Presented at the 6th Workshop on Omnidirectional Vision, Camera Networks, and Non-Classical Cameras (OMNIVIS 2005)

Multi-Viewpoint Uncompressed Capture and Mosaicking with a High-Bandwidth PC Camera Array

H. Harlyn Baker and Donald Tanguay
Hewlett-Packard Laboratories
Palo Alto, CA

Constantin Papadas
Integrated Systems Development, S.A.
Athens, Greece

Abstract

A newly developed high-performance multi-imager VGA camera array is described. Designed for synchronization, bandwidth, and economy within a videoconferencing application, and demonstrated initially through novel video mosaicking, the system shows promise for accelerating the new field of multi-viewpoint imaging. We describe the camera system's design, control, performance characteristics, use in the mosaicking application, and its commercialization path for community use.

1. Introduction

The processing of images in computer vision has developed from early days of single frame, through binocular stereo, multi-viewpoint stereo, streaming video, and up to streaming multi-viewpoint (MV) capture. Each of these source types places particular computational demands on their processing, and imposes corresponding restrictions on the character of the observations that may be made upon them. It can be argued that the less data available, the more difficult becomes reliable interpretation. Single views provide the least potent information, while streaming multi-viewpoint video provides the most. Note that our own human visual processing capabilities result from streaming binocular vision. This is the base imagery level we seek in our quest for both machine visual understanding and human-perception-quality interactive display.

This may be the decade of multi-viewpoint video. Numerous efforts are underway around the world in developing technologies that will support 3D TV and related uses of this imaging. 3DAV [19] is an international MPEG working group aiming to develop standards for three-dimensional audio-visual multi-viewpoint image transmission. Their concerns are broad, covering 3D sound and video for free-viewpoint television. The European Union supports two consortia of companies and academic sites: 3DTV, whose goal is integrating 3D technologies for general access, such as over the internet [11], and ATTEST, aimed at developing a backwards-compatible 3D TV capability using video supplemented by a synchronized depth layer to bring 3D TV to the home [12]. This technology is near ready for commercialization.

While having numerous insights on methods for coding and efficiently transmitting multiview-point video, these efforts uniformly lack a source for delivering adequate live imagery to a single PC. The challenges here lie in several directions, including: developing mechanisms on consumer-grade PCs to support the needed bandwidth; controlling the various cameras for synchronized acquisition so scene sampling does not degrade from multi-viewpoint into multi-time; and retaining as much of the signal quality as is needed for subsequent image analysis (lossless transmission is likely required).

Getting Bandwidth. PC I/O bandwidth has been generally insufficient for supplying the required level of image acquisition, with capture typically limited to the 1394a standard (400 megabits per second) or, more recently, 1394b Firewire (800 Mbps) protocols. CameraLink, a specialized bus architecture used for high resolution and high framerate cameras, has not been used as yet for multiple camera systems. A popular 1394a capture system from Point Grey Research (Dragonfly [10]) is typical of what is available, offering a maximum of four simultaneously active 30 Hz VGA (640 x 480) Bayer-format streams per 1394 card (the newer 1394b allows double this number). Multiple 1394 cards may be ganged onto a single PC for increased throughput, but synchronization becomes an issue, and the bandwidth limitation then moves to the PCI bus on which the grabber cards are situated. Most 1394 cards work on a 32bit/33MHz PCI bus which, for 8.8 MB/second Bayer-format VGA data, supports at most 15 video streams. These throughputs presume full utilization of the shared PCI bus; a more reasonable assumption is 50%, or 7 Bayer VGA cameras.

harlyn.baker,donald.tanguay@hp.com;
papadas@isd.gr

YUV422 format drops this to 3 and RGB brings it down to 2.

If color is not needed on the PC, can be converted later (such as at display time), or if the user wishes to do something more specialized with the components rather than use the conversion provided by the camera manufacturer, then the clear choice is to transmit Bayer format data.

While 4-to-8 VGA streams may seem an adequate basis for research developments, it prevents the insights that come from pressing against the data of real-time full-framerate MV applications. These applications, where observations from numerous perspectives are required, include those where computers are mediating with interacting people, large areas are to be covered in 3D and/or from multiple viewpoints, and latency and inaccuracy can limit the value of the experience.

1394a Firewire was available at 400Mbps in 1997 when 800Mbps 1394b design was begun, with 1998 deployment promised. Eight years have passed, and 1394b is only now beginning to be available on grabbers and cameras. The competition with USB2.0 has slowed the development of Firewire, and arrival of the promised higher bandwidths (up to 3200Mbps) is questionable. It seems that those wishing significantly higher bandwidths than these consumer-level standards will have to look elsewhere, for the present.

Multi-Viewpoint Capture. Approaches to MV capture have ranged from emulation, where a single camera is successively repositioned for capture from differing viewpoints [4, 16, 17], to typical stereo acquisition where two cameras work in unison, to high-quantity arrays for synchronized capture of compressed video [23]. Between are offline systems for multi-camera recording [14], systems for selective grabbing of video from a large array of available streams or using multiple PCs [8, 18, 25], those that remain in the analog domain [9], and novel single imager devices that use lenslets to synthesize an imager array [21].

Our laboratory has been investigating advanced videoconferencing for several years—initially with a focus on 3-D acquisition and display [1], and currently working toward high-resolution and high-bandwidth interaction. The issue of adequate capture quantities and rates has been with us from the beginning. To enable high bandwidth capture for experimentation and eventual deployment, HP, in collaboration with ISD S.A, Greece, has developed a system delivering uncompressed video from 24 synchronized VGA cameras to PC memory with no CPU overhead. The combination of lots of cameras, uncompressed VGA video, framerate operation, and synchronized capture to a single PC is unique, and brings a new level of possibility to multi-viewpoint image processing. Our intention is to use the system for multi-viewpoint studies, and to make it available to the community for further advancement in these areas.

2. The Capture System

The Herodion camera system (named after an amphitheater at the base of the Parthenon) is a high end multi-camera CMOS capture system built around a direct memory access (DMA) PCI interface that streams synchronized video to PC memory through a three-layer tree structure: a camera layer, a concentrator layer, and a frame grabber layer. Cameras, grouped in 6's (see Figure 1), are attached to *leaf* concentrators. Two leaf concentrators connect to a *middle* concentrator, up to two of which can connect to the PCI-bus *frame grabber*. Different configurations are supported, down to a single camera.

The camera head consists of three small boards stacked in a 30x30x30 mm configuration (Figure 2). These boards contain a color VGA CMOS sensor (VVL6500), a co-processor, and a low-voltage differential-signalling (LVDS) chip interfacing to the rest of the system. It is connected to a concentrator board through a 10-wire UTP connector over an LVDS bus that carries captured video data, control signals, and power (LVDS is used in flat panel digital displays and CameraLink communication [7]). An FPGA layer is available for head-level processing.

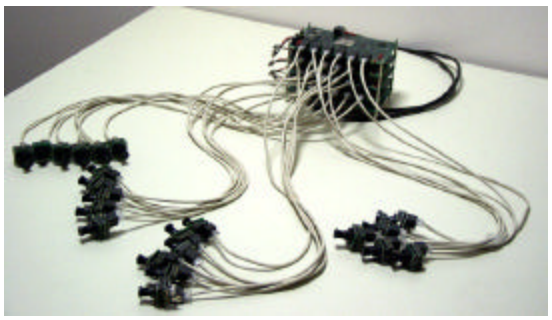

Figure 1: Twenty-two camera multi-imager system

Figure 2: CMOS imager head

The hardware layers (Figure 3) have different bandwidths, with the cameras using 18-bit LVDS operating at 192Mbps, the concentrators (accepting either cameras or other concentrators) joined by 16-bit 768Mbps LVDS, and a frame grabber tethered to its concentrators with two 1.15 Gbps 16-bit LVDS ports, giving a total of 2.3 Gbps to the PC. The concentrators and frame grabber use Altera Flex FPGAs, and the frame grabber is built on a QuickLogic 5064 64-bit DMA engine designed for a 64-bit PCI bus. This bus is the start of a range termed PCIx, with speeds of 33MHz and up. Our interface card currently operates at 33MHz on this bus, with theoretic utilization of about 0.25GB/sec. Redesign for PCI-express (PCIe, up to 16x PCI speeds, or about 4.4GB/s) is planned, which should give us the possibility of running almost 350 simultaneous synchronized VGA streams. While it is clear that some redesign in the data path would be needed to reach this figure, the grabber architecture is scalable, and future implementations will provide additional input ports to support a larger number of concentrators.

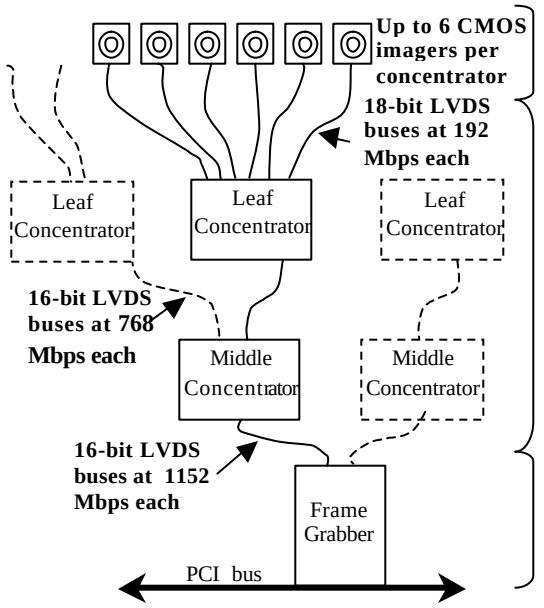

Figure 3: Layered architecture

The efficiency of the DMA can be observed in considering that CPU load is zero during video capture, even when all system components are active delivering 30 frames per second per camera to the PC main memory.

A special characteristic of the system interface—appealing to those who have worked with camera arrays—is its operational simplicity. I2C communication runs between all devices over the LVDS buses, with imagers controlled individually or in a broadcast mode. Bypassing the usual juggling needed to synchronize and organize time-stamped data, a Herodion image grab returns the data from all active cameras simultaneously, in interleaved or disjoint form, into a pre-allocated triple-buffer ring.

3. Multi-viewpoint Video Mosaicking

The camera array was designed for multi-view processing tasks including 3D reconstruction [4], image based rendering [1], and free-viewpoint and 3D TV [19]. Before applying the system in these areas, we have selected a panoramic video-conferencing capture task to demonstrate early capabilities. Here, we combine the observations from several imagers to emulate the performance of a much costlier multi-megapixel wide-angle video camera.

Our requirements differ from those of more standard mosaicking. While others address the stitching of images from nearby but arbitrary viewpoints (such as in handheld capture), we configure a set of cameras and wish to determine the stitching parameters once, before use, and then apply them repeatedly. These parameters will map all images to a common frame, composing a single wide-field-of-view image.

Figure 4: Video mosaicking 3-VGA camera.

Figure 5: Panoramic camera configurations: (a) 1 MPixel, (b) 2 MP, (c) 6MP

Line-based Calibration. One of our goals has been to build the high-resolution wide-angle panorama from a minimum number of standard resolution cameras. This drives our analysis since, for a particular resolution and field of view, minimizing cameras means minimizing the overlap among them. Traditional mosaicking involves finding corresponding *points* in the contributing images. These determine the homographies that relate their image perspectives [20], and typically require up to 50% image overlap. Our approach has been to develop a solution using not point features but *lines*. This possibility arises from the duality between lines and points on the projective plane. For homography H relating two images I and I' with corresponding points x and x' and lines l and l', we have:

$$x' = Hx$$
$$l' = H^{-T} l \qquad (1)$$

The line homography H^{-T} is the inverse transpose of the point homography H, and we are free to solve either way [13]. Since lines extend across several images, their use removes the requirement of having significantly overlapping imagery for producing the correspondences—images can even be disjoint.

For lines defined in Hessian normal form ($ax + by - c = 0$), constraints for line equations l and l' in two images—$l = (a,b,c)$ and $l' = (a',b',c')$—enter the optimization as:

$$\begin{bmatrix} 0 & a'c & -a'b & 0 & b'c & -b'b & 0 & c'c & -c'b \\ -a'c & 0 & a'a & -b'c & 0 & b'a & -c'c & 0 & c'a \end{bmatrix} \tilde{h} = 0 \qquad (2)$$

where $\tilde{h}$ is a linearized version of the line homography H^{-T}. There is a pair of such constraints for each line observed in two images.

This use of lines for transform estimation has sufficiency conditions similar to that of points: collinear lines are like coincident points, parallel lines are like points that are collinear, and a minimum of four lines in general position are needed to form an homography with 8 degrees of freedom (for stability, we use many more lines). The extended spatial support of line fitting presents an added advantage over a point-based solution in that localization of a line is more robust. This means that, when presented with the same number of observations of lines or points, we can expect better estimates from the lines [2].

Setting Up the Solution. The camera array is aimed toward a large projected display that fills its field of view. Images of linear patterns are presented on the display and acquired by the individual cameras of the array. These are corrected for lens distortion (also determined using projected lines), parameterized by a least-squares fitter, and passed to the homography solver. All lines are placed in the constraint equation (2) for each pair of cameras. We precondition the solution by performing the common normalization of the constraint matrix [13], solve for the desired homography, then denormalize the result. An initial solution is formed with a predetermined pattern of lines, then lines are presented selectively to optimize the solving.

Paired Image Solution. A pairwise hierarchy can be defined over camera adjacency relationships, with each camera being related to a chosen reference camera through a chain of homographies. This makes for simple and effective mosaicking with 1-D linear camera arrangements (the 3-camera system of Figure 4 and 5a). 2-D layouts, (Figures 5b and 5c), cannot be well solved with paired-image constraints alone. Figure 7 shows the paired-image adjacency relationships used in the solution that gives us the mosaic of Figure 6 (imager 9 is the reference frame). The constraints follow the links, i.e., there are none between 0-1, 1-2, etc. Gaps are present where the imagers' fields of view do not overlap.

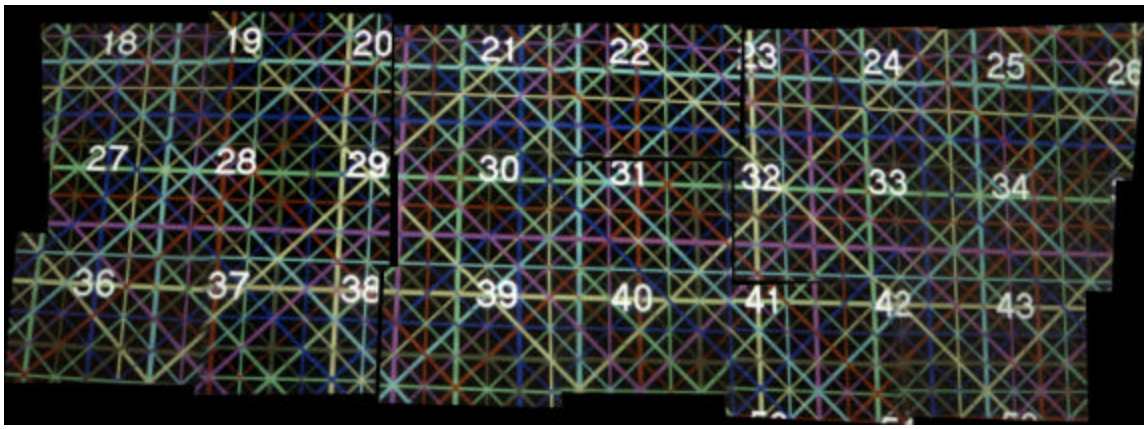

Figure 6: Local constraints over 3x6 camera array (notice errors at "23" and beneath "37")

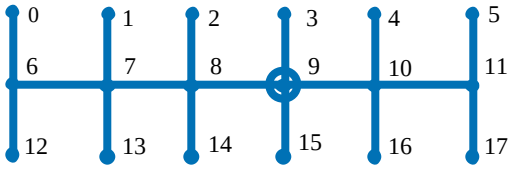

Figure7: Adjacency structure for computing homographies of Figure 6

Global Solution. The solution with this structure is local and piecewise. It is clear that we must incorporate higher-connectivity relationships in our solving so that everywhere that images abut, they are constrained. We have developed a linear solution to this that capitalizes on higher-order relationships among imagers. Where a cycle of length 3 exists between a set of adjacent imagers (Figure 8, left), we enforce the implied mutual constraints. This binds the unconnected relationships of Figure 7. Figure 8, right, shows the constraint involving quad relationships. Figure 9 indicates the improvement that comes from use of these higher degree relationships. The paired-image mosaicking is underconstrained, causing the tears, but the increased degree of solution improves the results.

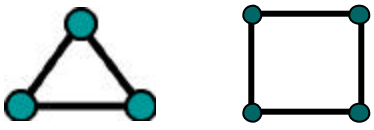

Figure 8: Higher-order constraints (degrees 3 and 4)

When moving up to a global solution, our line-based mosaicking has added advantage over those involving point features: Since lines can extend across the full field, camera views need not be adjacent to share observations of a line. Our next step is to use these extended constraints in global optimization that enforces line continuity over the full image set.

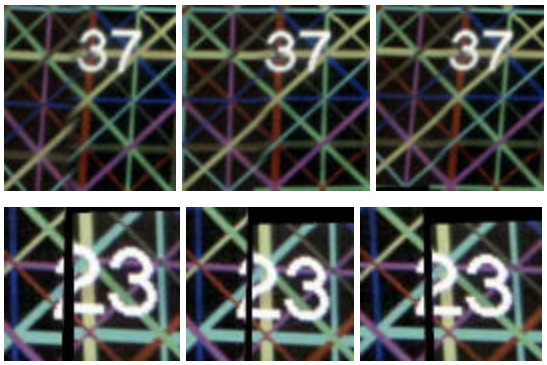

Figure 9: Increasing constraints 2-, 3-, and 4-degree

Constructing the Images. In building the mosaic, we select a base frame of reference, such as the one with minimal RMS distance to all others, determine a cropping of the resultant image, and compute a mapping table of values for filling this image from its contributing components. The table carries bilinear interpolation indices and weights for composing each destination pixel. The indices map resultant pixels through each imager homography back to their appropriate source image, mapping the selected point to account for any observed lens distortion.

By design, the vast majority of pixels will be seen by only one imager; where several imagers see a pixel, their contributions must be blended. Our metric for this blending is simple bilinear interpolation – regions of overlap are determined, and pixel weights are computed by distance from that source image's boundaries. Figure 4a showed the source images contributing to the desired

composite mosaic of Figure 4c. Figure 4b showed this view without blending. Good geometric and color registration suggest more complex blending (such as [6]) may not be necessary.

Photometric Correction. Before processing our video streams, we present a color chart to the individual cameras. Detection of the calibrated elements of this chart lets us compute transform matrices that map each camera's color output for agreement with measured values. Figure 10 shows an image as originally sampled, and its correction following this analysis. At image boundaries, luminance vignetting will give the resulting frame a patchwork appearance, so we need to adjust here as well. Wider-field color blending is needed to prevent color discontinuities at image seams [5].

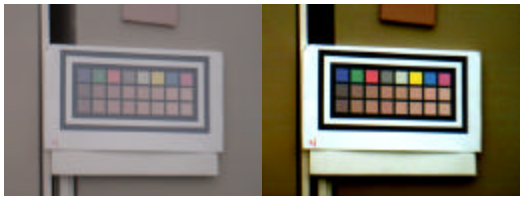

Figure 10: Original sampling; Color corrected

Exploiting the GPU. We use the PC's graphics processing units (GPUs) for the image resampling, treating vectors derived from the resampling transforms as static image *geometry* and the streaming video as dynamic *textures*. By downloading the geometry (that reshapes the source images), we can stream the subsequent video to the display and let the GPU hardware handle its compositing and blending [3, 24]. Color conversion from Bayer to RGB and color correction (a 3x4 matrix multiply at each pixel) are both done on the GPU. These reduce the bandwidth burden of the PCIe bus by a factor of 3, and remove the color computation from the CPU. In this configuration, the CPU has almost no role in the capture-to-mosaic display, and is free for other tasks.

Mosaicking Inaccuracies. When mosaicking cameras are colocated—having the same center of projection (as they would using mirrors in a catadioptric arrangement)—they share the same viewpoint and images can be tiled without error, otherwise the images contain disparate information. For any plane in the world, however, there exist mappings that will permit multiple cameras to construct a single integrated view of the features in that plane without error. These mappings are the linear homographies we compute. The plane is both our reference for calibration and where we expect our subjects to be located for correct mosaicked display.

In this context, error in the image mosaicking refers to the blur observed when features in the combined view are not on that homography plane, so produce double images. We can examine the impact of this off-plane imaging by reviewing the relationships among the geometric parameters that affect the process.

Understanding Double Images. The homography (H) between two images maps the content of one so that features that lie on the reference plane coincide when combined – they have zero disparity. All points on that plane will be combined through H to have this zero disparity. Figure 11 sketches the situation for two cameras, C_1 and C_2, viewing point P (on plane $?$) at P_1 and P_2, respectively (which are defined relative to their image centers). The effective disparity from the two viewpoints is $D = P_1 - P_2$. For a baseline B, with P at a distance Z, and cameras with focal length f, similar triangles tell us that, with $B\text{-}D = B\text{-}(P_1 - P_2)$, $B/D = Z/f$, or $D = Bf/Z$. H will bring P_1 and P_2 to the same point in the combined image.

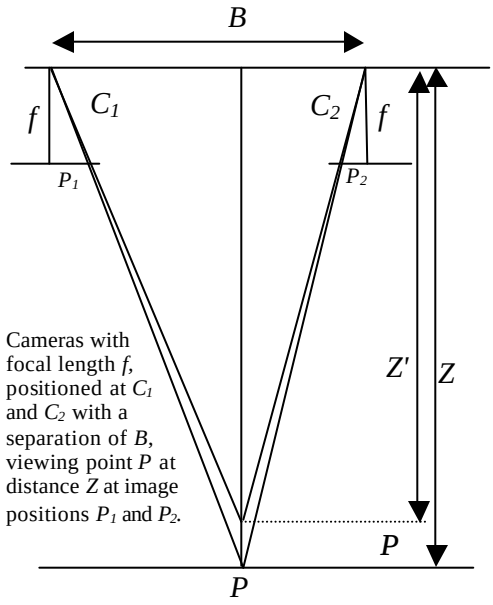

Figure 11: Mosaic rectification imaging geometry.

The converse of having any point on $?$ having zero disparity when transformed by H is that any point not on $?$ will have a non-zero disparity, i.e., it will appear not coincident but blurred. If we change the distance to Z', we have $D' = Bf/Z'$, and the blur we observe will arise from the change in disparity, $d = D - D'$, as we vary Z'. Notice $D' = D = 0$ when $Z' = Z$ for a homographically mapped

image—features have no disparity on the reference plane.

For: $D_1 = Bf/Z_1; D2 = Bf/Z_2$
We have: $d = D_2 - D_1 = Bf/Z_2 - Bf/Z_1$

Herodion Geometry. Our Herodion imagers are about 30mm apart. A standard figure of merit for resolution of displayed imagery is one part in 1000 [15]—a milliradian. For a 720 pixel-wide display of 45-inch width viewed at ten feet, the milliradian suggests a blur of 1.4 pixels will be imperceptible for human viewers. For six 25mm mosaicked imagers covering 60 degrees horizontally, we can expect no detectable blur within about a foot of the plane.

With the control over focal length, field of view, range in the scene, and imager density and size, we anticipate the blur issue not being a major dissuader from use of this technology, when placed in contrast with the cost and flexibility characteristics that this PC-based solution offers. Note that, while the homography is formed with respect to a single plane, the plane+parallax technique [22] can be used to alter the location of this plane—dynamically, if one wishes.

Bandwidth Observations. We conducted timing and performance experiments on HP xw9000 series workstations running Windows XP, dual 3+ GHz P5 processors, plenty of RAM, and NVIDIA Quadro FX3400 PCIe graphics boards, using the camera configurations of Figure 5.

Working with the 6-VGA camera (Figure 5b, with 0.88 MB per frame set) at 30 Hz, all-CPU mosaicking ran at 10Hz with 62% CPU utilization. GPU mosaicking used ~0% of the CPU while producing output at full 30 Hz frame rate.

The 3x6-VGA system (5.53MB per frame) ran in CPU-only mode at 4 Hz, with 70% CPU utilization. The GPU version ran at 24Hz (1.06Gb/s) with 50% CPU utilization (presumably from the high data handling and driver effects).

Earlier tests on an AGP bus GPU indicated that the texture mapping was the bottleneck in performance, but this is not the case with the PCIe GPU. Other factors constrain performance, including display configuration and driver options. The implication for us is that display of these high bandwidth data requires careful consideration of the graphics card, driver levels, and use of multiple GPUs.

4 Conclusions

We comment on both our experience with the camera system, and its use in this video mosaicking test application.

Mosaicking on a PC. For the subject distances with which we work – about eight to ten feet from cameras to conference participants – we find the planar homographies to provide quite acceptable video presentation. Some ghosting appears from distant objects that lie away from the calibration plane, but the blending makes them nearly unnoticeable. Few objects lie in the foreground. Review of the images produced by the three-imager mosaicking suggest the results are comparable with those from a broadcast-quality digital video camera costing over an order of magnitude more.

Our point is not to simply increase capture resolution – time will provide that. Whatever the native resolution of acquisition – there will always be interest in acquiring still higher-quality images. The methods we present serve these possible capture scenarios – both now and as imager resolutions rise.

Herodion Multi-imager Camera System. The newly developed camera array provides VGA imagery at the highest data rates of any multi-camera digital video system. Our experiments demonstrate that we can sustain grabbing over extended periods with moderate PCI competition. Our compositing tests showed that the available CPU allowed real time computation on the data.

The mosaicking facility demonstrated here is the first of what we hope to be a series of capabilities exploiting the multiple imager system. In order to increase performance scalability, we are migrating other elements of multi-viewpoint processing, such as lens correction and epipolar rectification, toward the camera heads, and are looking for further gains in GPU directed processing in support of multi-viewpoint display.

Commercialization. The hardware side of this collaboration (ISD, S.A.) are now offering configurations of the current camera system to the community, in size forms of 6, 12, 18 and 24 cameras. We continue to work together on bus and design changes to increase the number of cameras supported, decrease the cost, provide easier control, and enable use of higher resolution imagers.

Acknowledgements. Peter McGuinness of ST-Microelectronics and George Dramitinos of ISD were instrumental in the design and development of this system.

References

[1] Baker, H., Bhatti, N., Tanguay, D., et al, "Understanding Performance in Coliseum, an Immersive Videoconferencing System, *ACM Trans. on Multimedia Computing, Communications and Applications (TOMCCAP)*, 1:2, 2005.

[2] Baker, P., A. Ogale, C. Fermüller, "The Argus Eye", *IEEE Robotics and Automation Magazine*, 11:4, 2004.

[3] Bax, M.R., "Real-time lens distortion correction: 3D video graphics cards are good for more than games," Stanford Electrical Engineering and Computer Science Research Journal, 2004, http://ieee.stanford.edu/ecj/spring04.html.

[4] Bolles, R., H.H. Baker, D.H. Marimont, "Epipolar-Plane Image Analysis: An approach to determining structure from motion," *Intl. Jrnl. Computer Vision*, 1:1, 1987.

[5] Brown, M., A. Majumder, Ruigang Yang, "Camera-Based Calibration Techniques for Seamless Multiprojector Displays," *IEEE Trans. on Visualization and Computer Graphics*, 11:2, 2005.

[6] Burt, P.J., E.H. Adelson, "A multi-resolution spline with application to image mosaics," *ACM Trans. on Graphics*, 2:2, 1983.

[7] CameraLink Spec., http://www.compumodules.com/imageprocessing/euresys/CLinkHomepage.shtml.

[8] Deering, M.F., "High resolution virtual reality", *ACM SIGGRAPH*, 26, 1992.

[9] Dodgson, N.A., J.R. Moore, S.R. Lang., "Time-multiplexed autostereoscopic camera system," *Proc. SPIE Symposium on Stereoscopic Displays and Applications VIII*, 3012, San Jose CA., 1997.

[10] Dragonfly camera system, Point Grey Research, Vancouver, B.C., Canada, http://www.ptgrey.com/products/dragonfly/index.html.

[11] EU Three-dimensional television consortium-3DTV http://www.iti.gr/db.php/en/projects/3DTV.html

[12] Fehn, C., P. Kauff, M.O. De Beeck, F. Ernst,., W. Ijsselsteijn, M. Pollefeys, L. Van Gool, E. Ofek, I. Sexton, "An evolutionary and optimised approach on 3D-TV," in *Proc. International Broadcast Conference*, 2002.

[13] Hartley, R., A. Zisserman, *Multiple View Geometry in Computer Vision*, Cambridge University Press, 2000.

[14] Kanade T, P.J. Narayanan, P.W. Rander, "Virtualized reality: Concepts and early results," *Proc. IEEE Workshop on Representation of Visual Scenes*, 1995.

[15] Kingslake, R., *Optics in Photography*, SPIE Optical Engineering Press, Bellingham, WA., 1992.

[16] Levoy, M., P. Hanrahan, "Light Field Rendering," *ACM SIGGRAPH*, New Orleans, USA, 1996.

[17] Moravec, H.P., "Rover Visual Obstacle Avoidance," *IJCAI-7*, Vancouver, B.C., 1981

[18] Oi, R., T. Hamamoto, K. Aizawa, "Real-time arbitrary-view acquisition system by using random access IBR camera array," *IEEE Conf. Multisensor Fusion and Integration for Intelligent Systems*, Tokyo, Japan, 2003.

[19] Smolik, A., H. Kimata. Report of 3DAV exploration. ISO/IEC JTC1/SC29/WG11 Doc. N5878, 2003.

[20] Szeliski, R., H-Y. Shum, "Creating full view panoramic image mosaics and environment maps," *ACM SIGGRAPH*, 251-258, 1997.

[21] Tanida, J., et al., "Thin observation module by bound optics (TOMBO): an optoelectronic image capturing system," *Proc. SPIE Vol. 4089*, Optics in Computing, 2000.

[22] Vaish, V., B. Wilburn, N. Joshi, M. Levoy, "Using Plane + Parallax for Calibrating Dense Camera Arrays," *IEEE Conf. Computer Vision and Pattern Recognition*, 2004.

[23] Wilburn, B., M. Smulski, H-H. Kelin Lee, M. Horowitz, "The Light Field Video Camera," *Proc. Media Processors*, SPIE Electronic Imaging, 4674, 2002.

[24] Woetzel, J., R. Koch, "Real-time multi-stereo depth estimation on GPU with approximate discontinuity handling," *1st European Conference on Visual Media Production*, London, U.K., 2004.

[25] Yang, J.C., M. Everett, C. Buehler, L. McMillan, "A real-time distributed light field camera,'' in *Proc. 13th Eurographics Workshop on Rendering*, Eurographics Association, 2002.