Vector Color Filter Array Demosaicing

Maya R. Gupta and Ting Chen

Information Systems Laboratory
Department of Electrical Engineering, Stanford University, CA 94305, USA

ABSTRACT

Single-sensor digital cameras spatially sample the incoming image using a color filter array (CFA). Consequently, each pixel only contains a single color value. In order to reconstruct the original full-color image, a demosaicing step must be performed which interpolates the missing colors at each pixel. Goals in CFA demosaicing include color fidelity, spatial resolution, no false colors, no jagged edges, and computational practicality. Most demosaicing algorithms do well for color fidelity, but there is often a trade-off between a sharp image and the so-called "zipper effect" or jagged edge look. We propose a novel demosaicing algorithm called Vector Demosaicing that interpolates missing colors jointly by selecting the color vector that minimizes the sum of distances to the surrounding pixels. The selected color vector is a vector median of the surrounding pixels. The vector median forms an "average", but preserves sharp edges. We will discuss the theory behind our approach and show experimentally how the theoretical advantages manifest themselves to improve edge resolution while retaining smoothness. Computational complexity is shown to be possibly quite low, and we discuss how different approximations may affect the output.

Keywords: Color Interpolation, Color Demosaicing, Color Filter Array (CFA), Vector Median

1. INTRODUCTION

Usage of digital cameras is spreading widely as they are convenient image input devices. The increasing popularity of digital cameras has provided motivation to improve all elements of the digital photography signal chain. To lower cost, digital color cameras typically use a single image detector. Color imaging with a single detector requires the use of a Color Filter Array (CFA) which covers the detector array. In this arrangement each pixel in the detector samples the intensity of just one of the many color channels. The recovery of full-color images from a CFA-based detector requires a method of calculating values of the other color channels at each pixel. These methods are commonly referred as color interpolation or color demosaicing algorithms. Goals in color demosaicing include color fidelity, spatial resolution, no false colors or color fringe, no jagged edges, and computational practicality.

Many algorithms[1–17] have been proposed over the years, ranging from simple linear algorithms to sophisticated adaptive ones which usually require some local feature detections. Though adaptive algorithms are generally believed to outperform non-adaptive algorithms in terms of the quality of color reproduction, the performance gain often comes at the expense of tougher system computational requirements which imposes challenges in hardware implementations and prevents its use in certain applications such as real-time high speed imaging. On the other hand, simple algorithms such as the popular bilinear interpolation are very computational efficient, but they don't bring good tradeoffs between the sharpness and the so-called "zipper effect" or jagged edge look of the image. This motivates the search for simple and efficient algorithms that balance sharpness and complexity.

In this paper we propose a novel demosaicing algorithm called Vector Demosaicing that interpolates missing colors by selecting the color vector that minimizes the sum of distances to the surrounding pixels. The selected color vector is a vector median of the surrounding pixels. The vector median forms a smooth "average" but preserves sharp edges. It is thus a naturally edge-adaptive algorithm. We also show that the vector demosaicing can be implemented at low computational cost.

It is apparent that the color demosaicing algorithm is closely related to the CFA pattern. In this paper we have chosen to work exclusively with the Bayer pattern,[18] mainly due to its popularity. Figure 1 describes

Correspondence: Email: guptama@stanford.edu, tingchen@stanford.edu; Telephone: 650-496-5735; Fax: 650-723-8473

its configuration. Note that Red, Green and Blue (RGB) are the three primary colors that we will work with and the numerical indices are included to aid the algorithm description in section 2. Adapting the algorithm to work with other color filter arrays, such as Zhu and Parker's blue noise CFA's[19] , would be a straightforward extension.

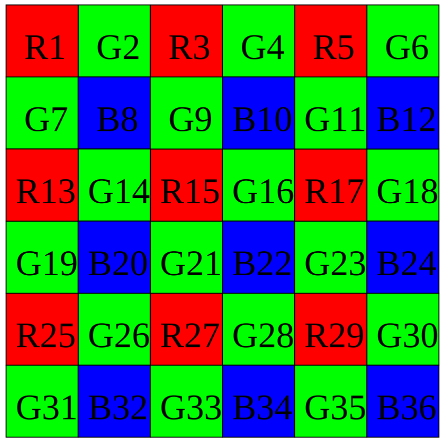

Figure 1. Bayer Pattern

The remainder of this paper is organized as follows. In section 2 we describe the vector demosaicing algorithm and explain the theories behind it. In section 3 we discuss results of comparative experiments. In section 4 we consider some possible future extensions to this vector demosaicing algorithm.

2. ALGORITHM DESCRIPTION

We would like a demosaicing algorithm to fill in the missing color values using relevant local information. Two extremes of this are to choose the color of a nearest-neighbor, or to take the mean of local values (bilinear interpolation). Nearest neighbor algorithms lead to jagged edges; smoothing algorithms, like bilinear interpolation, lead to blurred images.

We propose an algorithm with two paradigm shifts from previous work. We propose solving for the colors at a pixel jointly - that is solving for a vector instead of finding the colors pixel plane by pixel plane. Secondly, we may choose to replace the original color value known at a pixel with a value that matches better with the other color dimensions. This may sound counter-intuitive, but imagine for example that there is a sharp edge and the pixel in question is on the boundary. Then the estimate of the other two colors may be for the near side of the edge whereas the original color estimate is for the far side of the edge. Then the estimated pixel will contain color planes from different sides of the edge and a false color will result. However, we estimate the whole pixel at once which may result in throwing away the original color value. This means an edge may artificially be moved one pixel, but false colors are less likely to occur.

We begin by forming a notion of pseudo-pixels, such as are found on CRT monitors, where we group neighboring Red, Green, and Blue values together and refer to them as a pseudo-pixel. We will use a column vector to represent such a pseudo-pixel. For example, in Figure 1, one pseudo-pixel would be $\begin{bmatrix} R15 & G16 & B22 \end{bmatrix}^T$. Another pseudo-pixel would be $\begin{bmatrix} R15 & G16 & B10 \end{bmatrix}^T$.

Since we don't know which pseudo-pixels (if any) correspond to the actual edge information in the image, we can form an unbiased estimate by replacing an unknown color value with the average of its surrounding pseudo-pixel values. However, merely taking an arithmetic average of the psuedo-pixels is equivalent to bilinear interpolation and leads to smearing. Instead, in order to preserve edges, we take the vector median of the pseudo pixels to be the missing color vector.

For instance, to estimate the color values at location 21, we can find the vector median of the eight neighboring psuedo-pixels, as shown in equation (1), with each column representing one psuedo-pixel.

$$\begin{bmatrix} R21 \\ G21 \\ B21 \end{bmatrix} = \text{Vector Median} \left\{ \begin{bmatrix} R15 & R15 & R15 & R15 & R27 & R27 & R27 & R27 \\ G21 & G21 & G16 & G14 & G21 & G21 & G28 & G26 \\ B22 & B20 & B22 & B20 & B22 & B20 & B22 & B20 \end{bmatrix} \right\} \tag{1}$$

To estimate pixels where we know the red or blue color value, we use 16 surrounding pseudo-pixels, for pixels where we know the green value, we use 8 surrounding pseudo-pixels.

A vector median is that vector A which minimizes the sum of L^2 distances (Euclidean distances) to each point in the set. Let $\{x_1, x_2, ...x_N\}$ be a set of $K - dimensional$ vectors, then

$$\text{vector median of } \{x_1, x_2, ...x_N\} \text{ is the vector } A \text{ that solves } \arg\min_A \sum_{i=1}^{N} \left(\sum_{k=1}^{K} (A_k - x_{ik})^2 \right)^{1/2} \tag{2}$$

where A_k and x_{ik} are the k-th element of vector A and x_i, respectively. The vector median is easy to visualize. Imagine that you have a set of points, (as an example, on the plane), then pick a point and meaure off string from your new point to each point in the set. How much string is needed? Pick another point and try it again. The vector median is the point that minimizes the total length of needed string.

The vector median as found from Equation (2) reduces to the median in the scalar case. It is not separable in the vector components. The vector median is insensitive (though not blind) to outliers. Hence, it is a well-suited average for certain types of signal processing, including those tasks where false smoothing of abrupt edges is undesirable. Astola et. al promoted the use of this average for color image processing in their 1990 work.[20]

To contrast the vector median with the vector mean, the vector mean is the vector that minimizes the sum of squared L^2 distances to each point in the set. Figure 2 illustrates the difference between vector mean and vector median via two simple examples. We calculated the vector mean and vector median for eight two-dimensional vectors. When the set of vectors indicates a smooth spatial distribution, the vector mean and vector median differ little. When an edge is present, however, significant difference occurs. Although the vector mean is easily found by adding up the points and dividing by the size of the set, it can also be represented as the vector B that solves

$$\arg\min_B \sum_{i=1}^{N} \sum_{k=1}^{K} (B_k - x_{ik})^2 \tag{3}$$

Note that the vector mean is separable into the vector components, thus unlike the vector median, there is no difference in considering it plane-by-plane or jointly.

Although the idea of an average of a set being that point which minimizes the sum of distances to all points in the set seems intuitive, there is no closed form solution to Equation (2). The vector median can be found by an iterative method and can be approximated in a variety of ways. We consider an "exact" vector median to be the one that is accurate to within some ϵ, where ϵ is smaller than the quantization error of the imaging system. Statisticians have considered the problem of finding the vector median, or as they call it, the Fermat-Weber point, for many years. The theoretically most efficient method is Vardi and Zhang's algorithm.[21] The Vardi and Zhang algorithm is monotonic and guaranteed to converge. Numerical evidence suggests that it possesses near linear convergence.

An obvious way to approximate the vector median is to execute fewer iterations by setting ϵ bigger. ϵ becomes a knob that you can tune to achieve the accuracy-computation trade-off point desired. Color filter researchers Plataniotis and Venetsanopoulos have suggested the approximation[22] in which only the points in the data set are candidates for the vector median. Another way to approximate the algorithm is to use fewer of the surrounding pseudo-pixels in Equation (1), perhaps only those within the 3x3 box surrounding the pixel being estimated.

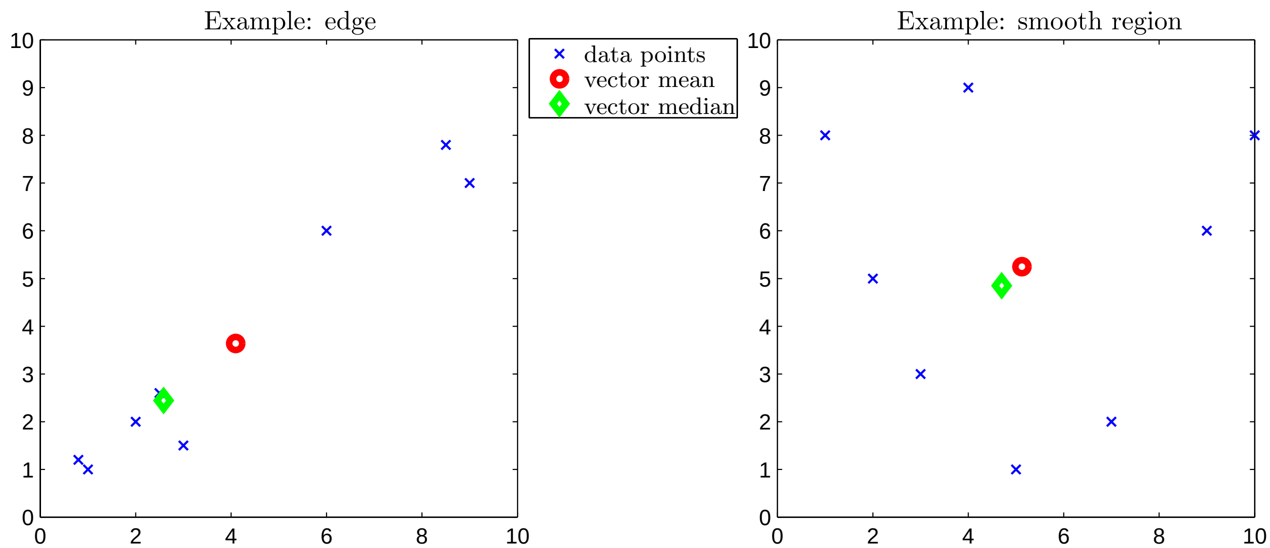

Figure 2. Vector Mean versus Vector Median

3. EXPERIMENTAL RESULTS

We compared the algorithm on a number of natural images to the results obtained using two other algorithms, bilinear interpolation and scalar median demosaicing. We did all of our processing in RGB color space with a Bayer CFA. Image examples are shown in Figure 3,4,5 and 6. Each image is shown in full scale first, then a selected region is zoomed in for close examinations.

Edges:

We found that our algorithm naturally adapted to edges and smooth regions. Edges were almost as sharp as in the original image with little zipper effect. In smooth regions the algorithm reconstructs the soft undulations of the original images as smoothly as bilinear interpolation.

False Colors:

Most edges did not result in false colors, unlike bilinear interpolation. For example, in the flowers image, the edges have much worse false color effects than with the vector median. However, in images with a lot of edge transitions close together, our observation reveals that false colors are still a significant problem.

Scalar Median:

To show that the vector median leads to different results than applying a scalar median, we compare to scalar median interpolation. Our scalar median interpolation is a plane-by-plane algorithm that takes the scalar median of the same color values in a 3x3 block centered on the pixel to be estimated. Given only two values to interpolate Red and Blue, the scalar median is not unique. As is commonly done with scalar medians, in those cases we used the mean of the two values.

Judging from these figures, it is clear that the scalar median interpolation leads to quite different results than the vector median. The images appear more blurred and there are more false color and zipper problems at edges. We believe that this combination of bad effects is due to the plane-by-plane nature of the algorithm optimizing differently at each plane, thus resulting in a locally jumbled reconstruction.

Distance Metric:

In the algorithm presented the vector median minimizes the sum of L^2 distances. The vector median we presented is strictly the Fermat-Weber vector median, there are other less common ways to define a vector median. Generalizing Equation (2), the vector A that minimizes the sum of L^p distances is considered the L^p vector median. We experimented with values of p different than 2. For values close to 2 there is very little difference in the output. However for very large values of p, such as the L^∞ vector median, the resultant image looks as though it was painted with a very thick brush, and is not suitable for anything but artistic use. We have not investigated the matter, but solving for the L^1 vector median may be easier than solving for the L^2 vector median, with little difference in resultant image quality.

Approximations:

We found all the approximations detailed in section 2 to result in images that were still substantially cripser than bilinear interpolation, but which were not as clean looking as the exact vector median images. False color problems increase with every approximation. For the approximation proposed by Plataniotis et. al, the resulting performance in this application was sharper than bilinear but looked significantly worse than the exact vector median estimation. Run-time was only roughly half. On the other hand, increasing ϵ appears to bring better tradeoff between complexity and performance. Figure 7 and 8 show an example where ϵ in the approximated vector median method has been increased by a factor of 25 in comparison to the exact solution. Notice both images look much sharper than bilinear interpolation, and the approximated case is only slightly worse than the exact. Such an ϵ reduces the computation time by 70%.

4. CONCLUSION

Vector median color filter demosaicing naturally adapts to edges and smooth regions to provide excellent sharpness without jaggedness. However, false colors may still be a significant problem with this algorithm, and we are investigating how to ameliorate this issue, possibly with a post-processing filter. We are also interested in how results may change when processed in other color spaces.

ACKNOWLEDGEMENTS

The work reported in this paper was started as a class project in EE392B at Stanford University in spring of 2000, therefore the authors would like to acknowledge Prof. Abbas El Gamal for the great opportunity and helpful discussions. We would also like to thank Anna Gilbert for sharing her implementation of Vardi and Zhang's algorithm.

REFERENCES

1. N. Ozawa, "Chrominance signal interpolation device for a color camera," *U.S. Patent 4,716,455* , 1987.
2. D. R. Cok, "Single-chip electronic color camera with color-dependent birefringent optical spatial frequency filter and red and blue signal interpolating circuit," *U.S. Patent 4,605,956* , 1986.
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* , 1987.
4. R. G. Keys and et al., "Cubic Convolution Interpolation for Digital Image Processing," *IEEE Transactions on Acoustic, Speech and Signal Processing* **ASSP-29**, pp. 1153–1160, 1981.
5. R. H. Hibbard, "Apparatus and method for adaptively interpolating a full color image utilizing luminance gradients," *U.S. Patent 5,382,976* , 1995.
6. C. A. Laroche, "Apparatus and method for adaptively interpolating a full color image utilizing chrominance gradients," *U.S. Patent 5,373,322* , 1994.
7. J. F. Hamilton, Jr. and J. E. Adams, Jr., "Adaptive color plane interpolation in single sensor color electronic camera," *U.S. Patent 5,629,734* , 1997.
8. J. E. Adams, Jr. and J. F. Hamilton, Jr., "Adaptive color plane interpolation in single sensor color electronic camera," *U.S. Patent 5,506,619* , 1996.
9. D. R. Cok, "Signal processing method and apparatus for sampled image signals," *U.S. Patent 4,630,307* , 1986.
10. X. Wu and et al., "Color Restoration from Digital Camera Data by Pattern Matching," in *Proceedings of SPIE*, vol. 3018, pp. 12–17, (San Jose, CA), February 1997.
11. E. Chang and et al., "Color Filter Array Recovery Using a Threshold-based Variable Number of Gradients," in *Proceedings of SPIE*, vol. 3650, pp. 36–43, (San Jose, CA), January 1999.

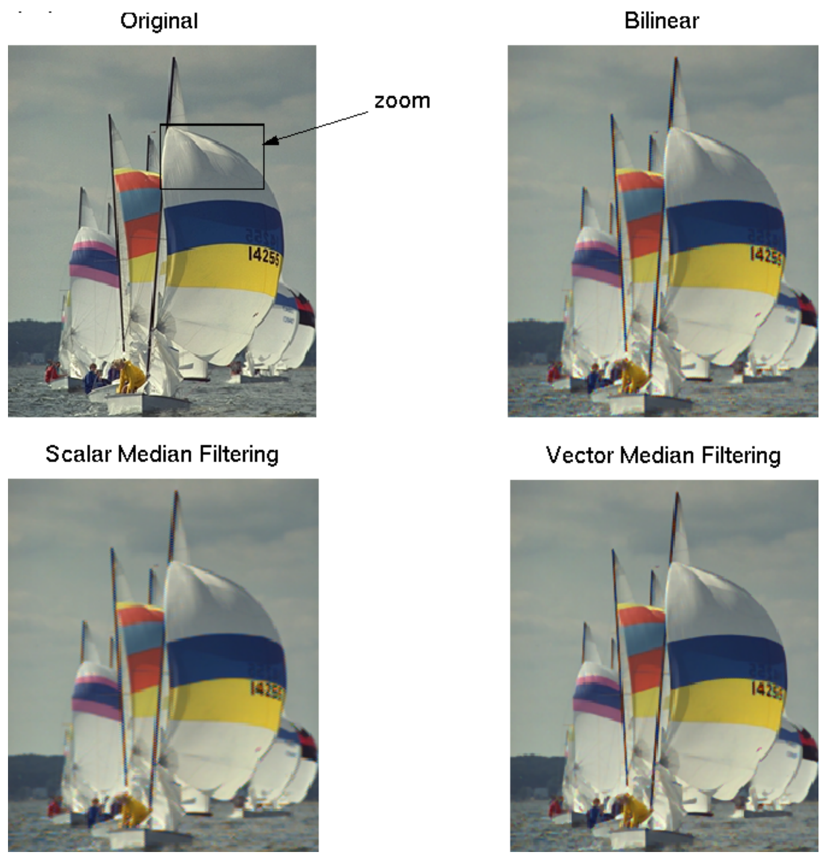

Figure 3. Example I

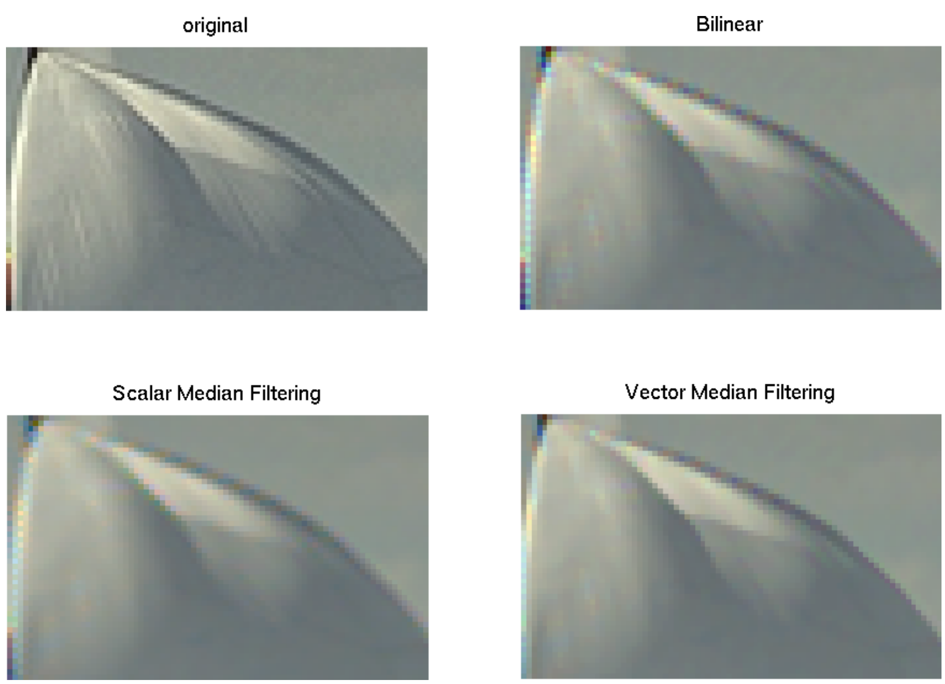

Figure 4. Example I : zoomed view

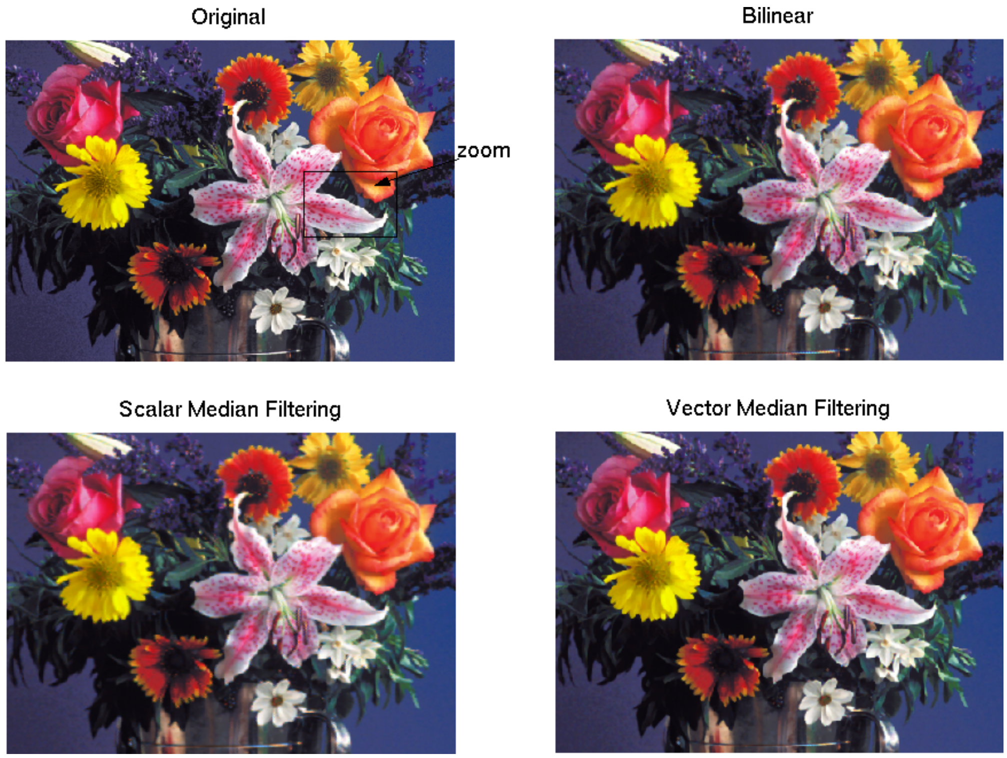

Figure 5. Example II

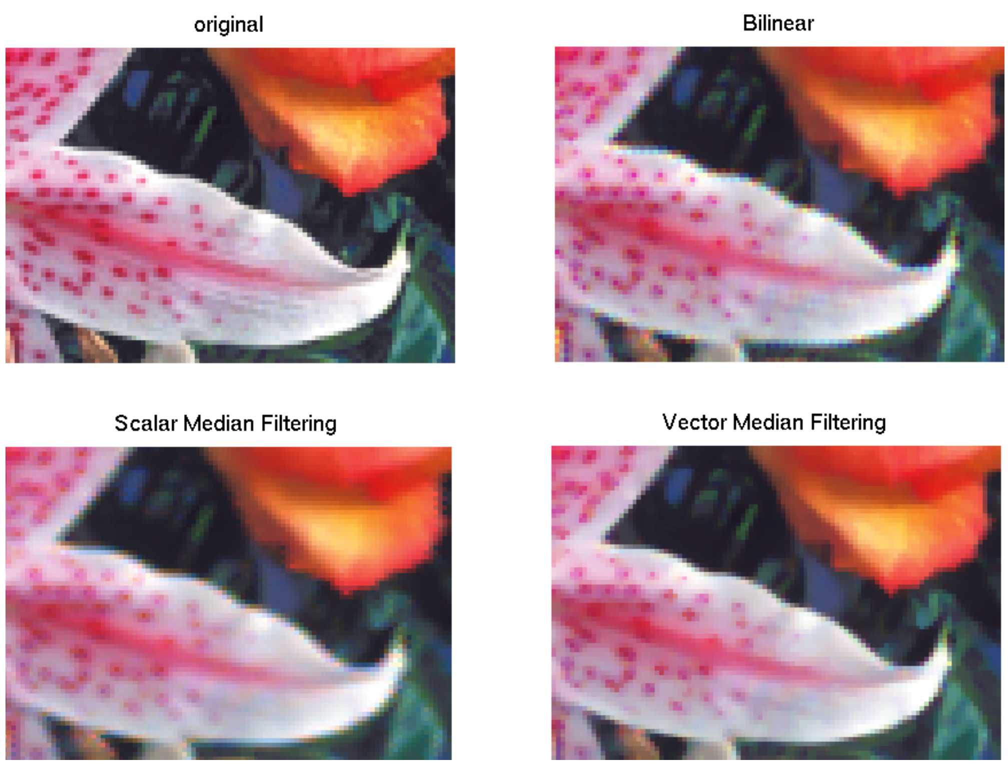

Figure 6. Example II : zoomed view

Figure 7. Example III

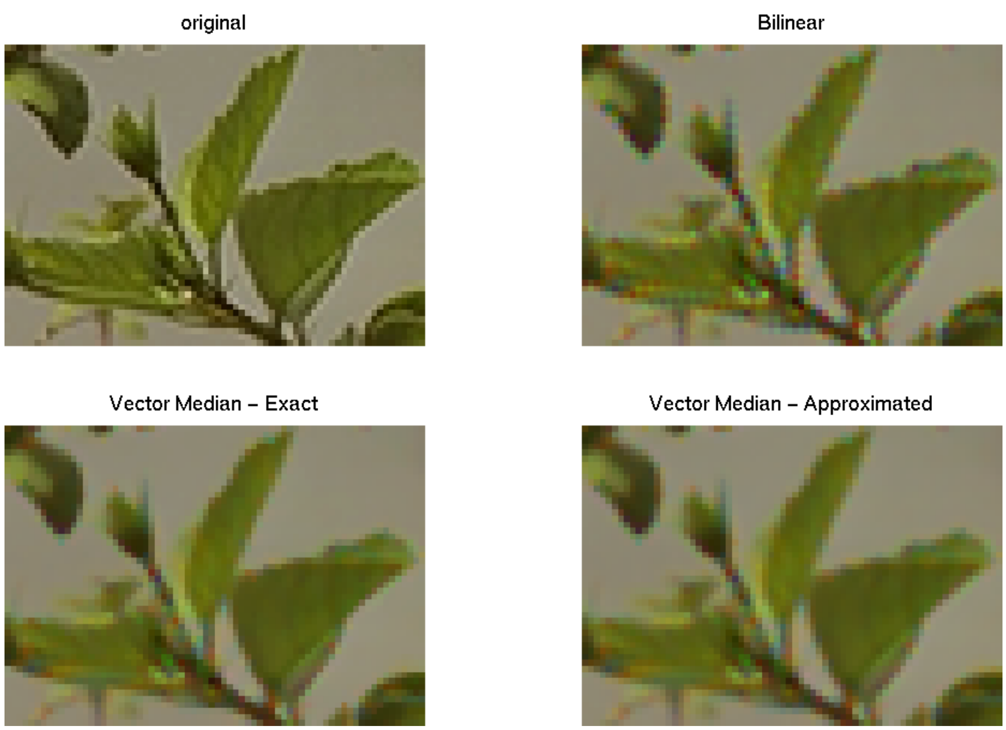

Figure 8. Example III : zoomed view

12. H. S. Hou and et al., "Cubic Splines for Image Interpolation and Digital Filtering," *IEEE Transactions on Acoustic, Speech and Signal Processing* **ASSP-26**, pp. 508–517, 1987.
13. W. T. Freeman, "Method and apparatus for reconstructing missing color samples," *U.S. Patent 4,663,655* , 1987.
14. M. A. Wober and et al., "Method and apparatus for recovering image data through the use of a color test pattern," *U.S. Patent 5,475,769* , 1995.
15. T. A. Matraszek and et al., "Gradient based method for providing values for unknown pixels in a digital image," *U.S. Patent 5,875,040* , 1999.
16. D. H. Brainard and et al., "Bayesian method for Reconstructing Color Images from Trichromatic Samples," in *IS&T 47th Annual Conference*, pp. 375–380, 1994.
17. R. Kimmel, "Demosaicing: Image reconstruction from color CCD samples," *IEEE Transaction on Image Processing* **8**(9), pp. 1221–1228, 1999.
18. B. E. Bayer, "Color imaging array," *U.S. Patent 3,971,065* , 1976.
19. W. Zhu and K. Parker, "Color filter arrays based on mutually exclusive blue noise patterns," *Journal of Visual Communication and Image Rpresentation* **10**, pp. 245–267, 1999.
20. T. Astola, P. Haavisto, and Y. Neuvo, "Vector Median Filters," *Proceedings of IEEE* , pp. 678–689, 1990.
21. Y. Vardi and C. Zhang, "The multivariate l1-median and associated data depth," *Proc. Nat. Acad. Sci. USA* **97**(4), pp. 1423–1436.
22. Sangwine and Horne, *The Color Image Processing Handbook*, Chapman and Hall, 1993.