

Subpixel Rendering of Bayer-Patterned Images

Bradley Atcheson

Department of Computer Science
The University of British Columbia

Abstract

A common workflow in digital photography involves viewing high resolution images on LCD displays - either computer monitors or camera viewfinders. To minimise cost and complexity, most cameras use a single sensor overlaid with a Bayer filter to selectively sample various wavelengths at interleaved locations. Missing data must then be interpolated, which is known to introduce artefacts. In order to show the image on a display with a lower resolution, it must then be downsampled, which introduces more artefacts. This process usually does not take into account the fact that LCD displays are composed of horizontally displaced subpixels, which can be individually addressed to achieve a higher effective horizontal resolution. A new downsampling algorithm is proposed, which works directly within the Bayer domain and exploits subpixel resolution to produce images of higher detail than the naïve demosaicking and downsampling combination.

1 Introduction

Currently available entry-level digital cameras are capable of multi-megapixel resolution, while display resolutions have remained relatively static in the recent past. In particular, camera viewfinders and handheld devices have displays of vastly lower resolution, yet we would still like to show as much detail as possible on them. Subpixel rendering allows us to do this by treating the display not as an array of full-colour pixels, but rather as an array of horizontally displaced monochromatic subpixels, triads of which form logical pixels¹. Figures 1 (a) and (b) show the layout of a typical LCD display. Because each subpixel is individually addressable, if we ignore colour we are theoretically capable of achieving a threefold increase in horizontal luminance resolution [Daly 2001].

This fact has been exploited since LCD displays first became available [Gibson 2003], however it has recently gained much popularity through Microsoft's ClearType™ font rendering technology [Platt 2000]. It is not possible to realise the full 3x performance improvement in practice since chroma aliasing is introduced. Fortunately the human

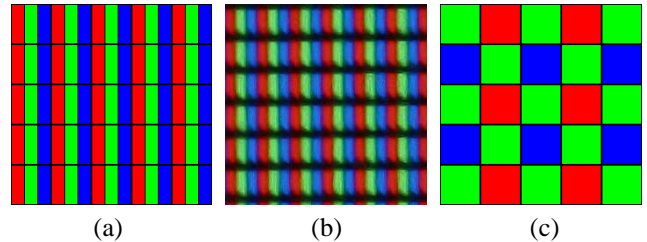


Figure 1: (a) LCD subpixel layout; (b) Close-up photograph of LCD panel; (c) Bayer CFA.

visual system (HVS) is less sensitive to chroma than it is to luma, so for a given viewing distance there is a certain amount of invisible chroma aliasing that can be taken advantage of to sharpen the image [Daly 2001].

ClearType renders high resolution glyphs on a lower resolution display by applying a near-optimal filter designed to remove frequencies that would lead to visible aliasing [Betrissey et al. 2000]. The same idea can be applied to general imagery, if we know the tristimulus values at each sample point. For most digital cameras this is not the case since a colour filter array (CFA) is placed over the sensor, allowing each element to record only a single primary colour. Figure 1 (c) shows the most popular CFA in use today - the Bayer pattern [Bayer 1976]. A *demosaicking* algorithm must be applied to recover the missing data from neighbouring pixels.

Since demosaicking generally produces artefacts, it may be advantageous to perform image processing operations directly in the Bayer domain when possible. Lukac et al. show that this approach produces good results and is computationally efficient when applied to colour image zooming [Lukac et al. 2005]. A demosaicking algorithm specifically designed to take advantage of subpixel rendering while downsampling is presented here, which produces images of higher quality than existing demosaicking algorithms when viewed on an LCD display. Since subpixel rendering is only effective in this situation, the algorithm is not suitable for use with other types of displays (including printed media) or when further image processing is to be applied.

The following section reviews demosaicking, some existing algorithms for performing it, and shows the problems that can occur with it in practice. Section 3 reviews subpixel rendering, showing how it can be used to achieve higher effective horizontal resolution without incurring vis-

¹To avoid confusion, the term “subpixel” will always be used to refer to the individual R,G,B elements.

ible chroma artefacts. Section 4 then links these two ideas and presents the main contribution of this paper. Results and an evaluation of the algorithm are presented in section 5, followed by the conclusion in section 6.

2 Demosaicking

In order to minimise their cost and complexity, most digital cameras use a single CCD or CMOS sensor overlaid with a colour filter array (CFA) to selectively sample various wavelengths at interleaved locations. The Bayer pattern is the most popular CFA in use, and allows for sampling green data at twice the resolution of red and blue, in accordance with the spectral sensitivity of the HVS [Bayer 1976].

Demosaicking is the process of recovering the missing colour data by interpolating between nearby samples. There are many algorithms available for doing this, and they can be divided into two categories: adaptive and non-adaptive [Chen 1999]. Non-adaptive algorithms are conceptually simpler, as they apply the same operation at each point. Nearest neighbour and bilinear interpolation are two such algorithms, which, although efficiently implementable, tend to produce the highly noticeable chroma aliasing visible in Figures 2 (b) and (c). These simple algorithms interpolate each colour channel independently and thus cause artefacts when edges are located between samples of a particular colour. Adaptive algorithms on the other hand, intelligently select the samples to interpolate between based on the image contents at that location. The edge-sensing algorithm illustrated in Figure 2 (d) compares the absolute difference between vertical and horizontal neighbours of each green pixel to be interpolated and uses the average of the pair with the smaller difference, under the assumption that an edge lies between the other pair. Blue and red pixels are interpolated to maintain a smooth transition in hues R/G and B/G [Adams 1995]. The errors produced by the adaptive algorithm are less visible but the computational cost is much greater [Chen 1999].

Dubois [Dubois 2005] uses a frequency domain analysis to elegantly show how the chromatic aliasing comes about. Aliasing is an unavoidable consequence of the Bayer-pattern sampling, and we cannot hope to eliminate it when viewing images at a 1:1 zoom level between captured samples and displayed (logical) pixels. However, if we downsample the image to fit it onto a lower resolution display, and employ subpixel rendering to raise the output resolution, then the effective sampling frequency may be high enough to avoid noticeable chroma aliasing.

When downsampling by large factors, it becomes unnecessary to even demosaick at all, since we have more than enough samples corresponding to each subpixel. Interpolated values are likely to contain some error which can be eliminated by skipping explicit demosaicking. In section 5 it is shown that the algorithm presented here can produce better quality results, at lower cost, than edge-detection demosaicking combined with bicubic downsampling.

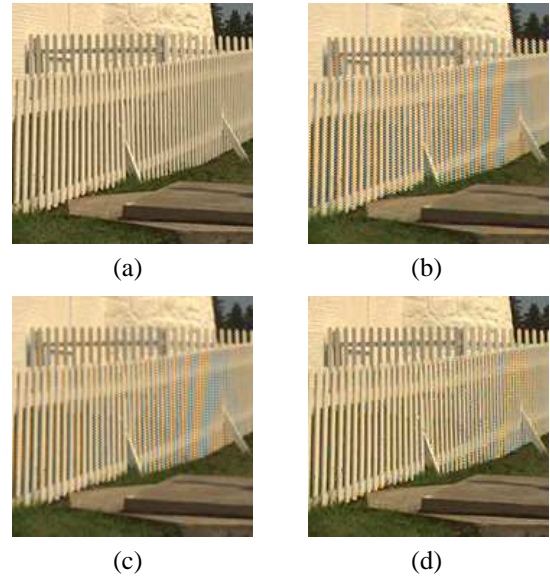


Figure 2: (a) Initial image; (b)-(d) After Bayer filtering and (b) nearest neighbour demosaicking, (c) bilinear demosaicking, (d) edgesensing demosaicking.

3 Subpixel Rendering

Most LCD panels are colour matrix displays, which consist of individually addressable red, green and blue subpixels. Most rendering algorithms treat triads of subpixels as single logical pixels, due to the legacy of CRT displays where this interpretation was enforced by the underlying technology. A white pixel is usually displayed by setting all 3 subpixels of a logical pixel to their maximum intensity. The HVS, which is not able to resolve each subpixel at normal viewing distances, merges these into a single white spot. However, there is nothing to stop us from rendering a white spot using the GBR triad that spans two logical pixels. Such an image would still appear white, yet it can be positioned more precisely than the logical pixel grid allows for. Figure 3 illustrates this idea used to render an angled line. With logical pixels we see the infamous jaggies, but being able to turn on only one or two thirds of a pixel allows us to greatly reduce them. To the viewer the image appears as a smooth, white triangle. Colour fringing can be observed when viewing the image from very close up, since some edges of the image cover only 1 or 2 of the subpixels and so will be composed only of the corresponding primary colours. The effective use of subpixel rendering is made possible by the fact that at normal viewing distances, these chromatic artefacts are invisible to the human viewer [Messing and Kerofsky 2004].

3.1 Hiding chroma aliasing

Experiments using gratings of varying contrast have been used to determine the sensitivity of the HVS to contrast at various spatial frequencies [Mullen 1985]. Figure 4

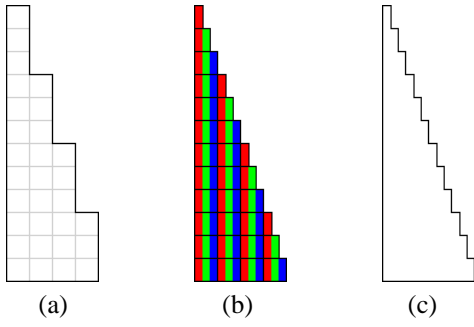


Figure 3: (a) Angled line rendered using logical pixels; (b) Smooth edge produced when rendering with subpixels; (c) The perceived image is white [Gibson 2003].

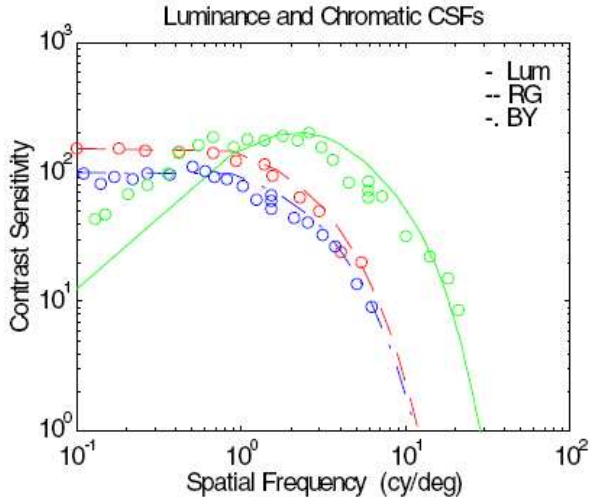


Figure 4: Luminance and opponent colour CSFs [Daly 2001].

shows the contrast sensitivities reported for the luminance, red/green and blue/yellow channels of the HVS. Notice that we are able to perceive luminance detail at frequencies beyond which chromatic detail cannot be detected. This suggests the use of what Daly calls the *advantage region* - a range of frequencies where chromatic aliasing is effectively invisible - to safely render at higher resolutions than can be achieved with logical pixels [Daly 2001]. This is only useful beyond some minimum viewing distance, since moving closer to the image results in the subpixels subtending a larger visual angle, reducing the apparent frequencies of the hidden aliased chromatic signals to the point where they are again detectable.

The Nyquist rate when rendering with logical pixels is 1/2 cycles/pixel. If R, G and B contributed equally to luminance then the Nyquist rate would be raised to 1/2 cycles/subpixel = 3/2 cycles/pixel with subpixel rendering. The aliased components of each colour would mutually cancel to produce an alias-free image [Messing and Daly 2002]. However this is not the case (green contributes far more luminosity

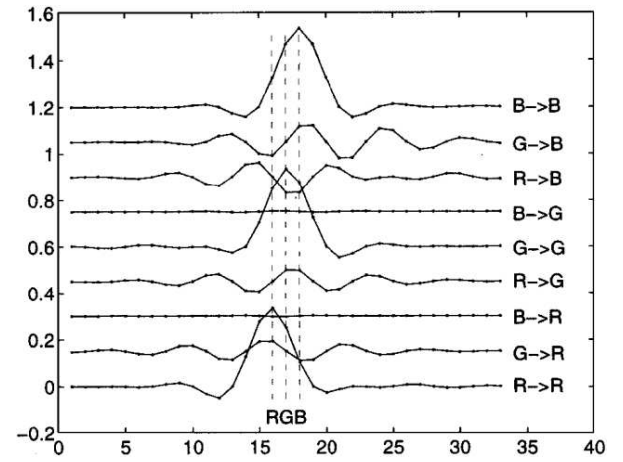


Figure 5: Optimal filters for subpixel antialiasing. Filters are displaced vertically for clarity [Platt 2000].

than red and blue), so this maximum resolution cannot be achieved in practice. By applying an antialiasing filter to remove frequencies above a point between 1/2 and 3/2 cycles/pixel we can make a tradeoff between increased luminance resolution and chromatic aliasing.

3.2 ClearType

Platt [2000] determined the optimal filters for use in the context of font rendering by minimising a perceptual error metric. The error between the original $N \times N$ full-colour samples γ_{kd} (spatially indexed by k) and the subpixel output α_k is defined as

$$E_{ck} = M_{ck}\alpha_k - \sum_{d \in \{R, G, B\}} C_{cd}\gamma_{kd} \quad (1)$$

where M_{ck} and C_{cd} map α_k and γ_{kd} respectively into an opponent colour space (Platt used XYZ coordinates for standard NTSC phosphors and estimated HVS coordinates to define the colour space). The spatial arrangement of the α_k are embedded within M_{ck} . The error is then transformed into frequency space, and weighted by experimentally determined CSFs W_{cn} . Determining the optimal filter then becomes a problem of minimising the importance weighted amplitudes of the error frequencies

$$\epsilon = \sum_{c=0}^2 \sum_{n=0}^{N-1} W_{cn} |E_{cn}| \quad (2)$$

It turns out that solving the linear optimisation problem is equivalent to applying a set of linear filters (shown in Figure 5) to the image. For each of the R, G and B channels, three filters are applied, one to each colour channel, to produce a single output. Notice the horizontal offset in the peaks of filters mapping each colour to itself, which reflects the relative position of those subpixels.

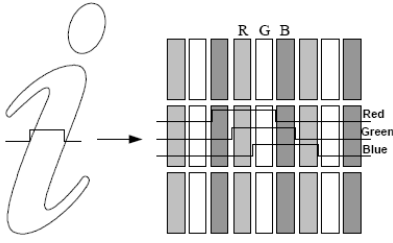


Figure 6: RGB decimation filters. R, G and B samples from the image are known at each of the subpixel locations [Betrissey 2000].

In developing a real-time implementation of this, Betrissey et al. noted that filters between differing channels have insignificant weight compared to the $R \rightarrow R$, $G \rightarrow G$, and $B \rightarrow B$ filters [Betrissey et al. 2000]. In addition, these reflexive filters look remarkably similar to each other. This suggests an efficient approximation, which they term *RGB Decimation*. It ignores the 6 insignificant filters and approximates the 3 remaining ones with a 1 (logical) pixel-wide box filter, as shown in Figure 6. This is reported to produce results nearly as good as optimal filtering, at a greatly reduced cost.

4 Bayer Subpixel Rendering

Two different approaches were attempted to take advantage of subpixel resolution, only the second of which produced favourable results. The initial attempt was a simple extension of existing demosaicking algorithms, which interpolated red and blue values at horizontally offset locations to account for the subpixel layout. One adaptive (edgesensing) and two non-adaptive (nearest neighbour and bilinear) algorithms were modified to do this. Results were only slightly different from the unmodified algorithms' and could not be said to improve image quality at all. A possible explanation for this disappointing result is that subpixel rendering requires a higher resolution input than what is available from a Bayer image. Each colour plane is sampled at a rate less than once per pixel, so rendering the image at subpixel resolution is not going to produce useful results when there is not enough input information available to take advantage of the increased resolution. This observation suggests another approach - artificially inflating the input sampling rate by rendering a zoomed out version of the image. The remainder of this section deals with the downsampling oriented algorithm.

ClearType relies on having a high-resolution image to render with samples for all 3 colours are available for each subpixel. For vector-based images, including fonts, this is certainly the case. However, for images that have been Bayer filtered, we do not have enough information available. Figure 7 overlays the Bayer pattern on the RGB decimation filtering diagram. To compute the middle R subpixel value, we need R values from the 3 marked subpixels. For G subpixels

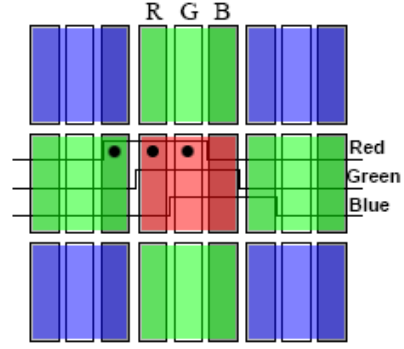


Figure 7: RGB decimation cannot be directly applied to Bayer filtered images.

this is actually not a problem since the optical system in the camera collects light over the entire pixel. The R value is also an area-based sample, but it belongs to an area not centered over that subpixel. To apply the approximate filter we need red samples at the 3 marked subpixels.

Subpixel rendering is used to improve image quality under typical viewing conditions, and one could argue that 1:1 zoom levels are not typical for digital photographs. Without any resampling, even relatively high resolution 1600×1200 displays are not capable of showing an entire 3 megapixel image, which even entry-level digital cameras are able to produce. Camera viewfinders require much greater levels of downsampling to fit the full image. At 1:1 scale there are $1/3$ Bayer samples/subpixel. Downsampling by a factor of 3 effectively increases this to the requisite 1 Bayer sample/subpixel for subpixel rendering. A factor of 6 (in each dimension) is used here since it simplifies the implementation - adjacent 6×6 blocks of Bayer data are identical, while 3×3 blocks contain alternately more red or blue samples. The approach used by the algorithm could be used for any higher downsampling factors as well. Strictly speaking we need only downsample on the horizontal axis but to maintain the original aspect ratio it is also applied along the vertical dimension.

Figure 8 illustrates the proposed algorithm. The bold outlined boxes represent a single logical pixel divided into 3 subpixels. The coloured cells behind it represent the Bayer samples, of which there are more than one per channel for each subpixel. Only 3 rows crossing the R subpixel contain any original data so it is only these rows that we use to obtain a value for that subpixel. Looking at the middle row, the red RGB decimation filter requires that we know the red values over the range $a - f$. Values for a , c and e are directly available, while b , d and f can be linearly interpolated from their neighbours. Finding the red subpixel's value is thus done as

$$CD = \frac{AB}{3} + \frac{CD}{3} + \frac{EF}{3} \quad (3)$$

$$= \frac{a + \frac{a+c}{2}}{3.2} + \frac{c + \frac{c+e}{2}}{3.2} + \frac{e + \frac{e+g}{2}}{3.2} \quad (4)$$

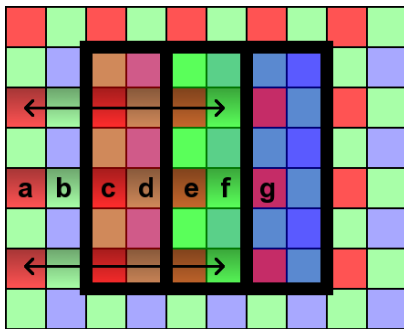


Figure 8: Proposed demosaicking scheme. Arrows and labels indicate the 3 rows to be filtered to produce a value for the red subpixel.

$$= \frac{a}{4} + \frac{c+e}{3} + \frac{g}{12} \quad (5)$$

which may be implemented by horizontally filtering with a kernel of the shape $[0, 1/4, 1/3, 1/3, 1/12]$ and then subsampling every third value. The 3 rows are assigned equal weight by vertically filtering with a kernel of the shape $[1/3, 1/3, 1/3]$.

A symmetric filter is used for the blue subpixels. Since there are green samples in all 6 rows, we have a choice as to how many we wish to use. 3 rows spanning the full subpixel height provide a balance between image quality and processing speed.

Note that there is an implicit linear demosaicking step used to find b, d and f . When these interpolated values are displayed directly on the monitor's pixels, it produces the aliasing shown in Figure 2 (c). However in this downsampled situation, these errors manifest themselves as much higher frequency chromatic aliasing components and are thus invisible under normal viewing circumstances. As the downsampling factor increases, these aliased signals become even less visible.

5 Results

By its very nature, subpixel rendering only improves the quality of images when viewed on an LCD panel. As Platt [2000] notes, printing is a subtractive process, so subpixel rendered images must be viewed on an LCD to fully appreciate its benefits. Text is also a better candidate for subpixel rendering than photographs, which exhibit obvious improvement only in regions of high horizontal frequency.

Subjective evaluation was performed using images where accurate RGB values are available at all pixels. Figures 9 and 10 show the synthetic zoneplate and the Kodak *lighthouse* test images that were used. Samples corresponding to the Bayer pattern were extracted and supplied to the demosaicking algorithms (nearest neighbour, bilinear and edge-sensing). Demosaicked data was then downsampled using nearest neighbour, bilinear and bicubic interpolation. The

proposed algorithm was implemented by filtering the original Bayer samples both horizontally and vertically as described in section 4, and then extracting from each 6×6 block, a single R, G and B value at a location corresponding to the position of the respective subpixel. The implementation was done using MATLAB.

Improvements were visible on both a high resolution laptop display, and a low resolution PDA. Figure 9 shows that, as expected, the regions of high horizontal frequency exhibited the most noticeable improvement when rendered using the subpixel-based approach, while vertical aliasing was not removed at all. Chroma aliasing was visible in images produced using the new algorithm, although it was less objectionable than the loss of detail caused by not taking advantage of subpixel resolution. There are two explanations for the chroma aliasing being visible. Firstly, the photograph was taken from very close to the image (60mm, much closer than what would be considered normal) and secondly, RGB decimation, rather than optimal filtering, was used. The optimal filters have transfer functions very close to ideal low-pass, whereas the decimation filter has a sinc transfer that admits some aliasing [Betrissey et al. 2000].

In Figure 10 the fence is rendered more accurately with the new subpixel algorithm than it is by edgesensing demosaicking combined with bicubic downsampling. This is despite the fact that the subpixel algorithm implicitly uses a linear interpolation step, which generally performs worse than adaptive demosaicking. Notice that the roof of the front left house is very jagged in both photographs. Although subpixel rendering does not improve these high vertical frequency regions, it does not appear to noticeably degrade quality there either.

For a quantitative evaluation, reference images were computed by applying RGB decimation to the original (non Bayer-filtered) images. The PSNR between the reference lighthouse image and that produced by the new algorithm was 30.34. The zoneplate, which represents somewhat of a worst-case scenario given the high frequency content, produced a PSNR of 22.64. It is unknown at this stage how well these RGB decimated Bayer images correspond to optimally filtered original images.

Despite not being implemented with the utmost of efficiency in mind, the algorithm performed favourably compared to other demosaicking/downsampling combinations. Table 1 shows the execution times for the various algorithms on a 512×768 image. This includes additional processing (*i.e.*, Bayer filtering) so these numbers do not reflect expected real-world performance. Although the subpixel algorithm performs a relatively expensive filtering step, it does not need to demosaick each and every pixel, as the naïve nearest neighbour algorithm does, and the two algorithms turn out to be on a par, performance wise. In terms of image quality, the subpixel algorithm produced better output than the edgesensing demosaicking combined with bicubic downsampling, yet it ran in only 20% of the time.

Demosaicking	Downsampling		
	Nearest	Bilinear	Bicubic
Nearest	0.86	1.42	1.44
Bilinear	1.73	2.4	2.38
Edgesense	3.62	4.16	4.28
Subpixel	0.85		

Table 1: Performance results (seconds) for demosaicking and then downsampling the lighthouse image.

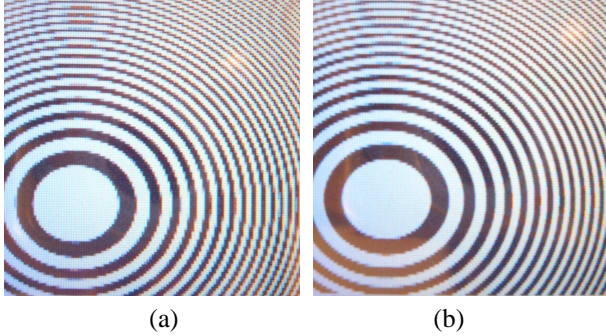


Figure 9: Photographs of zoneplates on PDA display. (a) Naïve bilinear demosaicking and downsampling; (b) Bayer subpixel downsampling. Identical camera settings were used for both images. Improved resolution in the horizontal dimension reduces the jagged edges, but some colour fringing is visible.

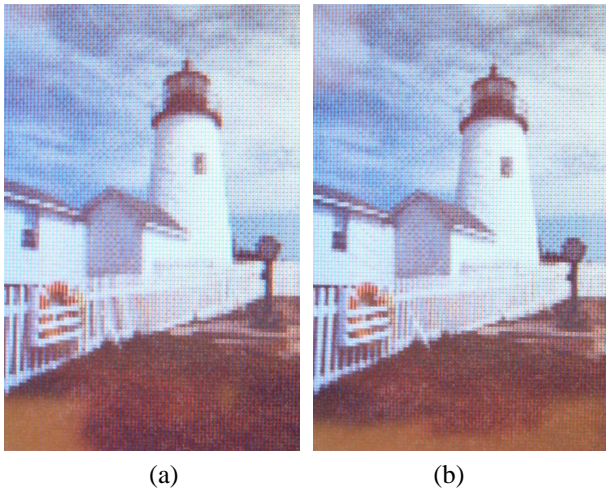


Figure 10: Photographs of lighthouse on PDA display. (a) Naïve edgesense demosaicking and bicubic downsampling; (b) Bayer subpixel downsampling. Identical camera settings were used for both images. The fence just to the right of the safety belt is reconstructed more clearly with subpixel rendering, and although not clearly visible in these photographs, the sides of the lighthouse and thin railings on top were sharper.

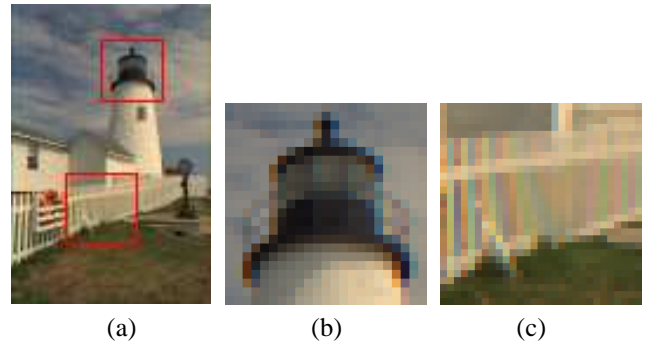


Figure 11: Close-ups of lighthouse. The sky to the right of (b) is lighter than the black structure, so only the right-hand (blue) subpixels on the boundary are set high, producing a blue fringe there. The coloured stripes in the fence in (c) add detail when they are rendered with subpixels

6 Conclusion

A new algorithm for producing downsampled versions of Bayer-filtered images was developed and tested. Following on from the work on subpixel font rendering, it exploits the fact that LCD panels are composed of individually addressable subpixels which can be utilized to improve horizontal luminance resolution without incurring objectionable chromatic aliasing. The novel aspect of this algorithm is that it operates within the Bayer domain, and does not rely on having accurately demosaicked values at all locations. The relatively large downsampling factor imposed by it is not a severe restriction given the application area of viewing digital photographs on lower resolution LCDs.

Results showed that the algorithm was able to retain more horizontal detail than the naïve combination of a downsampling a demosaicked image. This is despite the fact that a more intelligent demosaicking algorithm was used in the latter case (the subpixel algorithm performs some implicit linear demosaicking). Slight chromatic aliasing was present in the output, which can be attributed to the fact that non-optimal filters were used. The resultant artefacts were less objectionable than the loss of detail evident in the traditionally processed images.

For simplicity, the algorithm has only been implemented for downsampling by a factor of 6. Other arbitrary factors (greater than 3) could be allowed as a natural extension of this work. Other future work could look at implementing the optimal filters reported by Platt [2000] to see if it produces noticeably better results. Although efficient to implement, the transfer functions of the RGB decimation filters used by Betrissey et al. are fairly poor approximations of the optimal ones. If desired, the chromatic aliasing could be reduced by using other (wider) filters. An adaptive approach could also be employed to estimate where edges lie by looking in other colour planes, and then allowing this to information to dynamically shape the filter kernel.

In this framework of subpixel rendering of Bayer data, it becomes more natural to think of stored images as being composed of only the low density Bayer samples, and not performing any demosaicking at all. If a large enough down-sampling factor needs to be applied to view the entire image on the display, then better results can be achieved (in less time) by performing the image processing (rotation etc.) in the Bayer domain and applying RGB decimation (or a variant thereof) before outputting the image to an LCD.

References

- ADAMS, J. E. 1995. Interactions between color plane interpolation and other image processing functions in electronic photography. In *Proc. SPIE*, vol. 2416, 144–151.
- BAYER, B. E., 1976. Color imaging array, July. US Patent number 3971065.
- BETRISSEY, C., BLINN, J. F., DRESEVIC, B., HILL, B., HITCHCOCK, G., KEELY, B., MITCHELL, D. P., PLATT, J. C., AND WHITTED, T. 2000. Displaced filtering for patterned displays. In *SID Symposium Digest of Technical Papers*, vol. 31, 296–299.
- CHEN, T., 1999. A study of spatial color interpolation algorithms for single-detector digital cameras. Psych221/EE362 Course Project, <http://ise.stanford.edu/~tingchen/>.
- DALY, S. 2001. Analysis of subtriad addressing algorithms by visual system models. *Society for Information Display Symposium Digest of Technical Papers* 32, 1 (June), 1200–1203.
- DUBOIS, E. 2005. Frequency-domain methods for demosaicking of bayer-sampled color images. In *IEEE Signal Processing Letters*, vol. 12, 847–850.
- GIBSON, S., 2003. Sub-pixel font rendering technology. <http://grc.com/cleartype.htm>, Updated Nov 28, 2003.
- LUKAC, R., PLATANOTIS, K. N., AND HATZINAKOS, D. 2005. Color image zooming on the bayer pattern. In *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 15, 1475–1492.
- MESSING, D. S., AND DALY, S. 2002. Improved display resolution of subsampled color images using subpixel addressing. In *IEEE International Conference on Image Processing 2002*, vol. 1, 625–628.
- MESSING, D. S., AND KEROFSKY, L. 2004. Subpixel rendering on colour matrix displays with 2d geometries. In *SID Symposium Digest of Technical Papers*, vol. 35, 1474–1477.
- MULLEN, K. 1985. The contrast sensitivity of human colour vision to red-green and blue-yellow chromatic gratings. *Journal of Physiology* 359 (Feb.), 381–400.
- PLATT, J. C. 2000. Optimal filtering for patterned displas. *IEEE Signal Processing Letters* 7, 7 (July), 179–181.