Projective Demosaicing using Multiple Overlapping Images

James Fung and Steve Mann *
{*fungja,mann*}*@eecg.toronto.edu*
University of Toronto, EyeTap Personal Imaging Lab
Dept. of Elec. and Computer Engineering

Abstract

This paper presents a demosaicing approach which combines the Bayer patterns of multiple overlapping images. Multiple images of the same subject are taken, where the camera is free to pan, tilt, and rotate around its optical axis. The images are spatially registered and a Bayer pattern mosaic is created by combining each image's Bayer pattern. In the region of overlap, each additional image "fills in" the gaps in the Bayer pattern for each color channel, creating a completely filled Bayer mosaic. The presented method is implemented in graphics hardware, which provides hardware acceleration. Results are shown in which increased color channel resolution is achieved in the overlapping regions of multiple images.

Figure 1. A VideoOrbit. Two images have been spatially registered using the VideoOrbits algorithm.

1. Introduction

Digital cameras are able to provide Bayer patterns as output. The Bayer pattern is an array of red, green, or blue output where a single colour is available at each location. Since only one color component is available at a given location, the other color components must be interpolated.

The art of combining multiple pictures of the same subject matter to obtain higher definition composites has been previously explored. In a 1993 paper, multiple differently exposed pictures, in which the camera moved to include different overlapping subject matter, and in which the exposure of the camera changed, were combined together to get images of greater spatial and tonal resolution [6]. The notion of "being undigital" with digital cameras was further explored in a sub-

sequent 1995 paper [8], and [9]. However, no previous work exists in which multiple exposures are registered to fill in the gaps between their color lattice (e.g. Bayer) patterns.

Different methods [5, 3, 2, 11] have been proposed for constructing colour images from the Bayer patterns. This process is typically called "demosaicing". Common methods typically examine color information in a local region of the Bayer pattern, and use a variety of methods to interpolate output RGB color values at each location. Bilinear interpolation, for instance, is a simple method which can be applied to obtain color channel values at each array location. Other methods include detecting color edges.

This paper presents a method which instead uses the Bayer patterns of multiple images to "fill in" the gaps in the Bayer pattern. The method presented here spatially registers the Bayer patterns of multiple images. Each additional image provides more color channel samples at any given location. Thus, the gaps in the Bayer pattern of a single image can be filled with the Bayer pattern of another, overlapping image. In

G	R	G	R	G
B	G	B	G	B
G	R	G	R	G
B	G	B	G	B
G	R	G	R	G

Figure 2. A typical Bayer pattern. R,G,B denote the colour channels (red (R), green (G) and blue (B)) available at each location.

the sense that a "mosaic" may be generally considered as an aggregation of small inlaid elements, the combination of additional Bayer samples creates a final "mosaic" of Bayer color information. This composite mosaic then provides a dense set of color channel samples which can then be used to reconstruct a final output image.

2. Projective Demosaicing

The approach presented here uses the Bayer patterns of multiple, overlapping images to fill in the gaps in the colour channels. In order to achieve this, the images must be registered. We apply a projective image registration algorithm called VideoOrbits [7].

VideoOrbits[1] is an image registration algorithm which calculates a projective coordinate transformation between pairs of images of a static scene, taken with a camera that is free to pan, tilt, rotate about its optical axis, and zoom. The technique solves the problem for two cases: 1. images taken from the same location of an arbitrary 3-D scene, or 2. images taken from arbitrary locations of a flat scene [7].

The projective coordinate transformation is given by:

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \frac{\begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix}\begin{bmatrix} x \\ y \end{bmatrix} + \begin{bmatrix} b_1 \\ b_2 \end{bmatrix}}{\begin{bmatrix} c_1 & c_2 \end{bmatrix}\begin{bmatrix} x \\ y \end{bmatrix} + 1} \quad (1)$$

where a_{11}, a_{12}, b_1, a_{21}, a_{22}, b_2, c_1, c_2 are the 8 parameters describing the projective transformation, solved for by VideoOrbits, x, y are the original image coordinates, and x', y' are the projected coordinates. Denoting the operation of projection as P,

then the above is represented compactly as $[x', y']^T = P([x, y]^T)$. Figure 1 shows two images registered with VideoOrbits, where the lower left image has the identity projection, and the upper right image has been projected to spatially register it with the lower left.

Our method is as follows:

1. Given a set of overlapping Bayer outputs, perform standard interpolation on each to create a set of images, I_i.

2. Use VideoOrbits to determine the projective coordinate transformations, P_i, which register each image, I_i, with respect to a reference coordinate system. Typically, one image, I_0 is chosen to have the identity transformation, and the other images are registered with I_0.

3. Apply each P_i to each of the Bayer patterns to register each with respect to the same coordinate system. This combines the multiple Bayer patterns to form a final mosaic which has a denser set of Bayer samples in the region of overlap.
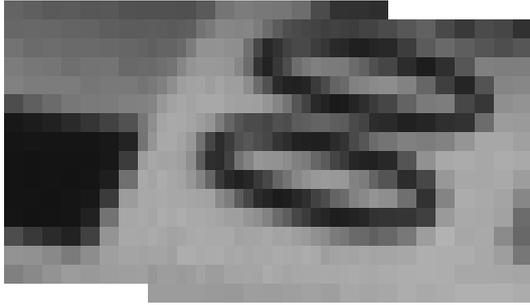
After a set of projective transformations P_i has been determined (as, for instance, in figure 1) a fully filled, composite Bayer mosaic is created examining local contributions of all of the input Bayer images. For any given blank array value, rather than interpolating between Bayer channels of a single image, we instead search all of the projected, overlapping input images for the Bayer value that lies closest to the current array position. For simplicity, we begin by taking the nearest neighbouring Bayer value of any of the projected input images.

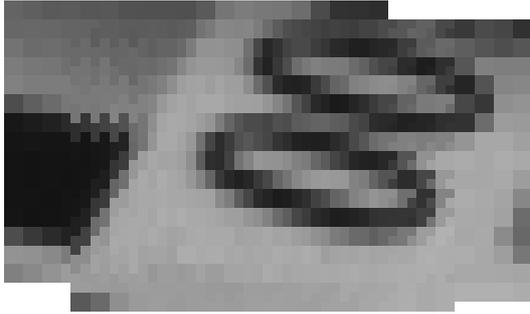### 2.1. Graphics Processing Unit (GPU) hardware implementation

The proposed algorithm has been implemented on commodity desktop graphics hardware. Modern graphics hardware now incorporates a Graphics Processing Unit (GPU) which is designed to perform fast mathematic operations. Additionally, the GPU is now highly programmable [4]. The VideoOrbits algorithm, used in the image registration step, has been implemented completely in graphics hardware, and has been shown to run faster on the GPU than the CPU[1][2].

A fragment shader program was written in Cg [10] to implement the Bayer mosaicing on the graphics hardware. In the graphics pipeline, a fragment may be considered as an output pixel with depth and color information. A fragment shader program is essentially a

---

[1]VideoOrbits is a set of Free/Open Source programs, and may be downloaded from http://comparametric.sourceforge.net

[2]Parallel GPU implementations may be obtained through the OpenVIDIA project at http://openvidia.sourceforge.net
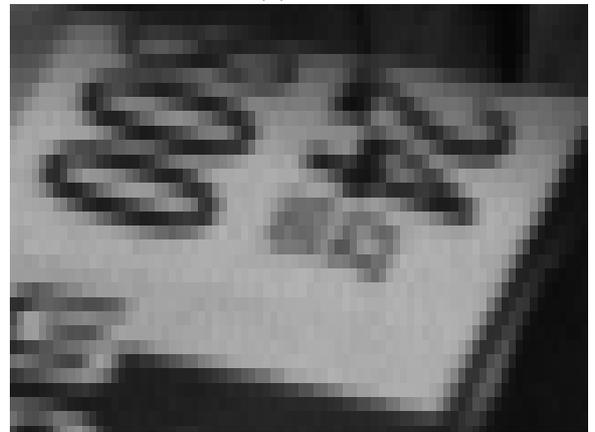
(a)

(b)

Figure 3. Overlapping Bayer patterns are registered. In the area of overlap, there is higher frequency of samples present, adding detail. (a) shows two overlapping Bayer patterns, and (b) shows three overlapping Bayer patterns

(a)

(b)

(c)

Figure 4. Comparison of adding additional Bayer patterns to created a composite mosaic. (a) shows a single Bayer pattern, interpolated by taking the nearest neighbour. (b) shows two Bayer patterns registered, and combined by taking using the nearest available Bayer value from either Bayer pattern. (c) shows 3 Bayer patterns registered and combined.

small, user–defined program which runs on each fragment produced by the graphics card. Our implementation takes care to ensure that each output fragment corresponds to exactly one pixel of the input image with the identity transformation. This is achieved by texture mapping the input image I_0 onto a quadrilateral, and displaying at the appropriate resolution. For our purposes, we employ a floating point buffer extension, which renders the output in an offscreen buffer (stored in video RAM), and uses full 32-bit IEEE floating point precision. The projective transformations P_i are input to the fragment shader program. The additional images $I_1, I_2, ...$ are stored in additional texturing units which the fragment shader can access.
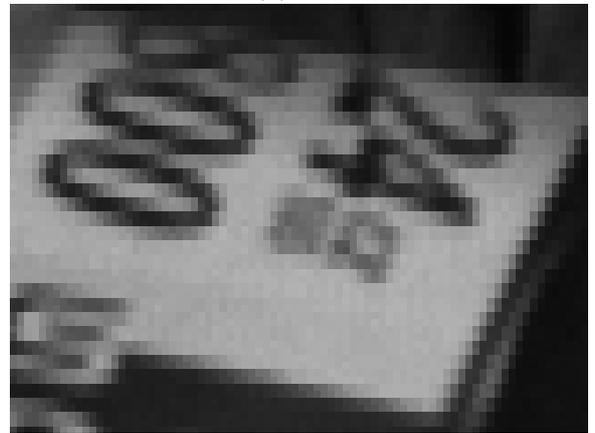
For each pixel, the fragment shader is given the appropriate texture coordinates $[x_0, y_0]^T$ of I_0 which are to be displayed. This corresponds in our case, to the image coordinates of I_0. The fragment shader then determines the nearest Bayer values for the other images as $[x_1', y_1']^T = P_1([x_0, y_0]^T)$. The distances of the current position to each of the nearest Bayer values from all overlapping images is compared, and the nearest
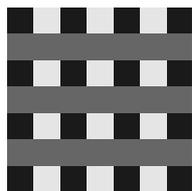
Figure 5. Mosaic contributions. This figure shows the contributions of three images to the final composite mosaic. Here, each of the three distinct grey levels represents which of the input image Bayer values is used in the composited Bayer mosaic.

neighbour is thus chosen. The output of the fragment shader program is the Bayer value of this nearest neighbour, and this is stored in the floating point buffer, and read out to main memory to be saved as an image. Additionally, the final image is rendered to the screen allows for real–time zoom and pan exploration of the images (of resolution 2482×1648).

3. Results

Figure 3 shows the composite Bayer mosaics created from small, rectangular Bayer images which have been registered. The individual Bayer images overlap in the centre. It can be seen that in the area of overlap, greater resolution is obtained. The registration has allowed for the gradient of the lettering to be correctly filled in by the additional images.

Figure 4 shows a larger area with text present in the image. It can be seen that as the number of overlapping images increases, more resolution is obtained.

Figure 5 shows which of three Bayer patterns contributions at each location of a composite mosaic.

4. Conclusion

A demosaicing approach was presented. The approach combined the Bayer patterns of multiple overlapping images. The approach was demonstrated by using multiple images of the same subject matter taken by a camera, where the camera was free to pan, tilt, and rotate around its optical axis. The images are spatially registered using the VideoOrbits algorithm, and a composited Bayer pattern mosaic was created by combining each image's Bayer pattern. In the region of overlap, each additional image "filled in" the gaps in the original Bayer pattern for each color channel, creating a completely filled Bayer mosaic. The presented method was implemented in graphics hardware, which provided hardware acceleration. The results shown demonstrate that the method can achieve increased color channel resolution in the overlapping regions of multiple images.

References

[1] J. Fung and S. Mann. Computer vision signal processing on graphics processing units. In *To appear in the Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP 2004)*, pages V93–V96, Montreal, Quebec, Canada, May 17–21 2004.

[2] J. J. E. Adams. Design of practical color filter array interpolation algorithms for digital cameras. *Proc. SPIE*, 3028:117–125, Feb. 1997.

[3] R. Kimmel. Demosaicing: image reconstruction from color ccd samples. *IEEE Trans. on Image Processing*, 8:1221–1228, Sept. 1999.

[4] E. Lindholm, M. J. Kilgard, and H. Moreton. A user–programmable vertex engine. In *Computer Graphics, Proc. of SIGGRAPH 2001*, pages 149–158, 2001.

[5] H. S. Malvar, L. wei He, and R. Cutler. High-quality linear interpolation for demosaicing of bayer-patterened color images. In *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP 2004)*, Montreal, Quebec, Canada, May 17–21 2004.

[6] S. Mann. Compositing multiple pictures of the same scene. In *Proceedings of the 46th Annual IS&T Conference*, pages 50–52, Cambridge, Massachusetts, May 9-14 1993. The Society of Imaging Science and Technology. ISBN: 0-89208-171-6.

[7] S. Mann. Wearable computing: Toward humanistic intelligence. *IEEE Intelligent Systems*, 16(3), May/June 2001.

[8] S. Mann and R. Picard. Being 'undigital' with digital cameras: Extending dynamic range by combining differently exposed pictures. In *Proc. IS&T's 48th annual conference*, pages 422–428, Washington, D.C., May 7–11 1995. Also appears, M.I.T. M.L. T.R. 323, 1994, http://wearcam.org/ist95.htm.

[9] S. Mann and R. W. Picard. 'virtual bellows': Assembling video into high quality still images. TR 259, M.I.T. Media Lab Perceptual Computing Section, Cambridge, Massachusetts, Jan 1994.

[10] W. Mark, R. Glanville, K. Akeley, and M. Kilgard. Cg: A system for programming graphics hardware in a c–like language. In *Proceedings of ACM SIGGRAPH. ACM Press, 2003*, volume 22, July 2003.

[11] K. Plantaiotis and R. Lukac. An efficient demosaicing approach with a global control of correction steps. In *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP 2004)*, pages III–469 – III–472, Montreal, Quebec, Canada, May 17–21 2004.