PiCam: An Ultra-Thin High Performance Monolithic Camera Array

Kartik Venkataraman*, Dan Lelescu, Jacques Duparré, Andrew McMahon, Gabriel Molina, Priyam Chatterjee, Robert Mullis
Pelican Imaging Corporation

Shree Nayar
Columbia University

Figure 1: From left to right - (a) The PiCam Camera Array Module (b) Raw 4×4 array images each 1000×750 pixels (c) Parallax corrected and superresolved high resolution 8MP Image; (d) A high resolution filtered 8MP depth map

Abstract

We present *PiCam* (Pelican Imaging Camera-Array), an ultra-thin high performance monolithic camera array, that captures light fields and synthesizes high resolution images along with a range image (scene depth) through integrated parallax detection and superresolution. The camera is passive, supporting both stills and video, low light capable, and small enough to be included in the next generation of mobile devices including smartphones. Prior works [Rander et al. 1997; Yang et al. 2002; Zhang and Chen 2004; Tanida et al. 2001; Tanida et al. 2003; Duparré et al. 2004] in camera arrays have explored multiple facets of light field capture - from view-point synthesis, synthetic refocus, computing range images, high speed video, and micro-optical aspects of system miniaturization. However, none of these have addressed the modifications needed to achieve the strict form factor and image quality required to make array cameras practical for mobile devices. In our approach, we customize many aspects of the camera array including lenses, pixels, sensors, and software algorithms to achieve imaging performance and form factor comparable to existing mobile phone cameras.

Our contributions to the post-processing of images from camera arrays include a cost function for parallax detection that integrates across multiple color channels, and a regularized image restoration (superresolution) process that takes into account all the system degradations and adapts to a range of practical imaging conditions. The registration uncertainty from the parallax detection process is integrated into a Maximum-a-Posteriori formulation that synthesizes an estimate of the high resolution image and scene depth. We conclude with some examples of our array capabilities such as post-capture (still) refocus, video refocus, view synthesis to demonstrate motion parallax, 3D range images, and briefly address future work.

CR Categories: I.3.7 [Computer Graphics]: Digitization and Image Capture—Applications I.4.4 [Image Processing and Computer Vision]: Restoration—Inverse filtering I.4.8 [Image Processing and Computer Vision]: Scene Analysis—Range data;

*kartik@pelicanimaging.com

Keywords: plenoptic acquisition, computational camera, light field, array camera, parallax detection, superresolution, depth map

Links: [DL](#) [PDF](#) [WEB](#) [VIDEO](#)

1 Introduction

A fundamental problem in designing cameras for mobile devices such as smartphones and tablets is the industrial design constraints that have to be met. For any given XY sensor format there is a corresponding optical format which, through standard optical design constraints, defines the lower limit of the camera module z-height automatically. This is a key industrial design constraint.

We describe an approach to solve the form factor problem by replacing a single aperture in the camera with a number of smaller apertures (lens array) with largely overlapping fields of view. With the smaller optical format (aperture), the focal length and therefore the camera module z-height, are correspondingly reduced. The sensor array together with the lens array are packaged to form an integrated camera array with a thin form factor. We refer to this basic camera array module as *PiCam* (Figure 2). The choice of aperture size and number of apertures in the *PiCam* is guided by the need to regain the resolution that is suppressed in the aperture size reduction through a post-capture superresolution process. The *PiCam* camera architecture specifications are presented in Section 3.

ACM Reference Format

Venkataraman, K., Lelescu, D., Duparré, J., McMahon, A., Molina, G., Chatterjee, P., Mullis, R., Nayar, S. 2013. PiCam: An Ultra-Thin High Performance Monolithic Camera Array. ACM Trans. Graph. 32, 6, Article 166 (November 2013), 13 pages. DOI = 10.1145/2508363.2508390
<http://doi.acm.org/10.1145/2508363.2508390>

Figure 2: (a) The PiCam module (b) lens array (c) sensor array

The concept of array cameras is not novel in itself and there has been considerable prior work in designing array cameras for various applications [Rander et al. 1997; Yang et al. 2002; Zhang and Chen 2004; Tanida et al. 2001; Tanida et al. 2003; Duparré et al. 2004; Bruckner et al. 2010]. However, our approach is unique in that we are the first to envision an array camera on a single monolithic sensor substrate with a corresponding integrated lens array, that targets the equivalent of multiple-megapixel cameras. Here, each sensor is capable of being independently controlled with respect to gain, exposure, and trigger times. The design of our sensor array is presented in Section 3.2.

To eliminate the color crosstalk that is endemic to all sensors with Bayer filters, each camera in the array is sensitive to a single spectral color and is optically isolated from its neighbors. Eliminating color crosstalk improves color fidelity and reduces the need for aggressive color correction in all lighting conditions. In addition, the relatively narrow spectral sensitivity of each camera (for example: blue=400nm to 500nm, green=470nm to 600nm, or red=570nm to 700nm) relaxes optical design constraints. Our lens array design and module integration are presented in Section 3.1.

Finally, our approach is unique in that it provides an efficient software imaging solution capable of being implemented on a mobile processor. The imaging software pipeline synthesizes a high resolution color image by taking an integrated approach to parallax detection/correction and superresolution and introducing new elements in the image restoration process, while adapting to imaging conditions. A key contribution of our approach is also that we compute the scene depth for each pixel, making this not only an imaging system but also potentially a 3D acquisition system. The software pipeline is presented in Section 4.

Our approach represents a new paradigm in mobile imaging by providing a depth modality in a small form factor design. The system follows a symbiotic design approach guided both by the needs of image post-processing and the trade-offs in hardware design, ultimately delivering an ultra-thin imaging system that is well-balanced in terms of overall resolving power, signal-to-noise ratio (SNR), dynamic range, spectral fidelity, and depth accuracy. The independent control of each camera in the array provides new flexibility and delivers a versatile solution that in combination with depth could enable a suite of applications. Results from a selected set of these applications, such as refocus, and point cloud generation are presented in Section 5.

2 Related Work

The very first vision of an array camera was by Lippmann [1908] where he combined a lenticular array with photographic film in order to capture and reproduce stereoscopic imaging. This is what is known today as *integral photography*. The more recent efforts in array cameras resulted from a need to perform still and video view

synthesis [Levoy and Hanrahan 1996; Taylor 1996]. These image-based rendering techniques provided a way to bridge the gap between synthetically generated scenes and photorealism. Levoy et al. provided a system with rotating camera gantry, which allowed one to capture a single static scene from multiple perspectives. Taylor extended that idea to dynamic scenes by using a linear array of still cameras synchronized to trigger simultaneously. This allowed one to record video scenes, which appear frozen in time, but with the ability to view this frozen scene dynamically from a moving viewpoint. A synchronous system with 51 video cameras surrounding the scene in a hemisphere is discussed in Rander et al. [1997]. The system enabled computing range images using a multi-camera stereo process as well as an intensity image. While both of these early systems were primarily used to synthesize novel views of the captured scene, neither explored the synthesis of images that were in any way enhanced in resolution from the original images.

Other approaches, most notably [Yang et al. 2002] and [Zhang and Chen 2004], used off-the-shelf cameras without explicit synchronization between them. This resulted in artifacts during view synthesis. The work in Wilburn et al. [2005] designed a more general purpose system capable of being extended to large arrays of cameras with the ability to synchronize between the cameras as well as carefully assign their spatial positions. More recently, a method was proposed in Nomura et al. [2007] to capture scene collages from multiple points of view using a flexible array design. They used a plastic frame onto which sets of cameras can be mounted in a variety of layouts and which can be physically flexed to vary the shape of the array. This configuration allowed for significant creative control in capturing very wide fields of view and from which interesting scene collages could be synthesized. These allow one to capture sufficient context that it becomes easy to perceive the overall scene structure. None of these approaches, however, lend themselves easily to small form factor designs.

It is also important to acknowledge the works in [Ng et al. 2005; Georgiev et al. 2011] on the plenoptic camera, given that they capture light fields. Ng introduced a lenslet array at the focal plane of the main lens and positioned the sensor at the focal plane of the lenslet array. Their optical design most resembles the original plenoptic camera in Adelson and Wang [1992]. Each lenslet captures a sub-aperture of the main lens and provides a depth of field that is correspondingly larger than that produced by the full aperture of the lens. Refocusing is achieved by a summation of the shifted versions of the sub-aperture images albeit at a considerable cost to overall image resolution. In contrast, our approach extends the depth of field by shrinking the aperture and the focal length of each lens in the array. In addition, since our approach has all lenses with overlapping fields of view, it offers us the potential for recovering the resolution encoded in the downsampled and aliased images of the array. In Georgiev et al. [2011] a slightly modified approach is suggested, where the lenslet array is moved beyond the focal plane of the main lens. This allows one to capture non-overlapping content interleaved in the pixels under the lenslets. This focused plenoptic camera takes a step in the direction of super-resolving the light field as this enables them to perform a fusion super-resolution (interleaved fusion) compared to Ng's work. However, this approach does not consider the impact of pixel blur and SNR, Bayer crosstalk, and lens blur, and does not use a formal stabilized, non-blind, and complete image restoration process as we do in our approach.

The first array camera that was developed with the goal to reduce the track length of an electronic imaging system was the TOMBO-architecture [Tanida et al. 2001; Tanida et al. 2003]. Here, the single aperture in the camera was divided into several smaller and more or less identical optical channels optically isolated from one another. Post-processing software used an iterative back-projection

(IBP) technique to reproduce an image of higher resolution than that captured by any single aperture on the array. However, the IBP is essentially implementing a Maximum Likelihood (ML) solution to the superresolution problem. As is well known, the ML-solution in noise is unstable, and indeed noise amplification was noted. In Horisaki et al. [2008] an irregular lens arrangement was used to ensure that component images were non-identical and used super-resolution algorithms to restore a high-resolution image. However, neither of these approaches considered an integrated approach to parallax detection/correction and superresolution, choosing instead to recover the resolution at a fixed depth from the camera. In addition, the demonstrated resolutions were low compared to today's expectations for a mobile imaging system.

Motivated by compound eyes in insects, Duparré et al. [2004] developed an array camera using sophisticated micro-optical fabrication techniques. A single pixel was applied per lens and many small lenses were fabricated using lithographic means, with each lens having a different viewing direction. However, the system was not very scalable as it required a non-economically large footprint in order to achieve resolutions of the order of several megapixels. Unlike our system, which is capable of meeting industry expectations for higher resolution (such as 5MP or 8MP), all of these approaches produce low output resolutions and do not offer the full complement of use cases, such as high-resolution stills and 720P/1080P video that are available in the mobile cameras today.

3 PiCam Architecture

We use the term legacy camera to refer to a conventional single aperture design such as cameras in currently available mobile phones. In such cameras, the sensor has a 2x2 Bayer filter pattern over the entire pixel array. Overlaid on top of the sensor is a 4- or 5-element lens. A voicecoil motor is included to provide autofocus. In contrast, Figure 3 shows that the *PiCam*'s focal plane sensor is divided into a number of smaller focal planes, which in turn are overlaid with a lens array. There are several significant changes that are immediately visible: *Optical Stack Height*: Firstly, the height of the

Figure 3: The *PiCam* architecture

optical stack is reduced by the smaller-aperture lens and a corresponding reduction in the number of lens elements needed. The benefits and problems of scaling a lens (changing its focal length at constant F-number and field of view) on its information capacity are fully explored by [Lohmann 1989; Fischer et al. 2008]. It is shown there that the magnitude of wavefront aberrations scales with the lens size up to the power of 4. Thus, smaller lenses can more easily approach the diffraction limit with a smaller number of optical elements.

Color Filters: Secondly, *PiCam* does not use a Bayer color filter pattern. In this *PiCam* instantiation, the filters are on the optical stack. Each camera in the *PiCam* array is optically isolated and sensitive to just one color of light: red, green, or blue. As we will see in Section 3.1, the reduction in spectral bandwidth of each channel enables one to design higher performance lenses. The choice of

the color filter pattern is largely decided by the spectral coverage around a reference camera (which defines our rendering viewpoint) in the different camera use modes - still capture and video. In video mode, to save on power consumption, only 9 of the 16 cameras are active, while in still capture mode all 16 cameras are activated for the increased resolution requirement. The chosen filter pattern needs to be optimal for both. The 'occlusion' or 'visibility' problem manifests itself in *PiCam* when capturing a foreground object against a background object. The areas immediately around the foreground object are not visible to all the cameras in the array due to the fact that different cameras are capturing the scene from different viewpoints and the foreground object occludes parts of the background scene. It is therefore important that irrespective of the orientation of the occluded zone around the foreground object, the area is sampled by at least one camera of each spectral color. The chosen filter pattern (shown in Figure 2 (c)) has two overlapping 3×3 sub-groups (the top left and bottom right), wherein the red and blue spectral channels are symmetrically disposed around a reference green camera in the center. This satisfies the spectral uniformity criteria in the 3×3 video use case. Additionally, it also allows for redundancy in that any one of these two sub-groups could be used for generating video. The top right corner blue filter and the bottom left corner red filter complete the uniformity requirement in the 16 camera still image use mode.

Lens Focal Length: Finally, the focal length of the *PiCam* array and its corresponding hyperfocal distance is shorter as compared to that of an equivalent legacy camera. The autofocus actuator may not be a necessity because its lens has a shorter hyperfocal distance which fundamentally determines its ability to focus on near objects [Kingslake 1951]. The hyperfocal distance is defined as the closest distance at which a lens can be focused while keeping objects at infinity acceptably sharp. It is a function of the focal length and F-Number and is approximated by $h = \frac{f^2}{F_c} + f$, where f is the focal length of the lens, F is its F-number, and c is the circle of confusion. When the lens is focused at this distance, all objects at distances from half the hyperfocal distance out to infinity will be acceptably sharp. Thus, the near focus distance of the lens becomes half the hyperfocal distance. The resulting image for the *PiCam*, with its significantly shorter focal length, is an "all in focus" image.

Given the above differences, the final imaging performance of the camera array is a function of (a) the resolution of the optics, (b) the optical format of each camera in the array, (c) the number of cameras in the array, (d) the pixel pitch and sensor resolution, and (e) the superresolution factor. Sections 3.1, 3.2, and 3.3 below describe the key features of the lens and sensor array and how the trade-offs impact the final resolution and performance of *PiCam*.

3.1 Lens Array

The primary challenge in lens design for array cameras is in preserving sufficient signal at high spatial frequencies to enable effective superresolution from individual low resolution images. As mentioned earlier, each lens in *PiCam* is designed only for a relatively narrow band of the overall visible spectral band of wavelengths. We performed an analytic study wherein we varied the full width at half maximum (FWHM) of the spectral channel bandwidth over which the lens was designed. Optimal diffraction limited lenses were designed for FWHM's of 0nm, 30nm, 60nm, and 100nm width respectively. The resulting lenses showed a monotonic decrease in both on-axis and off-axis MTF with increasing FWHM of the spectral channel. A substantial drop of greater than 25% occurs when the FWHM increases from 60nm to 100nm, showing the potential efficiency in MTF achieved when designing lenses for narrow spectral bands. We observe that when the spectral band of each channel that has to be corrected is substantially reduced, the color blur is

Figure 4: Optical MTF plots versus frequency in linewidths per picture-height (lw/ph) plots with varying F-number and optical formats specified by the number of pixels and pixel size it covers

also correspondingly reduced. Thus for a given lens blur, correspondingly less achromatization efforts have to be taken, which in turn implies lower material combinations and, therefore, fewer lens elements. Our supplementary materials show the variation in MTF and blur for increasing FWHM.

For a diffraction limited optical system, the diameter of the diffraction ring (as characterized by the lens PSF) to the first zero is a function of the wavelength of light (λ) and F-number of the lens and defines the optical cut-off frequency. In such a system, the PSF is the square of the Fourier transform of the lens aperture and the MTF is the magnitude of the Fourier transform of this PSF. The monochromatic diffraction limited optical MTF is given by $MTF_{optics} = \frac{2}{\pi} [\arccos(x) - x\sqrt{1-x^2}]$, where x is the normalized spatial frequency defined as $x = \frac{u}{u_c}$, u is the absolute spatial frequency, and u_c is the diffraction cutoff spatial frequency governed by the wavelength of light and the F-number of the lens as $(\lambda * F)^{-1}$. Therefore, the optical format (or F-number) taken together with the PSF determines the total space-bandwidth product of the diffraction limited optics or, in other words, the total information content in the optical image.

Figure 4 shows the diffraction limited optical MTF and its variation of F-number and optical format respectively. We choose an F3.1 optics design with an optical format equivalent to 1000×750 pixels (that is, $\frac{1}{7}$ inch) as a reference design to illustrate our performance analysis throughout the paper. The figure shows this reference design as the solid blue curve in the graph. Varying the F-number (from F3.1 to F2.6 and F2.0), while holding the optical format at 1000×750 pixels is shown by the two other solid curves in different shades of blue. Adjusting the optical format from 1000×750 to 800×600 or 1280×1024 pixels, while simultaneously holding the F-number constant at F3.1, is shown by cross-hatched and dotted curves. Finally, we show another curve with an optical format of 800×600 using a $1.4\mu m$ pixel and an F2.4 lens. These show a couple of different ways to achieve a higher optical cutoff and MTF. The first option of increasing the optical format increases the cost of XY-size of the camera array, but has the advantage that the optical design is less challenging. The second option of decreasing the F-number of the lens benefits a lower XY-footprint but it would come at the cost of increased lens design complexity. Nonetheless, both of these options are tractable and gives the camera designer a choice based on the various mitigating design factors. As a comparison, we show the MTF of the diffraction limited F2.4 lens for the iPhone5 wherein the larger optical format of the iPhone5 provides a larger space-bandwidth product and permits a much larger amount of information to be transmitted to the focal plane. The gap between

the achievable resolution in the large aperture conventional cameras (such as in an iPhone5) and that in a *PiCam* camera array is what needs to be addressed through a combination of carefully chosen parameters for the optics, sensor, and superresolution software.

3.2 Pixel and Sensor Architecture

The sensor architecture for the *PiCam* array has many aspects that need to be considered for ensuring optimal system performance in terms of overall resolution, optimality in die area, power consumption and data readout, and flexibility in camera control. Much of the camera array functionality is achieved through the use of a novel image sensor architecture. Although the image sensor we use is much like a typical CMOS image sensor from a device physics and circuit design perspective, the architecture of the device was carefully chosen to maximize the feature-set and performance of the camera array.

3.2.1 System Resolution and Pixel Size

Under certain simplifying assumptions, the system MTF is essentially a product of the optical MTF and geometric pixel MTF, wherein each of the optics and sensor stages lowpass filter the input signal (scene) thereby degrading the response at higher frequencies [Holst 1998]. The challenge for array cameras results from the requirement that superresolution should be able to recover a higher resolution final output image than the intrinsic resolution in the input component images. This requires that it is not just the optics MTF that matters but that the overall system MTF has to be maximized and requires consideration of the pixel architecture and size.

For an image sensor, the pixel pitch determines the spatial sampling rate, and the corresponding Nyquist frequency (N_y) is simply one half of the reciprocal of the center-to-center pixel spacing. Thus, for an image sensor with 1000×750 pixels using a 1.75μ pixel, the Nyquist frequency is 286 line-pairs per mm or 750 line-widths per picture height (a line-pair is equivalent to two line-widths). For a legacy camera, very little contrast is desired at spatial frequencies larger than the sensor's N_y , in order to avoid aliasing in the final output image. In fact, in high-end cameras, often the design employs an optical lowpass filter to attenuate higher order frequencies to mitigate the aliasing. This is where our camera array differs from legacy camera. Here, it is desirable to have content above N_y such that when multiple copies of an aliased signal are present, as is the case in a camera array, it is possible through superresolution to use the information that is inherently present in the aliasing to reconstruct a higher resolution signal. The differences between the aliasing patterns in the different camera images are primarily due to the array's sampling diversity resulting from the different viewing directions, which are either intentionally introduced or result from (positional) manufacturing tolerances. In order to attain a certain minimum MTF in the superresolved high resolution (HR) image, we need to provide sufficient contrast in the aliased low resolution (LR) input images from the camera array, which in turn implies a certain threshold for the lens MTF. These limits and thresholds become evident in the graphs that exemplify the optics and sensor design in Figures 4, 6, and 7.

3.2.2 Pixel Stack, Pixel Apertures, and Pixel Blur/MTF

While the pixel size influences the sampling rate and, hence, the aliasing, the structure of the pixel stack itself influences the amount of pixel blur, pixel crosstalk, and the pixel MTF. It is desirable to see what efficiencies can be enabled by exploiting the camera array architecture that is developed here. Figure 5 shows a typical pixel stack, in which light is incident on the microlens at the very top of the stack. The top oxide layer separates the microlens layer from

Figure 5: (a) A legacy pixel stack with a color filter. (b) A PiCam pixel stack with a thinner clear color filter. (c) and (e) Spatial box filters of width $2W_1$ and $2W_2$ respectively, capture the area sampling of the pixel geometry for legacy and PiCam pixel stacks (d) The corresponding sinc filter in frequency domain with cutoffs $1/(2W_1)$ and $1/(2W_2)$ for the legacy and PiCam pixels

the color filter layer and provides a suitable surface for effective microlens formation. The nitride passivation layer and bottom oxide layer provides support and isolation for the metal interconnects that are used to connect the various parts of the sensor. Finally, at the very bottom is the silicon substrate with the active area (*photodiode*) which collects the photons incident on it. The main task of the microlens is to gather all the light incident on its surface and focus that light onto the photodiode in the bottom most layer. While it is convenient to think of a pixel as a point sample, in reality it is an area sampling construct. The microlens spatially samples light over a given area, focusing it on the active photodiode, and in essence averages the light over the area covered by the microlens. In the spatial domain, the square pixel can be modeled as a box filter which, in the frequency domain, is the *sinc* function. Quite simply, larger the width of the pixel (i.e., spatial box filter), the larger the pixel blur and, hence, a smaller cut-off frequency in frequency domain.

In actual implementation, pixel crosstalk reduces the pixel MTF from that defined by its ideal *sinc* filter. In Figure 5(a), the angular incoming light is focused onto the pixel photodiode at the bottom of the stack. However, a certain percentage of the light is lost through crosstalk to the neighboring pixel along the common shared boundary of the neighboring pixel. It is therefore desirable to reduce the height of the pixel stack to reduce the amount of light lost due to crosstalk. Since the *PiCam* camera array employs monochromatic cameras wherein each camera in the array is sensitive to just one color of light, it is possible to move the color filter out of the pixel stack and deposit it on the optical stack or the sensor cover-glass instead. The advantage is that the size of the pixel stack is reduced by the removal of the color filter from the pixel stack. From Figure 5(a) and (b) it can be seen that this is almost equivalent to a 30% reduction, thereby reducing crosstalk by a significant amount. In addition, bringing the photodiode closer to the microlens enables the pixel to accept a wider cone of incoming light, thus providing increased design degrees of freedom to the optics design. Moreover, as shown in Figure 5(b), we shaped the microlens to be narrower, resulting in a reduction of the pixel blur, since the microlens is now averaging light over a relatively smaller area. This reduces light sensitivity at the expense of improving the pixel MTF, but given the significant improvements in sensitivity due to the removal of the

Figure 6: Overall system MTF is the product of the optics MTF and pixel MTF and is plotted for varying F-numbers - F3.1, F2.6, F2.4, F2.0 - and pixel sizes - 1.75μ, 1.4μ, 1.1μ

color filter we felt that this was a reasonable trade-off. The reference design was done on a front-side illuminated pixel, but similar arguments hold for the next generation backside-illuminated pixels as well.

Figure 6 shows the overall system MTF when both the MTF of the diffraction limited optics and the pixel MTF are taken into account. For each of the configurations listed in Figure 4, we show the corresponding system MTF and it can be seen that the blur induced by the pixel area sampling reduces the system MTF from that of the corresponding optics MTF alone. For example, for an F2.0 lens with an optical format of 1000×750 , 1.75μ pixels, the MTF₂₀ (the spatial frequency at which MTF is 20%) is 1627lw/ph, while the corresponding MTF₂₀ value after the pixel sampling blur is taken into account is around 1000lw/ph. This drastic reduction is one of the main reasons why care must be taken to optimize the pixel MTF. Thus, if one were to use the 1.1μ pixel instead of the 1.75μ pixel in this case, the corresponding MTF₂₀ increases from 1000lw/ph to around 1300lw/ph.

The sensor array is designed on a single substrate with multiple sub-arrays such that for each image projected through the apertures of the lens array, it provides temporally coherent captures of these sub-images where the relative temporal offset/phase are controllable. Achieving such functionality necessitates each of the focal plane pixel groups to have its own independent row control logic. This enables features such as single shot HDR where different sub-images could have different exposure times, and high-speed video captures through sequential staggering of the capture times of the individual cameras.

Given there is a plurality of focal planes simultaneously generating pixel data, it is also necessary to multiplex this output data onto the common output through an industry standard camera serial interface (MIPI CSI2). We developed a container format that enables the interleaving of pixel data from each camera (so as to minimize the on-chip storage requirements) as well as provide the receiver with sufficient information to interpret the received data (for example, which cameras are enabled, relative readout position of each camera, the multiplexing scheme used, the gain and integration time).

3.3 Superresolution Factor

Two aspects differentiate the *PiCam* from a conventional camera in terms of system resolution. The first, explained in Section 3.1, was the requirement of sufficient optical MTF above the Nyquist sampling limit of the sensor for superresolution. The second aspect is the superresolution factor itself - how much superresolution can

Figure 7: (a) Graph shows the MTF of the LR images from PiCam, the HR super resolved image, and the aliased part of the spectrum (b) The high resolution super resolved image shows the increased resolution of the PiCam (c) The low resolution image of the ISO12233 frequency chart from one of the cameras in the array (d) Image captured by iPhone5

we expect from the signal processing? Perhaps the most relevant study in this respect was that of [Baker and Kanade 2002], who addressed the limits to the resolution improvement one could achieve using linear SR reconstruction. Here, we investigate the effects of aliasing and analyze the superresolution factor while taking into account the amount of aliasing present in signal. Figure 7(a) shows the MTF curves of the low resolution (LR) image using a solid blue line and the corresponding high resolution (HR) superresolved image, using our superresolution algorithm (Section 4.2), using a solid green line. These represent the MTFs for our reference design using an F3.1 optics and a 1000×750 , 1.75μ sensor. The Nyquist frequency for this is shown by the red line at 750lw/ph.

Signals in the LR image above the Nyquist are aliased and appear as signals of lower frequency than the Nyquist and are represented by the dashed blue curve reflected about the red Nyquist line at 750lw/ph. Thus, at the Nyquist frequency all of the LR signal is aliased, and at frequencies below the Nyquist the dashed line represents the percentage of aliased signal present in the LR signal. To determine the maximum resolvable frequency in the LR domain, we choose that frequency where the amount of aliased signal is no more than 20% of the total signal present. This coincides with the frequency of 480lw/ph indicated by the vertical solid blue line in the graph. The lower part of the solid blue vertical line is shaded red to indicate the amount of aliased signal present. This correlates very well visually with the LR image of the ISO12233 chart, which shows the different frequencies imaged by a single camera in the array. We choose a similar threshold of 20% contrast for the MTF of the superresolved HR image to determine the maximum resolvable output HR frequency. The superresolved HR image corresponding to the HR MTF curve is shown in Figure 7(b) and indicates the maximum resolvable frequency to be 1140lw/ph. The ratio of maximum resolvable HR frequency to that of the maximum resolvable input-LR frequency (as determined by the amount of aliasing present in the signal) is taken to be the true superresolution factor. In our case, this works out to a factor of $\frac{1140}{480} = 2.4$.

4 PiCam Software Pipeline

Figure 8 shows the software processing pipeline that is used to synthesize an HR image from the multiple LR images that are generated from the camera array. The large synthetic aperture of the

array provides us with multiple apertures and, therefore, multiple points of view. In generating an HR image, one could generate the view from any one of the multiple viewpoints or from any point in between. For purposes of illustration, we choose an aperture in the middle of the array (typically the green camera closest to the array center) as the reference viewpoint, and generate the HR image as seen by that viewpoint. In the description that follows, the green camera corresponding to this viewpoint is denoted as the *reference camera*. The sections below outline the key aspects of the above stages and highlight how our approach to processing images from the PiCam differs from prior art.

Figure 8: PiCam Software Processing Pipeline

4.1 Normalization and Parallax Detection

Normalization: Individual cameras in the array are subject to *photometric* and *geometric* variations that arise during the manufacturing and module assembly process. The photometric normalization step corrects for transmission differences in individual lenses by capturing a scene of uniform intensity (i.e., a “flat field”). For each camera in the array, a spatially-varying correction factor is then computed, which reduces variations in photometric responses. Additionally, a pre-calibrated geometric normalization surface G characterizes offsets in geometry between cameras due to differences in lens distortion. The surface is generated by detecting corresponding corners in an image of a black and white checkerboard chart (30x25 squares). The residual distortion vectors in G are later used during parallax and superresolution to account for lens distortion variations. The output of this stage is a photonormalized set of LR images that are input to the parallax detection stage of the pipeline.

Parallax Detection: The parallax processing determines pixel correspondences for non-reference images with respect to the reference camera. Our approach builds upon past work done in the area of stereo correspondence [Szeliski 2010]. Because the cameras in the array are coplanar, in a non-reference camera, scene objects will simply shift along an epipolar line in a direction determined by the baseline separation between the cameras.

A discrete number of depths D are searched that are distributed in equal increments up to a desired maximum disparity. The number of depths to search is large enough to enable detection to sub-pixel levels of accuracy. This is essential for the fusion and super-resolution steps in later stages of the pipeline. For each pixel (i, j) in the reference camera, the best depth must be detected. To make this determination, at a candidate depth d , the pixels in non-reference cameras that correspond to (i, j) are used to calculate a matching cost. We use the sum-of-absolute-differences (SAD) metric where lower costs indicate higher correspondence between pixels across multiple viewpoints and higher costs indicate that the depth is likely a poor match. The depth corresponding to the lowest matching cost is selected as the depth of the pixel (i, j) .

The SAD matching cost metric is applied only between pixels from cameras with similar color filters. For the green cameras in the array, a list C_G contains cameras that are to be compared against the reference green camera. The list C_{Rcomb} enumerates combinations of red cameras to compare to other red cameras, and a list C_{Bcomb} exists for the same purpose in blue. Combinations of cameras in these lists are compared using the SAD metric and the results are added into the combined matching score below. The even distribution of red and blue cameras around the reference green camera represents another key point of differentiation from prior art such as the TOMBO approach where no special consideration was given to the placement of the red, green, and blue filters [Tanida et al. 2003]. Our approach ensures all pixels captured by the reference green camera are also seen by at least one red and one blue camera. Given these lists, a matching score E is calculated for each candidate depth d for a given pixel (i, j) in the reference camera, using the following equation

$$\begin{aligned}
 E(i, j, d) = & \sum_{\{c\} \in C_G} |\hat{y}_c(i_c, j_c) - y_{ref}(i, j)| \\
 & + \gamma_R \sum_{\{r, s\} \in C_{Rcomb}} |\hat{y}_r(i_r, j_r) - \hat{y}_s(i_s, j_s)| \\
 & + \gamma_B \sum_{\{r, s\} \in C_{Bcomb}} |\hat{y}_r(i_r, j_r) - \hat{y}_s(i_s, j_s)|,
 \end{aligned} \quad (1)$$

that sums contributions from red, green, and blue pixel correspondences across multiple cameras. The coordinates (i_p, j_p) represent the pixel coordinates in camera p corresponding to the reference pixel location (i, j) at the depth d , and $y_p(i_p, j_p)$ represents the pixel value at that location in camera p . Similarly, $y_{ref}(i, j)$ represents the intensity of the pixel (i, j) in the reference camera. The camera combinations $\{r, s\}$ represent individual pairs of cameras to be compared in the latter two summation terms of matching costs. The pixel values from the image locations $\hat{y}_c$, $\hat{y}_r$, and $\hat{y}_s$ are obtained through 2D interpolation since the locations (i_c, j_c) , (i_r, j_r) , and (i_s, j_s) do not map to integer grid locations in the cameras y_c , y_r , or y_s . Additionally, γ_R and γ_B are weighting terms that can be used to modulate the influence of the red or blue matching scores relative to the influence of the green cameras. Spatial filtering is applied to the matching scores before the decision in order to increase spatial support and reduce spurious noise artifacts in the final depth map, as is common practice in many stereo algorithms.

PiCam has an intrinsic advantage over stereo cameras in that with the increase in the number of cameras comes a robustness to

noise in depth searches. Additionally, combining matching scores from multiple orthogonal baselines, as explored in Okutomi and Kanade [1993] previously, can resolve ambiguities in cases that are traditionally problematic for binocular stereo, such as detecting the depth of a feature that is oriented parallel to the baseline. If matching scores from camera pairs with both horizontal and vertical baselines are aggregated into the same matching score, such situations can be resolved because the cameras with vertical baselines will be able to detect the depth of the horizontal features, even if the cameras with horizontal baselines cannot.

The depth map provided to super-resolution is the same size as a single, low-resolution array image, and reflects the viewpoint of the reference camera. The depth map contains, for each pixel, the depth value which yields the lowest value of E when aggregating all cameras included in the lists C_G , C_{Rcomb} , and C_{Bcomb} . In the next super-resolution step, this depth (parallax) information will be used along with the known pixel correspondences to fuse and restore a higher resolution, full-color version of the reference camera image.

4.2 Superresolution and Image Restoration

Once the LR images are registered to the reference camera, we rely on a superresolution (SR) process to obtain a high-resolution image from the captured low-resolution images. The progression of this process is illustrated in Figure 9. By design, the *PiCam* system allows aliased frequencies to be captured in each LR image (see Section 3.3) to serve this process. Our SR approach is designed in a Bayesian framework, and takes full advantage of our knowledge of system parameters. This information is used in the form of an *imaging prior*, along with the statistical *image prior* needed for image restoration. The imaging prior includes the sensor analog gain, exposure time, sensor-noise characteristics, optics PSFs and vignetting, geometric distortion, and knowledge of array camera geometry (i.e., baselines). In what follows, we present our superresolution restoration, where the main contributions include: a) an original approach for determining the likelihood-gradient for restoration; b) a new *uncertainty processing* approach to mitigate the errors in the geometric registration of array images; and c) a statistical prior that regularizes the solution and guides the restoration towards a noise-suppressed HR image.

Modeling the Image Formation process: The high resolution image we need to restore is a 2D projection of the 3D scene onto the image plane of a virtual, higher resolution reference camera. For clarity of presentation, we will restrict our discussion to estimating a single color channel (specifically, green channel). Let $\mathbf{x}$ be the desired, lexicographically ordered (vectorized), HR image captured on a high resolution grid corresponding to a selected *reference LR camera*. The goal is to estimate $\mathbf{x}$ from the lower-resolution observations $\{\mathbf{y}_p\}$ (where p indexes the cameras), given knowledge of our imaging system. The images $\{\mathbf{y}_p\}$ feature geometric displacements with respect to the reference image, which include optical distortions (scene independent), and parallax disparity (scene dependent). Mathematically, we can define the forward imaging model generating each observed LR image $\mathbf{y}_p$ as

$$\mathbf{y}_p = \mathbf{D}\mathbf{H}\mathbf{W}_p\mathbf{x} + \mathbf{n}_p, \quad p = 1 \dots P, \quad (2)$$

where the shift matrix $\mathbf{W}_p$ captures the total (geometric distortion and parallax shift) displacement of the image seen from the p^{th} camera, with respect to the reference camera. The matrix $\mathbf{H}$ represents the total blur due to optics and sensor, which is assumed spatially-invariant¹. The decimation matrix $\mathbf{D}$ reflects the sampling

¹Although the pixel blur is spatially invariant, the lens blur is not so in practice. However, the spatial variation is usually smooth enough to allow us to apply the model in Equation 2 in smaller areas.

Figure 9: From left to right: LR image, output of the fusion SR stage, output of MAP-SR (green channel), and the final color HR image

by the sensor of the high resolution image, while $\mathbf{n}_p$ is the imaging noise assumed to be i.i.d. Gaussian. Note that, in general, $\mathbf{W}_p$ and $\mathbf{H}$ are not commutative even though $\mathbf{H}$ is block-Toeplitz (and thus can be made block-circulant) when spatially localized, as the matrix $\mathbf{W}_p$ is not necessarily block-Toeplitz.

Maximum-a-Posteriori Formulation: Estimating $\mathbf{x}$ can be done through different SR approaches (e.g., subspace methods [Vandewalle et al. 2007], sparse representation of patches [Yang et al. 2008], compressive [Sen and Darabi 2009], moving least squares SR [Bose and Ahuja 2006]). Specifically targeting SR of light field images captured by plenoptic cameras, researchers have recently proposed SR algorithms that exploit specific image capture methods [Wanner and Goldluecke 2012; Georgiev et al. 2011]. To exploit our knowledge of the system characteristics and parameters, we employ a Bayesian restoration approach. More precisely, we formulate a maximum-a-posteriori (MAP) approach, that is derived specifically for our system. MAP-based solutions are also popular in the SR literature, see [Hardie et al. 1997; Farsiu et al. 2004; Protter and Elad 2009; Pickup et al. 2007], and references therein. More recently, such Bayesian approaches taking into account the imaging system characteristics have been applied successfully to superresolve light field images by Bishop and Favaro [2012].

To restore the high resolution image $\mathbf{x}$, we iteratively minimize a convex objective function

$$J(\mathbf{x}) = \sum_p \rho(\mathbf{DHW}_p \mathbf{x} - \mathbf{y}_p) + \psi(\mathbf{F}(\mathbf{x})) \quad (3)$$

that consists of a likelihood $\rho(\cdot)$ and a prior $\psi(\cdot)$ term. Here ρ and ψ denote norms, and $\mathbf{F}(\cdot)$ is a statistical prior. The gradient of $J(\mathbf{x})$ with respect to $\mathbf{x}$ is computed as

$$\nabla J(\mathbf{x}) = \sum_p \mathbf{W}_p^T \mathbf{H}^T \mathbf{D}^T (\mathbf{DHW}_p \mathbf{x} - \mathbf{y}_p) + \nabla \psi(\mathbf{F}(\mathbf{x})) \quad (4)$$

when the L_2 -norm is used for the likelihood term (under Gaussian noise assumption). Here ∇ is the gradient operator.

The initial estimate of $\mathbf{x}$ is obtained using a *fusion SR process*. A fused HR image is formed by transferring pixels from the LR images to the reference camera's HR grid, using the total displacements calculated during optical distortion calibration and parallax detection. Pixels initially fused at non-integral HR grid positions are interpolated using any interpolation technique (for example, nearest neighbor), at integer HR grid positions. Thus, the outcome of the initial fusion is a *stack* of a variable number of pixels (including no pixels) at each integer HR grid position. To generate a proper initial HR image estimate, at each HR grid position stacks of pixels are interpolated to generate one pixel value. This may include a spatial interpolation of neighboring grid positions (if a stack ends up empty).

To keep track of the stack of pixels that are used to initialize each HR pixel, we introduce a data structure S at each integer position

on the fused HR grid to store information about the pixel's sources in the LR cameras (camera index and pixel coordinates, color channel). Since we determined the geometric registration relating *all* cameras in the array and the reference camera, the information contained in S also captures the array camera geometry. In fact, for a given reference camera, we can easily map geometric information between cameras in the array, regardless of color channel, using the stacks S and the calibrated baseline information. Thus, S essentially stores a richer geometric information than what is commonly used. One key contribution of our SR process lies in how we exploit this in estimating a new likelihood gradient that allows us to perform superresolution under unconstrained imaging environments.

4.2.1 Likelihood gradient estimation

Let us first concentrate on the gradient of the likelihood term, i.e., first term in Equation 4. To compute this gradient a forward transformation model (Equation 2) is first applied to the HR estimate to obtain the simulated LR images $\tilde{\mathbf{y}}_p = \mathbf{DHW}_p \mathbf{x}$. We then form the likelihood gradients in the LR domain as $\mathbf{g}_p = \mathbf{DHW}_p \mathbf{x} - \mathbf{y}_p$. These LR gradients are then back-projected into the current HR estimate $\mathbf{x}$, using the corresponding per-camera backward transformation matrix $(\mathbf{DHW}_p)^T$. This process essentially corrects the HR estimate iteratively to make it agree with the observed LR images under a determined imaging model [Irani and Peleg 1991]. In our approach, we extend this popular likelihood estimation mechanism to exploit the rich information captured in S , as well as to allow for inaccuracies in the estimation of $\mathbf{W}_p$. We do so through a novel *uncertainty processing* in both the HR and LR domains.

An important point needs to be raised about the commutativity of $\mathbf{H}$ and $\mathbf{W}_p$ matrices here. For our small baselines, the only transformation that the images of scene features will undergo in different cameras of the array is a translation along epipolar lines (no perspective foreshortening). This would appear to imply that the order of warping and blurring prior to the decimation needed to form gradients is inconsequential. However, this is not so when dealing with practical scenes with depth discontinuities where different objects in the scene would shift differently in each LR camera. Inverting the blurring and warping operations in such cases will not map the images correctly to the non-reference camera viewpoints, and the gradients would consequently be inaccurate.

Uncertainty processing on the HR grid: As mentioned earlier, information captured in S at each HR pixel location, in conjunction with camera calibration information (baselines and lens distortion), allows us to relate each HR pixel with the LR images. By exploiting this rich set of information, we can estimate the warp matrices ($\mathbf{W}_p$) that allow us to project the HR estimate onto the viewpoint of all the LR cameras (see left and middle columns of Figure 10). Specifically, the corresponding entries $\mathbf{W}_p(\cdot)$ of the shift matrices are applied to the $\{s, t\}$ pixel position, for all p , even if some of the cameras are not directly linked in the stack S at $\{s, t\}$. Note that the warp estimates $\mathbf{W}_p(\cdot)$ are local (per pixel) and are applied in the HR grid location. Following this warp, we blur and decimate each

Figure 10: Fusion and forward transformation process. For warping, we project pixels onto the nearest integral grid location.

HR warped image to obtain an estimate of $\mathbf{g}_p$ corresponding to each camera $\mathbf{y}_p$. These LR gradients are then upsampled, blurred by $\mathbf{H}^T$ and inverse shifted onto the coordinates of the reference HR image, where they are accumulated and normalized, as per Equation 4.

Using S to locally relate *all* LR cameras even when some may not contribute directly to estimating the initial fused pixel allows us to correct for registration inaccuracies that may result in incorrect image fusion. Additionally, such localized warping (and gradient estimation) makes our likelihood more robust to handling practical cases where depth discontinuities may exist in the scene. In such cases, we cannot relate the LR images with simple registration models as commonly done in literature [Irani and Peleg 1991; Vandewalle et al. 2007].

The approach described until now is computationally expensive (requiring visiting of every LR camera, for every HR grid position). Instead, we restrict projecting the reference HR image at a given pixel position $\{s, t\}$ to only those LR cameras that are contained in $S(s, t)$ which provides a more localized fusion information. Additionally, we warp neighboring pixels of position $\{s, t\}$ to cameras mapped in $S(s, t)$ using their respective stacks S (and camera baseline and optical distortion information) to form patches large enough to blur with $\mathbf{H}$ and decimate. Thus, each HR pixel still contributes to the gradient formation from multiple cameras.

Some of the stacks on the HR grid positions may still be empty after the initial SR fusion. We handle such sparsely populated S distributions by visiting LR cameras (and pixel positions inside those images) based on the neighboring grid positions' stacks. We restrict ourselves to a relatively small (3×3) spatial neighborhood around each empty $\{s, t\}$ on the HR grid. Although this spatial localization is motivated from our need to obtain accurate likelihood gradient estimates even for local sparsely fused HR grids, we could certainly exploit spatial stack-neighborhoods to reduce uncertainty in likelihood gradient estimation for all HR grid locations.

Uncertainty processing on the LR grid: Another contribution of our SR mechanism lies in filtering the likelihood gradient in the LR domain. In literature filtering against the imaging noise and outliers is sometimes applied to the likelihood term to improve estimation quality. For example, in Zomet et al. [2001] a median filtering was used, while El-Yamany and Papamichalis [2008] applied a more robust M-estimator to the likelihood term. In Protter and Elad [2009], a video super-resolution with no explicit motion compensation is described, by forming weighted, patch-based gra-

dients using bilateral weights. In contrast to our work, deblurring is usually performed separately, decoupled from the rest of the SR. Importantly, our motivation for the developments in this section is to not only address the impact of imaging noise and outliers on the likelihood gradients, but also to guard against spatial registration errors reflected in the stacks S and implicitly in $\mathbf{W}_p$.

In our work, instead of computing, and then back-projecting into the current HR estimate the gradients with respect to only the LR pixel positions ($\{a_p, b_p\}$ in Figure 10) indicated by shift matrices $\mathbf{W}_p$, we also visit neighboring LR pixels around those positions. Since in most cases our parallax errors lie within the support of a single LR pixel, we visit only a small 3×3 LR neighborhood.

Referring again to Figure 10, at each decimation position $\{s'_p, t'_p\}$ in the warped and blurred HR images, we obtain an LR gradient $g_p(s'_p, t'_p, m, n)$ for each position $\{m, n\}$ within a 3×3 window $U(a_p, b_p)$ centered at $\{a_p, b_p\}$ in $\mathbf{y}_p$. These contribute now to a filtered gradient at position $\{s'_p, t'_p\}$:

$$g_p(s'_p, t'_p) = \sum_{m, n \in U(a_p, b_p)} \alpha_p(s'_p, t'_p, m, n) g_p(s'_p, t'_p, m, n). \quad (5)$$

The weight $\alpha_p(s'_p, t'_p, m, n)$ is computed from the set of pixels in a 5×5 window $\mathbf{B}_p(a_p, b_p)$ around position $\{a_p, b_p\}$ in $\tilde{\mathbf{y}}_p$, corresponding to $\{s'_p, t'_p\}$, and the block $\mathbf{B}_p^{lr}(m, n)$ centered at each $\{m, n\}$ in $\mathbf{y}_p$ as

$$\alpha_p(s'_p, t'_p, m, n) = \exp \left\{ - \frac{\|\mathbf{B}_p(a_p, b_p) - \mathbf{B}_p^{lr}(m, n)\|}{\sigma} \right\}, \quad (6)$$

where σ is the noise standard deviation.

For each decimation position $\{s'_p, t'_p\}$ in the warped and blurred HR image p , the sum of the weights $\alpha_{p, sum}(s'_p, t'_p) = \sum_{m, n \in U(a_p, b_p)} \alpha_p(s'_p, t'_p, m, n)$, is also recorded alongside the filtered gradient from Equation 5. All the gradient values $g_p(s'_p, t'_p)$ and their corresponding sum of weights $\alpha_{p, sum}(s'_p, t'_p)$, are back-propagated onto the grid of the reference HR image, by upsampling, blurring, and inverse-shifting them. The normalization of the multiple gradients thus formed at $\{s, t\}$ uses the sums of weights $\alpha_{p, sum}(s'_p, t'_p)$, to obtain the final likelihood gradient at $\{s, t\}$. The use of such neighborhood information to filter the likelihood gradient makes our MAP-based SR process stable even with (up to 1 LR pixel) inaccuracies in registration.

4.2.2 Statistical Prior

Since the problem of super-resolution is ill-posed, we use a statistical prior to regularize the HR estimate. Current priors used in image restoration include versions of the Total Variation (TV) [Rudin et al. 1992] approach, including bilateral-TV [Farsiu et al. 2006], autoregressive model prior [Bishop et al. 2009], and Gaussian Mixture Models [Mitra and Veeraraghavan 2012]. Although TV priors encourage the preservation of edges in the restored image, they are based on global image statistics of edges in an image. The locally-adaptive prior we introduce instead has the form

$$\psi(\mathbf{F}(\mathbf{x})) = \|\mathbf{x} - \mathbf{R}(\mathbf{x})\|_{\Lambda, q}^q, \quad (7)$$

where $\mathbf{R}(\cdot)$ is an edge-preserving, local nonlinear operator (the bilateral filter) and the norm is $q = 2$. The gradient of this prior is computed, and added to the likelihood gradient in Equation. 4, then applied to the current HR estimate. This locally-adaptive prior enforces preservation of edges under the operator $\mathbf{R}(\cdot)$. There is a subtle, yet important difference, with the bilateral prior introduced

in [Farsiu et al. 2006]. In that approach, spatial HR-image gradients of different orders are weighed using a bilateral weighting. In our case, the prior gradients are computed between the current HR estimate and its bilaterally-filtered version. In other words, our restored image needs to feature the properties of a bilaterally-filtered HR image.

The data fidelity (likelihood) term by itself provides a good estimate of the HR image in high SNR regions, and the statistical prior helps estimate the low SNR regions better, and ensures solution stability. However, although not done in this paper, we can also incorporate a diagonal weight matrix Λ to weigh the norm in Equation 7, as a function of the SNR. Thus, Λ would cause a higher relative weight for the data fidelity term for high SNR pixels, while for low SNR the image prior would be more significant.

Until now, we have restricted our discussion to estimating a single color channel, namely the green channel. As per-channel estimation is not optimal, a cross-channel approach is always preferable. We could easily incorporate estimating multiple channels in the framework described by injecting a cross-channel image prior. However, due to computational constraints, we first perform a channel difference-based fusion of the red and blue channels with respect to the restored green channel. This is followed by a transfer of weighted MAP-gradients from the green channel into the fused red and blue channels to obtain a high resolution RGB image.

5 Results and Applications

We developed the camera array as a 4x4 sensor array where each sensor supports 1000×750 pixels in a 4:3 aspect ratio. The synthesized high resolution image output by our processing is an 8 megapixel image. The lens array consists of 3 types of lenses - red, green, and blue- where each has F-number 3.1, a diagonal field of view of 56° and a focal length of 2mm. The overall camera module height is less than 3.5mm as compared to a height of anywhere between 5.2mm and 6.0mm for a legacy camera module. A key

Distance (m)	.20	.40	.60	.80	1.00	2.00	3.00	4.00	5.00
Error (%)	1.1	2.3	3.5	4.7	5.9	12.6	20.2	28.9	38.9

Figure 11: Variation of depth resolution with object distance

challenge in creating a robust imaging system is the estimation of accurate depth by the parallax stage and the resulting resolution of the depth map. The least count (this is the highest degree of accuracy of measurement) of the disparity between two cameras is a function of the system noise and blur and, in practice, we have found this to be $\frac{1}{3}$ of an LR pixel. This puts a limit on the depth map accuracy and how the depth resolution changes with distance. Since disparity is a function of depth, if we let $d_{u \rightarrow v}(z_1)$ represent the disparity between two cameras, u and v , at a distance z_1 and $d_{u \rightarrow v}(z_2)$ represent the disparity at distance z_2 (where $z_2 > z_1$), then $d_{u \rightarrow v}(z_1) - d_{u \rightarrow v}(z_2) \geq \frac{1}{3}$. This constraint can be used to determine how the estimated value of depth changes with increase in object distance. Figure 11 shows this variation of the accuracy of the depth estimate with object distance.

Obviously a key aspect of how successful this approach is in being a reasonable alternative to conventional camera design lies in how effective the superresolution is at recovering the resolution encoded in the aliased signals of the LR images in the array, under all lighting and scene conditions. Figure 7 shows our system behavior in capturing high frequency images. The ability of the camera array to capture high frequency signal is fundamentally defined by the lens F-number (MTF), optical format of lens, pixel size (pixel

MTF is in Line widths per picture height (lw/ph)	Pixel Type and Size (Array Size)				
	1.4 μ BSI 800x600	1.75 μ FSI 1000x750	1.75 μ FSI 1280x1024	1.4 μ BSI 1280x1024	1.1 μ BSI 1600x1300
F#	3.1	1050	1417	1144	1220
	2.4	950	1180	1325	1470
	2.0	-	1281	1473	1640

Figure 12: Varying F-number, optical format, and pixel size enables one to achieve different resolutions in the PiCam

Figure 13: (a) Without and (b) with, our likelihood and uncertainty processing

MTF), and the superresolution factor. Figure 12 shows how the resolution of the final super-resolved image changes with varying values of these parameters by extrapolating from the current reference design (highlighted in red) using a F3.1 lens, and 1000×750 , 1.75μ front-side illuminated (FSI) pixels per camera and using the information presented in Figure 6. The table can be used to determine how much resolution is being traded off by the array camera in comparison with the legacy camera in gaining depth information. If we consider the sensor size (3264×2448), pixel size ($1.4\mu m$), and F-number (F2.4) of the iPhone5, a corresponding array camera of similar specification would be a 4×4 array with each sub-array having 800×600 , $1.4\mu m$ pixels and an F2.4 lens. The achievable resolution (MTF20) of such a comparable system would be 950 lw/ph (highlighted in blue). In comparison, the achievable resolution (MTF20) of the iPhone5 is 2000lw/ph (Figure 6) that tells us we are trading resolution by a factor of 2 for depth information.

In Figure 13, we also illustrate the effects of not using some of the new developments discussed in Section 4.2 (namely the likelihood gradient formation and filtering, and the uncertainty processing), on the quality of restoration. This is akin to performing a more classical likelihood computation, and it basically translates into forming single likelihoods (no filtering) at fusion-populated HR grid positions, where we used a median filtering of any pixel stacks formed, and interpolating any holes on the HR grid. This degrades image quality, and may even interfere with the correct reconstruction of frequencies, as errors in the operations preceding the MAP processing start playing a larger role.

Figure 15 shows a comparison of the noise performance of the camera array against that of the iPhone5 at 100Lux illumination. We offset the exposure time on the camera array to compensate for the faster lens on the iPhone5. The figure shows, in 3 zoomed out regions, the relative noise difference between the two cameras. Both visually and using the SNR metric (both measured YSNR of about 39dB) we see that our camera array performs at close to the noise performance of the iPhone5. It must be pointed out, however, that a smaller exposure time also leads to more motion blur and in the

Figure 14: First two rows show example image comparisons - (a) PiCam Hi Res 8MP image, 1/30sec (b) iPhone5 8MP image, 1/30sec (c) 100% zoomed region of interests showing differences between PiCam and iPhone5. Third row shows the post capture refocus ability of the PiCam - (a) Foreground (b) Middle and (c) Farthest, subject in focus (insets show zoomed background subject)

final analysis is not an adequate solution and one that can only be addressed by a corresponding faster lens for the PiCam. We have also compared our camera against iPhone5 both indoors and outdoors. All of these scenes were captured at the same time by the PiCam and by the iPhone5 under the same illumination conditions.

Figure 14, in the top two rows, shows a couple of examples of an indoor and outdoor shot, and the last row shows an example of a post capture refocus application. By virtue of having a relatively small optical format ($\frac{1}{7}$) and a shorter focal length compared to a legacy 8MP camera in a mobile device, the hyperfocal distance of our imaging system is at 40cm. Thus, in addition to all objects from 20cm to ∞ being captured in focus, the computed depth map allows one to achieve a degree of post-capture refocus that provides creative flexibility to the photographer. The iPhone5 has a larger aperture and a correspondingly large hyperfocal distance (≈ 100 cm). In the top row image, the iPhone5 is focused on a near object (< 50 cm) and has a correspondingly short depth of field so that the background picture has defocus blur. By contrast, the PiCam has everything in focus and the background appears sharper than it does on the iPhone5. A second point to consider is that the degree of refocus ability is clearly dependent on the resolution of

Figure 15: 100 Lux noise performance comparison - (Left) PiCam F3.1, shutter 1/12sec (Right) iPhone5 F2.4, shutter 1/20sec

the depth map. Thus, an object at a given distance from the camera may be refocused against its background as long as the background object is at a distance larger than the depth resolution for that distance (see Figure 11). The last row of Figure 14 shows an example of an image being refocused, first with the foreground in focus and

Figure 16: (a) Image captured by PiCam, (b) Point cloud

then with the background in focus. Our synthesized high resolution images show traces of chromatic fringing that could be improved with better color tuning and tone mapping and is an area for future work.

Finally, *PiCam* is a 3D capable capture device given its ability to compute depth of textured surfaces. This should enable a handheld digitization device capable of creating point clouds that lend themselves to subsequent meshing. We captured an object (a Chinese dragon), shown in Figure 16(a), on a tabletop at a distance of 50cm and converted the depth map from the object to a point cloud by filtering the depth map to reveal just the high confidence depth values. A total of 87000 points were rendered with texture and the resulting figure is shown in Figure 16(b). In addition to the above applications, the *PiCam* architecture also enables view synthesis from any position on the array and also in front or behind the array. The synthesis procedure consists of three steps - calculating the displacement of each image in the array to the virtual viewpoint, computing the depth map for the virtual viewpoint, and forming the color image at that viewpoint. The method enables one to showcase motion parallax and generate stereo left and right views. Being able to generate different viewpoints, when combined with real-time gyroscope readings on a mobile device, provides the viewer with a sense of scene depth and a rich interactive experience. We show examples in the accompanying video.

6 Future Work and Conclusion

Our approach to computational array cameras has shown robustness in generating high resolution color images under varying imaging conditions. In addition, *PiCam* provides a significant difference over the legacy camera in that it relaxes the z-height constraint considerably. The resolution of our system is a function of the optical format and F-number of the lens, the pixel size, and the number of cameras in the array. Depending on the application, an appropriate choice of the imaging system and processing parameters allows one to scale the resolution of the system without hitting the z-height constraint. Additionally, the potential for assembling systems with multiple distributed arrays allows for greater range imaging and view synthesis, which could benefit applications in security and automotive. Our system is also fault tolerant to an extent since the failure of 1 or 2 cameras in the array leads to a graceful degradation of system performance.

At the same time, there are areas of improvement both in terms of image quality and processing speed. Artifacts from incorrect depth estimates, along the contours of objects close to the camera, are still possible depending on scene content and background texture. The correction of these artifacts, while not a focus of this paper, remains an important area of continuing research. The algorithms are computationally intensive and while efficiencies are being exploited to allow the pipeline to scale to mobile devices, this remains an active area of work due to the constrained nature of power and compute

availability in these form factors. Nevertheless, we believe that with continued work in both of these areas this architecture has the potential to remove some of the limitations to camera design for small form factor devices, and enable new imaging applications.

Acknowledgments

It is a privilege to be part of the team that helped create the *PiCam* computational camera array. This camera is the basis of Silicon Valley startup Pelican Imaging Corporation founded by the author Kartik Venkataraman with assistance from colleagues Dan Lelescu, Jacques Duparré, Robert Mullis, and Bedabrata Pain. The authors wish to thank the people whose hard work has made this computational camera array possible, including the engineering team led by Purnam Sheth. The sensor and lens arrays are based on the architectural specification of Pelican Imaging Corporation and manufactured by our industry partners - Aptina Imaging [Cho et al. 2013], Heptagon and others. We also extend thanks to those that helped with the preparation of this paper, including Herve Hornung, Lisa Garvey, and Stephen Williams. Finally, we would like to acknowledge the support and encouragement of Chris Pickett.

References

- ADELSON, T., AND WANG, J. Y. A. 1992. Single lens stereo with a plenoptic camera. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 14, 2 (Feb.), 99–106.
- BAKER, S., AND KANADE, T. 2002. Limits on super-resolution and how to break them. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 24, 9 (September), 1167 – 1183.
- BISHOP, T. E., AND FAVARO, P. 2012. The light field camera: Extended depth of field, aliasing, and superresolution. *IEEE Transactions on Pattern Recognition and Machine Intelligence* 34, 5 (May), 972–986.
- BISHOP, T., ZANETTI, S., AND FAVARO, P. 2009. Light field superresolution. In *Proceedings of International Conference on Computational Photography (ICCP)*, IEEE.
- BOSE, N. K., AND AHUJA, N. A. 2006. Superresolution and noise filtering using moving least squares. *IEEE Transactions on Image Processing* 15, 8 (Aug.), 2239–2248.
- BRUCKNER, A., DUPARRE, J., LEITEL, R., DANNBERG, P., BRAUER, A., AND TUNNERMANN, A. 2010. Thin wafer-level camera lenses inspired by insect compound eyes. *Optics Express* 18, 24 (Nov.), 24379–24394.
- CHO, K.-B., TU, N., BRUMMER, J., AZAD, K., HSU, L., WANG, V., KIM, D., PALLE, K., MIAO, T.-M., CHEN, Y., HONG, C., BAO, T., SHARMA, V., FONG, Y., IRKAR, K., HASHMI, S., SUKUMAR, V., KABIR, S., ROSENBLUM, G., GAO, Y., AHN, K.-H., KO, H.-J., WATSON, J., AND KENOYER, C. 2013. A 12MP 16-focal plane CMOS image sensor with 1.75 μ m pixel: Architecture and implementation. In *Proceedings of International Image Sensors Workshop*.
- DUPARRÉ, J., DANNBERG, P., SCHREIBER, P., BRAUER, A., AND TUNNERMANN, A. 2004. Thin observation module by bound optics: Concept and experimental verification. *Artificial apposition compound eye fabricated by micro-optics technology* 43.
- EL-YAMANY, N. A., AND PAPAMICHALIS, P. E. 2008. Robust color image superresolution: An adaptive m-estimation framework. *EURASIP Journal in Advances in Signal Processing* 2008 (Feb.), Article ID 763254.

- FARSIU, S., ROBINSON, D., ELAD, M., AND MILANFAR, P. 2004. Advances and challenges in super-resolution. *International Journal of Imaging Systems and Technology* 2004, 14 (Aug.), 47–57.
- FARSIU, S., ELAD, M., AND MILANFAR, P. 2006. Multiframe demosaicing and super-resolution of color images. *IEEE Transactions on Image Processing* 15, 1 (Jan.), 141–159.
- FISCHER, R. E., GALEB, B. T., AND YODER, P. 2008. Optical systems design.
- GEORGIEV, T., CHUNEV, G., AND LUMSDAINE, A. 2011. Superresolution with the focused plenoptic camera. *Computational Imaging IX*, 243–254.
- HARDIE, R. C., BARNARD, K. J., AND ARMSTRONG, E. E. 1997. Joint MAP registration and high-resolution image estimation using a sequence of undersampled images. *IEEE Transactions on Image Processing* 6, 12 (Dec.), 1621–1633.
- HOLST, G. C. 1998. *CCD Arrays, Cameras and Displays*, 2nd ed. JCD Publishing and SPIE, Winter Park, Florida, March.
- HORISAKI, R., NAKAO, Y., TOYODA, T., KAGAWA, K., MASAKI, Y., AND TANIDA, J. 2008. A compound-eye imaging system with irregular lens-array arrangement. *Proceedings of the SPIE* 7072 (Sept.).
- IRANI, M., AND PELEG, S. 1991. Improving resolution by image registration. *CVGIP: Graphical Models and Image Processing* 53, 3 (September), 231 – 239.
- KINGSLAKE, R. 1951. *Lens design fundamentals*. Academic Press.
- LEVOY, M., AND HANRAHAN, P. 1996. Light field rendering. In *Proceedings of SIGGRAPH 96*, Annual Conference Series, 31–42.
- LIPPMANN, G. 1908. Épreuves réversibles. photographies intégrales. *Comptes Rendus de l'Académie des Sciences* 9, 146 (Mar.), 446–451.
- LOHMANN, A. W. 1989. Scaling laws for lens systems. *Applied Optics* 28, 23 (December), 4996–4998.
- MITRA, K., AND VEERARAGHAVAN, A. 2012. Light field denoising, light field superresolution and stereo camera based refocusing using a GMM light field patch prior. In *Proceedings of Computer Vision and Pattern Recognition Workshops (CVPRW)*, IEEE, 22–28.
- NG, R., LEVOY, M., BRODIE, M., DUVAL, G., HOROWITZ, M., AND HANRAHAN, P. 2005. Light field photography with a handheld plenoptic camera. *Stanford University Computer Science Tech Report CSTR-2005-02*.
- NOMURA, Y., ZHANG, L., AND NAYAR, S. 2007. Scene Collages and Flexible Camera Arrays. In *Proceedings of Eurographics Symposium on Rendering*.
- OKUTOMI, M., AND KANADE, T. 1993. A multiple-baseline stereo. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 15, 4 (Apr.), 353–363.
- PICKUP, L. C., CAPEL, D. P., ROBERTS, S. J., AND ZISSERMAN, A. 2007. Overcoming registration uncertainty in image super-resolution: Maximize or marginalize? *EURASIP Journal in Advances in Signal Processing* 2007, 2 (June), Article ID 2356.
- PROTTER, M., AND ELAD, M. 2009. Super resolution with probabilistic motion estimation. *IEEE Transactions on Image Processing* 18, 8 (Aug.), 1899–1904.
- RANDER, P., NARAYANAN, P., AND KANADE, T. 1997. Virtualized reality: Constructing time-varying virtual worlds from real events. *IEEE Proceedings on Visualization*, 277–283.
- RUDIN, L. I., OSHER, S., AND FATEMI, E. 1992. Nonlinear total variation based noise removal algorithms. *Physica D* 60 (Nov.), 259–268.
- SEN, P., AND DARABI, S. 2009. Compressive image super-resolution. In *Proceedings of 43rd Asilomar Conference on Signals, Systems, and Computers*, IEEE, 1235–1242.
- SZELISKI, R. 2010. *Computer Vision, Algorithms and Applications*, 2011 ed. Springer, Springer-Verlag, London Limited.
- TANIDA, J., KUMAGAI, T., YAMADA, K., MIYATAKE, S., ISHIDA, K., MORIMOTO, T., KONDOU, N., MIYAZAKI, D., AND ICHIOKA, Y. 2001. Thin observation module by bound optics: Concept and experimental verification. *Applied Optics* 40, 11 (Apr.), 1806–1813.
- TANIDA, J., SHOGENJI, R., KITAMURA, Y., YAMADA, K., MIYAMOTO, M., AND S, S. M. 2003. Color imaging with an integrated compound imaging system. *Optics Express* 11, 18 (Sept.), 2109–2117.
- TAYLOR, D. 1996. Virtual camera movement: The way of the future? *American Cinematographer* 77, 9 (Sept.), 93–100.
- VANDEWALLE, P., SBAIZ, L., VANDEWALLE, J., AND VETTERLI, M. 2007. Super-resolution from unregistered and totally aliased signals using subspace methods. *IEEE Transactions on Signal Processing* 55, 7 (July), 3687–3703.
- WANNER, S., AND GOLDLUECKE, B. 2012. Spatial and angular variational super-resolution of 4D light fields. In *Proceedings of European Conference on Computer Vision*, Springer, vol. 7576 of *Lecture Notes in Computer Vision*, 608–621.
- WILBURN, B., JOSHI, N., VAISH, V., TALVALA, E.-V., ANTUNEZ, E., BARTH, A., ADAMS, A., HOROWITZ, M., AND LEVOY, M. 2005. High performance imaging using large camera arrays. *ACM Trans. Graph.* 24, 3 (July), 765–776.
- YANG, J. C., EVERETT, M., BUEHLER, C., AND MCMILLAN, L. 2002. A real time distributed light field camera. *Eurographics Workshop on Rendering*, 77–86.
- YANG, J., WRIGHT, J., MA, Y., AND HUANG, T. 2008. Image super-resolution as sparse representation of raw image patches. In *Proceedings of Computer Vision and Pattern Recognition (CVPR)*, IEEE, 1–8.
- ZHANG, C., AND CHEN, T. 2004. A self-reconfigurable camera array. *Eurographics Symposium on Rendering*, 243–254.
- ZOMET, A., RAV-ACHA, A., AND PELEG, S. 2001. Robust super-resolution. In *Proceedings of Computer Vision and Pattern Recognition (CVPR)*, vol. 1, IEEE, 645–650.