

# Stereoscopic 3D Reconstruction using Motorized Zoom Lenses within an Embedded System

Pengcheng Liu, Andrew Willis, Yunfeng Sui

University of North Carolina at Charlotte  
9201 University City Blvd., Charlotte, NC 28223

## ABSTRACT

*This paper describes a novel embedded system capable of estimating 3D positions of surfaces viewed by a stereoscopic rig consisting of a pair of calibrated cameras. Novel theoretical and technical aspects of the system are tied to two aspects of the design that deviate from typical stereoscopic reconstruction systems: (1) incorporation of an 10x zoom lens (Rainbow-H10x8.5) and (2) implementation of the system on an embedded system. The system components include a DSP running  $\mu$ Clinux, an embedded version of the Linux operating system, and an FPGA. The DSP orchestrates data flow within the system and performs complex computational tasks and the FPGA provides an interface to the system devices which consist of a CMOS camera pair and a pair of servo motors which rotate (pan) each camera. Calibration of the camera pair is accomplished using a collection of stereo images that view a common chess board calibration pattern for a set of pre-defined zoom positions. Calibration settings for an arbitrary zoom setting are estimated by interpolation of the camera parameters. A low-computational cost method for dense stereo matching is used to compute depth disparities for the stereo image pairs. Surface reconstruction is accomplished by classical triangulation of the matched points from the depth disparities. This article includes our methods and results for the following problems: (1) automatic computation of the focus and exposure settings for the lens and camera sensor, (2) calibration of the system for various zoom settings and (3) stereo reconstruction results for several free form objects.*

## 1. INTRODUCTION

This paper describes an optical system whose purpose is to reconstruct 3D surface positions of surfaces observed by a pair of digital cameras, also known as stereoscopic 3D reconstruction. This topic has been a major focus within the field of computer vision for more than 30 years [1] and has been thought about for almost a century [2]. In fact, primates devote about half of their cerebral cortex to processing visual information [3] which may help explain why completing this task using a digital computer is computationally difficult. With applications for both lander spacecraft and rover spacecraft, stereoscopic reconstruction instruments have become critical components to robotic spacecraft.

A stereoscopic reconstruction instrument estimates 3D  $(x, y, z)$  positions of objects viewed by a pair of digital cameras. By knowing or estimating the image formation properties of each camera, their relative pose, and the projected image positions of the observed 3D surfaces in the digital images, one may invert the image formation process and find the 3D locations responsible for reflecting the light sensed by the camera pair [1]. Several problems arise in obtaining accurate 3D estimates, which have prompted an explosion of reconstruction techniques (the text [1] is entirely devoted to this subject and discusses in excess of 40 significant publications on this problem). This is due to the extremely large number of variables involved which, in addition to the geometric problem discussed previously, include *photometry*, i.e., how the scene is illuminated and how surfaces viewed within the scene reflect that illumination, and *environmental dynamics*, i.e., the dynamics of the environment through which both the illuminating light and sensed light passes. The most general approaches to the problem seek to estimate many of these parameters and generally have difficulty achieving high accuracy without using a large number of images to constrain the values of the unknown variables [4, 5]. However scientists have highly accurate knowledge of many of these parameters for custom-designed spacecraft which serves to constrain the problem and, when used properly, greatly simplifies the solution [6]. This simplification translates into reduced computational complexity for generating 3D data from a pair of 2D images and many such systems exist today both commercially, e.g., Point Grey Research sells two such systems : Bumblebee2 and Digiclops [7], and for space research, e.g., the Mars Exploratory Rovers (MERs) Spirit and Opportunity [8, 9].

This paper describes the implementation of an embedded stereo system including both the hardware design and software design. Hardware design includes the system devices, their communication links, and how they integrate to generate high-quality stereoscopic image data. Image data is processed to calibrate the cameras as part of the system setup and subsequently to perform image rectification, dense disparity maps, and 3D surface estimates.

---

Further author information: (Send correspondence to A. Willis)

A. Willis: E-mail: arwillis@uncc.edu, Telephone: 1 704 687 8420

## 2. RELATED WORK

The design of our stereoscopic system generally follows that of other FPGA-based stereoscopic systems in the literature such as those in [10, 11]. Such designs use a camera projection model to represent the image formation process that occurs in each of the two cameras. The variables of this model are referred to as the *intrinsic parameters*, since they specify the internal workings of the camera in terms of the lens and sensor. The relative joint geometry of the two cameras in the 3D world is modeled as a single rigid Euclidean transformation whose parameters are referred to as *extrinsic parameters*. Camera calibration techniques estimate the intrinsic parameters for each of the two cameras and the extrinsic parameters that specify their relative spatial location and orientation which is used to reconstruct the locations of 3D scene points.

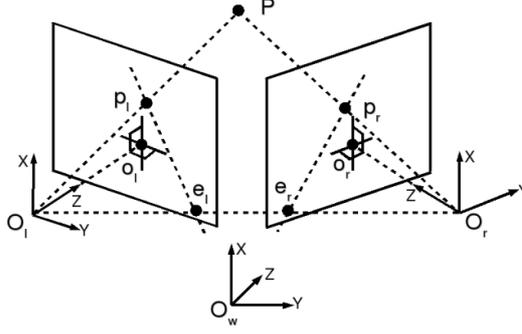
Our calibration process is typical and proceeds by placing an *a-priori* known calibration target having known geometry and appearance within view of one or more cameras that are to be calibrated. The locations of special points within the calibration pattern, referred to as calibration feature points, are measured with respect to a global coordinate system usually based on the calibration target, e.g., the upper-left hand corner of a chessboard pattern is taken as the origin and the row and column directions of the chessboard define the  $y$  - *axis* and  $x$  - *axis*. Using the known geometry of the pattern and the prescribed origin, 3D locations of the calibration pattern feature points are known with a high degree of accuracy. A cameras calibration parameters may be estimated by recording an image of the calibration pattern, estimating the projected locations of the pattern feature points in the camera image, and then finding the correspondence between the 3D calibration feature points and those detected in the image. For multiple camera systems such as the two-camera stereoscopic system, two cameras observe a single stationary pattern. Estimation of their calibration parameters defines their relationship to the calibration pattern which indirectly defines the relative orientation and position of the camera pair.

Camera calibration is a core computer vision problem that has been studied for over two decades. Seminal papers on the subject by Longuet-Higgins [12] and Tsai [13] and subsequent significant developments in [14–16] have brought the topic of camera calibration to some level of maturity (see [17] for a recent review of calibration techniques). Others researchers formulate the camera calibration problem in terms of a homography between the camera pair which incorporates both extrinsic and intrinsic parameters. The homography may be estimated using the point correspondence described [13–15] or by finding vanishing points in the image by detecting the projection of parallel 3D lines within each image [16, 18].

Knowledge of the camera calibration variables reduces the problem of 3D reconstruction to the following 1-dimensional search problem: *For each pixel in the left image, we must find the image of the same surface location in the right image.* In general this would be a 2-dimensional search problem, however, the epipolar constraint shows that, in fact, the corresponding pixel in the right image must lie along the *epipolar line*, i.e., the line defined by the intersection of the *epipolar plane* with the plane of the right camera CCD (see Figure 1). Researchers often perform image rectification which re-projects the recorded images for each camera into a special coordinate system such that pixels along a given row in the left image correspond to pixels along the same row of the right image. This transforms pairs of conjugate epipolar lines in the two cameras to become collinear and parallel to one of the image axes (see for a complete description of rectification techniques for any epipolar geometry [19]).

A solution to the aforementioned problem simplifies to taking each pixel in the left image and finding the corresponding pixel from the same surface point that lies in the same row of the right image at a different column location. This is known as the problem of *dense stereo matching* which is often the most difficult aspect of 3D reconstruction. Solutions to the stereo matching problem for rectified images provide a *disparity* for each matched pixel pair whose value is the column difference between the corresponding pixel locations in the left and right images. The image of column differences for all pixels is called the disparity map (see Figure 6(c) for an example).

Approaches to solving the correspondence problem can be broadly classified into two categories: the intensity-based matching and the feature-based matching techniques. In the first category, the process is applied directly to the intensity profiles of the two images, while in the second, features are first extracted from the images and the matching process is applied to the features. There are numerous proposed approaches to this problem using an expansive variety of pattern-matching techniques. Some methods use area-based intensity differences [20], graph cut methods [21], Bayesian approaches [22], multi-scale approaches [23], partial differential equations [24], dynamic programming [25], segmentation-driven methods [26], matching image regions [27], and color-weighted correlation [28]. Recent methods such as [26] integrate image segmentation techniques with smoothness constraints on matched disparity values to produce accurate solutions to this problem. Unfortunately, such techniques require resources and computational complexity that preclude implementation within an embedded system with modest memory and computational resources.



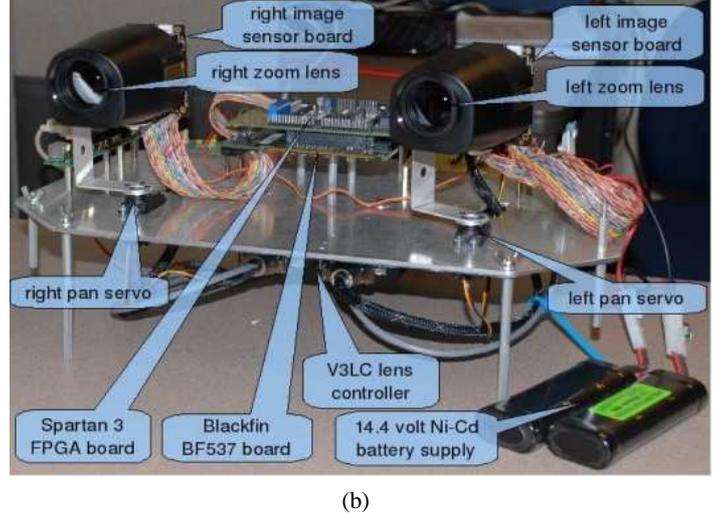
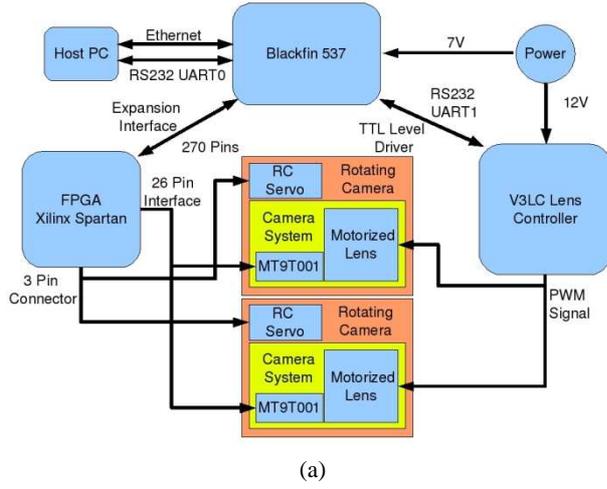
**Figure 1.** Epipolar geometry is constructed around the epipolar plane which is the plane containing a 3D surface position  $\mathbf{P}$  and the optical centers of the left and right cameras,  $\mathbf{O}_l$  and  $\mathbf{O}_r$ , respectively. Epipolar lines correspond to the intersection of this plane with the image sensors of the left and right cameras. The epipolar constraint requires that the projection of the 3D surface point in each camera image must lie along the epipolar line, i.e., the line in the camera image plane that also belongs to the epipolar plane (the line corresponding to the intersection of the two planes). In 3D reconstruction, we select a pixel from the left image,  $\mathbf{p}_l$ , and must compute the location of the pixel in the right camera coming from the same 3D location,  $\mathbf{p}_r$ .

From each matched pixel pair, we can compute the position of the projection point in 3D by triangulation as described in [29, 30]. Surface reconstruction proceed by reconstructing neighboring surface locations and using the neighborhood relationship of the image to determine the connectivity of a surface mesh in 3D. Reconstructed locations may be highly variable due to errors in the estimated calibration parameters or the computed disparities. Large errors appear as mesh outliers and create highly irregular variations in the reconstructed surface mesh.

### 3. SYSTEM MODEL AND NOTATION

We begin by defining an arbitrary 3D surface point  $\mathbf{P}$  that is viewed by a pair of cameras that measure the surface point at image locations  $\mathbf{p}_l$  and  $\mathbf{p}_r$ . All points lie in global coordinate system having origin at some arbitrary 3D point,  $\mathbf{O}_w$ , and orientation defined three mutually orthogonal vectors defining the coordinate system  $x, y, z$  axes. This global coordinate system is typically taken as a pre-defined position and orientation of the instrument (in some cases a robot) upon which the stereo system is mounted. We refer to the two cameras of the system as *left* and *right* based on their observed relative positions when one stands behind the imaging system between the two cameras. Variables associated with the left camera are indicated with a subscript  $l$  and variables associated with the right camera are indicated with a subscript  $r$ . Using this convention and the prescribed coordinate system, we denote the positions of the left and right cameras by the location of their optical centers  $\mathbf{O}_l = (O_{l,x}, O_{l,y}, O_{l,z})^t$  and  $\mathbf{O}_r = (O_{r,x}, O_{r,y}, O_{r,z})^t$  respectively which physically corresponds approximately to the center of the final optical lens element that focuses light onto the CCD center (see [29] for an exact description). Orientations of the left and right cameras are specified by  $3 \times 3$  rotation matrices  $\mathbf{R}_l$  and  $\mathbf{R}_r$  that depend upon 3 parameters each, i.e.,  $\mathbf{R}_l(r_{l,1}, r_{l,2}, r_{l,3})$  and  $\mathbf{R}_r(r_{r,1}, r_{r,2}, r_{r,3})$ . In practice, the orientations for all coordinate systems are parametrized using the 3-parameter Rodriguez representation,  $\mathbf{r} = (r_1, r_2, r_3)$ , from which one may compute a unique 3D rotation matrix  $\mathbf{R}(\mathbf{r})$  using the Rodriguez formula. Jointly, the parameters  $\mathbf{x}_{ext} = (\mathbf{R}, \mathbf{T})^t$  form the 6 unknown variables referred to as the extrinsic parameters of the system.

The extrinsic parameters of our stereoscopic system specify the relative position and orientations of the two cameras *where the left camera position and orientation define the world coordinate system origin and  $x, y, z$  - axes respectively*. The physical meaning of the camera orientation is specified in terms of three mutually orthogonal directions in 3D, the horizontal and vertical directions span the columns and rows of the camera sensor respectively and depth corresponds to forward/backward motions along the optical axis, i.e., the line passing through the lens center perpendicular to the plane of the camera sensor. In this coordinate system, the translation vector  $\mathbf{T} = (t_x, t_y, t_z)^t = \mathbf{R}_l^t(\mathbf{O}_r - \mathbf{O}_l)$  defines the position of the right camera with respect to the coordinate system defined by the left camera and  $\mathbf{R} = \mathbf{R}_l^t \mathbf{R}_r$  defines the orientation of the right camera with respect to the coordinate system defined by the left camera. In contrast to the typical stereoscopic system, each camera in our system is mounted on a servo motor which rotates the cameras in the horizontal plane; roughly the world coordinate system  $xz$  - plane (see Figure 2).



**Figure 2.** (a) A functional block diagram of the embedded system showing system devices and their electrical interconnections. (b) An image of the as-built system with devices labeled running from a 14.4V supply provided by 2 9-cell 7.2V Ni-Cd batteries (bottom right).

The intrinsic parameters of a camera within our system is referred to as the parameter vector  $\mathbf{x}_{int} = (\mathbf{m}, \mathbf{k})^t$  where  $\mathbf{m} = (f, s_x, s_y, o_x, o_y)^t$  specifies the lens focal length,  $f$ , the width and height dimensions of a pixel,  $(s_x, s_y)$ , and the location of the image center  $\mathbf{o} = (o_x, o_y)$  defined as the point where the optical axis intersects the plane of the camera sensor. The vector of parameters in  $\mathbf{m}$  determine the image formation model for a perfect (distortion-free) lens. The vector  $\mathbf{k} = (k_1, k_2, k_3, k_4)^t$  models the lens radial and tangential distortion using the Brown-Conrady lens model. Radial distortions are modeled by parameters  $k_1$  and  $k_2$  and describe the deviations of the projected pixel locations from their ideal (distortion-free) positions as a function of radius from the image center  $\mathbf{o} = (o_x, o_y)$ . Specifically, if we let  $\mathbf{p} = (x, y)^t$  denote a location in the image plane,  $k_1$  and  $k_2$  define the coefficients of the 1-dimensional polynomial  $f(k_1, k_2, r) = \sum_{i=1}^2 k_i r^{2i}$  where  $r$  is radius from the image center, i.e.,  $r = \sqrt{(x - o_x)^2 + (y - o_y)^2}$ . Parameters  $k_3$  and  $k_4$  model “decentering” distortions in the  $x$  and  $y$  directions. These distortions occur when the optical system is not properly centered over the image plane. The deviations of the projected pixel locations are modeled by  $\delta\mathbf{p} = \begin{bmatrix} 2k_3xy + k_4(r^2 + 2x^2) \\ k_3(r^2 + 2y^2) + 2k_4xy \end{bmatrix}$  (this model is used in [31] and was originally developed in [32–34]). Using this distortion model, we can map the ideal projection locations of a point  $\mathbf{p}$ , to its location within the distorted image  $\mathbf{p}_d$  by the following equation  $\mathbf{p}_d = (1 + k_1r^2 + k_2r^4)\mathbf{p} + \delta\mathbf{p}$ . Jointly,  $\mathbf{m}$  and  $\mathbf{k}$  define the 9 unknown variables of the vector  $\mathbf{x}_{int}$  that form the intrinsic parameters. We must estimate these parameters separately for each of the two cameras in our stereoscopic system which we denote  $\mathbf{x}_{l,int}$  and  $\mathbf{x}_{r,int}$  the the left and right cameras respectively.

#### 4. SYSTEM HARDWARE DESIGN

The hardware design of our system involved nine different devices integrated using a variety of communication interfaces for chip setup and data transmission. The overall system block diagram is shown in Figure 2(a) and a view of the as-built system with labeled components is shown in Figure 2(b). The system devices are listed below:

- (1) Blackfin BF537 Digital Signal Processor (DSP) (Analog Devices BF537-EZKit)
- (1) Xilinx Spartan 3 FPGA (Analog Devices BF537-FPGA Daughter card)
- (1) Lens Controller Module (Image Labs International V3LC)
- (2) Micron 3Megapixel CMOS digital image sensors (Micron MT9T001 on a MI300 development board)
- (2) Rainbow motorized zoom lenses (Rainbow H10x8.5)
- (2) Servo Motors (HiTec HS-322D)

The Blackfin BF537 EZKit is a development board distributed by the chip manufacturer, Analog Devices, for evaluation of their Blackfin processor product line. The BF537 can operate with a variable clock rate up to 600MHz and includes 132kB of on-chip SRAM and a full SIMD architecture which can greatly accelerate processing of image data. The development board integrates this processor with a number of peripheral devices including audio A/D and D/A converters (AD1871/AD1854), a 10/100MB ethernet interface (SMSC LAN83C185), an I<sup>2</sup>C inter-integrated circuit bus, many (>64) General Purpose I/O lines (GPIO), and two Universal Asynchronous Receiver/Transmitter serial ports (UARTs). A number of interfaces are provided that may be used to communicate and share data with external devices. Some available standard bus interfaces include the Serial Peripheral Interface (SPI) and the inter-integrated circuit bus (I<sup>2</sup>C). Other interfaces are specific to the Blackfin such as the Serial Port Interface (SPORT) and Parallel Peripheral Interface (PPI). Our project makes use of the PPI interface to transfer image data from the CMOS image sensors to the DSP system, both UART interfaces which allow the DSP to communicate with the motorized lens controller and allow the host PC to issue commands through a command line interface to the DSP embedded system (bash console), and the I<sup>2</sup>C interface to communicate directly with those devices the system integrated circuits which are the Xilinx FPGA and the two Micron CMOS image sensors (see Figure 2).

The FPGA daughter card manufactured by Analog Devices to interface with their Blackfin development boards connects to the DSP development board via a expansion interface connection consisting of 270 pins distributed across 3 high-density 90-pin connectors. Through this interface, the BF537 PPI and I<sup>2</sup>C interface connect to the Xilinx Spartan 3 which is integrated with the daughter card. Unfortunately, use of the  $\mu$ Clinux operating system and network interface have hardware-level conflicts with the FPGA daughter card. These conflicts were resolved by disconnecting 21 pins on the FPGA daughter card expansion interface connector. The majority of these pins are associated with the 10/100MB ethernet interface of the DSP development board and their disconnection does not impact our system functionality. The integrated FPGA is a Xilinx XC3S1000 which contains 1M gates, 17k logic cells, 120kbits of distributed RAM, 432kbits of block RAM and 24 dedicated multipliers that operate on a 25MHz clock (an external clock may be provided if necessary). There are numerous DIN connectors on the surface of the FPGA daughter card. Four system devices are controlled by the FPGA: a pair of servo motors that rotate each camera in the system independently and a pair of Micron MT9T001 CMOS image sensors which sense images from the 3D scene.

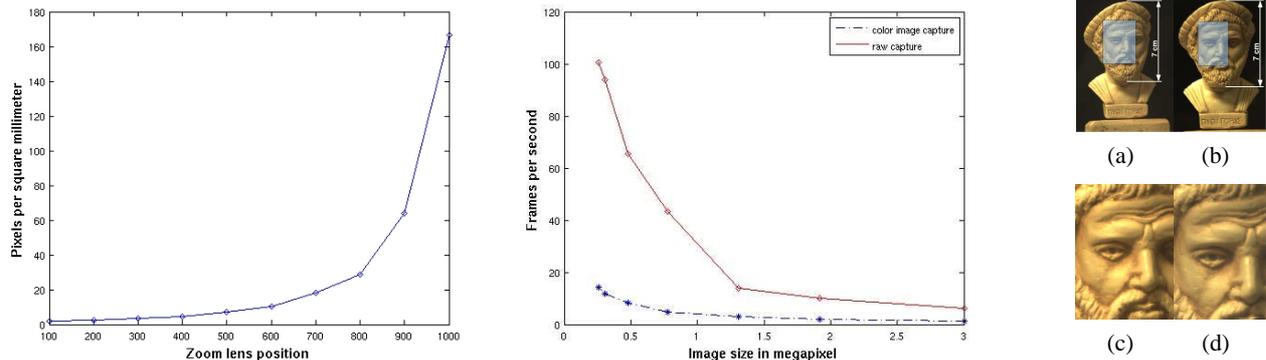
The system contains two HiTec HS-322D servo motors that are each controlled via a 3-wire connection that supplies power, signal, and ground connections to the servo. Servo positions are determined by a basic VerilogHDL block in the FPGA that generates a pulse width modulated (PWM) signal to the signal pin of the servo motor. Software running on the DSP writes position values to the FPGA using the I<sup>2</sup>C interface to specify the positions of each servo motor.

Each Micron CMOS image sensor connects to the FPGA through a short 26-pin ribbon cable and is capable of generating images with a maximum resolution of 2048x1536 at a rate of 12 frames per second (48MHz pixel clock). Software running on the DSP communicates through the FPGA to each MT9T001 image sensor which are assigned unique addresses on the system I<sup>2</sup>C bus. Register values are written to each of the image sensors to configure the image capture. There are numerous settings that may be manipulated such as resolution, pixel format, amplification, and pre-filtering (see the Micron MT9T001 datasheet for a complete list of settings). Software on the DSP also writes values to the FPGA via the I<sup>2</sup>C bus to select one of the cameras for continuous image capture. The selected camera passes its image data to the DSP main memory using the PPI interface where the FPGA simply bridges the PPI bus between the DSP and the selected camera. Our methods for interfacing the FPGA and PPI is similar to that described in [35].

A Rainbow H10x8.5 motorized zoom lens is mounted onto each of the system image sensors. This lens has three motors which control the lens zoom, focus, and iris via separate PWM signals. Such lenses allow the stereoscopic system to provide high-resolution images of structures at a distance by enabling large variations in the effective focal length of the optical system which can vary from 8.5mm (1x magnification) to 85mm (10x magnification).

The V3LC lens controller uses six PWM outputs to control the zoom, focus, and iris positions for the pair of motorized lenses. The DSP communicates with the lens controller using a RS-232 serial connection which the lens motors must be automatically adjusted to ensure high-quality image formation. In practice, a TTL line-driver (MAX232A) was necessary to amplify the RS-232 signal from the DSP logic levels of 0V-3.3V to RS-232 levels of -10V,10V. The lens controller implements a rich variety of functions that include commands for moving to absolute positions, incremental motions, and reading motor positions back from each of the lens motors.

The whole stereo system is mounted on an aluminum platform, and is powered by a battery (14V in practice). Internal system components require two different voltage levels: 12v for the V3LC lens controller and 7.2v for all other system devices. Power distribution is accomplished via two voltage regulators that provide the 12v and 7.2v outputs to power the system.



**Figure 3.** (left) A plot showing the number of pixels per square  $mm$  as a function of the zoom setting for a planar object  $2m$  from the instrument. (right) A plot showing the system frame rate in frames per second (fps) a function of the image size in mega-pixels. Images (a,b) are  $300 \times 400$  pixel image patches taken from a stereoscopic image pair at a distance of  $2m$ . Within each image, a blue box highlights the equivalent resolution surface region ( $300 \times 400$  pixels) when viewed at a  $10 \times$  magnification (c,d). 3D Reconstructions for these images are shown in Fig. 2(b,c,d).

## 5. SYSTEM SOFTWARE DESIGN

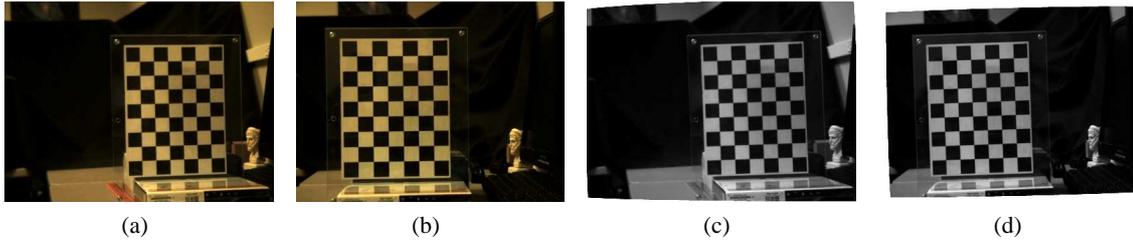
The system runs an embedded version of the  $\mu$ Clinux operating system developed and maintained by a very large online community (see <http://blackfin.uclinux.org> for details). The only permanent storage on the BF537 EZ-kit development board is a 4MB EEPROM. Boot-loader software, u-boot, is written to the EEPROM which performs the onboard initialization and initializes a trivial ftp (TFTP) transfer over the development board ethernet interface. This transfer takes a compressed copy of the operating system and the associated file system collectively referred to as a *system image* (with some abuse of terminology) from the host development PC and places it in the 64MB of RAM available on the development board. This operating system is then uncompressed into main memory and executed generating a fully functional linux operating system and filesystem which reside in the development board RAM. Resources are hence scarce since use of the disk and allocation of system memory both draw from the remaining RAM resources of the system.

A RS-232 serial connection between the host PC and the development board is used as a teletype (TTY) console within which users may execute commands on the embedded  $\mu$ Clinux system. To utilize the hardware added to the system, several modifications were necessary to the system image. Standard interfaces such as the RS232 interface, the I<sup>2</sup>C bus, and the PPI were all available and functioning as standard options available with the  $\mu$ Clinux operating system kernel. Hence, software access to all of the integrated hardware devices only required changes in the compilation settings of the system.

However, several applications were developed to access these devices through standard Unix `ioctl()` commands and to control the data flow. Towards this end special-purpose programs were developed and integrated into the standard system image. These special-purpose programs orchestrated the data flow from the cameras to the DSP system to generate 3D stereoscopic data. This complete reconstruction process involves calibration of the system cameras, rectification, dense matching between the stereoscopic image pair, and finally reconstruction of the observed 3D surface positions.

## 6. VARI-FOCAL STEREOSCOPIC RECONSTRUCTION

Our stereoscopic system integrates two zoom lenses with an embedded system for capturing stereoscopic images. Use of zoom lenses by a stereoscopic system offers powerful benefits. For example, data produced from the stereoscopic Pancam instrument on Mars rovers *Spirit* and *Opportunity* provide reasonably accurate 3D geometric measurements for objects close to the rover. Yet, accuracy decreases with distance from the rover. This is due in part to the “inverse square law” for the image sensor, i.e, the surface sample density at a radius of  $2r$  is roughly  $1/4$  the sample density at a distance  $r$  (in terms of *number of samples*/ $m^2$  of observed surface). Hence, high-resolution image capture of distant surfaces requires the rover to move closer to these surfaces which requires use of precious power resources and spacecraft mission time. Use of a zoom lens, at a minimum, allows a preliminary inspection of the distant surface by viewing the surface of interest under high-magnification. With sufficiently accurate camera calibration, this high-magnification image data may provide sufficient scientific information for analysis in which case it is a much more efficient utilization of both spacecraft time and power. For reasons such as these, the subsequent Mar Science Laboratory (MSL) mission to Mars, slated for launch in late 2011,



**Figure 4.** (a-b) an example of a calibration image pair captured by left camera and right cameras (a-b) respectively. (c-d) show the rectified versions of the images in (a-b) respectively.

includes a Mastcam device capable of recording images at 10 frames per second (fps) and includes zoom cameras with 10x magnification capable of imaging surfaces  $1km$  from the rover at a resolution of  $1cm$  per pixel.

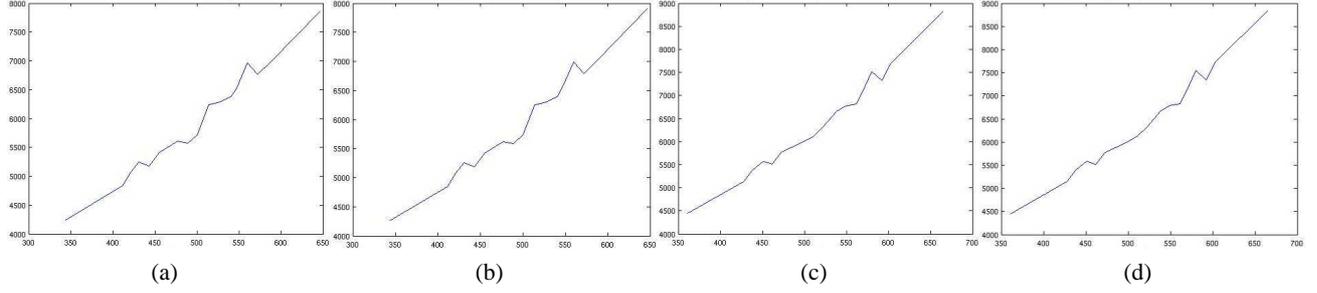
Figure 3 plots the density of sensed measurements at full resolution when viewing a planar surface perpendicular to the camera view at a distance of  $2m$ . The plot demonstrates that the designed system is capable of capturing dense sets of stereoscopic measurements from distant surfaces and with a maximum magnification of 10x, has a performance similar to the instrument included as part of the MSL.

The frame rate of the imaging system for capturing raw images and capturing color images are presented in Figure 3(a). This plot demonstrates that the system has acceptable image capture rates which are comparable to the aforementioned MSL. Reconfiguration of the image sensor via I<sup>2</sup>C allows the system to change the size of captured images. While the image capture of the system is very fast, processing the data incurs additional delays, the delay incurred by color interpolation is shown in Figure 3(b).

Proper image formation is necessary to obtain high quality reconstruction results. For our system, this requires that the object of interest must satisfy three criterion: (1) correct magnification, (2) correct focus, and (3) correct light regulation. Our system meets these criterion by setting the proper zoom, focus, and iris positions for the motorized lens on each camera. Automatic computation of the proper magnification for an object requires specification of the desired number of pixels per  $mm^2$  of observed object surface area. Since segmentation of objects within images is not of interest for this project, we accept the value for the system magnification as a user-specified constant. Having set the zoom position of the camera to the user-specified value, the object of interest is assumed to occupy a small  $100 \times 100$  pixel region in the center of both camera images. With this assumption in mind, we modify the image capture settings of the image sensors for each camera to restrict measurements to this small region which allows for an extremely high frame rate. We then continuously vary the focal position of the motorized lens until the observed intensity variation, i.e., the variance of the captured images, is maximized. This stopping point is assumed to be the correct focus position for the lens motor. In most cases, movement of the iris is not necessary since the image brightness may be controlled by altering the exposure time between image frames. However, for bright-light or low-light conditions, motion of the iris motor may be necessary. In such cases, we fix the exposure time of the image sensor and vary the position of the iris motor until an extreme position is met of the the average intensity of the smoothed image reaches half the dynamic range of the image.

Unfortunately, documentation for the zoom lenses does not specify the relationship between the effective focal length of the optical system and the lens motor position. Hence, we had to estimate this relationship by calibrating the stereo system for a number of pre-defined zoom positions and observing the relationship between the estimated the focal length and the lens motor position. Figure 5 shows our findings as a sequence of four plots. The  $x - axis$  of these plots show the reported motor position which may vary from 300 to 850. The  $y - axis$  of these plots is the effective focal length of the lens (in pixels). Image sensors may have rectangular pixel sensors which give rise to two different approximations of the focal length: (i) the focal length divided by the x-pixel dimensions and (ii) the focal length divided by the y-pixel dimension. A sequence of 13 calibrations were performed from which the linear relationship between the focal length and motor position shown in Figure 5 was computed. Note that variation away from a straight line occurs due to experimental noise, the change in the field of view for different focal lengths, and errors in the estimated parameters.

As previously mentioned, accurate calibration of the stereo system is critical for accurate 3D reconstruction of viewed surfaces. This becomes especially true when the image magnification increases when reconstructed surface locations are much more sensitive to small errors in the calibration parameters. Calibration of our zoom cameras is accomplished by collecting a sequence of stereoscopic image pairs of a chess board calibration pattern with known 3D geometry for a set of



**Figure 5.** Plots (a,c) show the estimated focal length of the zoom lens divided by the x-dimension of an image pixel,  $f_x$ , for the left and right cameras respectively. Plots (b,d) show the estimated focal length of the zoom lens divided by the y-dimension of an image pixel,  $f_y$ , for the left and right cameras respectively.

pre-defined zoom lens positions. Calibration settings for an arbitrary zoom setting are then obtained by interpolation of the camera parameters using the functional relationships estimated in Figure 5. Calibration parameters for each zoom setting were computed using the MATLAB camera calibration toolbox [31]. Unfortunately, the MATLAB toolbox requires the user to manually select four points on each of the calibration images before estimating the camera parameters which makes calibration tedious and time consuming, especially for a large number of different zoom settings. An example of images from the calibration process are shown in Figure 4(a-b) and the variation of the focal length parameters over a variety of zoom settings is shown in Figure 5(a-d). Rectification of stereoscopic image pairs is accomplished using classical techniques as specified in [29]. Figure 4(c-d) show two the rectified versions of the stereoscopic images taken in (a) and (b).

The rectified stereoscopic images are then provided as input to a dense stereo matching algorithm to compute the disparity map between the image pair. We use a custom adaptation of the dynamic programming algorithm proposed in [36] which requires little memory and provides high performance compared to other techniques while sacrificing accuracy by limiting the number of paths considered in the disparity space. The approach reduces the dense matching problem for a scan-line pair to a path computation in disparity space. A path is found by a number of local steps which assume continuity and deal with occlusions. Each step is taken based on a local information, but the current state of a path represents a global computation.

Subsequent 3D surface reconstruction is accomplished by classical triangulation of the matched points from the disparity map. The traditional triangulation method projects rays out from the left and right camera origins through the matched pixel pair in the left and right images. Ideally these rays should intersect at the true location of the 3D surface point (see Figure 1). However, due to noise in the estimated calibration parameters and the computed disparity value these 3D lines are generally skew and do not intersect. Hence, we take the point closest to the two rays as the reconstruct 3D surface location. Geometrically, this corresponds to the point that lies along the midpoint of the line segment which connects the two rays at their closest point.

## 7. RECONSTRUCTION RESULTS

Figures 6, 7, 8 and 9 are results generated from the described system. The four objects were chosen for reconstruction based on their relative sizes, geometry, and surface texture variations. All of the experiments were carried out with the object placed approximately  $1.8m$  from the stereoscopic camera pair. Each row of results corresponds to a single 3D reconstruction experiment. Proceeding from left to right, a row contains the left and right rectified images, the computed disparity map for the rectified image pair and the reconstructed 3D surface with the image texture superimposed on the 3D model.

Figure 6 shows reconstruction results for a decorated cylindrical container  $25cm$  high and  $18cm$  in diameter for three different zoom settings and is an example of a large object having simple surface geometry and relatively complex surface texture. The rectified stereo pair in (a) and (b) were processed by the dense matching algorithm to generate the disparity map (c) and the final 3D surface is shown in (d). The smooth geometry of the 3D surface and the colorful texture of the cylinder surface allow for the computation of a somewhat accurate disparity map as shown in (c). Yet, for some scan lines, the computed disparities are not valid. At these location the disparities are discarded and result in missing data within the 3D model that can be seen as horizontal stripes in the final 3D surface model. Note that this experiment tests reconstruction using lower magnification values for the zoom lenses as the image field of view captures the entire surface in all three reconstructions.

Figure 7 show reconstruction results for a large coffee mug 15cm in height and 12cm in diameter at it's largest point (the top) and represents a medium-sized object with simple geometry and a texture of moderate complexity. As in Figure 6, the color pattern on the surface provides useful cues for correct disparity values for both of the stereoscopic pairs. This is especially true for the second row of Figure 7 which corresponds to a higher magnification reconstruction of the object. The improvement in the quality of the reconstruction here may be attributed to detection of small color variations within the rectangular color patches that may have been aliased, i.e., blurred, for the stereoscopic image pair having lower magnification.

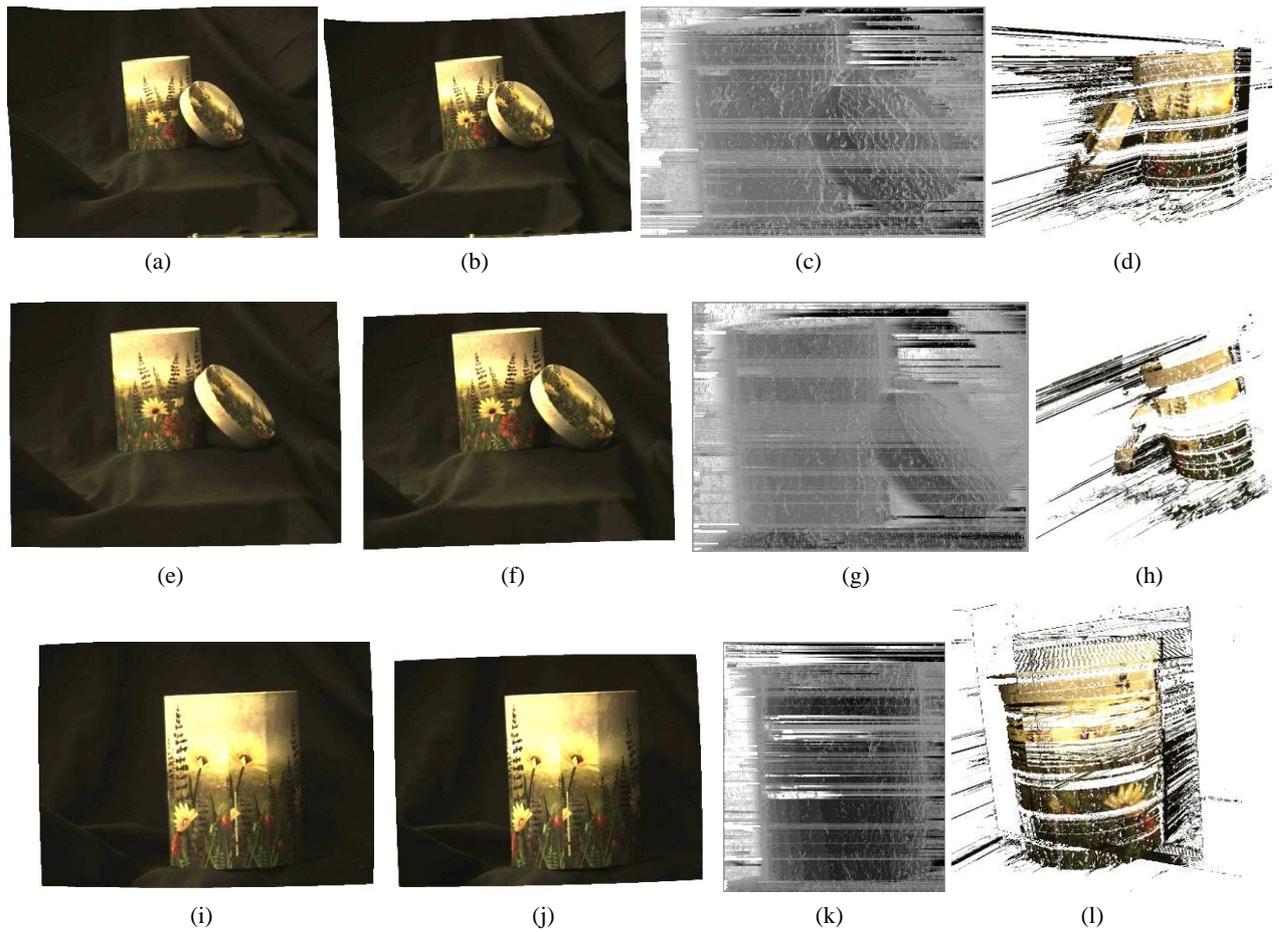
Figures 8 and 9 show reconstruction results for small objects: a figurine of Pythagoras 12cm in height and a carpenter's laser-level 8cm in width. The figurine has a complex geometry, homogeneous color and a nearly Lambertian albedo which provides very useful information for dense matching and, under higher magnification settings, provides relatively good results. The carpenter's laser level has strongly contrasting yellow and black regions and a simple smooth geometry. Our results for reconstruction on this object are relatively accurate within the yellow regions of the image and are poor in the dark regions of the image. Since intensity variations in the image are the basis of the dense matching algorithm, it is not surprising that it performs poorly in dark image regions where variations in the intensity due to geometry and due to noise become similar in magnitude, i.e., the signal to noise ratio in such regions is low.

## 8. CONCLUSION

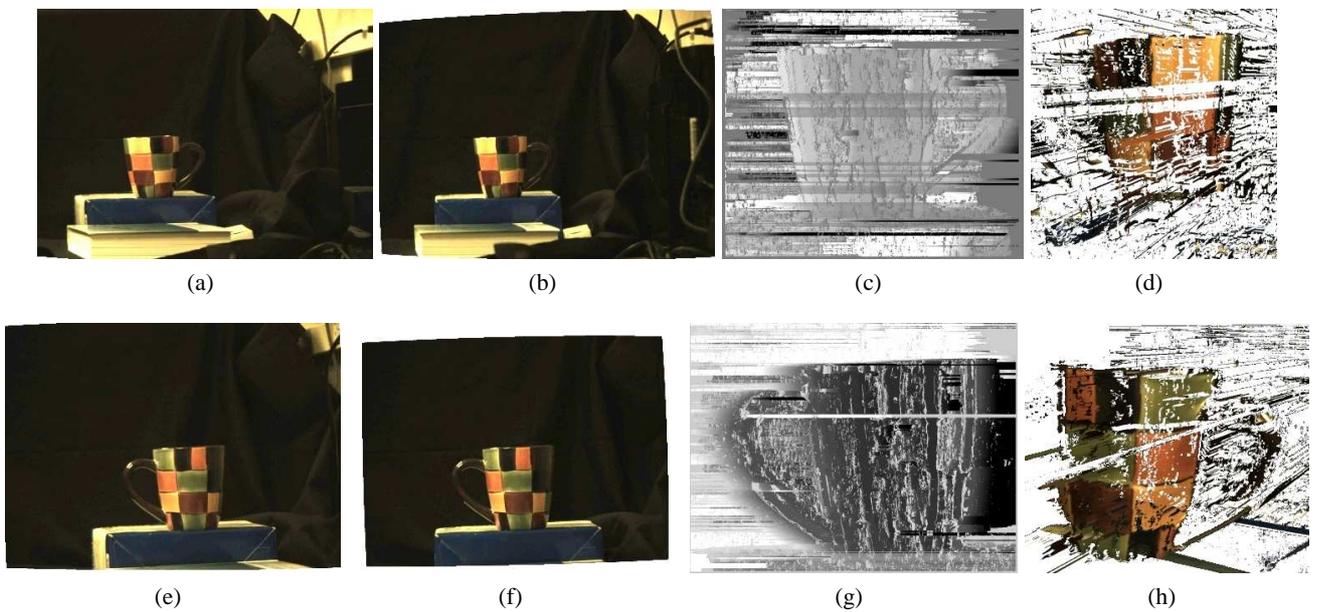
This paper describes the design and implementation of a stereoscopic 3D reconstruction system on an embedded DSP. Details were provided regarding the hardware and software design of the system which required special consideration of the integrated devices and the limited system computational and memory resources. The system also incorporates two 10x magnification zoom lenses which allow for collection of high resolution data from distant object. Such functionality is particularly useful in robotic systems where moving the stereo system has significant power and time costs. The complete system includes a DSP which orchestrates data flow within the system and performs complex computational tasks and an FPGA which provides interfaces between the DSP and the system components which consist of a CMOS camera pair and a pair of servo motors which rotate (pan) each camera. Use of the zoom lens in a stereoscopic system was described with particular emphasis on methods for automatically adjusting the lens motor positions to produce high quality images and methods for calibration zoom lenses and estimating their parameters for reconstruction at arbitrary magnification values. Surface reconstruction follows that of classical stereoscopic systems and consist of three steps: (1) rectify the measured stereoscopic image pair using the estimated calibration parameters, (2) perform dense stereo matching between pixels in each row of the image pair using a low-computational cost method, and (3) reconstruct the 3D surface by triangulating 3D surface locations from the computed depth disparities. Future work would consider integration of estimated geometries at different magnifications and improvements to the dense matching algorithm.

## REFERENCES

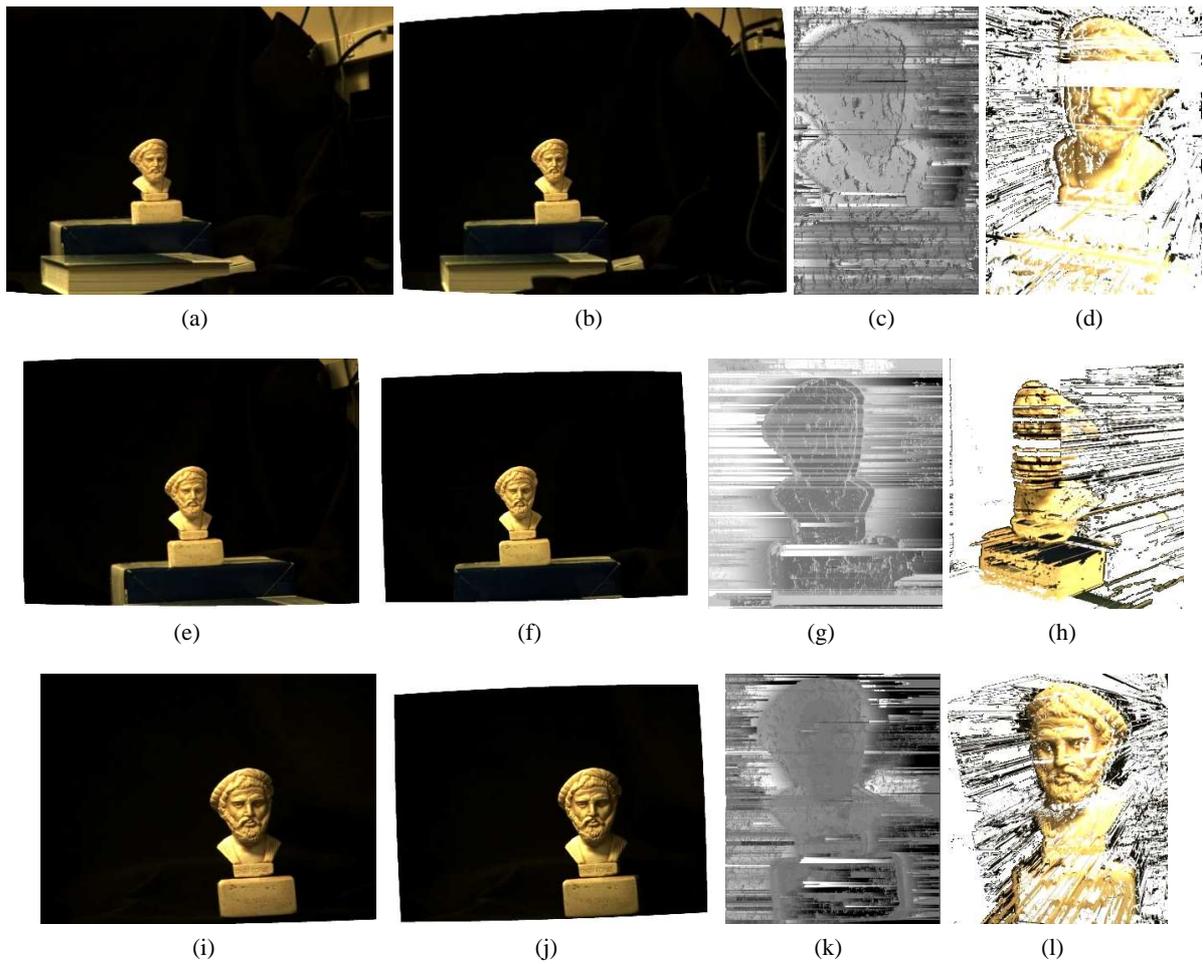
1. Y. Ma, S. Soatta, J. Kořecká, and S. S. Sastry, *An Invitation to 3-D Vision, From Images to Geometric Models*. Springer : New York, 2004.
2. E. Kruppa, "Zur ermittlung eines objektes aus zwei perspektiven mit innerer orientierung," *Sitz.-Ber. Akad. Wiss., Math. Naturw., Kl. Abt. Ila*, vol. 122, pp. 1939–1948, 1913.
3. D. J. Felleman and D. C. van Essen, "Distributed hierarchical processing in the primate cerebral cortex," *Cerebral Cortex*, pp. 1–47, 1991.
4. Q. Chen and G. Medioni, "Efficient iterative solutions to the m-view projective reconstruction problem," in *Proc. of the Int. Conf. on Computer Vision and Pattern Recognition*, vol. II, pp. 55–61, 1999.
5. A. Tirumalai, B. Schunck, and R. Jain, "Dynamic stereo with self-calibration," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 14, no. 12, pp. 1184–1189, 1992.
6. Y. Xiong and L. H. Matthies, "Stereo vision for planetary rovers: stochastic modeling to near real-time implementation," in *Conf. on Computer Vision and Pattern Recognition (CVPR)*, vol. 8, pp. 1087–1093, 1997.
7. "Point grey research." <http://www.ptgrey.com/products>, 2007.
8. Y. Yakimowski and R. Cunningham, "System for extracting 3d measurement from a stereo pair of tv cameras," *Journal of Computer Vision, Graphics and Image Processing (CVGIP)*, vol. 7, pp. 195–210, 1978.
9. L. H. Matthies, "Stereo vision for planetary rovers: stochastic modeling to near real-time implementation," in *Int. Journal of Computer Vision (IJCV)*, vol. 8, pp. 71–91.



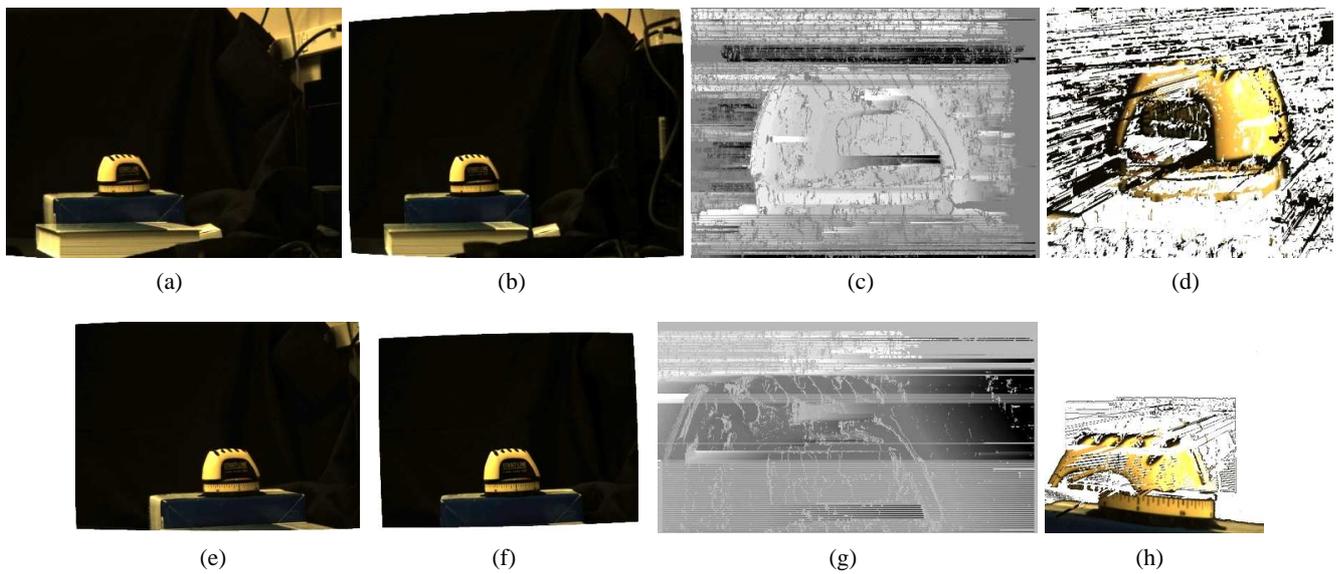
**Figure 6.** (a-l) show stereo reconstruction results for a cylindrical object for three different zoom positions (see §7 for details).



**Figure 7.** (a-h) show stereo reconstruction results for a coffee mug for two different zoom positions (see §7 for details).



**Figure 8.** (a-l) show stereo reconstruction results for a figurine for three different zoom positions (see §7 for details).



**Figure 9.** (a-h) show stereo reconstruction results for a carpenter's laser-level for two different zoom positions (see §7 for details).

10. Y. Jia, X. Zhang, M. Li, and L. An, "A miniature stereo vision machine (MSVM-III) for dense disparity mapping," in *Proceedings of the 17th International Conference on Pattern Recognition*, pp. 728–731, 2004.
11. C. Murphy, D. Lindquist, and A. M. Rynning, "Low-cost stereo vision on an FPGA," in *International Symposium on Field-Programmable Custom Computing Machines*, pp. 333–334, 2007.
12. H. C. Longuet-Higgins, "A computer algorithm for reconstructing a scene from two projections," *Nature*, no. 293, pp. 133–135, 1981.
13. R. Y. Tsai, "A versatile camera calibration technique for high-accuracy 3d machine vision metrology using off-the-shelf tv cameras," *IEEE Journal Robotics and Automation*, vol. 3, no. 4, pp. 323–344, 1987.
14. Heikkil and Silven, "A four-step camera calibration procedure with implicit image correction," in *CVPR*, pp. 1106–1112, 1997.
15. Z. Zhang, "A flexible new technique for camera calibration," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 22, no. 11, pp. 1130–1134, 2000.
16. R. Hartley and A. Zisserman, *Multiple View Geometry in Computer Vision*. Cambridge University Press, 2000.
17. F. Remondino and C. Fraser, "Digital camera calibration methods: considerations and comparisons," in *IEVM06*, pp. 266–272, 2006.
18. L. Grammatikopoulos, G. Karras, and E. Petsa, "An automatic approach for camera calibration from vanishing points," *ISPRS journal of photogrammetry and remote sensing*, vol. 62, no. 1, pp. 64–76, 2007.
19. D. Oram, "Rectification for any epipolar geometry," in *12th British Machine Vision Conference (BMVC)*, pp. 653–662, 2001.
20. H. Hirschmuller, "Accurate and efficient stereo processing by semi-global matching and mutual information," in *IEEE Conf. on CVPR*, vol. 2, pp. 807–814, 2005.
21. V. Kolmogorov and R. Zabih, "Computing visual correspondence with occlusions using graph cuts," in *International Conference on Computer Vision*, pp. 508–515, 2001.
22. J. Sun, H. Y. Shum, and N. N. Zheng, "Stereo matching using belief propagation," *IEEE Transaction on Pattern Analysis and Machine Intelligence*, vol. 25, no. 7, pp. 787–800, 2003.
23. S. Birchfield and C. Tomasi, "Multiway cut for stereo and motion with slanted surfaces," in *International Conference On Computer Vision*, pp. 489–495, 1999.
24. D. Scharstein and R. Szeliski, "Stereo matching with non-linear diffusion," *International Journal of Computer Vision*, vol. 28, no. 2, pp. 155–174, 1998.
25. S. Birchfield and C. Tomasi, "Depth discontinuity by pixel-to-pixel stereo," in *International Conference On Computer Vision*, pp. 1073–1080, 1998.
26. A. Klaus, M. Sormann, and K. Karner, "Segment-based stereo matching using belief propagation and a self-adapting dissimilarity measure," in *In CVPR*, vol. 3, pp. 15–18, 2006.
27. Y. Wei and L. Quan, "Region-based progressive stereo matching," in *In Conference on Computer Vision and Pattern Recognition*, pp. 106–113, 2004.
28. Q. Yang, L. Wang, R. Yang, H. Stewenius, and D. Nister, "Stereo matching with color-weighted correlation, hierarchical belief propagation and occlusion handling," in *In CVPR*, pp. 2347–2354, 2006.
29. E. Trucco and A. Verri, *Introductory techniques for 3-D computer vision*. Prentice Hall, 1998.
30. R. I. Hartley and P. Sturm, "Triangulation," in *In Proceedings of ARPA Image Understanding Workshop*, pp. 957–966, 1994.
31. J. Y. Bouguet, *Camera calibration toolbox for MATLAB*. [http://www.vision.caltech.edu/bouguetj/calib\\_doc/](http://www.vision.caltech.edu/bouguetj/calib_doc/).
32. D. C. Brown, "Close-range camera calibration," *Photogrammetric Engineering*, vol. 37, no. 8, pp. 855–866, 1971.
33. J. G. Fryer and D. C. Brown, "Lens distortion for close-range photogrammetry," *Photogrammetric Engineering and Remote Sensing*, vol. 52, no. 1, pp. 51–58, 1986.
34. D. C. Brown, "Decentering distortion of lenses," *Photometric Engineering*, vol. 32, no. 3, pp. 444–462, 1966.
35. S. Hasan and K. Lee, "Interfacing to a stratix FPGA using the blackfin parallel peripheral interface," tech. rep., Chameleonic Radio Technical Memo No.7, Virginia Tech, 2006.
36. G. G. Filho and Y. Aloimonos, "An optimal time-space algorithm for dense stereo matching," tech. rep., CS-TR-4839, Department of Computer Science, University of Maryland, 2003.