Optimal Hardware and Software Design of an Image-Based System for Capturing Dynamic Movements

Tara C. Hutchinson, *Member, IEEE*, Falko Kuester, *Member, IEEE*, Kai-Uwe Doerr, *Member, IEEE*, and David Lim

***Abstract*—In contrast to conventional image-capture systems, which attempt to minimize the amount of data collected during capture, typically by using hardware filters, the more general condition of using all information captured on a camera sensor is much more challenging and requires rigorous consideration of the hardware and software pipelines to obtain accurate tracking results. In this paper, this issue is specifically addressed by describing a unique hardware and software design implemented for use as a fully image-based capture system. An attempt is made to minimize the cost of this system by maximizing hardware control through software implementation. The hardware and software requirements are described in the context of the desired high-speed capture suitable for earthquake motions or other dynamic movements in a scene. Experiments are conducted and presented illustrating the good performance and stability of the system. This system is deemed suitable for the general condition of a building interior.**

***Index Terms*—Camera acquisition, camera arrays, dynamic movements, high-speed capture, multithreading.**

I. Introduction and Motivation

THE PERVASIVE nature of cameras (image-based sensors) has promoted their development and deployment as sensor networks that now span a broad range of scales and areas, encompassing everything from microscopy to space exploration. Examples include ultrahigh-speed cameras used in the area of particle physics that are capable of acquiring images at rates of 400 000 000 frames/s [1] as well as high-resolution applications such as 16-Mpixel sensor arrays (4000 $\times$ 4000 pixels) for direct electron-microscopy imaging of proteins [2] or the 2.5-Mpixel camera array of the Hubble Space telescope [3], [4]. More conventional applications (less specialized, but still highly unique) include, for example, work by Moon and Bernold [5] who used an image-based approach for controlling a robotic paint-removal system applied to bridge structures. This system required robot-joint positional information, which could be collected from images captured using a moderate-speed analog video camera and frame grabber. The pronounced physical constraints and associated difficult field working conditions makes the image-based approach in this context extremely appealing. Papanicolaou *et al.* [6] describe an image-based system for use in detecting the movement of sediments in a stream. This system was used in conjunction with a laser Doppler velocimeter to study the effects of turbulence on sediment transport. In a similar application, Gustafsson and Gustafsson [7] use image-analysis techniques to study particle movement by introducing and monitoring passive markers traveling through a flow pattern. In a recent application to structural measurements, Fu and Moosa [8] use a single monochrome charged-couple-device (CCD) camera for detailed monitoring of the external deformations of a simply supported beam. Successful application to damage detection and structural-health monitoring was demonstrated using this single CCD camera approach. These and other image-based approaches applied to civil-engineering applications use single cameras coupled with frame grabbers and digital image-processing techniques to identify two-dimensional scene anomalies specific to the application problem.

Of particular interest are remote-sensing applications, which have received increased attention and highlighted some of the critical research challenges. The availability of ever-improving imaging chips that provide increased resolution and data-acquisition rates is posing the problem that data are made available at rates and quantities that frequently exceed the capabilities of available data-acquisition systems. As a response to this problem, many conventional image-capture systems attempt to minimize the amount of data collected during capture by applying hardware filters that limit the type of information being recorded. A good example are light-based systems that use filters to limit recorded information to a very narrow wavelength band that matches that of light being reflected or emitted from passive or active markers placed in the target environment e.g., [9]. This approach allows select individual pixels to be recorded rather than entire images, dramatically reducing the dataset size, while allowing higher sampling rates. Unfortunately, a crucial strength of the camera sensor is sacrificed, since the target environment now has to be prepared before data can be acquired.

The more general condition, of using all the information captured by an image-based sensor is much more challenging and requires rigorous consideration of hardware and software pipelines needed to obtain accurate tracking results, particularly

Manuscript received September 18, 2004; revised June 21, 2005. This work was supported by a joint program between the Consortium of Universities for Research in Earthquake Engineering (CUREE) and Kajima Corporation (CUREE–Kajima Phase V).

T. C. Hutchinson is with the Department of Civil and Environmental Engineering, University of California, Irvine, CA 92697-2175 USA (e-mail: thutchin@uci.edu).

F. Kuester, K.-U. Doerr, and D. Lim are with the Department of Electrical Engineering and Computer Science, University of California, Irvine, CA 92697-2625 USA (e-mail: fkuester@uci.edu; kdorr@uci.edu; hdlim@uci.edu).

Digital Object Identifier 10.1109/TIM.2005.860872

for large amounts and rates of data. In this paper, we specifically address this issue by describing a unique hardware and software design implemented for use as a fully image-based capture system. The hardware and software systems are optimized in terms of specifications and cost effectiveness, with specific design requirements of high-speed capture (e.g., earthquake-induced motions or other dynamic movements in a scene). Experiments are conducted and presented illustrating the superior stability and performance of the system. This system is deemed suitable for the general condition of the interior building environment.

II. Image-Based System: Hardware Design

The hardware for the image-based capture system consists of an array of four high-speed CCD cameras, a suitable computational platform, data storage, and the necessary cabling and power supplies. The following sections discuss the selection of the camera array and the design of the computational platform for this system.

A. Camera Array Design

A four-camera array was selected to provide redundancy in image collection and to represent the idea of placing image sensors throughout a room for observation (such as in each of the four corners of a typical building room). This camera array allows us to determine motions in all three dimensions (translations and rotations) and overlapping fields-of-view (FOVs) provides redundancy, thereby increasing resolution.

For our work, CCD cameras were selected due to their high linearity, solid-state nature, and potential for high resolution and high speed. The terminology "CCD camera" comes from the premise that images are collected on a CCD chip, in charge packets. Light hitting the chip substrate (on individual photodetector sites located on the CCD chip) is moved in individual charged packets off of the chip. The CCD itself is manufactured on a light-sensitive crystalline silicon chip, with photodetector sites built onto the chip. Currently, there are three basic types of CCD architectures available: 1) full-frame CCD, 2) interline-transfer CCD, and 3) frame-transfer CCD. The variations in architecture define the mode of transfer of accumulated charge off of the CCD image-sensing device. Perhaps the most popular, which we have selected, is the interline-transfer CCD. In the interline-transfer CCD, every other column of the sensor is covered by an opaque mask, and information is moved laterally onto these masks then shifted (down the chip) to a serial register, as illustrated in Fig. 1. This mode of transfer is important since it means that no more than 50% of the chip area is light sensitive. It also means a tight spacing of light-sensitive areas is provided across the entire chip, helping maximize the chip's ability to retain the input light.

B. Survey of Available Camera Technologies

Based on the dominant range of frequencies of movement generated during earthquakes and satisfying optimal sampling rates, i.e., to ensure aliasing of images does not occur, data-acquisition rates of at least 30 frames/s are necessary. Higher frequencies (greater than 60 frames/s) are desirable to allow for the differentiation of acquired movement data to obtain velocities and accelerations. In addition, the vision sensors and associated processing system should be capable of capturing image data at high resolutions. Given only these specifications, there are hundreds of vision sensors available. However, given other desirable attributes, such as low cost, small size, high shock rating, readily available and compatible interfaces, such as IEEE Firewire or Universal Serial Bus (USB)-2 interfaces, the selection greatly reduces. Table I lists four potential models we considered for the application area of earthquake monitoring.

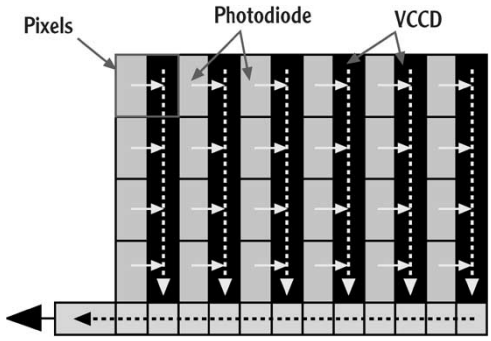


Fig. 1. Interline transfer of charge packets of photodiodes laterally and then vertically (into register) on the CCD chip (schematic courtesy of Kodak [16]).

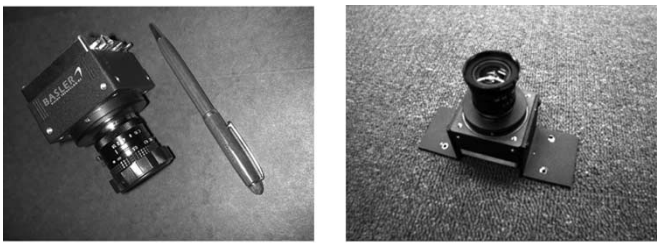
TABLE I
Comparison of Specifications and Cost of Different Available Camera Models

	Basler	Pulnix	Cooke	Point Grey
Model	A301fc	TM-6700CL	PixelFly VGA	Scorpion
Resolution	658x494	640x494	640x480	640x480
Frame rate	80 fps	60 fps	40 fps	60 fps
Color depth	8 bit	24 bit	24/36 bit	8/16 bit
Price (U.S.)	$1,895	$1,795	$6,090	$1,100

Although these models are rather expensive compared to regular off-the-shelf web cameras, the cost compared to the overall cost of a building is rather small, which makes the investment feasible. Other limiting factors for the overall system include interface-transfer hardware, e.g., PC-specific hardware such as the peripheral component interconnect (PCI) bus, hard drives, interface cards, etc. Any of these elements may have the potential for limiting the system bandwidth, and subsequently, the capture rate. Running full resolution at the highest speed may challenge PCI-bus read and write speeds, for example. Upon capturing these large volumes of data, compression and storage techniques will also need to be carefully considered.

C. Selected Camera Sensors

Based on the survey of available CCD cameras, we selected a suitable high-speed reasonable-resolution camera for use in the four-camera array. Camera selection was also based on the robustness of driver libraries available by the various manufacturers. The selected cameras are manufactured by Basler [10] and distributed locally by Aegis Electronics [11]. The specific cameras selected were the Basler A301fc (color) model, with a



(a) (b)

Fig. 2. Photographs of the Basler A301fc camera. (a) With pen for scale and (b) with mounting wings ready for assembly.

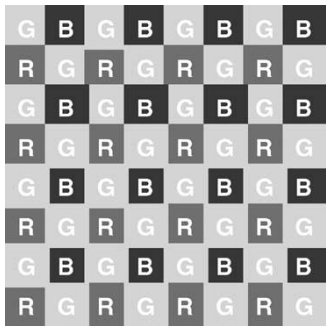


Fig. 3. Color sensitivity on the CCD chip (upon placement of the Bayer filter) for the Basler A301fc cameras.

rated 658 × 494 (8 bit/pixel) resolution, capable of 80 frames/s, and interfaced via IEEE 1394 (Firewire) technology. The gain, brightness, and exposure time of these cameras are programmable through the IEEE interface, adding flexibility and control within the software. The use of the IEEE interface also allows us to run relatively long cable lengths from the cameras (72 m with repeaters or 40 km with an optical link), while still providing flexibility to connect to laptop and desktop systems without the need for extra hardware, such as frame-grabber cards. The Firewire standard also allows devices to be powered through a data cable, further reducing the amount of required wiring. The cells within the CCD chip on these cameras contain square 9×9 μm pixels (12.7-mm-square CCD sensor), which are progressively scanned at once. The progressive area-scan nature of these cameras allows us to obtain the higher frame rate.

The camera itself, as shown in Fig. 2(a), is small (roughly 62 mm square body with variable length lens associated with the extension of the c-mount lens) and lightweight (310 g without a lens). The cameras are also rated for 10g in vibration, well above anticipated demands during seismic motion. Four steel-mounting "wings" [Fig. 2(b)] were built for each of the cameras to allow ease of attachment using wood or metal screws.

The silicon chip that the CCD is mounted on is unable to distinguish colors, therefore either color-separation optics (e.g., three CCD chips) or color-separation filters must be used to obtain coloring in the final image. Color filters must be placed in front of the chip, and after capturing an image, one can interpolate to obtain the desired coloring. The A301fc cameras have a color-separation filter known as a Bayer, or mosaic-style filter, in front of the CCD chip, allowing only one color to pass on any pixel [either red ($\lambda \geq 600$ nm), green ($\lambda = 500-600$ nm), or blue ($\lambda = 400-500$ nm)]. Across any 2×2-pixel array, two diagonally placed green-sensitive, one red-sensitive, and one blue-sensitive pixel may be found, as illustrated in Fig. 3. Using bilinear interpolation, the full-color information for all pixels can be computed, resulting in a colored image. These cameras cover a spectral range from $\lambda = 400-1000$ nm, well outside of the human-vision system ($\lambda \approx 400-800$ nm).

Two different sets of Canon lenses were purchased for these cameras with fixed focal lengths of 4.8 and 12.5 mm. With variable magnifications of $\beta = 1:1.8$ to $1:16$, these lenses allow millimeter resolution at distances of between 1.5–6.0 m. The lenses were equipped with locking screws to assure that they would remain fixed at their final settings during shaking.

TABLE II
COMMON PCI BUSES AND THEIR ASSOCIATED SPEED AND BANDWIDTH

Bus Type	Bus Width	Bus Speed	Bandwidth
PCI	32 bits	33 MHz	132 MB/sec
PCI	64 bits	33 MHz	264 MB/sec
PCI	64 bits	66 MHz	512 MB/sec
PCI	64 bits	133 MHz	1 GB/sec

D. Computational Platform and Data Flow

The full resolution and frame rate of these cameras results in acquisition of approximately 24.8 MB/s of data per camera, which must then be transferred across the IEEE 1394 (Firewire) bus through a computational platform and onto a storage media or device. Four cameras acquiring data at full resolution and acquisition rates at the same time result in a total data rate of approximately 99.2 MB/s. The Basler cameras use the Firewire 400 standard, which is limited to 400 Mb/s or 50 MB/s. Therefore, one Firewire PCI card per camera is used to guarantee the necessary bandwidth. Although Firewire 800 cards and cables (with 800 Mb/s) have recently become available, at the present time, cameras are not available to interface with this standard. Once available, multiple cameras could be interfaced with a single Firewire (800 standard) PCI card.

After data is acquired and passed to the Firewire PCI card, it must then be transferred through the systems' PCI bus. Table II lists the common PCI buses and their speeds in the current market. Of these, the 32-two-bit PCI bus running at 33 MHz is the most commonly found in current desktop computers. These buses have a theoretical peak transfer rate of 132 MB/s. However, using four cameras, simultaneous read and write operations across the PCI bus would require twice the full output from all four cameras or 198.4 MB/s, well in excess of theoretical rates for typical desktop computers. A more optimal solution is to allow read operations to occur across one bus and write operations to occur across another bus, suggesting the use of a motherboard with multiple PCI buses. Server-style PC systems in this case provide such a configuration, where at least two independent PCI buses are typically arranged on the primary motherboard. In this case, we have selected Intel's server mainboard SE7501BR2, which features three independent PCI buses, two of which are 64-bit-wide PCI-X buses running at 100 MHz. The wider bus architecture and faster clock frequency allows data-transfer rates up to six times greater than the conventional 32-bit PCI standard. A single Intel 2.8-GHz Xeon CPU is mounted on the board.

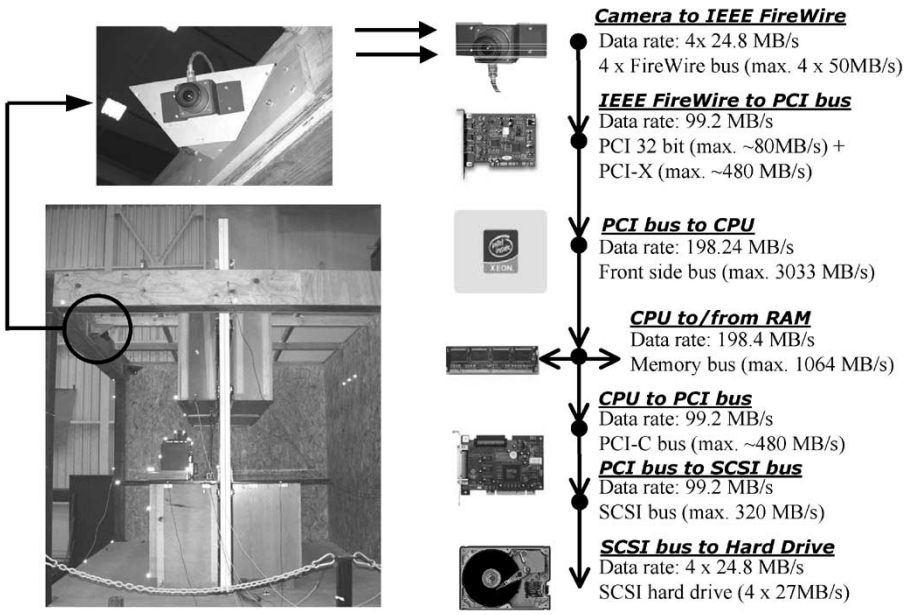


Fig. 4. Capturing and storing data using the image-based system. (Left) photographs of cameras located on the shake table and (Right) data flow inside the computer system after capturing.

Fig. 4 shows an overview of the dataflow inside the computational platform designed for capturing data from our four-camera array. The left photographs also show the layout of the cameras, resting mounted on a large shake table. Data rates produced by the capturing process are provided next to each bus in the computational platform. Below that, the maximum theoretical capacity of the bus at that location in the transfer is listed in parentheses.

As illustrated in Fig. 4, image data must first be moved from the camera via the IEEE interface to the PCI and PCI-X buses. Since the amount of data produced by the cameras already exceeds the bandwidth of a single 32-bit PCI bus, two of the four Firewire cards are attached to a 64-bit PCI-X bus, while the other two remain on the 32-bit PCI bus. Data must then transfer to the CPU, while moving across the main memory modules. This path involves data transmitted through the 533-MHz front-side bus and the 266-MHz DDR RAM memory bus. The bandwidth limitations of both bus systems are much higher than the demands of this application, therefore, these buses do not present any limitations, even though the images have to cross both buses twice. After moving across the processor and RAM modules, the data must be stored for future analysis. For this purpose, the third PCI-X bus is linked with a small-computer-system-interface (SCSI) bus and interfaced with fast SCSI hard drives. This third PCI bus is dedicated to data transfer to the hard drive, therefore its full bandwidth (480 MB/s) may be utilized.

Hard drives that are currently available typically have an advertised transfer rate of 60 MB/s or greater. Unfortunately, these speeds can only be attained while reading information. Observing the high-bandwidth requirements of our application, we equipped our system with fast Seagate ST336752LW SCSI hard drives. Benchmark tests showed that the transfer rate while writing an Audio Video Interleaved (AVI) stream to these devices is about 27 MB/s. Consequently, assigning a single hard drive per camera was possible, leaving a small margin of 2.2 MB/s between the write throughput of 27 MB/s and the demand (data rate per camera) of 24.8 MB/s. Finally, the architecture of the computational system designed and shown in Fig. 4 allows us to capture data at the full frame rate and resolution produced by the four-camera array attached to the system.

III. Image-Based System: Software Design

A primary component of our development work is centered around SceneIdentifier, our software framework for 1) image capture and acquisition; 2) image processing; and 3) data analysis and results preview. The system was implemented in C^{++} and uses a development platform known as QT[1] for its graphical user interface (GUI). QT provides the thread-management and event-handling features needed and is platform independent. The source code is developed using internal coding standards[2] and an autogeneration documentation system called Doxygen.[3] A revision-control system is used to adhere to the latest software developments.[4] The following sections present the image capture and acquisition process and the performance testing of this process.

A. Image Capture and Acquisition

Digital image/video data can be directly acquired from the Basler 301fc camera through its IEEE 1394 (Firewire)

[1]QT for GUI Development (http://www.trolltech.com).

[2]Visualization and Interactive Systems Group (VIS) coding standards (http://vis.eng.uci.edu/standards).

[3]Doxygen documentation system (http://sourceforge.net/projects/doxygen).

[4]Concurrent Version System (CVS) revision control (http://www.cvshome.org/docs/manual).

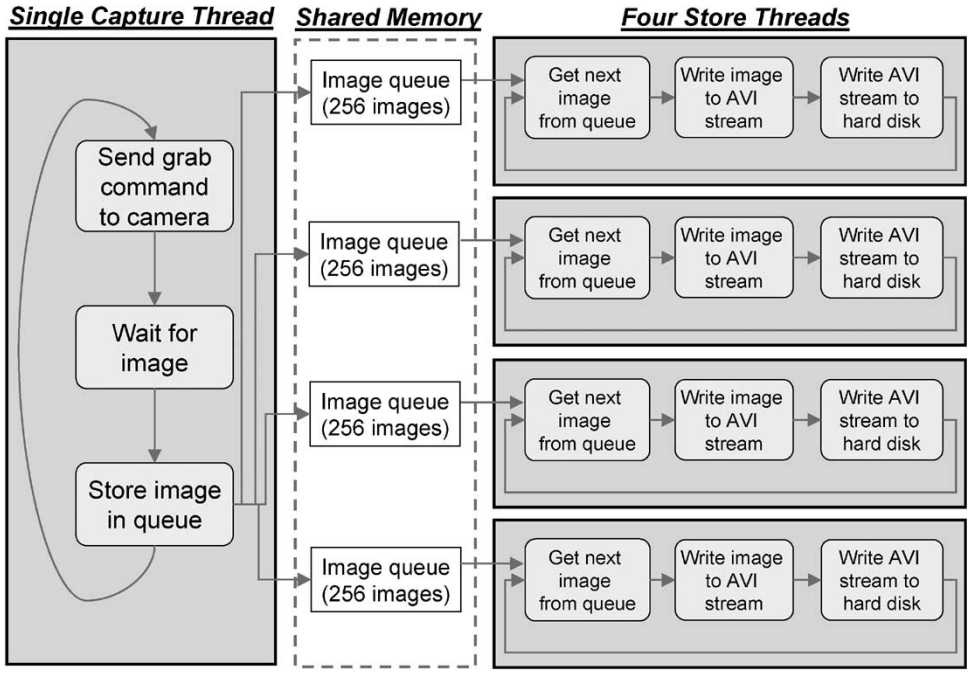


Fig. 5. SceneIdentifier's unique multithreaded capture and storing architecture.

interface. One advantage of this configuration is that no additional framegrabbers are required. Framegrabbers are generally needed for conventional video cameras, adding cost to the overall system configuration. However, as will be described, we designed a special software pipeline to bypass the use of hardware framegrabbers altogether. Prior to describing this, we highlight the capture process within the software.

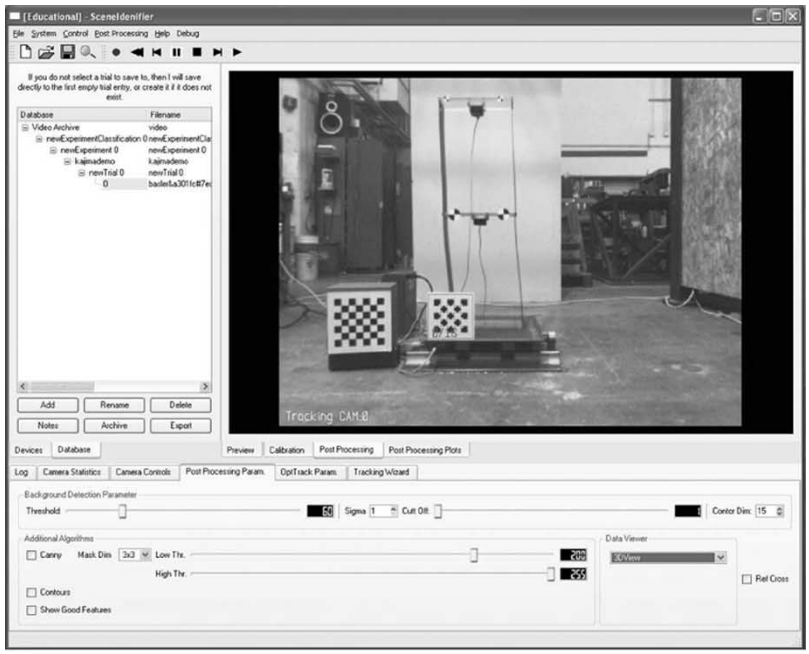
1) Capture Process: While the cameras can acquire images at speeds of up to 80 frames/s, they do not have the notion of video streams. To obtain videos, individual images have to be acquired and composited into a video. Microsoft's AVI format was selected as the preferred data format to manage these streams. AVI uses the resource-interchange file format (RIFF) to efficiently encode multimedia audio and video information. Our framework supports the sequential acquisition of still images from the Basler cameras at up to 80 Hz and the real-time generation of the corresponding AVI file. AVI encoding and decoding is included in our SiAviInput and SiAviOutput classes.

As described in Section II-B, it was desirable to control the entire sensor array (consisting of the four high-speed cameras), from a single host computer. To avoid unnecessary wait states and allow proper control of timing and sequencing of the image capture, we have developed a multithreaded application to handle data acquisition (DAQ). Fig. 5 illustrates the parallel threads running in our application. One thread is responsible for handling requests for images that are dispatched to the camera array and the other four threads (one for each camera) are responsible for transferring images from a shared buffer to the AVI stream that is stored on a camera-specific system disk. The storing threads run with real-time priority to ensure that all images are merged into the respective AVI stream as soon as they become available. If a thread observes delays in DAQ, it may also temporarily release computational resources to the other system threads to increase overall system performance, leaving enough computational cycles for all of the required operations.

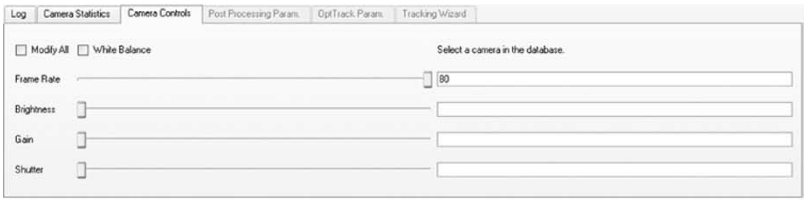
The image capture is synchronized with the image-transfer threads through multiple image buffers. One of these image-buffer threads is available for each camera and accessed by the corresponding transfer thread, allowing temporary storage of up to 256 acquired images, in the event that system resources or bandwidth become limited for a short period of time. System-performance tests, presented in Section IV, show that our software design successfully allows the capture of video data at full resolution and frame rate for the four-camera array, using the computational system previously described.

2) SceneIdentifier GUI (SIGUI): A user-friendly GUI was developed to provide complete control over all camera parameters and options such as the shutter speed, gain, acquisition rates, and white balance. Fig. 6(a) shows a snapshot of the SceneIdentifier with an image from an active camera in view. The camera-controls dialog allows the user to control specific camera parameters in the main window, as shown in Fig. 6(b). The main window shown in Fig. 6(a) includes tabs for preview, calibration, as well as image analysis, such as a playback of the original video sequence, after it has been interpolated to present color values. Image analysis includes feature detection and the analysis of time-varying waveforms of selected items. Time-varying movement of features detected in the images are also visualized in the viewer. Some of these features can be observed in real time, while others, such as object reconstruction, must be replayed from a previously acquired record. Fig. 6(c) shows an example of post-processed x- and y-direction time histories presented in SIGUI.

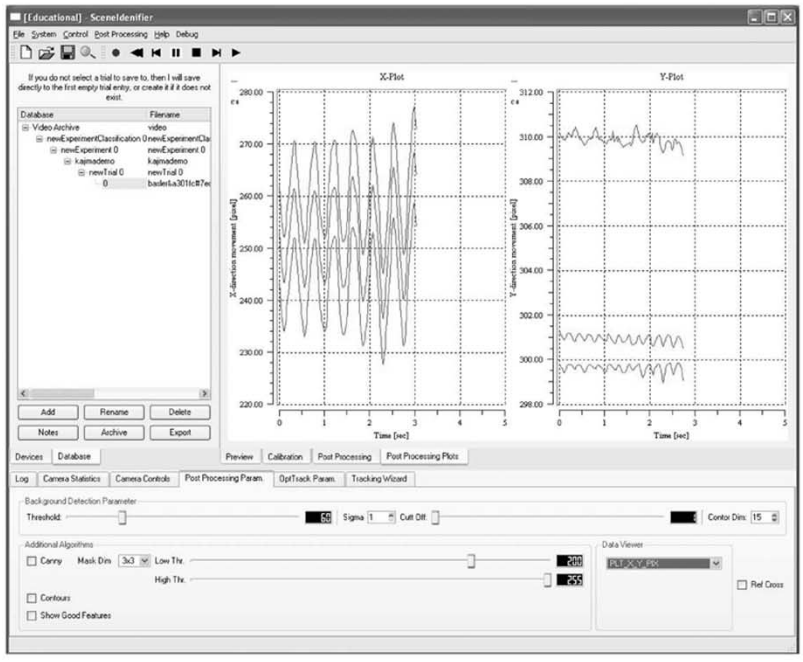
The user can also specify the desired location of the video file that will be generated by each camera, through SIGUI's



(a)



(b)



(c)

Fig. 6. Snapshot of the SIGUI. (a) Overall view with a camera view during postprocessing, (b) camera controls, and (c) postprocessing plots.

file-management system. In addition, the user interface of SceneIdentifier supports the following capabilities:

1) an arbitrary number of cameras (no hard coding);
2) autodetection of newly installed cameras;
3) camera listing for the user, including options to enable/disable individual cameras;

4) grouping of individual cameras observing specific parameters (frames per second, etc.).

3) Camera View and File Management in SIGUI: SIGUI has a unique camera view feature, which allows control over all of SceneIdentifier's data-acquisition capabilities. At startup time, the application performs a hardware query to determine the type and number of firewire cameras connected to the DAQ node. Available cameras are subsequently displayed in enumerated form in the camera-tree view, as shown below.

+ Cameras
  + −− Camera 1
  + −− Camera 2
  + −− Camera 3
  + −− Camera 4

In our framework, we use a database structure, which is extensible markup language (XML)-enabled for efficient data storage and management. The database view provides the user with access to all data that was previously acquired, including image streams and experiment logs. It provides the basis for users to manage files in logical sequence with experiments. Fig. 7 highlights the basic features of the GUI. The tree view presented can provide the following information:

− Experiment Classification
−− Experiment
−−− Experiment Configuration
−−−− Experiment Parameters
−−−−− Camera Data

For a typical experiment, the framework will look as presented at the bottom of the page.

Fig. 7. Snapshot of the SceneIdentifier data view mode from SIGUI.

B. Image-Based System Calibration

Camera calibration and image correction are fundamental steps that must be applied before reliable image processing and analysis can occur. A complete review of camera-calibration fundamentals and techniques applied is provided in [12] and herein we focus only on the description and identification of the intrinsic parameters for each camera. Of particular interest is the image correction for radial and tangential distortions, as it is known that all camera lenses introduce such distortions. The intrinsic camera parameters establish a relationship between the image and the camera (Fig. 8) and include the coordinates of the cameras' principal point (c_x, c_y), the focal length in the x and y directions (f_x, f_y), the radial-distortion coefficients (k_1, k_2) and the tangential-distortion coefficients (p_1, p_2).

To calibrate a camera, the calibration routine needs to be supplied with several views of a chessboard pattern of known geometry. Points on the model plane and their projection onto the image are then passed to the calibration routine. For every view, a minimum of 11 $M_i \leftrightarrow m_i$ point dependencies are needed for a general camera model, where M_i represents the known three-dimensional (3-D) pattern points and m_i represents the corresponding two-dimensional (2-D) image points. A rule of thumb is that for a good parameter estimation, the number of constraints (point measurements) should exceed the number of unknowns (the ten parameters for a CCD camera) by a factor of five [13]. The SI calibration tool can be configured

− "Field Test" (name of the data base entry)
−− "02072004" (name of the specific test sequence, e.g., date of experiment)
−−− "Configuration 1" (name of the experimental setup)
−−−− "GM 1" (name of the ground motion)
−−−−− "Camera xx" (the camera capturing the data for the above series)

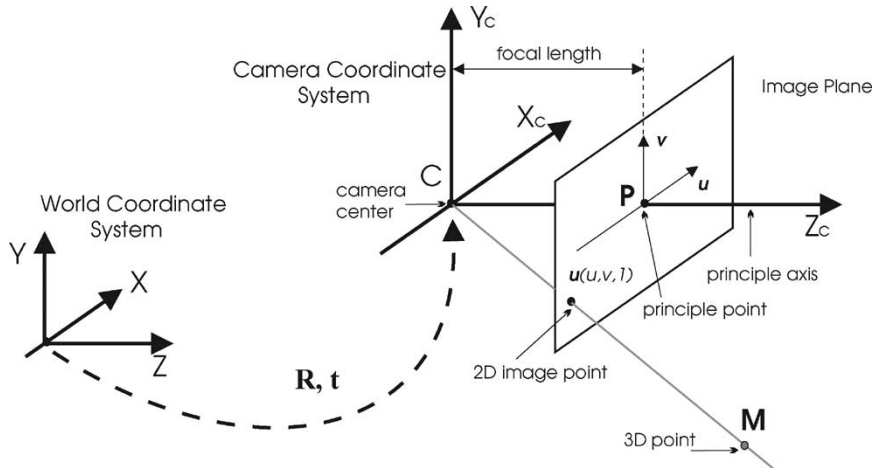


Fig. 8. Camera model.

to identify various chessboard (reference pattern) sizes up to 30×30 squares (900 reference points) to provide an adequate number of point dependencies. Using a flat chessboard pattern, the relation between a 3-D point $M_i = [X \;\; Y \;\; 0 \;\; 1]^{\mathrm{T}}$ and its image projection $m_i = [x \;\; y \;\; 1]^{\mathrm{T}}$ can be described as

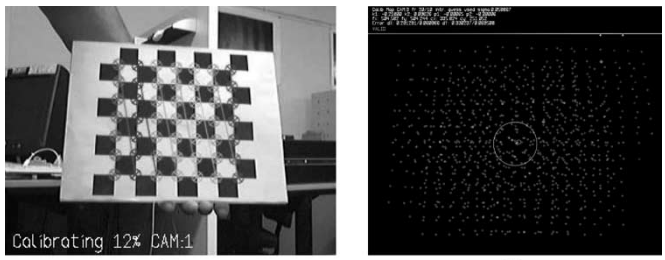
$$sm_i = HM_i \quad \text{with} \quad H = \begin{bmatrix} h_{11} & h_{12} & h_{13} \\ h_{21} & h_{22} & h_{23} \\ h_{31} & h_{32} & h_{33} \end{bmatrix} \tag{1}$$

where H is the homography matrix for relate the $\mathbb{R}^2$ image space with the $\mathbb{R}^3$ 3-D world space and s is the unspecified scalar factor. The Open Source Computer Vision Library (OpenCV) [14] used in this work has a four-step calibration function that finds this homography matrix and estimates the intrinsic and extrinsic camera parameters with the initial distortion parameters set to zero. In a final step, optimization is done by minimizing the point-reprojection error using all parameters with a maximum-likelihood estimation described in [15], by minimizing the following equation

$$\sum_{i=1}^{n} \sum_{j=1}^{m} \|m_{ij} - \check{m}(A, k_1, k_2, p_1, p_2, R_i, t_i, M_j)\|^2 \tag{2}$$

where, $\check{m}(A, k_1, k_2, p_1, p_2, R_i, t_i, M_j)$ is the projection of point M_j in image i according to (1) followed by the image correction with the distortion parameters. R_i represents the camera-rotation matrix, t_i the translation, and A the camera matrix with the intrinsic parameters focal length (f_x, f_y) and principal point (c_x, c_y). The calculated distortion parameters are used to correct the image in the image processing pipeline. To guarantee fast correction, the information on how each image pixel has to be corrected is stored in a data array matching the size of the image. Additional discussion regarding the image correction procedure may be found in [14].

Fig. 9 shows 1) the reference-point collection on a checkerboard pattern [Fig. 9(a)], and 2) the reprojection error after a calibration [Fig. 9(b)]. Fig. 10 illustrates the effect of the correction function when applied to a reference image. Small distortions can be seen by comparing the subtle differences in the gap between the two shelves or the perspective distortion of the lower edge of the two shelves seen in each image shown in Fig. 10(a) and (b).



(a) (b)

Fig. 9. Image correction. (a) Reference point collection and (b) reprojection error.

IV. Acquisition System Performance Evaluation

The specialty design of the camera hardware and software acquisition system required testing to assure that video streams could be properly collected. Of particular interest is the evaluation of the parallel thread architectures' capability to maintain synchronization and acquisition speeds during long capture runs. A number of experiments were performed to test these capabilities, and the results of these are presented in this section.

Two variables in the hardware and software will be sensitive to the final acquisition speed. First, we recognize that the cameras internal ability to maintain acquisition speed will be sensitive to the shutter speed settings. Therefore, different shutter speeds are selected for the test runs. To determine the selections, we seek the theoretical shutter speeds associated with the cameras optimal capture limit.

There are two main hardware limitations in the capture process, the first is the exposure time, the second is the time to transfer data from the sensor. The theoretical shutter speed v_{s} is simply the inverse of the exposure time t_{e}. For the camera sensors used in this work, t_{e} is defined as $(\mathrm{shutterValue} + 1) \times 20\ \mu\mathrm{s}$, where the shutterValue varies from $0 \rightarrow 4095$ [10]. This means that an optimal shutter value of 624 would result in the theoretical limit of the camera, i.e., 80 Hz. However, these

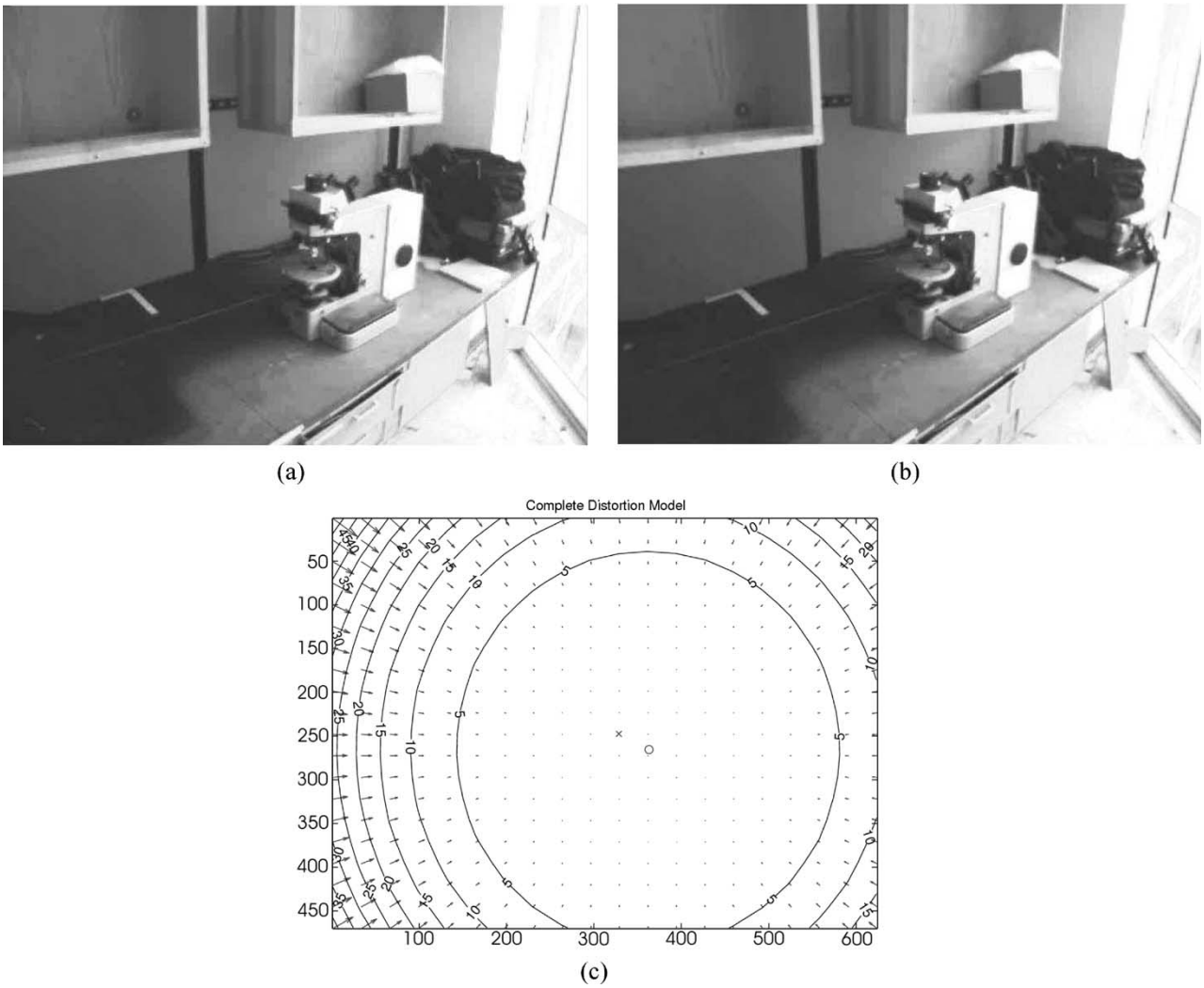


Fig. 10. Image correction example. (a) Image with lens distortion, (b) corrected image, and (c) correction model.

cameras require 11.7 ms to transfer data from the sensor. While a frame is being created, image data can be transferred in parallel, however, a second image cannot be processed. Therefore, to obtain the optimal 80-Hz capture rate, a maximum shutterValue of 584 can be set. Based on this argument, we select shutter values just near the maximum (500) and approximately half of the maximum (250) for these experiments.

Second, given the hardware pipeline described above, a likely bottleneck for acquisition speed is the number of cameras, as the computer must process the incoming data. We therefore consider configurations varying between one and four cameras. Thus, the final acquisition experiment matrix includes shutterValue $= 250, 500$; configurations consisting of one, two, three, and four cameras; five trial runs each; and a 60 s capture duration.

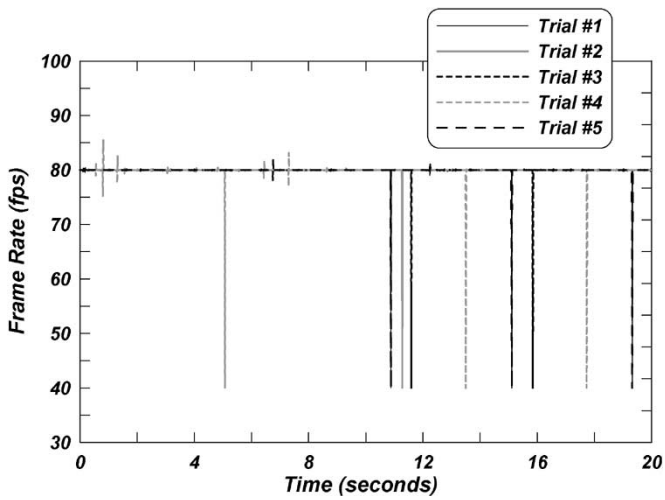


Fig. 11. Sample performance data for one camera with a shutterValue $= 250$.

A. Results and Discussion

Sample results for a single camera, with the shutterValue setting at 250 are shown in Fig. 11. This figure shows all five trial runs, where a target setting of 80 frames/s was set in the software. Note that periodic pulses above the rated 80 frames/s are attainable. This is likely due to a nominal uncertainty in the specifications of the camera sensor for individual frame transfer. Although in these performance runs, capture times were set to 60 s, for this plot, we truncate the x-axis to 20 s for clarity. Periodic single point "spikes" (drops to

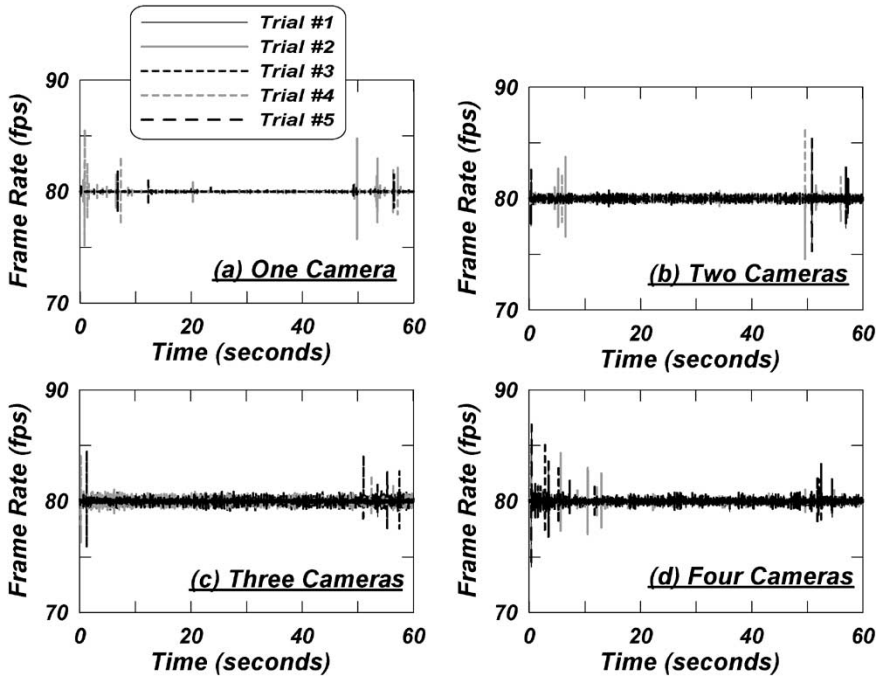


Fig. 12. Performance data summary plots for shutterValue $= 250$ for (a) one, (b) two, (c) three, and (d) four cameras.

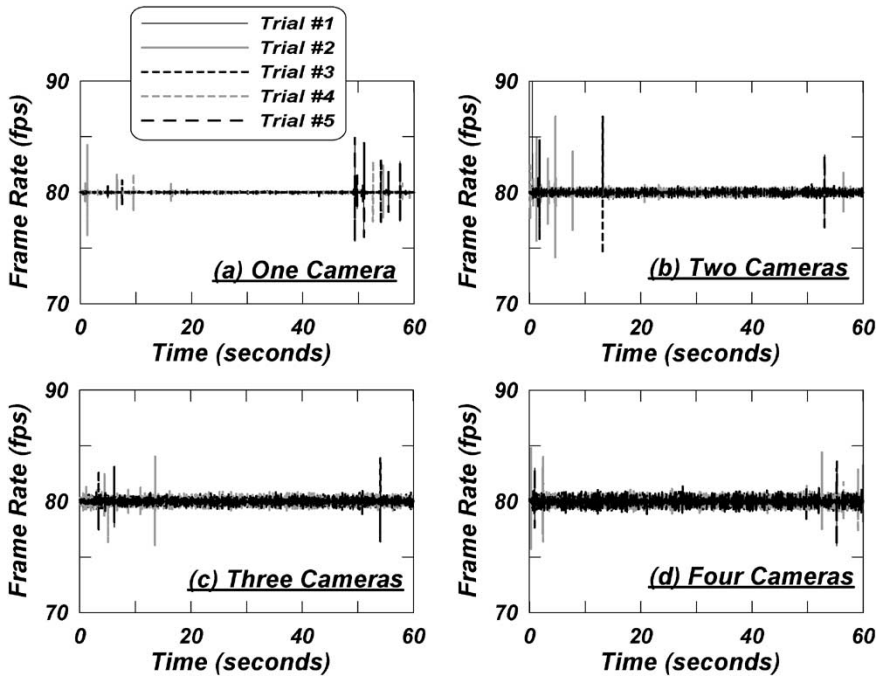


Fig. 13. Performance data summary plots for shutterValue $= 500$ for (a) one, (b) two, (c) three, and (d) four cameras.

40 frames/s) are evident, where we force the frame rate to split, indicating that we may have a data overload. These spikes are only instantaneous, however, and very minimal, occurring only once per approximately every 4–6 s. As such, they may therefore be easily removed. After postprocessing the data, the final results for the different configurations are shown in Figs. 12 and 13. These figures illustrate that as the number of cameras is increased, a nominal increase in the noise is observed [observing parts (a)–(d)] due to the added computational demands on the system. However, summary statistics provided in Table III illustrate the excellent performance of the system, with mean frame rates of above 79.99 attainable in each trial case. Moreover, the highest standard deviation σ of these runs was 0.246 frames/s, approximately 0.3% of the target 80 frames/s. Results indicate the lowest mean frame rate of these experiments as 73.19 frames/s. These results substantiate the good performance and stability of the camera hardware and software pipelines developed.

TABLE III
SUMMARY OF CAPTURE PERFORMANCE EXPERIMENTS CONSIDERING MULTIPLE CAMERA CONFIGURATIONS AND DIFFERENT SHUTTER EXPOSURE TIMES

Performance Run Series	Details	Number of Cameras	Ave Mean (m) (fps)	Ave Minimum (fps)	Standard Dev (σ) (fps)	m + σ (fps)	m - σ (fps)
1	5 runs at 250 shutter speed	1	79.998	76.428	0.099	80.097	79.899
	5 runs at 250 shutter speed	2	79.997	73.194	0.169	80.166	79.828
	5 runs at 250 shutter speed	3	79.998	77.012	0.181	80.179	79.817
	5 runs at 250 shutter speed	4	79.999	76.334	0.190	80.189	79.809
2	5 runs at 500 shutter speed	1	79.998	77.065	0.087	80.085	79.911
	5 runs at 500 shutter speed	2	80.000	76.025	0.191	80.191	79.809
	5 runs at 500 shutter speed	3	79.997	76.511	0.180	80.177	79.817
	5 runs at 500 shutter speed	4	79.995	76.536	0.246	80.241	79.749

V. CONCLUSION

In this paper, we describe hardware and software designs for an optimally low cost image acquisition system. The system is designed for use within a real building environment for object tracking, and as such, performance goals, namely high resolution and speed, are desirable. A unique multithreaded software framework was developed to avoid busy waits and to optimize the computational power of a standard server-style computer. Benchmark experiments considering different shutter speeds and camera configurations illustrate the good performance and stability of the software and hardware system developed. The image-based system presented has the potential to be used as a pixel-based sensor array for object tracking with proven reliability to maintain fast speeds and high resolutions. Although select aspects of the system are specialized, generalized aspects implemented, such as the multithreading concept and the hardware pipeline for optimizing speed, are easily extensible for other applications.

ACKNOWLEDGMENT

The authors gratefully acknowledge the helpful comments and suggestions provided by Dr. K. Kanda, Program Manager, and Dr. N. Kurato, both with Kajima Corporation, and Consortium of Universities for Research in Earthquake Engineering (CUREE) Oversight Program Managers Prof. B. Iwan and Prof. H. Shah.

REFERENCES

[1] S. Kleinfelder, Y. Chen, K. Kwaitkowski, and A. Shah, "High-speed CMOS image sensor circuits with in-situ frame storage," *IEEE Trans. Nucl. Sci.*, vol. 51, no. 4, pp. 1648–1656, Aug. 2004.

[2] N. Xuong, A. Milazzo, M. Ellisman, S. Peltier, J. Bouwer, F. Duttweiler, P. Leblanc, J. Matteson, H. Wieman, H. Matis, F. Bieser, P. Denes, and S. Kleinfelder, "First use of a high-sensitivity active pixel sensor array as a detector for electron microscopy," in *SPIE Electronic Imaging: Science and Technology, Sensors and Camera Systems Scientific, Industrial, and Digital Photography Applications V*, San Jose, CA, Jan. 2004, vol. 5301, pp. 242–249.

[3] Hubble Telescope Site, 2004. [Online]. Available. http://hubblesite.org/

[4] *The Hubble Project—NASA Goddard Space Flight Center*, 2004. [Online]. Available: http://hubble.nasa.gov/

[5] S. Moon and L. E. Bernold, "Vision-based interactive path planning for robotic bridge paint removal," *J. Comput. Civ. Eng., ASCE*, vol. 11, no. 2, pp. 113–120, Apr. 1997.

[6] A. N. Papanicolaou, P. Diplas, M. Balakrishnan, and C. L. Dancey, "Computer vision technique for tracking bed load movement," *J. Comput. Civ. Eng., ASCE*, vol. 13, no. 2, pp. 71–79, Apr. 1999.

[7] L. Gustafsson and P. Gustafsson, "Studying disk movements by novel image-analysis method," *J. Eng. Mech., ASCE*, vol. 121, no. 8, pp. 931–934, Aug. 1995.

[8] G. Fu and A. G. Moosa, "An optical approach to structural displacement measurement and its application," *J. Eng. Mech., ASCE*, vol. 128, no. 5, pp. 511–520, May 2002.

[9] T. C. Hutchinson and F. Kuester, "Monitoring global earthquake-induced demands using vision-based sensors," *IEEE Trans. Instrum. Meas.*, vol. 53, no. 1, pp. 31–36, Feb. 2004.

[10] Basler Vision Technologies, 2003. [Online]. Available: http://www.baslerweb.com

[11] Aegis Electronics Group Inc., 2003. [Online]. Available: http://www.aegis-elec.com/rep.html

[12] M.-C. Villa-Uriol, G. Chaudhary, F. Kuester, T. C. Hutchinson, and N. Bagherzadeh, "Extracting 3d from 2d: Selection basis for camera calibration," in *Proc. IASTED Computer Graphics and Imaging (CGIM)*, Kauai, HI, 2004, pp. 315–321.

[13] R. Hartley and A. Zisserman, *Multiple View Geometry in Computer Vision*, 2nd ed. Cambridge, U.K.: Press Syndicate Univ. Cambridge, 2003. CB2 2RU.

[14] Intel Research Group, *Open Source Computer Vision Library*, 2004. [Online]. Available: http://www.intel.com/research/mrl/research/opencv

[15] Z. Zhang, "Flexible camera calibration by viewing a plane from unknown orientations," in *Proc. IEEE Int. Conf. Computer Vision (ICCV)*, Kerkyra, Greece, 1999, pp. 666–673.

[16] Eastman Kodak Company, *Expanding the Horizons of Professional Digital Photography: Technical Discussion*, 2003. [Online]. Available: http://wwwse.kodak.com/

Tara C. Hutchinson (M'02) received the M.S. degree in civil engineering from the University of Michigan, Ann Arbor, in 1995, and the Ph.D. degree from the University of California, Davis, in 2001.

She is an Assistant Professor in structural engineering within the Department of Civil and Environmental Engineering, University of California, Irvine. Current research activities include both experimental and analytical studies primarily in earthquake engineering and emphasizing seismic soil-structure interaction and performance-based earthquake engineering. Integration of image-processing and computer-vision techniques for the development of new sensing and visualization approaches forms an important part of the experimental component of her work.

Falko Kuester (M'02) received the M.S. degrees in mechanical engineering and computer science and engineering in 1994 and 1995, respectively, from the University of Michigan, Ann Arbor. In 2001, he received the Ph.D. degree from the University of California, Davis.

He is currently an Assistant Professor in the Department of Electrical Engineering and Computer Science, University of California, Irvine. His research interests include image-based modeling and rendering, terascale scientific visualization, and virtual reality, as well as distributed and remote visualization. Application areas of his research include earthquake engineering, biomedical engineering, rapid prototyping, and command and control.

Kai-Uwe Doerr (S'01–A'01–M'03) received the Ph.D. degree from Darmstadt University of Technology, Darmstadt, Germany, in 2004.

He has expertise in the fields of virtual cockpit simulation, prototyping, and three-dimensional (3-D) database generation. Currently, he is a Post-Doctoral Researcher working jointly with the Departments of Civil and Environmental Engineering and Electrical Engineering and Computer Science at the University of California, Irvine. His current work focuses on image-based tracking algorithms and field-based applications of these tools.

David Lim received the B.S. degree from the School of Information and Computer Science (ICS), University of California, Irvine, in 2004.

During his undergraduate research program, he conducted research on image-acquisition techniques using minimal-hardware triggering methods.