{g}_{2,c} - \tilde{\mathbf{g}}'_{1,c} \end{pmatrix} \quad (20)$$

Here c is the result of computing the LHS of equation (17), and the values of $\mathbf{g}_2$ and $\tilde{\mathbf{g}}_1$ are altered equally at each point. The implementation of equation (20) requires filtering with $\mathbf{wa}'$ to obtain c and an update step to alter the value of $(g_1[\mathbf{i}], \mathbf{g}_2, \tilde{\mathbf{g}}_1)$ which requires one multiplication and one addition per sample, and thus has a similar level of complexity to the filtering operation. Likewise, the projection onto the hyperplane defined by equation (14) is

$$\begin{pmatrix} \tilde{g}_1[\mathbf{i}] \\ \tilde{\mathbf{g}}_1 \\ \mathbf{g}_1 \end{pmatrix} = \frac{-d}{(1 + \|L\|^2)} \begin{pmatrix} 1 \\ L \end{pmatrix} + \begin{pmatrix} \tilde{g}_{1,c}[\mathbf{i}] \\ \tilde{\mathbf{g}}_{1,c} \\ \mathbf{g}_{1,c} \end{pmatrix} \quad (21)$$

where L is the appropriately organized vector of linear predictor coefficients and d is the result of the direct application of the linear predictor at this position.

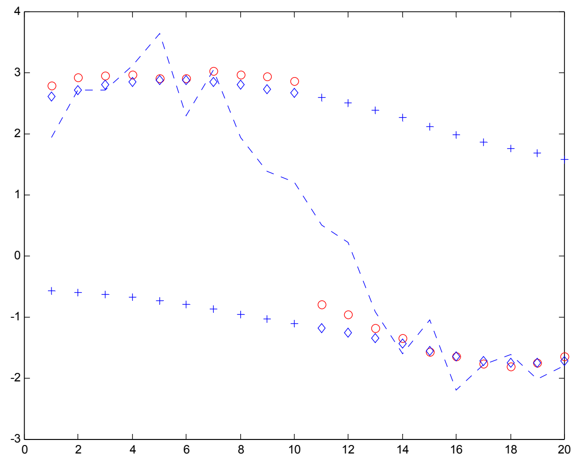

Fig. 1. One dimensional example showing the original signal (o), the recovered version ($\Diamond$), the extension to each region ($+$), and the blurred noisy version (dotted line).

While there are many other popular projection methods, such as variants of the conjugate gradients algorithm, they suffer from a variety of problems. Generally, convergence is not guaranteed unless the matrix $\mathbf{M}$ has properties such as positive definiteness which, in our case, it does not have. Each iteration is usually more complex than the SOR method, and may require multiplication by $\mathbf{M}^T$, which would be very difficult to compute. Finally, when they do converge they do so more slowly than the SOR method for this particular problem.

4. CONVERGENCE

The fastest convergence is obtained by applying the SOR method with $\omega = 1.1$, which typically gives a visually acceptable result after a single iteration, and shows no visible improvement after about 4 iterations. Unfortunately, in the 1D case the SOR method does not converge for certain choices of prior. While we have not observed divergent behaviour in the 2D case, it remains possible that this might occur, and demonstrating convergence is rendered difficult by the essentially infinite number of possible segmentations. However, this difficulty can be overcome by monitoring the behaviour of the solution at each iteration. In particular, the residual $\mathbf{r} = \mathbf{b} - \mathbf{Mg}$ provides a measure of the accuracy of the solution and is simple to calculate at each iteration. In the very rare cases where the SOR method diverges, we can fall back on the POCS method, which is guaranteed to converge, although somewhat more slowly.

5. RESULTS

Results are obtained for the one and two dimensional cases. Figure 1 shows a synthetic illustrative 1D example. For this example the first and second half of the original signal are independent realizations of a second order autoregressive process, with two poles in the z-plane at 0.9, and variance 1. The original signal is shown

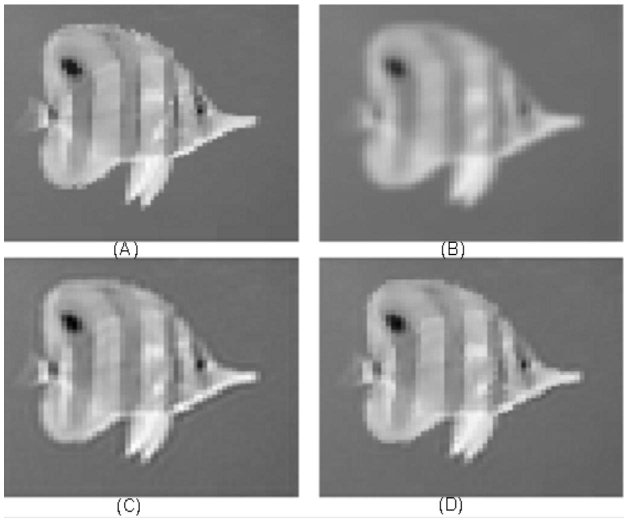

Fig. 2. (A) Original, (B) Blurred noisy image, (C) Recovered using Wiener filter, (D) Recovered using our proposed method

with the circles. A 5 sample uniform blur and noise with variance 0.1 was applied to generate the blurred, noisy signal shown with the dotted line. Our method is employed to recover the signal, using a Wiener filter with a support of 11 samples. The recovered signal at the 5th iteration of the SOR method is shown with the diamonds and the extension of each region is shown with the symbol (+). In this example we make use of our prior knowledge of the boundary position.

Figure 2 shows the method applied to an image. Figure 2A shows the original, Figure 2B shows the blurred noisy version, with a Gaussian blur (standard deviation 0.3 time the nyquist frequency) and an SNR of 25db, Figure 2C shows the image recovered with Wiener filtering, and Figure 2D shows the image recovered using our algorithm after 4 iterations of the SOR method. The method used for generating the segmentation is discussed further in the following section. The Wiener filter uses a 9x9 pixel kernel. Figure 2C clearly shows undesirable artifacts associated with the Wiener filter around the image edges. These artifacts are removed in Figure 2D.

6. DISCUSSION

Our proposed method requires segmentation of the image prior to running. While in some cases the segmentation may be known in advance, in many interesting applications it is not. As a proof of concept, we provide here a method for determining an appropriate segmentation. This task is greatly simplified by the fact that only high contrast edges need be considered, and these can be detected from the Wiener filtered image. Nonetheless, small errors in the segmentation can result in observable artifacts in the solution. In fact, we have found that for natural images it is necessary for the segmentation resolution to be at least 4x the resolution of the data ζ. The initial segmentation in Figure (2) was determined using a watershed algorithm applied to the gradient magnitude of the Wiener filtered image. The segmentation was then upsampled to 4 times the data resolution, and the segmentation boundaries smoothed with morphological operators. A small number of variations of the segmentation were generated by dilating each boundary a few pixels each way of the original segmentation. The cost function (6) was determined for the image recovered for each segmentation, and a composite segmentation generated based on the local cost. The total number of segmentations trialled was 10, with each run using just 3 iterations of the SOR method.

In general, the amount of work required to do accurate segmentation grows with the size of the blur. However, in many important problems the optical blur is quite small, covering only a few dozen pixels.

7. CONCLUSION

We have presented a computationally efficient algorithm for MAP estimation of images based on a filtering paradigm. The algorithm assumes a piecewise independent Gaussian prior, which results in artifact free recovery of high contrast edges in the image. The algorithm leverages the success of the Wiener filter solution to run in a single pass over the majority of the image and requires just a few iterations to converge in the vicinity of the high contrast edges.

While segmentation of the image is required, it has been shown that this can be done successfully based on the Wiener filtered image providing the blur and noise are not too large. We will show further color results at the conference, together with greater detail on the segmentation process.

8. REFERENCES

[1] R. Aykroyd and S. Zimeras, "Inhomogenous prior models for image reconstruction," *Journal of the American Statistical Association, Theory and Methods*, vol. 94, pp. 934–946, September 1999.

[2] S. Geman and D. Geman, "Stochastic relaxation, gibbs distributions and the baysesian restoration of images," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 6, pp. 721–741, June 1984.

[3] F. Jeng and J. W. Woods, "Compound gauss-markov random fields for image estimation," *IEEE Transactions on Signal Processing*, vol. 39, pp. 683–697, March 1991.

[4] J. Idier, "Convex half-quadratic criteria and interacting auxiliary variables for image restoration," *IEEE Transactions on Image Processing*, vol. 10, pp. 1001–1009, July 2001.

[5] D. Taubman, "Generalized wiener reconstruction of images from colour sensor data using a scale invariant prior," *ICIP*, vol. 3, pp. 801–804, September 2000.

[6] H. J. Trussel and R. E. Hartwig, "Mathematics for demosaicking," *IEEE Transactions on Image Processing*, vol. 11, pp. 485–492, April 2002.

[7] D. M. Young, *Iterative Solution of Large linear Systems*. New York, Academic Press, 1971.

[8] O. Axelsson, *Iterative Solution Methods*. Cambridge University Press, 1994.

[9] A. M. Tekalp, H. Kaufman, and J. W. Woods, "Edge-adaptive kalman filtering for image restoration with ringing suppression," *IEEE transactions on Acoustics, Speech and Signal processing*, vol. 37, pp. 892–899, June 1989.