Real-time Implementation of Gaussian Image Blending in a Spherical Light Field Camera

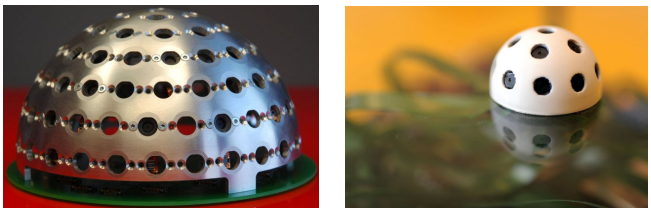
Vladan Popovic, Hossein Afshari, Alexandre Schmid and Yusuf Leblebici
Ecole Polytechnique Fédérale de Lausanne (EPFL), Lausanne, Switzerland.
vladan.popovic@epfl.ch, hossein.afshari@epfl.ch, alexandre.schmid@epfl.ch, yusuf.leblebici@epfl.ch

***Abstract*—Panoptic is a custom spherical light field camera used as a polydioptric system where imagers are distributed over a spherical geometry, each having its own vision of the surrounding and a distinct focal plane. The spherical light field camera records light information from any direction around its center. A novel Gaussian blending technique is presented for vision reconstruction of a virtual observer located inside the spherical geometry of this camera. This blending technique improves the output quality of the reconstructed image with respect to ordinary stitching techniques and simpler image blending algorithms. A hardware architecture based on Field Programmable Gate Arrays (FPGA) with the real-time implementation of the Gaussian blending algorithm using the spherical light field camera is presented, along with the imaging results.**

I. Introduction

A trend in constructing high-end computing systems consists of parallelizing large numbers of processing units. A similar trend is observed in digital photography, where multiple images of a scene are used to enhance the performance of the capture process. The most common applications relate to increasing image resolution [1] and obtaining high dynamic range images [2], [3]. Virtualized reality and view interpolation for creating the illusion of a three-dimensional scene is another use of multi-view systems [4].

Early systems for capturing multiple views were based on a single translating [5] or rotating [6] high-resolution camera for capturing and later rendering the light field. This idea was later extended to a dynamic scene by using a linear array of still cameras [7]. For capturing large data sets, researchers focused on arrays of video cameras. In addition to the synchronization of the cameras, very large data rates present new challenges for the implementation of these systems. The first camera array systems were built only for recording and later offline processing on Personal Computers (PC) [4]. Other such systems [8], [9] were built with real-time processing capability for low resolution and low frame rates. A general-purpose camera array system was built at Stanford University [10] with limited local processing at the camera level. This system was developed for recording large amounts of data and its intensive offline processing, and not for real-time operations. Recently, a camera system able to acquire an image frame with more than 1 gigapixel resolution was developed [11]. The system uses a parallel array of microcameras to acquire the image. Due to the extremely high resolution of the image, it suffers from a very low frame rate, even at low output resolution.



(a) (b)

Fig. 1. Side view of the fabricated Panoptic cameras with a hemispherical diameter of (a) $2\,r_{\odot} = 129\mathrm{mm}$ with 104 camera positions and (b) $2\,r_{\odot} = 30\mathrm{mm}$ with 15 camera positions.

Most developed camera array systems are bulky and not easily portable platforms. Their control and operation depend on multi-computer setups. In addition, image sensors on camera arrays are mounted on planar surfaces which prohibits them from covering the full view of their environment. Full view or panoramic imaging finds application in various areas such as autonomous navigation, robotics, telepresence, remote monitoring and object tracking. Several solutions for acquiring omnidirectional images and their application have been presented in [12]. Another acquisition principle using cameras arranged in a ring is presented in [13]. This system supports a low number of monochrome cameras, with the limited vertical angle of view of the omnidirectional image.

A new approach for creating a multi-camera system distributed over a spherical surface is presented in [14], [15]. This new multi-camera system is referred to as the Panoptic camera. The Panoptic camera is an omnidirectional imager capable of recording light information from any direction around its center. It is also a polydioptric system where each CMOS camera sensor has a distinct focal plane. Fig. 1 depicts two prototypes of a custom-made Panoptic camera.

The omnidirectional vision reconstruction algorithm using the nearest neighbor technique and the coverage analysis of the Panoptic camera is discussed in detail in [14]. A linear blending technique is presented in [16]. A novel blending algorithm based on a Gaussian weighted average is developed and presented in Section II-B. This blending algorithm significantly improves the quality of the final reconstructed image. The algorithm is implemented in real-time into the hardware platform presented in [14]. Imaging results and comparisons are presented in Section IV.

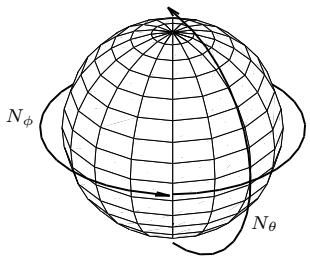


Fig. 2. Pixelized spherical surface $\mathcal{S}_d$ with $N_\theta = 16$ latitude pixels and $N_\phi = 16$ longitude pixels (total of 256 pixels).

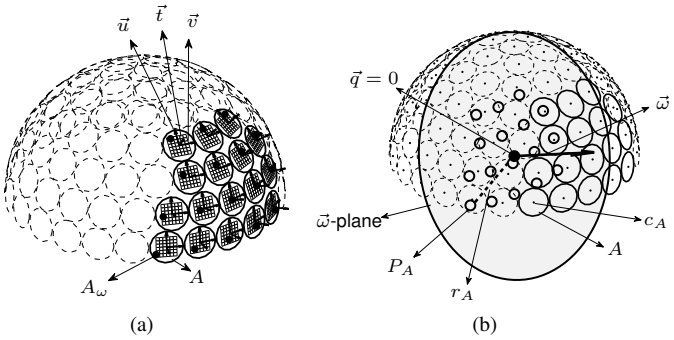


Fig. 3. (a) Cameras contributing to the direction $\vec{\omega}$ with their contributing pixels in the respective image frames, (b) projections of camera centers contributing in direction $\vec{\omega}$ onto planar surface normal to $\vec{\omega}$.

II. Omnidirectional Vision Reconstruction Algorithm

The omnidirectional vision of a virtual observer located anywhere inside the hemisphere of the Panoptic structure can be reconstructed by combining the information collected by each camera in the light ray space domain (or light field [5]).

In this process, the light field is estimated on a discretized spherical surface $\mathcal{S}_d$ of directions. The surface of this sphere is discretized into an equiangular grid with N_θ latitudes and N_ϕ longitudes samples, where each sample represents one pixel. Fig. 2 shows a pixelized sphere with sixteen pixels for N_θ and N_ϕ each. A unit vector $\vec{\omega} \in \mathcal{S}_d$, represented in the spherical coordinate system $\omega = (\theta_\omega, \phi_\omega)$, is assigned to the position of each pixel.

The construction of the virtual omnidirectional view $\mathcal{L}(\vec{q}, \vec{\omega}) \in \mathbb{R}$, where $\vec{q}$ points to the observation point, is performed in two steps. The first step consists of finding a pixel in each camera image frame that corresponds to the direction defined by $\vec{\omega}$. The second step consists of blending all pixel values corresponding to the same $\vec{\omega}$ into one. The result is the reconstructed light ray $\mathcal{L}(\vec{q}, \vec{\omega})$.

To reconstruct the light field, all the cameras having an $\vec{\omega}$ in their angle-of-view are first determined. To extract the light intensity in that direction for each contributing camera, a pixel in the camera image frame has to be found. Due to the rectangular sampling grid of the cameras, the $\vec{\omega}$ does not coincide with the exact pixel grid locations on the camera image frames. The pixel location is chosen using the nearest neighbor method, where the pixel closest to the desired direction is chosen as an estimate of the light ray intensity. The process is then repeated for all $\vec{\omega}$ and results in the estimated values $\mathcal{L}(c_i, \vec{\omega})$, where c_i is the radial vector directing to the center position of the i^{th} contributing camera's circular face. Fig. 3(a) shows an example of the contributing cameras for a random pixel direction $\vec{\omega}$ depicted in Fig. 3(b). The contributing position A_w of the camera A, providing $\mathcal{L}(c_A, \vec{\omega})$ is also indicated in Fig. 3(a).

The second reconstruction step is performed in the space of light rays given by direction $\vec{\omega}$ and passing through the camera center positions. Under the assumption of Constant Light Flux (CLF), the light intensity remains constant on the trajectory of any light ray. Following the CLF assumption, the light ray intensity for a given direction $\vec{\omega}$ only varies in its respective orthographic plane. The orthographic plane is a plane normal to $\vec{\omega}$. Such plane is indicated as the "$\vec{\omega}$-plane" in Fig. 3(b), and represented as a gray-shaded circle (the boundary of the circle is drawn for clarity purposes). The light ray in direction $\vec{\omega}$ recorded by each contributing camera intersects the $\vec{\omega}$-plane in points that are the projections of the cameras focal points on this plane. The projected focal points of the contributing cameras in $\vec{\omega}$ direction onto the $\vec{\omega}$-plane are highlighted by hollow points in Fig. 3(b). Each projected camera point P_{c_i} on the planar surface is assigned the intensity value $\mathcal{L}(c_i, \vec{\omega})$, that is calculated in the first step.

As an example, the projected focal point of camera A onto the $\vec{\omega}$-plane (*i.e.* P_A) in Fig. 3(b) is assigned the intensity value I_A. The virtual observer point inside the hemisphere (i.e. $\vec{q}$) is also projected onto the $\vec{\omega}$-plane. The light intensity value at the projected observer point (*i.e.* $\mathcal{L}(\vec{q}, \vec{\omega})$) is estimated by one of the blending algorithms, taking into account all $\mathcal{L}(\vec{q}, \vec{\omega})$ values or only a subset of them. In the given example, each of the seventeen contributing camera positions shown with bold perimeter in Fig. 3(b) provides an intensity value which is observed into direction $\vec{\omega}$ for observer position $\vec{q} = 0$. The observer is located in the center of the sphere and indicated by a bold dot. A single intensity value is resolved among the contributing intensities through a two-dimensional interpolation on its respective $\vec{\omega}$-plane.

A. Current Blending Algorithms

Applying the nearest neighbor technique in the second reconstruction step is explained in [14]. The light intensity at the virtual observer point for each $\vec{\omega}$ direction is set to the light intensity value of the best observing camera for the $\vec{\omega}$ direction. The nearest neighbor technique is expressed in (1) in mathematical terms:

$$\begin{aligned} j &= \text{argmin}_{i \in \mathcal{I}}(r_i) \\ \mathcal{L}(\vec{q}, \vec{\omega}) &= \mathcal{L}(c_j, \vec{\omega}) \end{aligned} \tag{1}$$

where $\mathcal{I} = \{i | \vec{\omega} \cdot \vec{t_i} \geq \cos(\frac{\alpha_i}{2})\}$ is the index set of contributing cameras for the pixel direction $\vec{\omega}$. A pixel direction $\vec{\omega}$ is assumed observable by the camera c_i if the angle between its focal vector $\vec{t_i}$ and the pixel direction $\vec{\omega}$ is smaller than half of the minimum angle of view α_i of camera c_i. The length

r_i identifies the distance between the projected focal point of camera c_i and the projected virtual observer point on the $\vec{\omega}$-plane. The camera with the smallest r distance to the virtual observer projected point on the $\vec{\omega}$-plane is considered the best observing camera. For example, such distance is identified with r_A and depicted by a dashed line for the contributing camera A in Fig. 3(b).

The linear blending scheme incorporates all the contributing cameras through a linear combination. This is conducted by aggregating the weighted intensities of the contributing cameras. The weight of a contributing camera is the reciprocal of the distance between its projected focal point and the projected virtual observer point on the $\vec{\omega}$-plane (*i.e.* r). The weights are also normalized to the sum of the inverse of all the contributing cameras distances.

The linear blending is expressed in (2) in mathematical terms.

$$\mathcal{L}(\vec{q},\vec{\omega}) = \frac{\sum\limits_{i\in\mathcal{I}} \frac{1}{r_i}\mathcal{L}(c_i,\vec{\omega})}{\sum\limits_{i\in\mathcal{I}} \frac{1}{r_i}} \tag{2}$$

B. Vignetting Correction and Gaussian Blending

Vignetting is an adverse effect observed in cameras, where the pixels located close to the image frame borders are significantly darker than the pixels located in the center. Vignetting also affects the reconstructed omnidirectional view; thus, pixel intensities in the reconstructed image alternatively vary, *i.e.* certain regions are darker and others are brighter.

Several methods are proposed for modeling the vignetting effect and its correction. The chosen model for the Panoptic camera is the Kang–Weiss model [17]. The Kang–Weiss model takes into account the pixel position in the camera image frame, the camera focal length and a camera constant named the vignetting factor. All pixels in each camera frame are corrected by multiplying the sampled pixel intensity with the factor given by:

$$I'(u,v) = I(u,v)(1-\alpha d)\frac{1}{(1+(d/f)^2)^2} \tag{3}$$

where α is the vignetting factor, f is the focal length, $I(u,v)$ is the pixel intensity at coordinates (u,v) and $d=\sqrt{u^2+v^2}$.

Even though the vignetting correction equalizes the brightness of the images across the cameras, the reconstructed image quality mostly depends on the blending algorithm. The presented algorithms have several disadvantages. An image reconstructed using the nearest neighbor method shows clear boundaries between the fields of view of different cameras. Although any brightness difference is reduced by the vignetting correction, the boundaries are still visible and create an unpleasant effect to the human eye. An example of such an image is shown in Fig. 5(a). On the other hand, linear blending resolves the problem of sharp boundaries to a certain extent. The pixels in the regions where cameras' fields of view overlap are blended using a weighted average, as expressed in (2). The intensity difference is reduced, but it still exists due to the linearly chosen weights as shown in Fig. 5(b).

Distributing the weights according to a Gaussian function with respect to a pixel distance appears to be an appropriate solution to limiting the brightness difference. The new weights in the weighted average expression are:

$$\begin{aligned} w(r_i,d_j) &= \frac{1}{r_i}\cdot\mathcal{G}(d_j,\sigma_d) \\ \mathcal{G}(d_j,\sigma_d) &= e^{-\frac{d_j^2}{2\sigma_d^2}} \end{aligned} \tag{4}$$

where r_i is the same distance as in (2), d_j is the distance of the j^{th} pixel in the camera image frame from its center and σ_d is the variance of the Gaussian distribution function.

By adding the Gaussian factor to the weighted average expression, the borders between cameras are not visible any more, as shown in Fig. 5(c). The variance is set to $\sigma_d = 100$. The Gaussian factor reduces the difference in brightness in the images from different cameras and the overlapping regions are now equalized with their respective surrounding.

III. Hardware Implementation

A custom FPGA board has been designed with a XILINX Virtex5 XC5VLX50-1FF1153C FPGA as a core processing unit in order to capture and process the video streams produced by the cameras in real-time. This board interfaces with twenty PIXELPLUS PO4010N single-chip Common Intermediate Format (CIF, 352×288) cameras with 66° minimum angle of view. The cameras of the Panoptic system have been calibrated for their true geometrical position in the world space, and lens distortion parameters are obtained. The extraction of their intrinsic parameters is also done *a priori*. To support higher number of camera interfaces, multiple identical boards of the same kind are stacked. For scalability and extension purposes, the designed board also contains high-speed Low-Voltage Differential Signaling (LVDS) serial links and extension connectors. The board is also equipped with a Universal Serial Bus (USB) 2.0 device chipset for external access and high-speed data transfer. The FPGA board contains two Zero Bus Turn around (ZBT) Static Random Access Memories (SRAM) with 36 Mb capacity and an operating bandwidth of 166 MHz, for each.

A. FPGA Architecture

The architecture of the FPGA is depicted in Fig. 4(a). The FPGA design consists of five major blocks. The arrow lines depicted in Fig. 4(a) show the flow of image data inside the FPGA. Image data streaming from the cameras enters the FPGA via the Camera input channel block. A time-multiplexing mechanism is implemented to store the incoming frame data from all the camera modules into one of the single data port SRAMs. Hence, the Data transmit multiplexer block time-multiplexes the data received by the Camera input channel block and transfers it to the Memory controller block for storage in one of the SRAMs. The SRAMs are partitioned into twenty equal segments, one for each camera. The Memory controller block interfaces with two external SRAMs available on the board. The Memory controller block provides access

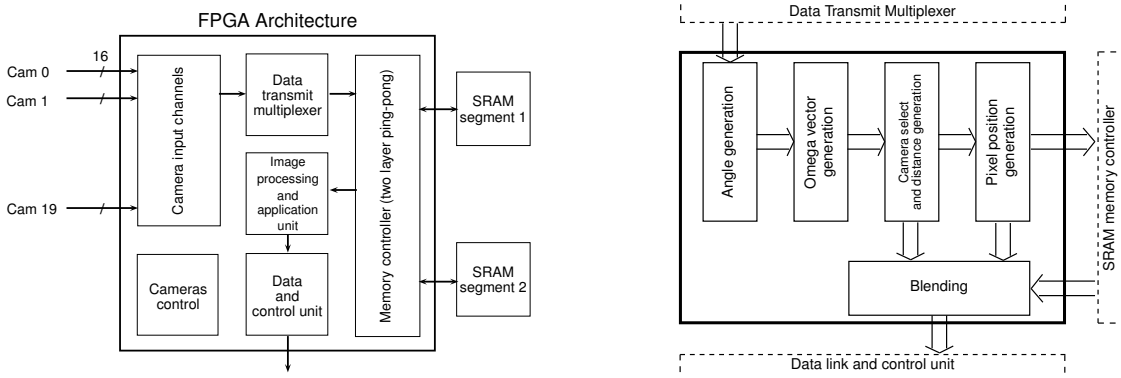


Fig. 4. (a) System-level architecture, (b) block diagram of the light field reconstruction unit inside the Image processing and application block.

for storing/retrieving the incoming/previous twenty frames in/from the SRAMs. The SRAMs swap their roles (*i.e.* one is used for writing and one for reading) with the arrival of each new image frame from the cameras. The Image processing and application unit block is in charge of signal processing and basic functionalities such as single video channel streaming, all channels image capture and omnidirectional view reconstruction. The Image processing and application unit block accesses the SRAMs via the Memory controller block and transfers the processed image data to the Data link and control unit block. The Data link and control unit block provides transmission capability over the external interfaces available on the board such as high-speed LVDS serial links or the USB 2.0 link. The Cameras control block is in charge of programming and synchronizing the cameras connected to the FPGA board.

B. Light Field Reconstruction Hardware

The light field reconstruction algorithm is implemented inside the Image processing and application unit. The block diagram is shown in Fig. 4(b). This image processing entity comprises five modules. The Angle generation module generates the spherical coordinates, *i.e.* $(\theta_\omega, \phi_\omega)$, of the $\vec{\omega}$ directions which are of interest for the reconstruction. The span and resolution of the output view is selectable within this module. It is possible to reconstruct a smaller portion of the light field with an increased resolution, due to the initial oversampling of the light field (*i.e.* the cameras record more samples than the reconstructed image has). Hence, a more detailed image with a limited field of view can be reconstructed while keeping the same frame rate. Such images are shown in Fig. 6. The maximum reconstruction resolution supported by the hardware is 32M pixels. Higher resolutions are achieved trading off the frame rate.

The Omega vector generation module calculates the unit radial vector pertaining to the spherical positions $(\theta_\omega, \phi_\omega)$ received from the Angle generation module. The Camera select and distance generation module identifies which cameras contribute (*i.e.* observe) to the construction of the pixel in $\vec{\omega}$ direction. The index of the contributing cameras are passed to the Pixel position generation module. Concurrently, this module computes the distance between the focal point projection and the virtual observer projection on the $\vec{\omega}$-plane, for each contributing camera in direction $\vec{\omega}$.

The Pixel position module calculates the true pixel position in the image frame of the cameras selected in the Camera selection block. The same block interfaces with the SRAM memory controller to retrieve the pixel value of the contributing cameras upon calculation of the true pixel position. The camera index originating from the Camera select module is used to access the correct segment of the SRAM where the image frame of a camera is stored. The true pixel position is used to access the target address within each segment inside the SRAM. The Pixel position module also calculates the distance of the selected pixel in the image frame from the image center. The distance is further used for image blending.

The Blending module receives the pixel light intensity values from all contributing cameras along with the two distances explained above. The distances are used to calculate the contributing weights using the expression (4). The final weight is the reciprocal value of the distance between the projected focal point and the projected virtual observer point on the $\vec{\omega}$-plane for each contributing camera multiplied by the pixel dependent Gaussian factor. The weights are used in the blending process of the second reconstruction step mentioned in Section II. Finally, the block estimates a single light intensity value for each $\vec{\omega}$ direction.

C. Scalability

Each FPGA board can interface with twenty cameras. In order to supporta higher number of cameras and increase the throughput of the Panoptic camera, multiple FPGA boards must be incorporated. Hence, the omnidirectional view reconstruction workload is distributed and the algorithm operates in parallel on all FPGA boards. Thus, a central FPGA is required to receive the output data from the FPGA boards, apply the final blending process and transfer the result to a PC for display. The processing part of the central FPGA only contains the Blending module. It calculates the final pixel values based on the pixel values and the weights calculated in the first stage FPGAs.

(a)

(b)

(c)

Fig. 5. A computer laboratory at the Swiss Federal Institute of Technology in Lausanne (EPFL, ELD227). Panoramic construction with a pixel resolution of $N_\phi \times N_\theta = 1024 \times 256$ (a) using the nearest neighbor technique, (b) using linear blending and (c) using Gaussian blending with $\sigma_d = 100$.

IV. Results

The Panoptic system with thirty embedded cameras presented in [14] is used for real-time image extraction and evaluation. The thirty camera system contains two FPGA boards for camera interfacing and one central FPGA. The FPGA device utilization summary of a single board is stated in Table I.

TABLE I
FPGA device utilization summary

Resources	Used	Available	Utilization
Occupied Slices	4223	7200	58%
Slice Registers	10076	28800	34%
BlockRAM/FIFO	23	48	47%
DSP48Es	48	48	100%

The operating frequency of the design implemented in the FPGA is 133 MHz. Hence, the output video streaming rate of each FPGA board is 6.6M pixels per second. Using this frequency, the total latency of the system is less than 1 μs. The power consumption of each FPGA board in operation is 5W.

Three captured snapshots of the same scene from the real-time output (*i.e.* 25 frame per second) of the Panoptic device with thirty embedded cameras are shown in Fig. 5. The horizontal and vertical directions in the shown panoramic constructions correspond to ϕ and θ spherical coordinates, respectively. Fig. 5(a) corresponds to the panoramic scene constructed for a virtual observer located at the center of the sphere using the nearest neighbor technique. No gain compensation or radiometric calibration has been used for the cameras. Hence, the boundaries between the cameras are apparent and high intensity changes are visible in Fig. 5(a). The linear blending technique improves the color intensity

Fig. 6. Detailed image parts obtained using grid refinement: a lamp magnified 8x, the books magnified 32x, a desk magnified 8x.

variations as observed in Fig. 5(b) and provides a scene with less sharp color transitions. The linear blending technique also results in a "ghosting" effect altering the objects that are close to the Panoptic system and that manifests itself as the duplication of their edges. Fig. 5(c) shows the omnidirectional view reconstruction using the Gaussian blending. The color transitions in the overlapping regions are significantly reduced. This effect is due to the applied vignetting correction and the Gaussian factor. The Gaussian blending is not a filtering operation, thus it does not influence the image sharpness as it only affects the inter-camera brightness differences.

Fig. 6 depicts the ability of the Panoptic to refine a pixelization grid in a selected portion of space. This results in increased detail, while keeping the same output image resolution. In Fig. 6 a lamp, a desk and books are shown in increased resolution. The quality of the magnified parts is proportional to the number of observing cameras.

V. Conclusion

The Gaussian blending technique for the omnidirectional view reconstruction of the Panoptic camera is discussed. The introduced Gaussian blending algorithm decreases the high light intensity variations in the reconstructed image. The applied vignetting correction further improves the output image quality. However, the "ghosting" effect for the close objects still exists. To further improve the output of the Panoptic camera, the real-time implementation of the multi-band blending technique [18], [19] is being considered. The architecture of an FPGA based system for the real-time deployment of the reconstruction algorithm is presented. Snapshots of the real-time output of the Panoptic system have been displayed. Furthermore, the ability to display image parts in more detail is also presented. Future work related to the Panoptic device focuses on real-time 3-D cinematography, high resolution real-time light field reconstruction and Application-Specific Integrated Circuit (ASIC) design of the current system.

Acknowledgment

This research has been partly conducted with the support of the Swiss NSF under grant number 200021-125651. The authors gratefully acknowledge the support of Xilinx, Inc., through the Xilinx University Program.

References

[1] R. Szeliski, "Image Mosaicing for Tele-Reality Applications," in *Proc. of the Second IEEE Workshop on Applications of Computer Vision*, dec 1994, pp. 44 –53.

[2] S. Mann and R. W. Picard, "On Being 'Undigital' with Digital Cameras: Extending Dynamic Range by Combining Differently Exposed Pictures," in *Proceedings of IS&T*, 1995, pp. 442–448.

[3] P. E. Debevec and J. Malik, "Recovering High Dynamic Range Radiance Maps from Photographs," in *Proc. of the 24th Conf. on Computer Graphics and Interactive Techniques*, New York, NY, USA, 1997, pp. 369–378.

[4] P. Rander, P. J. Narayanan, and T. Kanade, "Virtualized Reality: Constructing Time-Varying Virtual Worlds From Real World Events." in *IEEE Visualization'97*, 1997, pp. 277–284.

[5] M. Levoy and P. Hanrahan, "Light Field Rendering," in *Proc. of the 23rd annual Conf. on Computer Graphics and Interactive Techniques*, New York, NY, USA, 1996, pp. 31–42.

[6] Y. Schechner and S. Nayar, "Generalized Mosaicing," in *IEEE Inter. Conf. on Computer Vision (ICCV)*, vol. 1, Jul 2001, pp. 17–24.

[7] D. Taylor, "Virtual Camera Movement: The Way of the Future?" *American Cinematographer*, vol. 77, no. 8, pp. pp. 93–100, 1996.

[8] C. Zhang and T. Chen, "A Self-Reconfigurable Camera Array," in *Eurographics Symposium on Rendering*, 2004, pp. 243–254.

[9] J. C. Yang, M. Everett, C. Buehler, and L. McMillan, "A Real-Time Distributed Light Field Camera," in *Proc. of the 13th Eurographics workshop on Rendering*, Aire-la-Ville, Switzerland, 2002, pp. 77–86.

[10] B. Wilburn, N. Joshi, V. Vaish, and et al., "High Performance Imaging Using Large Camera Arrays," *ACM Trans. Graph.*, vol. 24, pp. 765–776, July 2005.

[11] D. J. Brady, M. E. Gehm, R. A. Stack, D. L. Marks, D. S. Kittle, D. R. Golish, E. M. Vera, and S. D. Feller, "Multiscale Gigapixel Photography," *Nature*, vol. 486, no. 7403, pp. 386–389, 2012.

[12] Y. Yagi, "Omni Directional Sensing and Its Applications," *IEICE Trans. Inform. and Systems*, vol. E82-D, no. 3, pp. 568–579, March 1999.

[13] J. de Villiers, "Real-time photogrammetric stitching of high resolution video on COTS hardware," in *Proc. of the Int. Symp. on Optomechatronic Technologies*, vol. 9, 2009, pp. 46–51.

[14] H. Afshari, L. Jacques, L. Bagnato, and et al., "The PANOPTIC Camera: A Plenoptic Sensor with Real-Time Omnidirectional Capability," *Journal of Signal Processing Systems*, pp. 1–24.

[15] ——, "Hardware Implementation of an Omnidirectional Camera with Real-Time 3D Imaging Capability," in *3DTV Conf.: The True Vision - Capture, Transmission and Display of 3D Video*, May 2011, pp. 1–4.

[16] H. Afshari, A. Akin, V. Popovic, A. Schmid, and Y. Leblebici, "Real-Time FPGA Implementation of Linear Blending Vision Reconstruction algorithm using a spherical light field camera," in *IEEE Workshop on Signal Processing Systems*, Quebec City, Canada, October 2012.

[17] S. B. Kang and R. S. Weiss, "Can We Calibrate a Camera Using an Image of a Flat, Textureless Lambertian Surface?" in *Proc. of the 6th European Conf. on Computer Vision - Part II*, 2000, pp. 640–653.

[18] M. Brown and D. Lowe, "Automatic Panoramic Image Stitching Using Invariant Features," *International Journal of Computer Vision*, vol. 74, no. 1, pp. 59–73, Aug. 2007.

[19] P. J. Burt and E. H. Adelson, "A Multiresolution Spline with Application to Image Mosaics," *ACM Trans. Graph.*, vol. 2, no. 4, pp. 217–236, Oct. 1983.