CMU Very Fast Range-Imaging System

Shigeyuki Tada, Andrew Gruss
and **Takeo Kanade**

10 October 1993
CMU-CS-93-179

School of Computer Science
Carnegie Mellon University
Pittsburgh, Pennsylvania 15213-3891

Abstract

We present a high-speed range-imaging system based on a VLSI computational sensor developed in the CMU Computer Science Department. The VLSI range sensor is a custom chip consisting of an array of cells which combine photo-sensing and computation. Unlike conventional "step-and-repeat" light-stripe range finders, our sensor gathers range images in parallel as a scene is swept by a continuously moving plane of light. A prototype range-finding system has been built using a second-generation sensor and is capable of acquiring a 32 $\times$ 32 point frame of 3-D measurements in a millisecond — two orders of magnitude faster than currently available range-finding systems. The accuracy and the repeatability of the acquired range data has been measured to be less than 0.2%.

In this paper, we discuss the range-finding system and present experimental results that measure its performance.

This research was supported in part by an AT&T Foundation Grant, the National Science Foundation, under grant MIP-8915969, and the Advanced Research Projects Agency, ARPA Order No. 7511, monitored by the NSF under grant MIP-9047590. The views and conclusions contained in this document are those of the authors and should not be interpreted as representing the official policies, either expressed or implied, of AT&T, NSF or the U.S. Government.

Keywords: Range imaging, VLSI, analog signal processing, computational sensing.

1. Introduction

Range-finding is a key component for many robotic applications. Numerous techniques have been developed to acquire the three dimensional profile of a scene as a range image[5]. Among these, light-stripe range-finding is one of the most robust and practical methods. A conventional light-stripe range finder operates in a step-and-repeat manner. The light-stripe, a plane of light, is projected at a known position, and an intensity image is taken using a CCD camera. Then the position of the stripe in the image is extracted. If the camera geometry is known, range information of the points lit by the stripe is obtained via triangulation. To acquire a complete range image the stripe is stepped and the process is repeated until entire scene has been scanned. Usually it takes on the order of a second to acquire a single range image. Such a long acquisition time has restricted the use of this type of range finder.

We developed two generations of cell-parallel light-stripe sensors, taking advantage of recent VLSI technology[1,2,3] to process the monitored intensity signal at each sensing element. As the light-stripe sweeps continuously over the scene, each element measures the time at which it sees the peak intensity reflected from a point on an object surface. As the stripe position is controlled precisely, the measured time is converted to the stripe position, which is used to determine the point in three-dimensional space where the object, the stripe and the sensing element line-of-sight ray intersect. In this method, the acquisition time is determined by time required for the stripe to sweep the scene, limited by the bandwidth of the photoreceptor used in the element. Arranging these sensing elements in a two-dimensional array, a range image can be acquired very rapidly. All sensing elements operate in parallel so that acquisition time does not depend on number of measuring points.

We incorporated the VLSI range sensors into a prototype range-imaging system[4]. This system consists of a sensor chip and its interface electronics, a light-stripe generating assembly, and a three degree-of-freedom (3-DOF) positioning system. We also developed a new calibration algorithm which maps the measured range data into the world coordinate system.

In Section 2, we present the cell-parallel algorithm which we employ and in Section 3 review its VLSI implementation. In Section 4, we describe the configuration of the prototype range-imaging system which we have developed. System calibration is presented in Section 5, and, in Section 6, we show experimental results and summarize the range system performance.

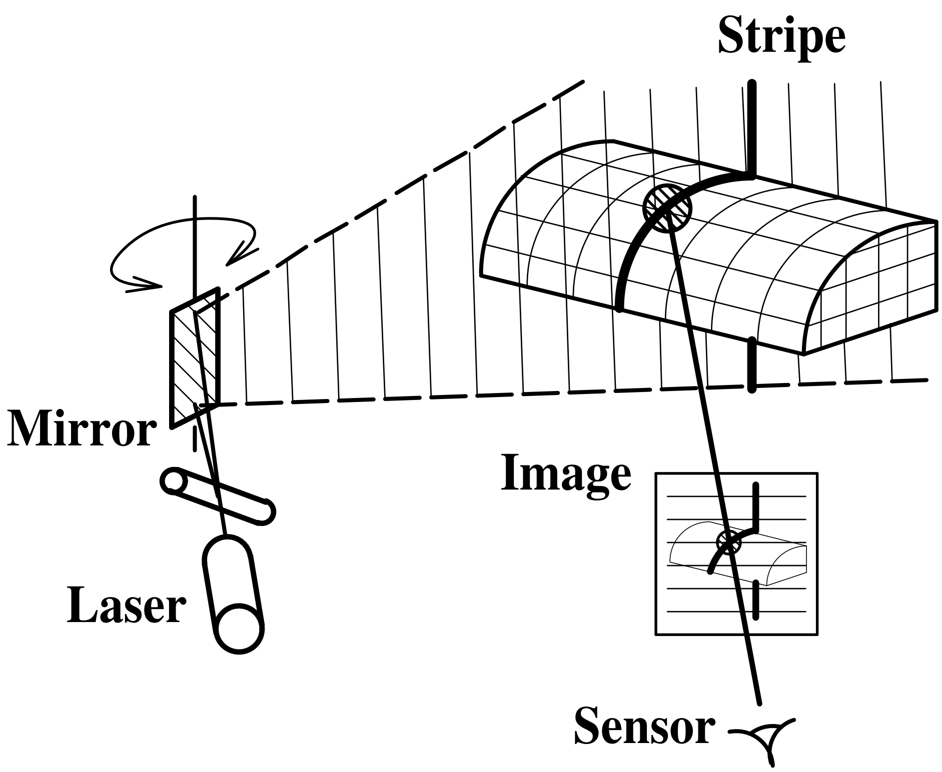


Figure 1. Geometry of a light-stripe range finder.

2. Cell-Parallel Light-Stripe Range Imaging

Figure 1 shows the principle on which a conventional "step-and-repeat" light-stripe range finder is based. When a light-stripe is projected onto an object, the sensor sees a bright line which follows the contour of the object surface intersected by the light stripe. If the projector and the imaging sensor geometry are known, range information at all points lit by the stripe can be determined via triangulation. To acquire a complete range image using this method, one must perform the following procedure in a step-and-repeat manner until entire scene is covered. First, a light stripe of known geometry is projected onto the scene. Second, range data from points lit by the stripe are extracted from an acquired intensity image. Finally, the stripe is stepped to a new position. If an $N \times N$ range image is desired, N intensity images, each at a different stripe position, must be acquired and processed. The total time T_f^{Step} to acquire a range frame is —

$$T_f^{Step} = N \cdot T_f^{Video}. \tag{1}$$

Assuming $T_f^{Video} = 1/30$ seconds and $N = 100$, $T_f^{Step} = 3.3$ seconds is required. More time is needed for higher spatial resolution range images. The long acquisition time is a serious problem for this method when applied to real-time problems, for example, manipulation of moving objects.

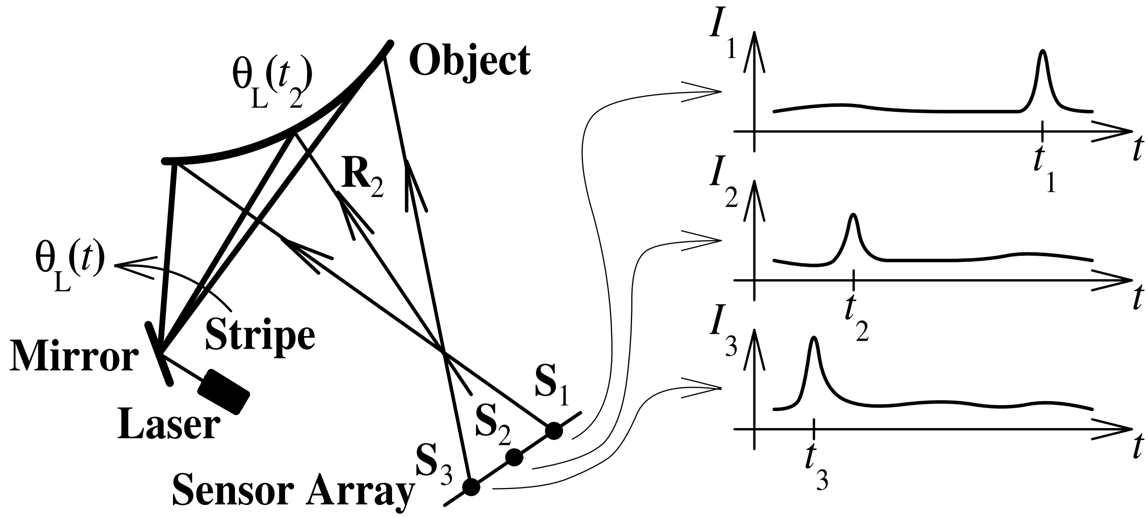


Figure 2. Cell-parallel light-stripe range imaging.

2.1. The Cell-Parallel Method

The cell-parallel technique, in which photoreception and signal processing are performed in each sensing element, is a better approach for light-stripe range-finding[6]. Figure 2 shows how a cell-parallel light-stripe range finder operates. In this figure, three sensors arranged in a line are shown. The light-stripe is perpendicular to the page and swept across the scene from right to left. Each sensor measures the *time* at which the stripe is seen along its line-of-sight ray. For instance, cell $\mathbf{S}_2$ monitors the light intensity $\mathbf{I}_2$ returned along its line-of-sight ray $\mathbf{R}_2$. The intensity peaks when the stripe crosses a point at which the line-of-sight ray intersects the object surface. If the projection angle $\theta_L(\mathbf{t}_2)$ is controlled precisely, range is a function of time $\mathbf{t}_2$. Cell $\mathbf{S}_2$ performs peak detection on the measured intensity and reports the *time-stamp* $\mathbf{t}_2$ at which the peak occurred. The three-dimensional coordinates of the object point are uniquely determined from the time-stamp as the intersection of the line-of-sight $\mathbf{R}_2$ with the stripe plane at $\theta_L(\mathbf{t}_2)$ on the surface of the object.

A sensor which acquires a range image can be built by arranging sensing cells in a two-dimensional array. The cells work in parallel during a single pass of the stripe over the scene. Therefore, acquisition time is independent of the spatial resolution —

$$T_f^{Cell} = T_f^{Stripe}. \tag{2}$$

The acquisition time T_f^{Stripe} of a cell-parallel sensor is determined by the bandwidth of the photoreceptor used in its sensing elements. The photodiodes used in our cell design have a bandwidth on the order of megahertz. So very high frame rates $(1/T_f^{Cell})$ are achieved.

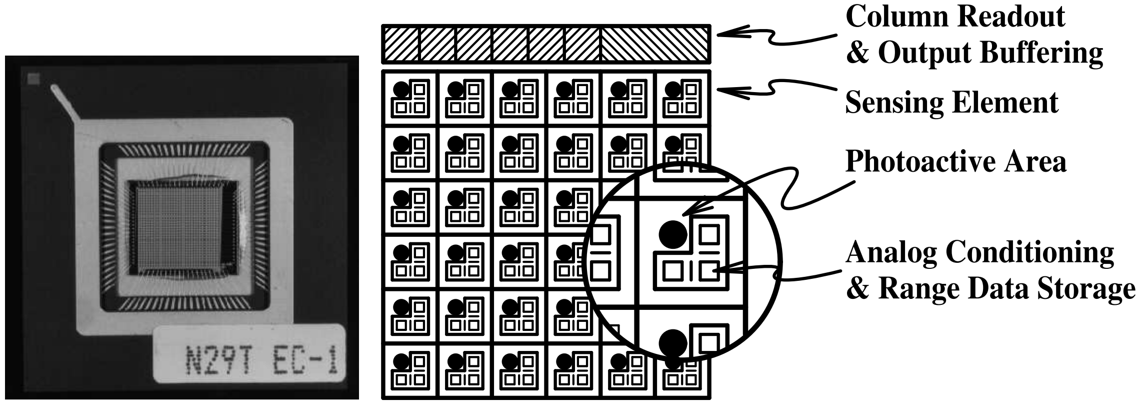


Figure 3. Range sensor integrated circuit (second generation).

3. The VLSI Range Sensor

In order to measure range using a very high-speed stripe, the interface between the photoreceptor and signal-processing circuitry in each sensing element must have high bandwidth. A computational sensor provides a local high-bandwidth connection in each element of what can be a large sensing-element array. Silicon VLSI technology provides the means for building such a sensor.

We have fabricated two generations of multi-pixel cell-parallel range sensors. In the first-generation chip, thresholding is used for peak detection (Appendix A). The second-generation range sensor, shown in Figure 3, incorporates several advantages over the first-generation design. The die area of the new cell is 216 μm $\times$ 216 μm, 40% smaller than that of the cells of the first-generation sensor (photoreceptor area has been kept constant). Stripe detection is done in a more robust manner and the range data readout circuitry has been simplified. In addition, the new cell provides a means to record and readout the peak intensity seen when a range-data sample is acquired. This intensity image is used in the system calibration process as described in a Section 5.1. The second-generation sensor chip has 1,024 range-sensing elements arranged in a 32×32 array.

3.1. Sensor Implementation

Figure 4 summarizes the operation of each element in our cell-parallel sensor array. The incident light intensity is converted into a voltage signal and the peak of this waveform is detected. The time at which the peak occurred (the time-stamp) is recorded in a memory element within the cell. This time-stamp is later read out of each element, providing raw range-image measurements.

Circuitry found in each sensing cell of the second generation chip is outlined in Figure 5. Shown are the photoreceptor (***PDiode***), photocurrent transimpedance amplifier (***Photo-Amp***), peak follower (***PeakHld***), comparison stage (***Comp***), and time-stamp sample-and-hold (***PGate***).

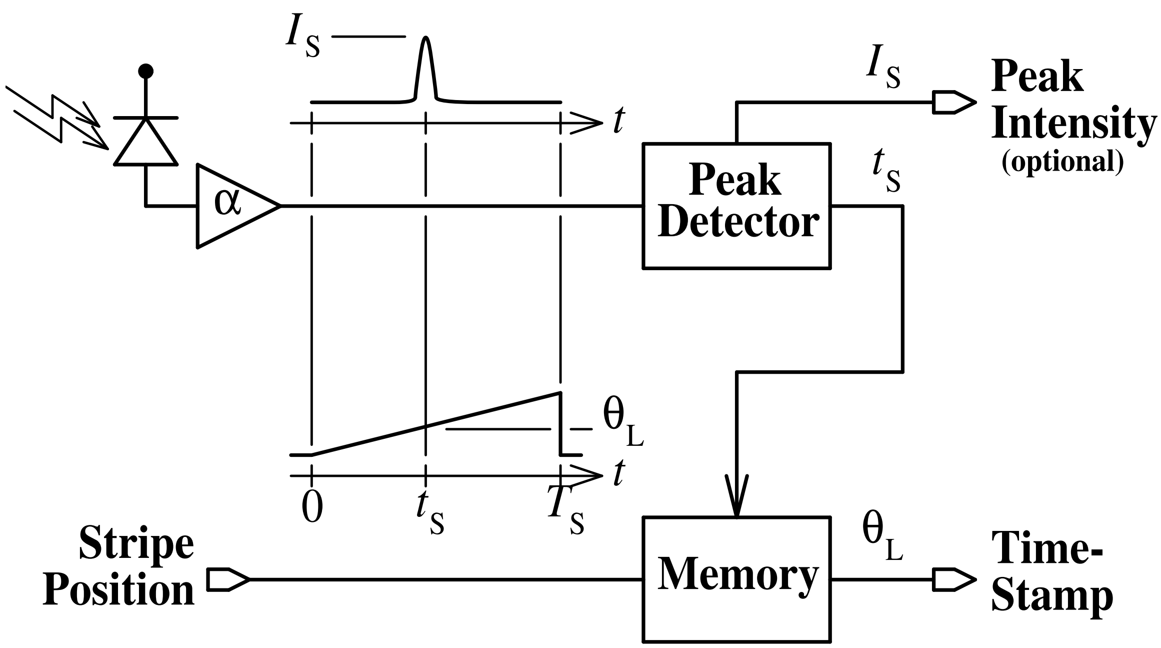


Figure 4. Basic sensing element block diagram.

3.2. Sensor Operation

In operation, the sensing cells cycle between two phases — *acquisition* and *readout*. During the acquisition phase, the light intensity from the stripe reflected back from the object is monitored. Photocurrent output is amplified and fed to the input of the peak-following circuit, recording ($\mathbf{V}_f$ on capacitor $\mathbf{C}_I$) the maximum light intensity observed during this stripe scan. When the intensity peaks, the value of the time-stamp voltage is held, recording the time of the stripe detection ($\mathbf{V}_{ts}$ on $\mathbf{C}_\theta$).

The acquisition phase ends when the stripe completes its scan. At that time, the cells have recorded a range image in the form of held time-stamp values, as an analog voltages stored on the capacitors found in each cell.

The readout phase immediately follows the acquisition phase. The raw range data is read from the chip in time-multiplexed fashion. Charge held in each cell is gated through pass transistors and integrated using an on-chip amplifier to produce a buffered output voltage. Cells are addressed for readout using row and column select lines. An $N \times M$ array is read using N row enables lines and M column enables. A given cell is read by asserting the row and column enables that correspond to its location in the array.

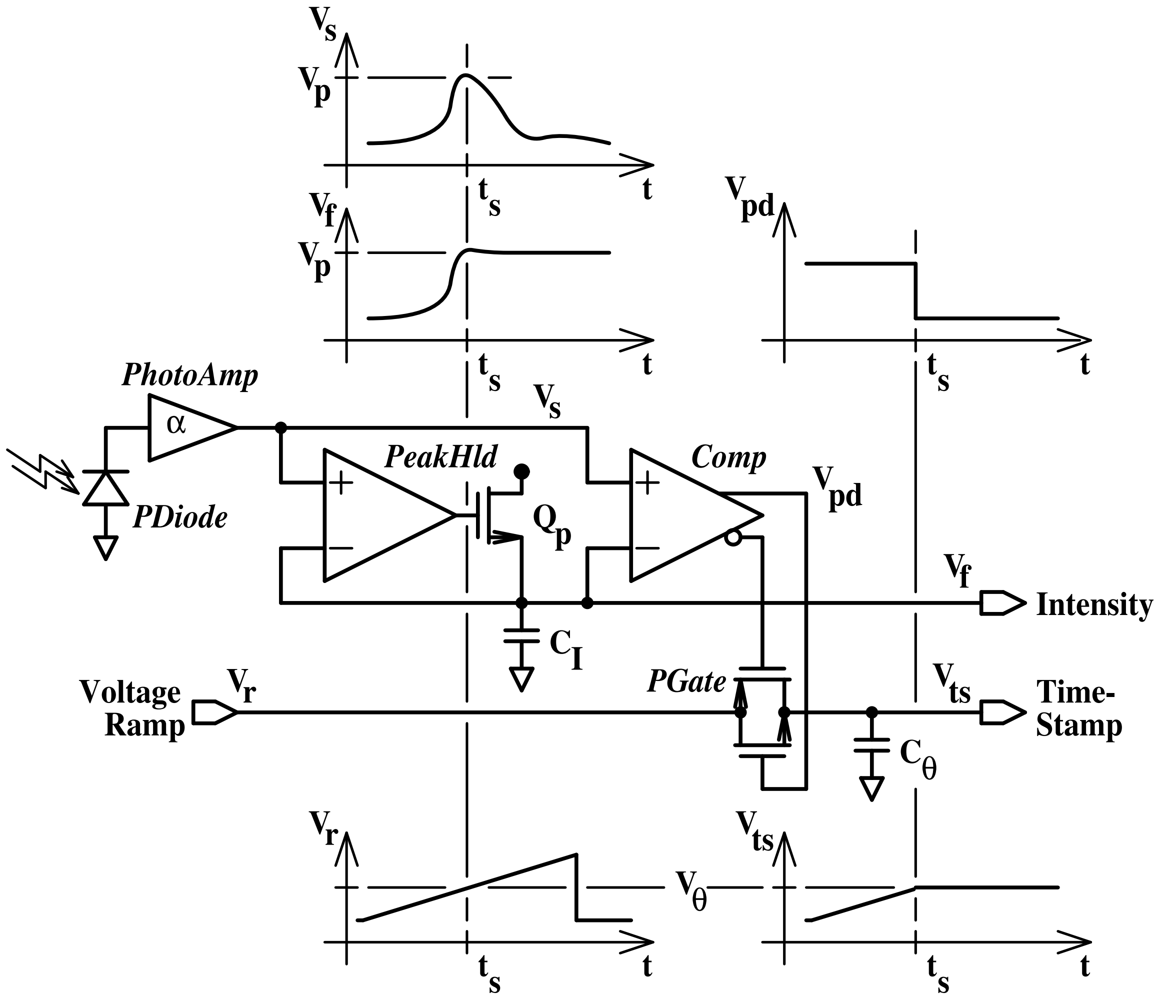


Figure 5. Sensing element circuitry (second generation).

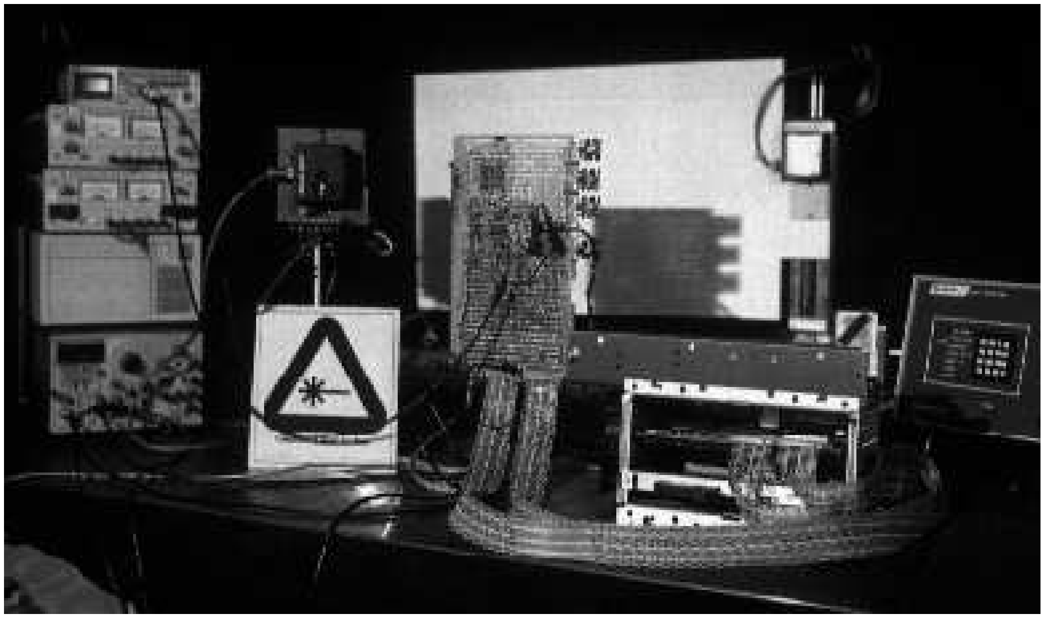


Figure 6. The cell-parallel range-finding system.

4. The Range-Imaging System

We developed a prototype range-imaging system using the VLSI range-sensor chip described in Section 3. Figure 6 shows the system which consists of a light-stripe generator (just left of center in the photo), the VLSI sensor chip and its interface electronics (center of the photo), a host computer (SUN Sparc-2, not shown), and a calibration target (white object in background) mounted on the 3-DOF positioning system (lower right). The system configuration is summarized in Table 1.

Baseline	300 mm	
Operating Depth Range	340 - 500 mm	
Laser Source	Laser Diode (Collimated)	
	Wavelength	830 nm
	Output Power	30 mW
	Stripe Width	1 mm
Sweep Assembly	Rotating Mirror	
	Actuator	Galvanometer
	Sweep Angle	40°
Sensor Optics	1” - Format CCD Lens	
	Focal Length	25 mm
	f - number	f / 1.4

Table 1. Cell-parallel sensor system summary.

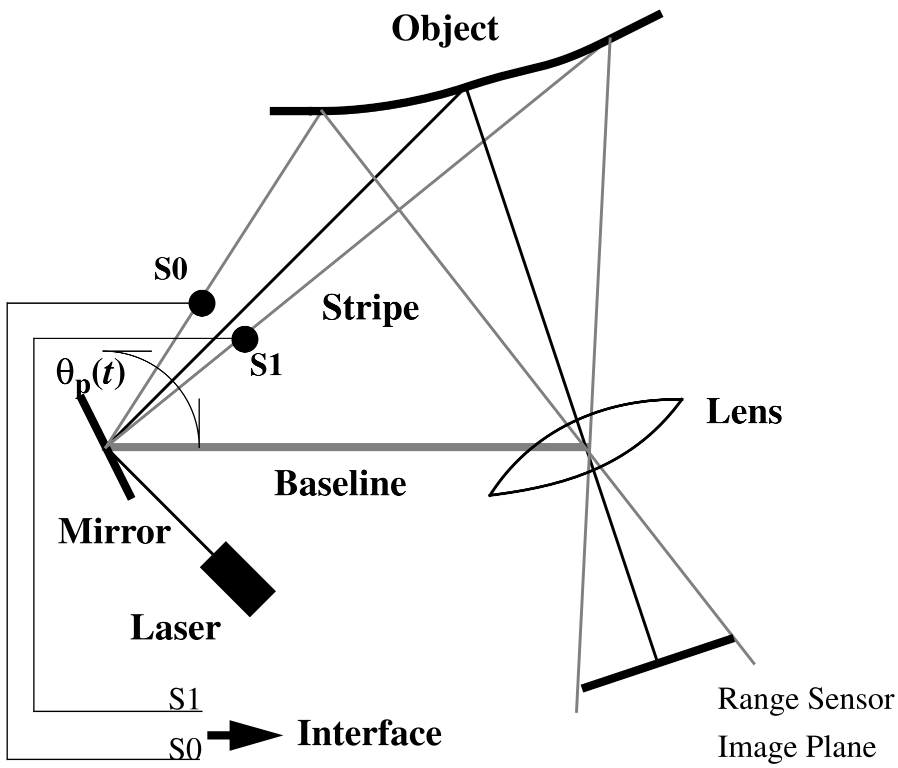


Figure 7. Scan detectors for system synchronization.

4.1. Scan Generation

The scan-generation assembly emits a plane of light and scans it over the scene. A laser diode, which emits near-infrared (780 nm) beam, is used as the light source. The beam from the diode is first collimated and then fanned into the stripe using a cylindrical lens. The stripe is swept using a mirror attached to a spinning motor. Sensing elements obtain range information by measuring the time at which each sees the projected stripe. Thus, the position of the light-stripe as a function of time must be controlled precisely.

As shown Figure 7, two photo sensors, **S0** and **S1**, measure stripe geometry. One provides a start-of-scan (SOS) (time origin) signal and the other an end-of-scan (EOS) (duration) signal. (For left-to-right scans, **S0** measures SOS and **S1** measures EOS.) **S0** and **S1** are located at a fixed angle of $\Delta\theta_{SS} = 40°$ around the axis of rotation of the motor. The system records the time, $\mathbf{t}_{\mathrm{SOS}}$ and $\mathbf{t}_{\mathrm{EOS}}$, when each detector sees the stripe. The angular velocity of the motor is constant and so stripe position $\theta p(t)$ is —

$$\theta_p(t) = \left(\frac{\Delta\theta_{SS}}{t_{\mathrm{EOS}} - t_{\mathrm{SOS}}}\right)(t - t_{\mathrm{SOS}}) = 2\omega_M(t - t_{\mathrm{SOS}}) . \qquad (3)$$

The time between SOS and EOS is measured precisely for each range measurement. The result is used to adjust the time-stamp data read from the sensor to compensate for variations in the speed of the stripe-scanning motor.

The angular velocity f_M required for the motor to achieve a scan-time of T_{SCAN} is —

$$\theta_p = \frac{1}{2} \cdot \frac{\Delta\theta_{SS}}{360°} \cdot \frac{1}{T_{SCAN}}. \tag{4}$$

The factor of $1/2$ appears because the reflected stripe swept two degrees for every one degree of motor rotation. Assuming $\Delta\theta_{SS} = 40°$ and $T_{SCAN} = 1$ ms, angular velocity $f_M = 3,300$ rpm is required. If a single-faceted mirror is used, the frame rate is only 60 Hz even though the acquisition speed is high. A polygonal mirror is one mechanism for increasing the frame rate. Our system uses another mechanism — a single-faceted mirror mounted on a galvanometer. The galvanometer is driven by a triangular wave provided from an external signal generator. A galvanometer-based sweep mechanism produces left-to-right and right-to-left scans. Both are used for range measurement. The relative timing of the SOS and EOS signals identify the direction of a given scan. By choosing the triangle wave frequency and the mirror excursion carefully, a high range frame rate is achieved while maintaining the one millisecond scan.

4.2. Sensor Interface

High-accuracy range measurements at high frame rates necessitates specialized interface electronics for the sensor chip. As described in Section 2.2, each sensing element records range as a time-stamp voltage held on a capacitor. During the acquisition phase, the time signal, a voltage ramp broadcast to each element, must be precisely synchronized with the stripe scan. Also, the held time-stamp voltage must be read out quickly to minimize errors due to charge leakage. Finally, a system flexible enough to be used with the first-generation sensor, as well as future sensor designs, without major modification was required.

To address these needs, we developed interface electronics based on a microcoded engine. This interface permits high-accuracy operation of the device at high speed. We achieved a range accuracy of 0.5 mm using one millisecond stripe scans using it. In addition, the interface supports readout of the intensity images provided by the second-generation chip and row/column based readout.

As shown in Figure 8, the chip interface consists of two boards — the *master board* and the *slave board*. The master board, which is installed in the host computer, controls the system operation and communicates with the host computer. The microcoded engine consists of a memory and a program counter, which is driven by an 8 MHz clock. Interface operations are programmed using microcode. This structure provides very fast operation and accurate time management. A memory is provided for data storage which enables the system to acquire up to 511 range image frames continuously without host interacting with the host.

The sensor chip and its optics are mounted on the slave board. Two 12-bit analog-to-digital (A/D) converters and two 12-bit digital-to-analog (D/A) converters are incorporated on this board. The signals from the scan detectors, **S0** and **S1**, are processed with logic found here to obtain the SOS and EOS signals. Sensor chips with different pin assignments are easily supported by rewrapping short wires.

The master and slave boards are connected by 10-bit half-duplex parallel buses, which

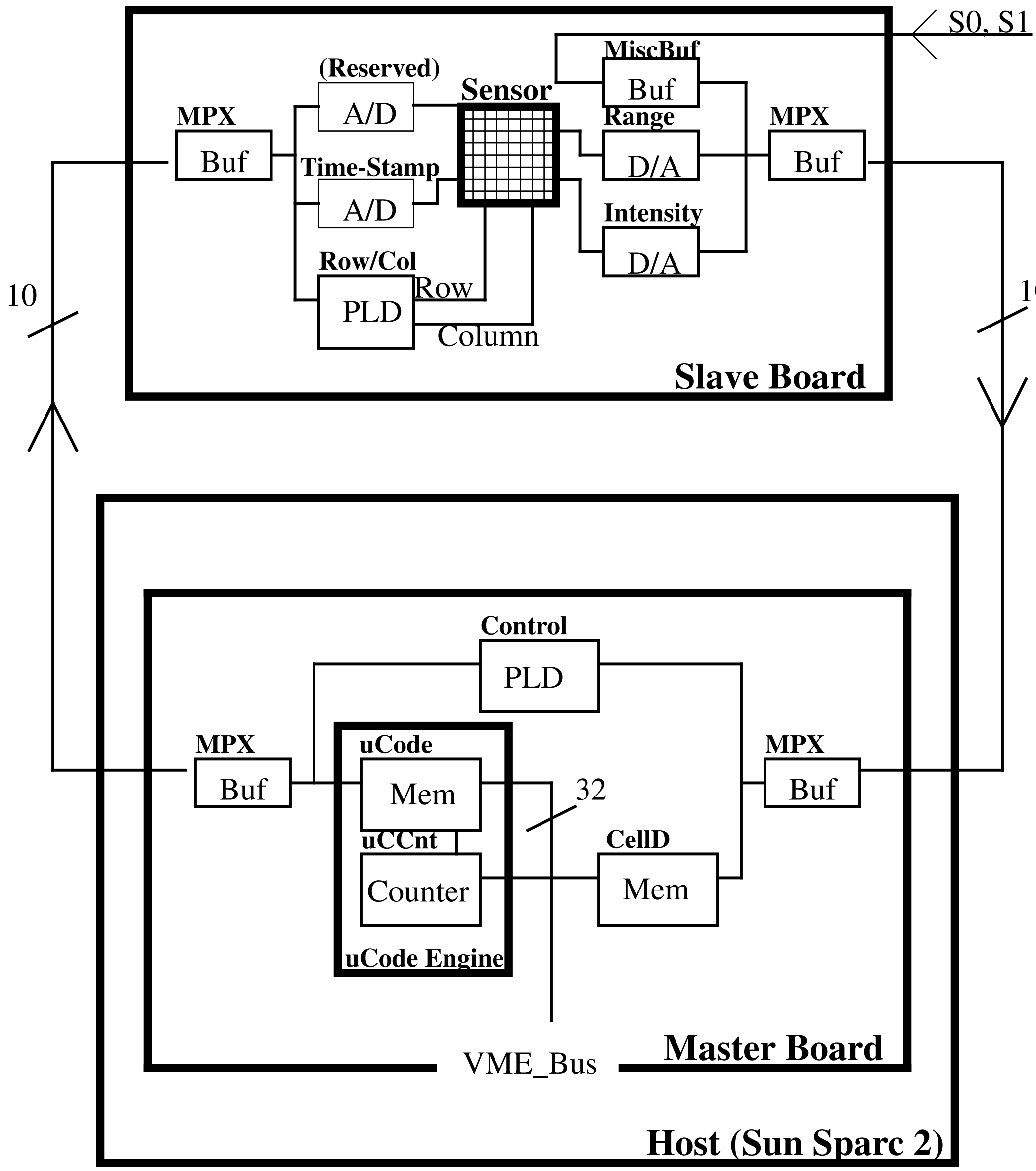


Figure 8. Chip-interface block diagram.

can transfer data at 10 Megabytes per second. Using this circuitry, all 1,024 sensor cells are read in approximately 3.5 msec, with the on-chip integration[1] and the A/D conversion dominating the readout time.

5. Calibration

Calibration provides the complete specification of system geometry necessary for converting cell time-stamp data into range data. The calibration parameters consist of two models. The first is the *imager model*, which describes the geometry and optical parameters of the sensor chip. The second is the *stripe model*, which maps time-stamp values to distance for all sensing elements.

5.1. Imager-Model Calibration

The imager model takes the form of a 4×3 projection matrix **C** which maps three-dimensional points (x, y, z) onto two-dimensional sensor-image plane points (u, v) [7]. This projection is written as —

$$u = \frac{xc_{11} + yc_{21} + zc_{31} + c_{41}}{xc_{13} + yc_{23} + zc_{33} + c_{43}} \tag{5}$$

and

$$v = \frac{xc_{12} + yc_{22} + zc_{32} + c_{42}}{xc_{13} + yc_{23} + zc_{33} + c_{43}}, \tag{6}$$

where C_{XX} are the components of matrix **C**. Practically, C_{43} is set to one, leaving eleven parameters to be solved for in this step. As sensing cells are arranged in square grid, the rows and columns of the array are interpreted as coordinates in the image plane. To perform the imager model calibration, a data set consisting of (x, y, z) and (u, v) is collected over the entire field of view using a planar reference object manipulated with the accurate 3-DOF positioning device built into the range system. From the measurements, parameters of C are determined using a least squared-error fitting.

Intensity images from the second-generation sensor are used to measure (u, v). (The intensity data was measured to be reliable enough (see Section 6.2) to be used for the imager-model calibration.) To collect the data, a white planar target, on which a black dot of 3 cm diameter has been drawn, is used as shown in Figure 9. The target is mounted on the positioner so that its surface is parallel to the world-*xy* plane. An intensity image is taken for each target position. The centroid of the dot in the image plane is calculated after thresholding the image to a binary image. Figure 10 shows the centroid detection result when the target is moved along the X_W or Y_W axes at a given Z_W. The dashed line in the plot shows a fitted line which represents the real position of the mark. The accuracy of the centroid detection (i.e — the error from the fitted line) was calculated to be within 0.1 pixels in the u and v directions. The standard deviation was less than 0.05 pixels. These results show that data collection using the intensity images provided by the second-generation sensor yield accurate and stable data sets for imager-model calibration.

In order to incorporate measured imager-model data, equations (5) and (6) are written in matrix form as follows —

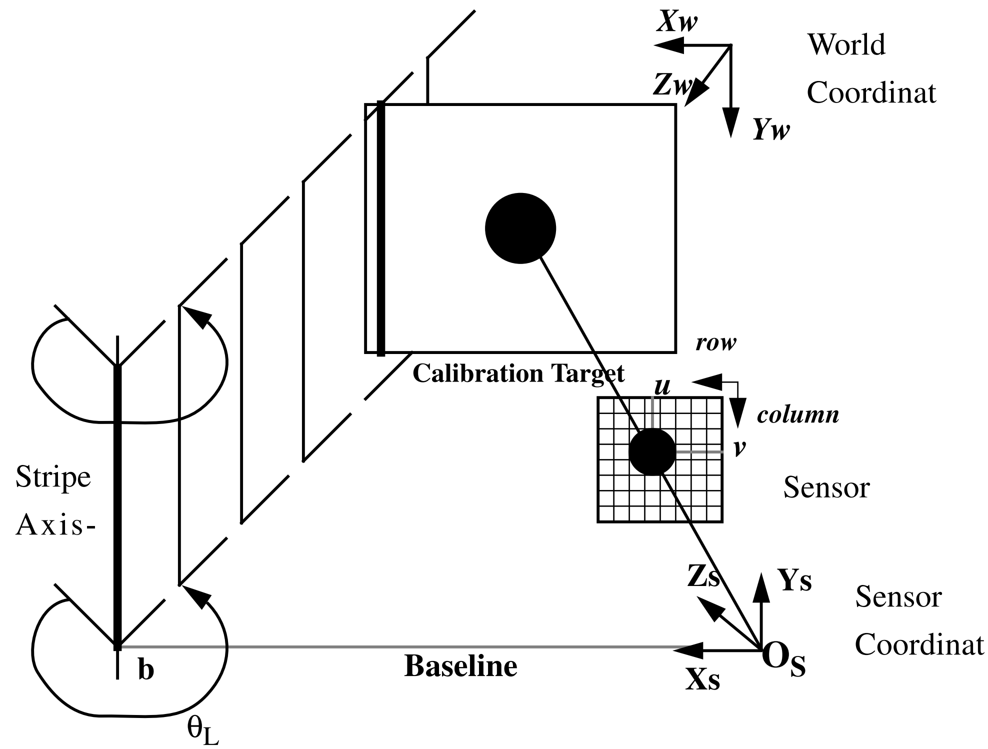
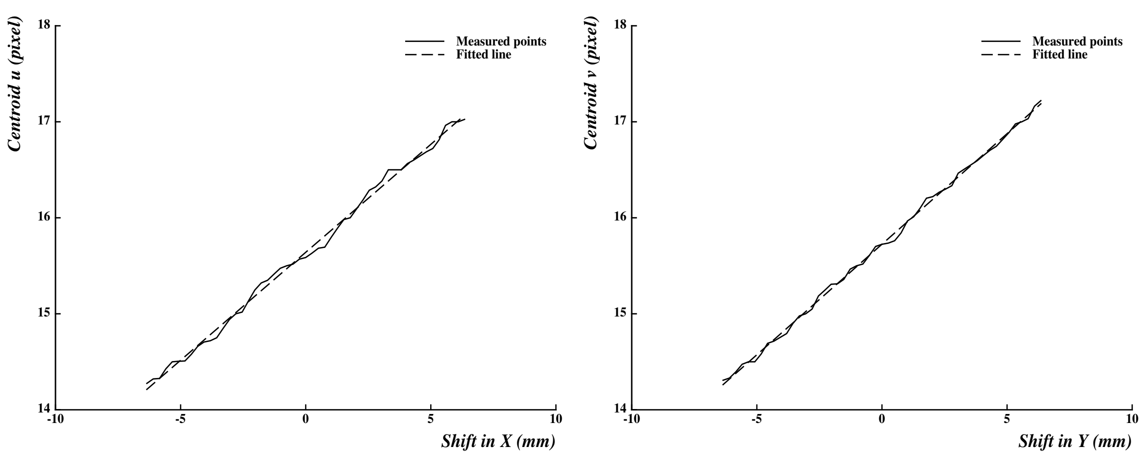


Figure 9. Imager-model calibration.



(a) Accuracy in *u* direction. **(b) Accuracy in *v* direction.**

Figure 10. Accuracy of mark detection.

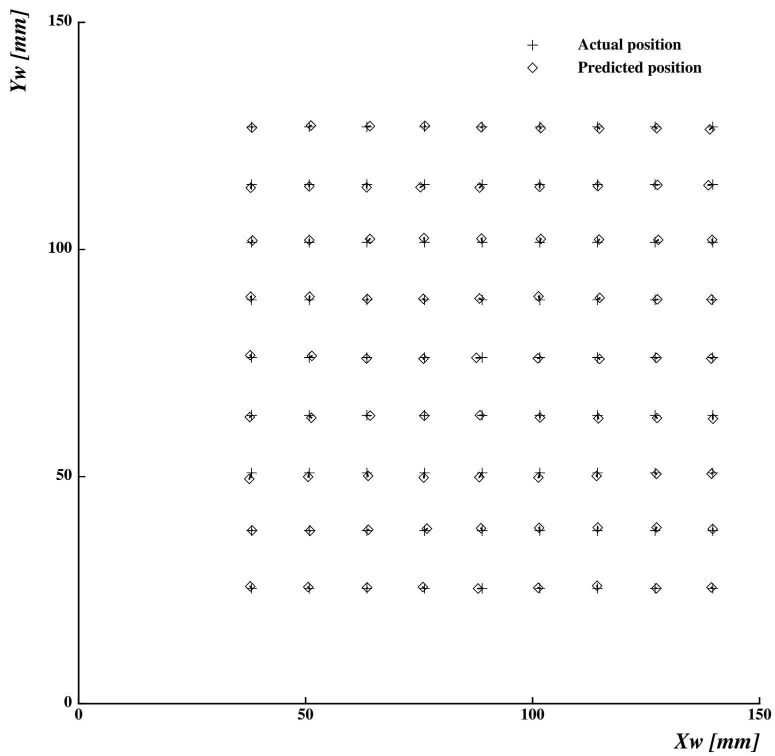


Figure 11. Imager-model calibration result at Z_W = 0.

$$\begin{bmatrix} x^1 & y^1 & z^1 & 1 & 0 & 0 & 0 & 0 & -u^1x^1 & -u^1y^1 & -u^1z^1 \\ 0 & 0 & 0 & 0 & x^1 & y^1 & z^1 & 1 & -v^1x^1 & -v^1y^1 & -v^1z^1 \\ x^2 & y^2 & z^2 & 1 & \dots & & & & & & \dots \\ \dots & & & & & & & & & & \dots \\ 0 & 0 & 0 & 0 & x^n & y^n & z^n & 1 & -v^nx^n & -v^ny^n & -v^nz^n \end{bmatrix} \begin{bmatrix} C_{11} \\ C_{21} \\ \dots \\ C_{33} \end{bmatrix} = \begin{bmatrix} u^1 \\ v^1 \\ \dots \\ u^n \\ v^n \end{bmatrix} \tag{7}$$

where (u^i, v^i) is the centroid detected for the ***i***-th target position (x^i, y^i, z^i) and n is the total number of measured points. If more than six points are used, the eleven calibration parameters are determined using a pseudo-inverse method (Appendix. C).

To evaluate the imager model obtained, we calculated the target position in world coordinates by substituting measured centroid position (u^i, v^i) and z^i into equations (5) and (6) and compared the computed result with measured centroids. Figure 11 shows the accuracy of the imager model at $Z_W = 0$ mm. A “+” indicates the actual target position (x^i, y^i) and “◊” shows the predicted position — a plot of (u^i, v^i) obtained via imager-model calibration. The averaged restoration error is 0.5 mm, roughly 0.1% of range, adequate for the desired 0.1% range measurement accuracy.

The imager-model geometry is now fully specified through calibration.

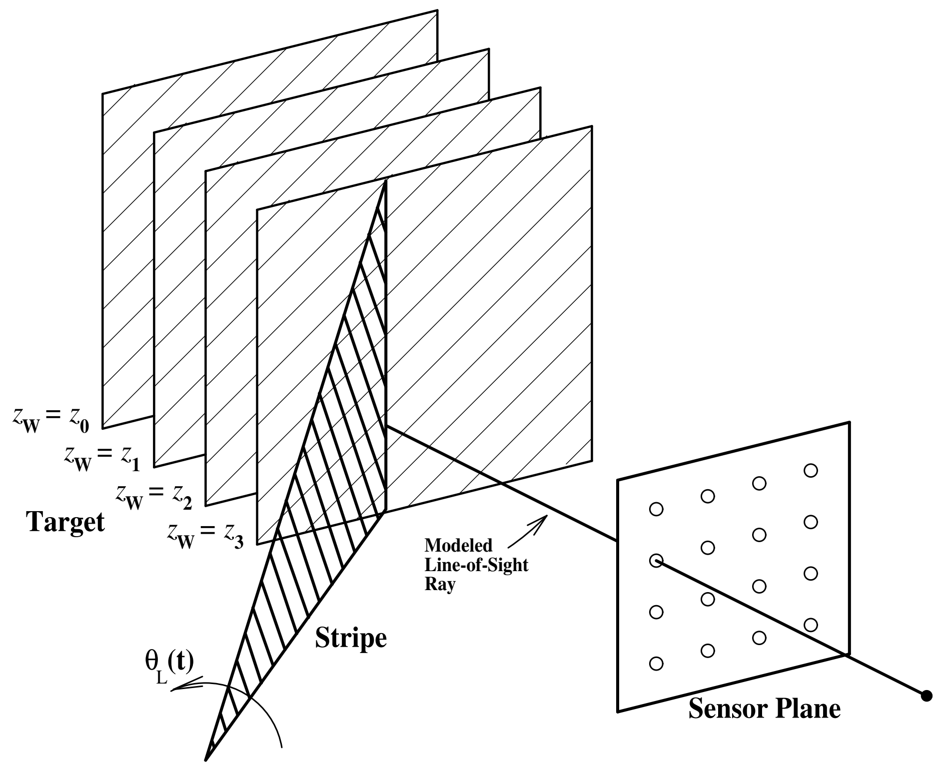
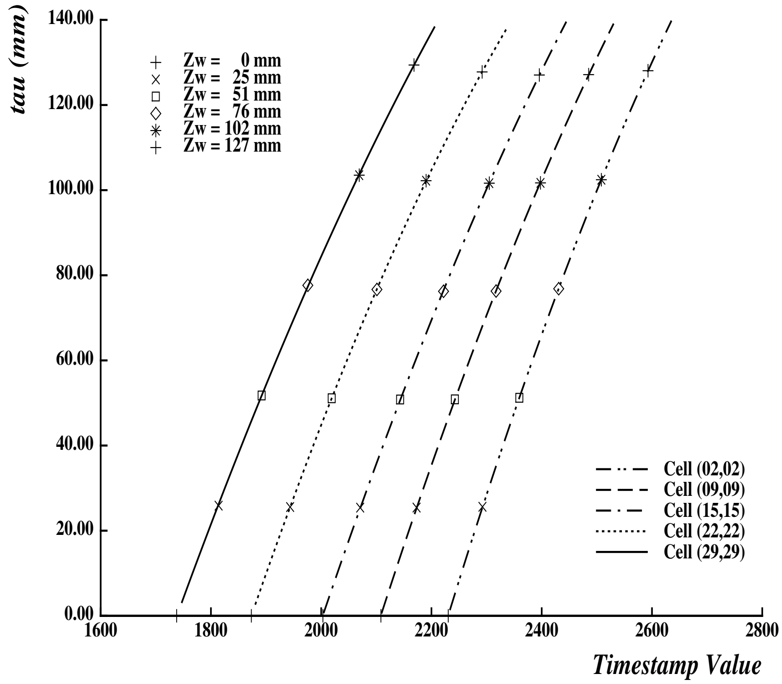


Figure 12. Stripe-model calibration.

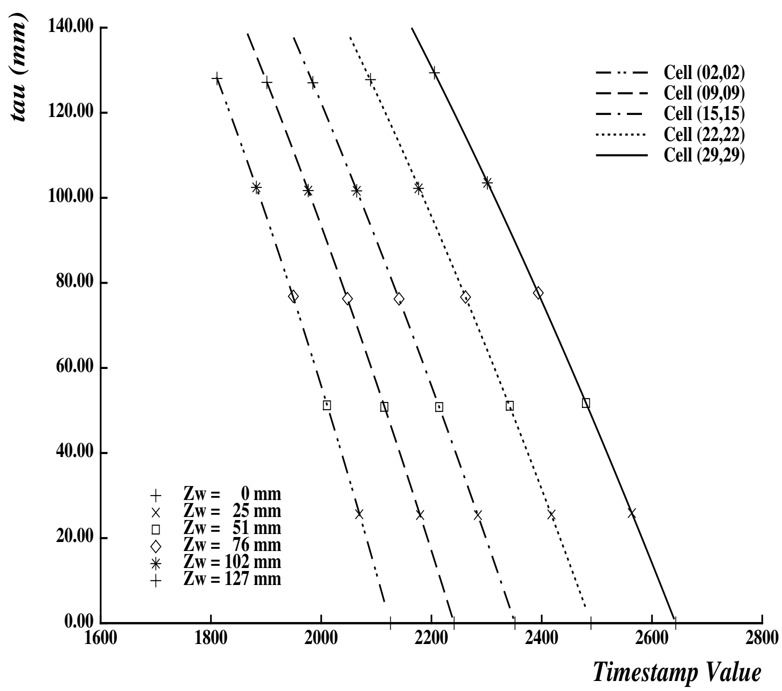
5.2. Stripe-Model Calibration

The second part of the calibration procedure determines the stripe-model — a mapping between time-stamp data and range along the line-of-sight ray of every sensing element. As shown in Figure 12, a planar white target is used to perform the calibration. The target is held at a known world-z position, parallel to the xy plane, and time-stamp readings θ_S from all sensors are recorded. This process is repeated for many z positions. Using this information, the function which maps the time-stamp value θ_S read from a given cell into the distance τ along the cell's line-of-sight is approximated via a parabolic fit. A system that scans the light stripe in two directions (left-to-right scan and right-to-left) needs two sets of parameters for each cell. During range measurement, the direction of each scan is recorded in order to specify the parameter set to be used for the range conversion. Experimental data, showing the fitted τ versus θ_S functions for several sensing elements, is shown in Figure 13.

Calibration of the cell-parallel range sensor is now complete.



(a) Left-to-right scan.



(b) Right-to-left scan.

Figure 13. Stripe-model calibration result.

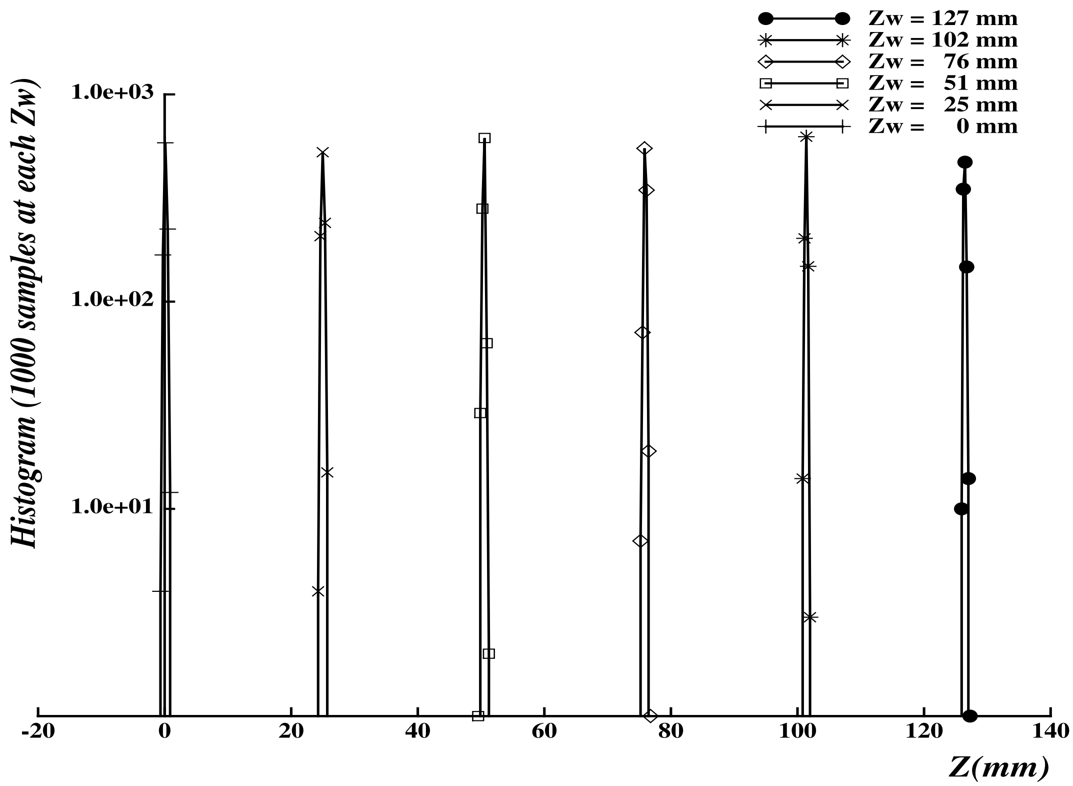


Figure 14. Cell (15, 15) range-data histograms.

6. System Performance

6.1. Range Accuracy and Repeatability

The quality of the range data produced by the cell-parallel range sensor was evaluated. In this measurement, a planar target was held at a known world-z position with the 3-DOF positioning device. The world-z axis heads almost directly toward the sensor with z_W = 500 mm roughly away. Analog time-stamp values from the sensor array were digitized, using a 12-bit analog-to-digital converter (A/D), and recorded for 1,000 trials. Light-stripe sweep (acquisition phase) time for each scan was 6 msec and the frame rate was 100 Hz (includes 4 msec/frame readout time).

A histogram of the range data reported by one cell is plotted in Figure 14. The horizontal axis represents the digitized time-stamp value, converted to world-z distance through the calibration model. Data for six world-z positions are combined in this plot. The vertical axis shows the number of times (plotted logarithmically) out of the 1,000 trials that the sensing element reported that world-z distance. The sharpness of each peak is an indication of the stability (repeatability) of the range measurements.

Averaged statistical data for all sensing elements is plotted in Figure 15. In order to measure accuracy and repeatability, the position of the target, as reported by the cell-paral-

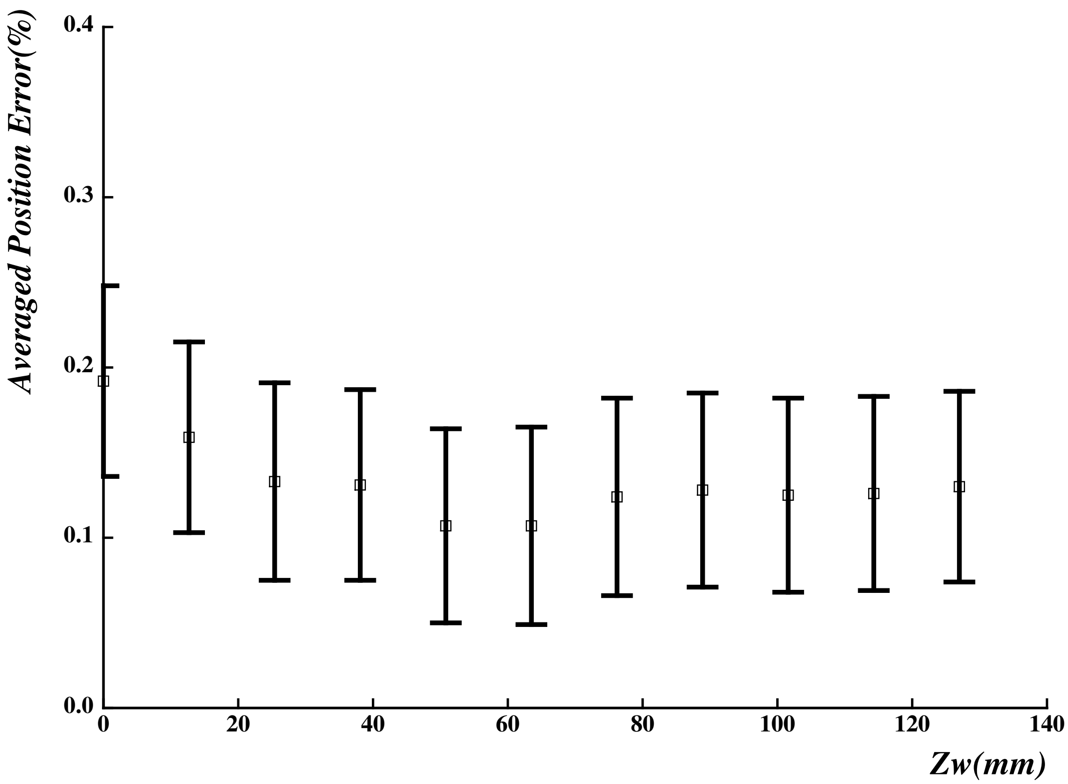


Figure 15. Range data accuracy and repeatability.

lel sensor, is compared to the actual target z position. The "boxed" points in the plot represent the mean absolute error, expressed as a fraction of the world-z position and averaged for all elements at Z_W. One standard deviation of "spread", also normalized with Z_W, is shown above and below each box.

The experiments show the mean measured range value to be within 1 mm at the maximum 500 mm z — an accuracy of 0.2%. The aggregate distance discrepancy between world and measured range value remains less than 1 mm over the entire 360 mm to 500 mm z range. The cell-parallel sensor repeatability is found by computing the standard deviation of the distance measurements. The measured repeatability of histogram data is less than 0.5 mm — 0.1% of the maximum 500 mm positioner translation.

6.2. Measurement of Scene Reflectance

As shown in Section 3.1, intensity data is acquired by the second-generation cells. This sensor directly measures scene reflectance as an artifact of the ranging process and is unlike intensity information returned by, for example, a CCD camera. In order to extract stripe timing, the peak intensity value observed during the stripe scan is recorded. The second-generation sensing cell provide the means to read these values from the chip. The scene is swept by a light source of known power and distance to the target object is measured. Thus, scene reflectance can be extracted from the peak intensity value provided. We use these "pseudo-intensity" images in the system calibration process (Section 5.1).

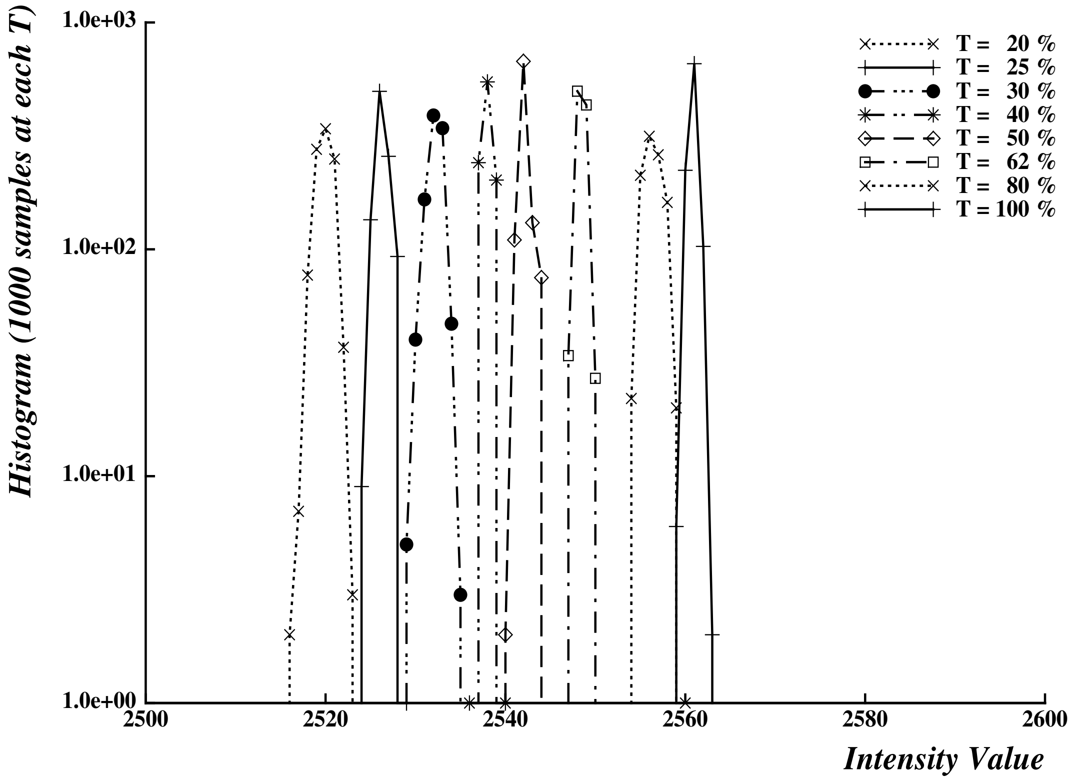


Figure 16. Intensity histogram at cell (15, 15).

The quality of the peak-intensity data has been evaluated. Figure 16 shows a histogram of the data measured by one cell. In this experiment, a planar target was placed a fixed distance from the sensor, roughly parallel to the world-*xy* plane. One thousand frames of intensity data were acquired. To control the incident intensity level from the stripe projector, neutral-density filters with known transmittance were placed between the stripe light source and the target. In the figure, the horizontal axis represents the intensity value (digitized to 12-bits) and the vertical axis shows the number of times a given 12-bit value was seen out of the 1,000 trials. Data for eight filters of different transmittance (the eight peaks) are combined in this plot (filter transmittance used for each peak is labeled “T”).

The incident intensity change is clearly seen in Figure 16. The sharp peaks in the plot indicate the high stability of detected intensity. The standard deviation for each measurement is calculated to be less than one bit (intensity-value gain used for the experiment could have been higher). This result supports the reliability of system calibration done using the pseudo-intensity images returned by the chip.

6.3. Range-Image Samples

Figure 17 shows a range image acquired by the sensor, plotted using a wire-frame representation. The imaged target is a hand placed in front of a planar background held perpendicular to the world-*z* axis. The range sensor is looking directly at the hand from a distance of approximately 500 mm. This range image was acquired using a one millisecond stripe scan. Two extended fingers and two folded fingers are clearly visible. Each grid

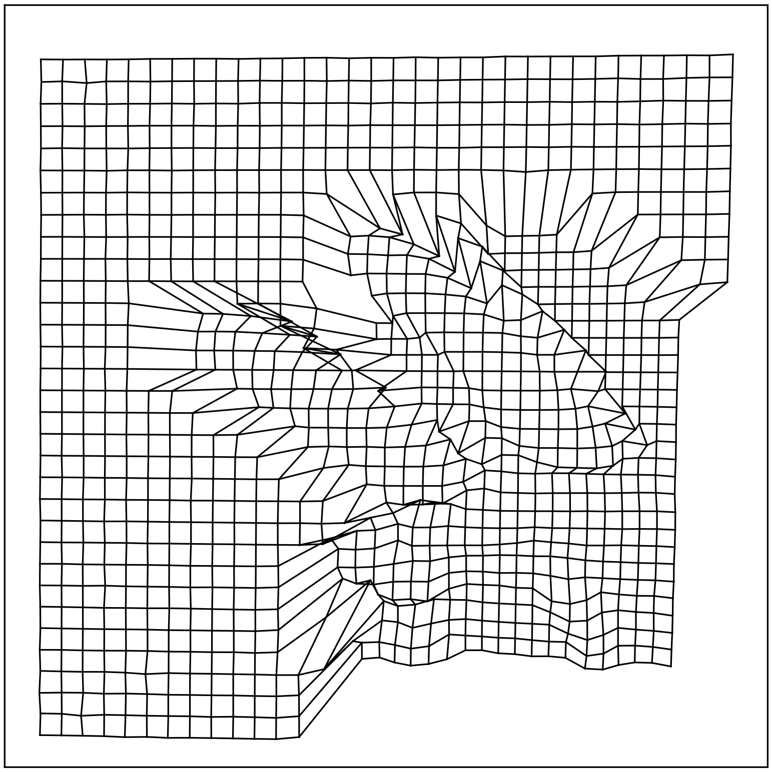


Figure 17. Range image (hand).

point, representing measured range data at a given sensing element, lies on the 3-D line-of-sight ray which passes through the center of the sensing element and optical center of the sensor lens. Grid point spacing becomes denser as object surface closes in on the sensor. Therefore, smaller “squares” represent object surface patches closer to the sensor and may make the object shown appear to be a “mold” of a hand at first glance. Figure 18 shows the pseudo-intensity (reflectance) image acquired at the same time. As described in Section 3.1, this image is the peak-intensity levels recorded by sensing cells during the stripe scan.

Figure 19 shows a “range movie” — a sequence of range images of a moving hand. The stripe scan time and the interval time (time between frames) were 850 microseconds and 15 msec, respectively. The result demonstrates the speed with which the sensor acquires range images.

6.4. Interdependence of Range and Intensity

In the second-generation sensor, the range data is dependent on the level of reflected light incident on the sensing cell. To measure how intensity and range data are interrelated, time-stamp value was measured with changing the intensity levels. Figure 20 (a) and (b) show the experimental results from one cell. In the experimental setup, a planar target was held parallel to the world-*xy* plane. The projected intensity level was changed by putting a neutral-density filter between the light source and the target. In Figure 20, the time-stamp value is plotted as a function of the measured intensity value. The target was placed at six

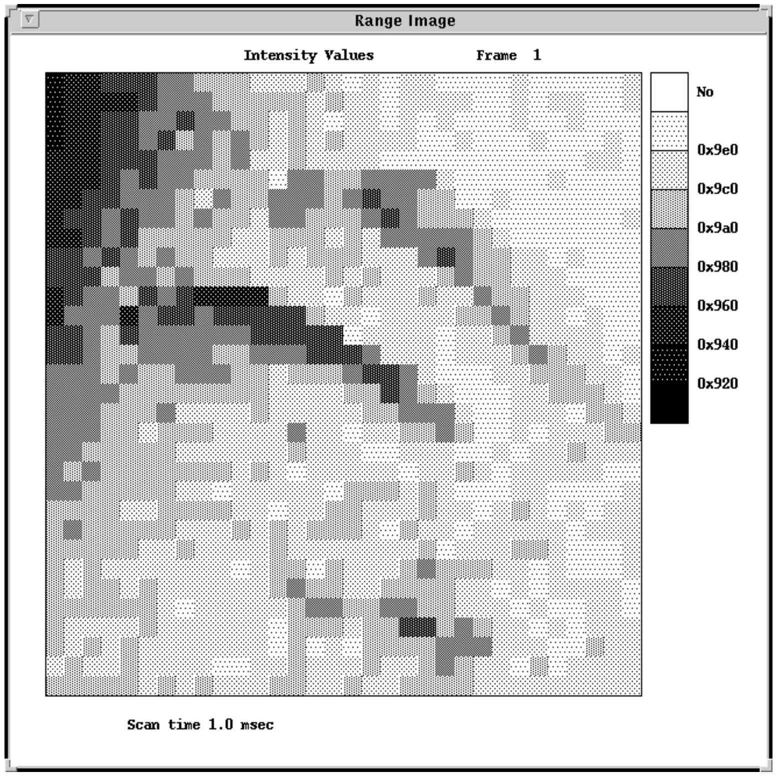


Figure 18. Intensity image (hand).

different world-z positions where the intensity level was varied using the neutral density filter set (transmittance was varied in a 20% to 100% range). The horizontal axis represents the digitized intensity value, and the vertical axis shows the digitized time-stamp value, converted to range using previously measured calibration parameters. The figures show that range measurement is affected by the intensity level. The problem is exacerbated at low intensity and at high stripe scan rates.

This phenomenon is caused by the comparator in the peak detector in the sensing element (see Section 3.1). More time is needed to decide that the intensity has peaked when the peak voltage of the input signal is low due to insufficient light. We attempted to compensate for this effect by correcting range measurement using the reported intensity value from the cell. However, we were unable to obtain an adequate model for the effect. The interrelation of range and intensity differs cell by cell and many factors cause intensity variation, for example, the scanning speed of the stripe, the distance between the light source and the object, the distance between the object and the sensor, the surface properties of the object.

When the stripe scan-time is reduced to 6.5 msec, the time-stamp value changes by only five counts of the 12-bit A/D output when the intensity changes from 50% to 100%. As per Figure 13, this corresponds to a 1.6 mm difference in the measured range. Furthermore, the intensity value is stable as shown in Figure 16 and can be used as a criterion during operation to judge the confidence of each acquired range data value.

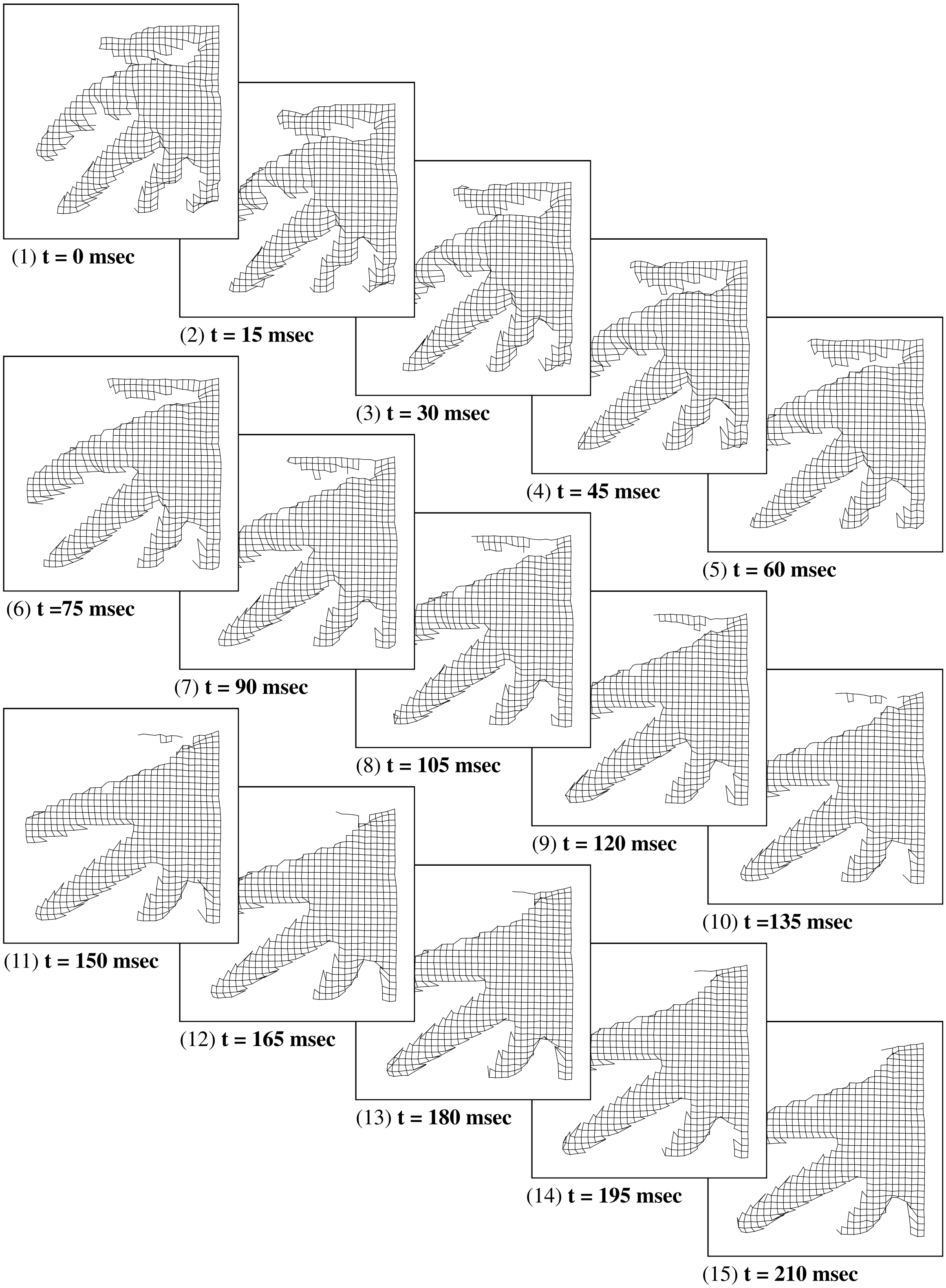
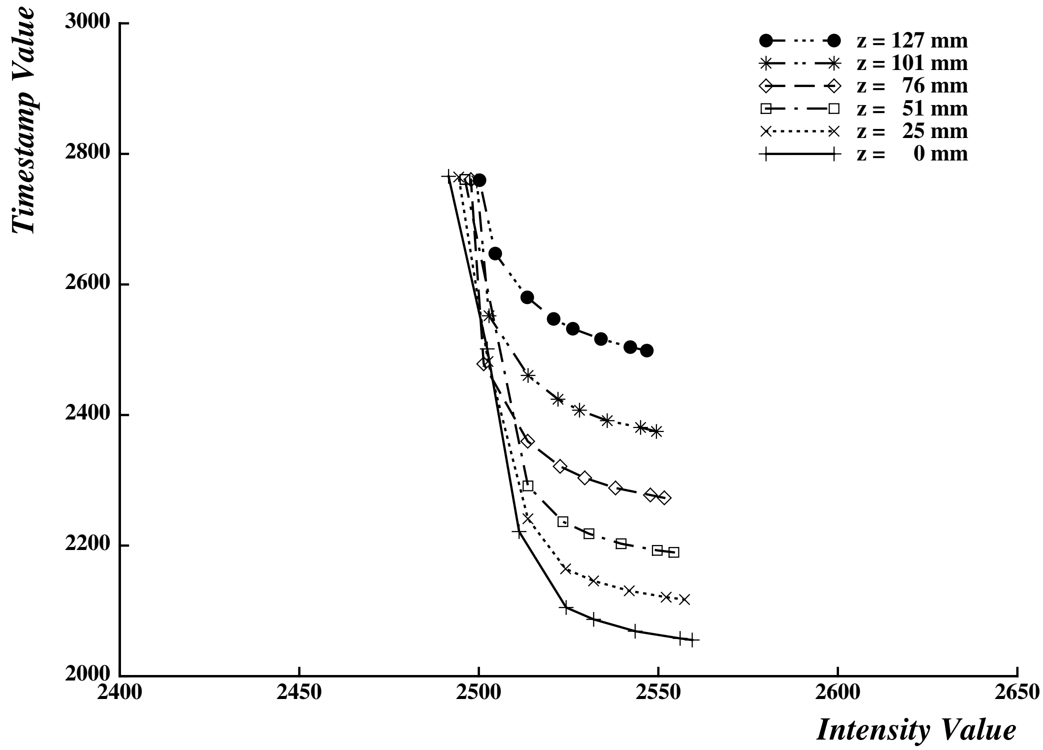
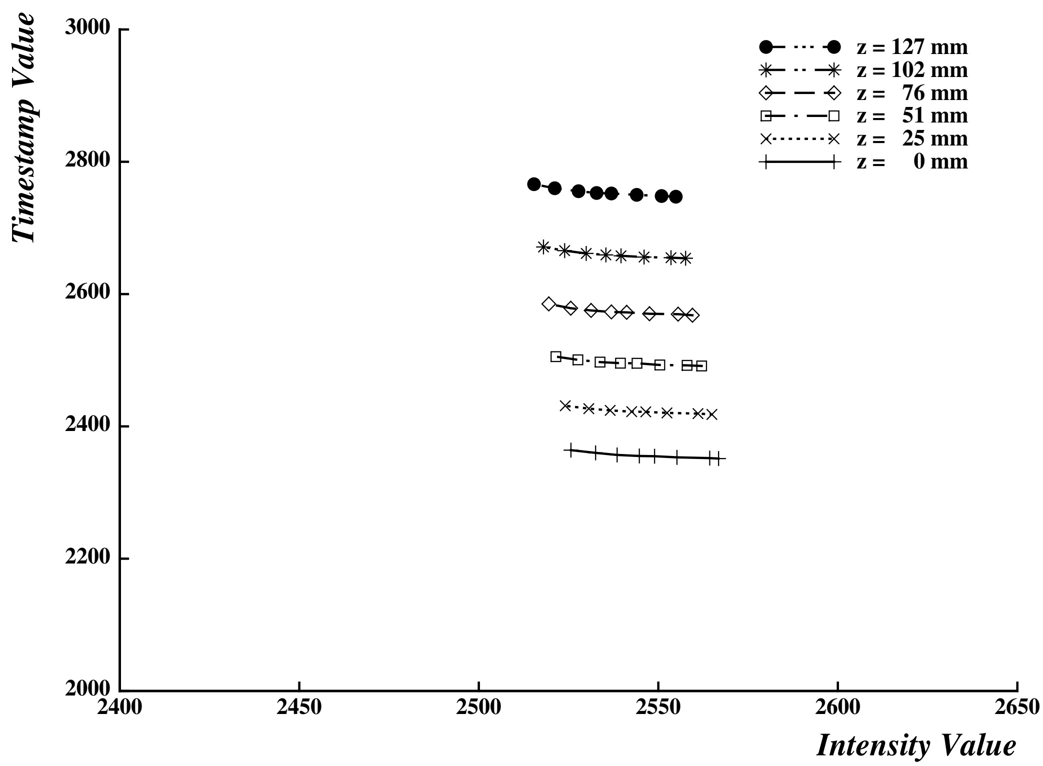


Figure 19. Range-image series (hand).



(a) Stripe scan time = 1.3 msec.



(b) Stripe scan time = 6.5 msec.

Figure 20. Intensity vs. time-stamp at cell (15, 15).

7. Conclusion

We have reviewed the design and implementation of a very high speed VLSI range-image sensor and have presented a high performance range-imaging system based on the device. This system acquires a 32×32 range image in a few milliseconds. We presented experimental results that show its range accuracy to be less than one millimeter at the systems 500 mm maximum operating distance. Repeatability of individual range measurements was within 0.5 mm. The cell-parallel ranging algorithm employed makes the spatial resolution of the device independent of the range image acquisition time.

The VLSI range sensor has demonstrated a successful application of the computational sensor approach, in which processing is performed locally on sensed data. Despite the promise of the computational sensing methodology, few practical implementations have been reported. The work reported here convincingly demonstrates the power of this approach.

Acknowledgment

We thank Rick Carley for his assistance and useful comments, Mark DeLouis for helping us build the scan-generation assembly, and David Simon for being the first to use the sensor in a practical application.

References

[1] A. Gruss, *A VLSI Smart Sensor for Fast Range Imaging*. Ph.D. thesis, Carnegie Mellon University, November 1991.

[2] T. Kanade, A. Gruss, and L.R. Carley, “A very fast VLSI remainder,” in *Proceedings of the 1991 IEEE International Conference on Robotics and Automation*, Sacramento, CA, pp. 1322-29, April 1991.

[3] A. Gruss, T. Kanade, and L. R. Carley, “Integrated sensor and range-finding analog signal processor,” *IEEE Journal of Solid-State Circuits*, vol. 26, pp. 184-191, March 1991.

[4] A. Gruss, S. Tada, and T. Kanade, “A VLSI smart sensor for fast range imaging’”, in *Proceedings of the 1992 IEEE/RSJ International Conference on Intelligent Robots and Systems*, Raleigh, NC, pp. 349-358, July 1992.

[5] P. J. Besl, “Range imaging sensors,” Research Publication GMR-6090, General Motors Research Laboratories, March 1988.

[6] K. Araki, Y. Sato, and S. Parthasarathy, “High speed rangefinder,” in *Optics, Illumination, and Image Sensing for Machine Vision II*, vol. 850, pp. 184-188, SPIE, 1987.

[7] D.H. Ballard and C.M. Brown, *Computer Vision*. Prentice-Hall, Inc., 1982.

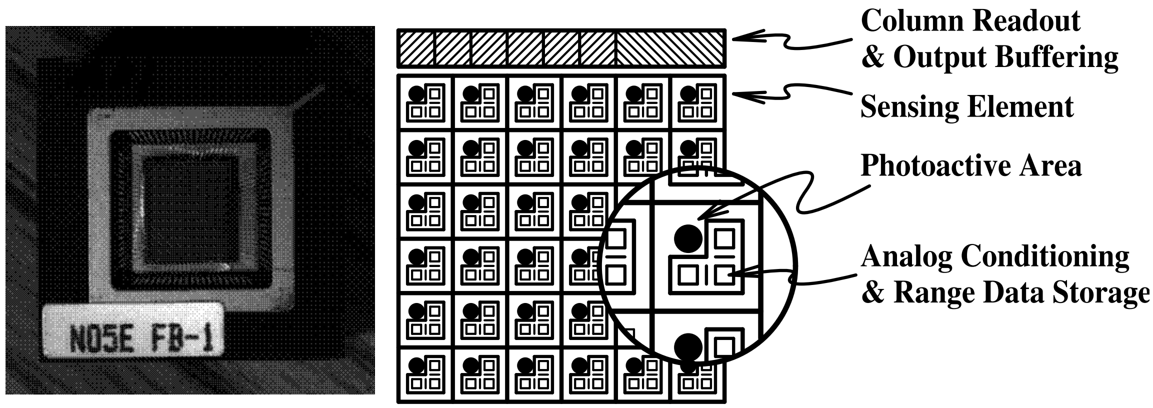


Figure A-1. Range-sensor integrated circuit (first generation).

Appendix A. The First-Generation VLSI Range Sensor

Our first-generation VLSI cell-parallel range sensor chip is shown in Figure A-1. This device consists of 896 sensing elements arranged in a 28×32 array. It was fabricated in a 2 μm p-well CMOS double metal, double poly process and measures 9.2 mm × 7.9 mm (width × height). Of the total 73 mm^2 chip area, the sensing element array takes up 59 mm^2, readout column-select circuitry 0.37 mm^2, and the output integrator 0.06 mm^2. The remaining 14 mm^2 is used for power bussing, signal wiring and die pad sites.

The architecture chosen for the range sensing elements is shown in Figure A-2. Areas of interest in the diagram include the photoreceptor (**`PDiode`**), the photocurrent transimpedance amplifier (**`PhotoAmp`**), threshold comparison stage (**`n2Comp`**), stripe event memory (**`RS_Flop`**), time-stamp track-and-hold circuitry (**`PGateI/CCell`**) and cell readout logic (**`PGate0/TokenCell`**).

In operation, sensing elements cycle between two phases — *acquisition* and *readout*. During the acquisition phase, each sensing element implements the cell-parallel procedure of Figure 4. The photodiode within each cell monitors light energy reflected back from the scene. Photocurrent output is amplified and continuously compared to an external threshold voltage $\mathbf{V_{th}}$. When photoreceptor output exceeds this threshold, the “stripe-detected” latch in the cell is tripped, holding the value of the time-stamp voltage at that instant.

The acquisition phase is synchronized with stripe motion and ends when the stripe completes its scan. At that time, the sensing elements of the array have recorded a range image in the form of held time-stamp values. The raw range data must now be read from the chip.

A time-multiplexed readout scheme off-loads the acquired range image in raster order through a single chip pin. Held time-stamp charge is gated onto a two-level analog readout bus using dual n/p-transistor pass gate structures. They permit the use of rail-to-rail time-stamp voltages, maximizing the dynamic range of the analog time-stamp data. The charge is integrated on chip to present a buffered output voltage on the time-stamp raster output pin.

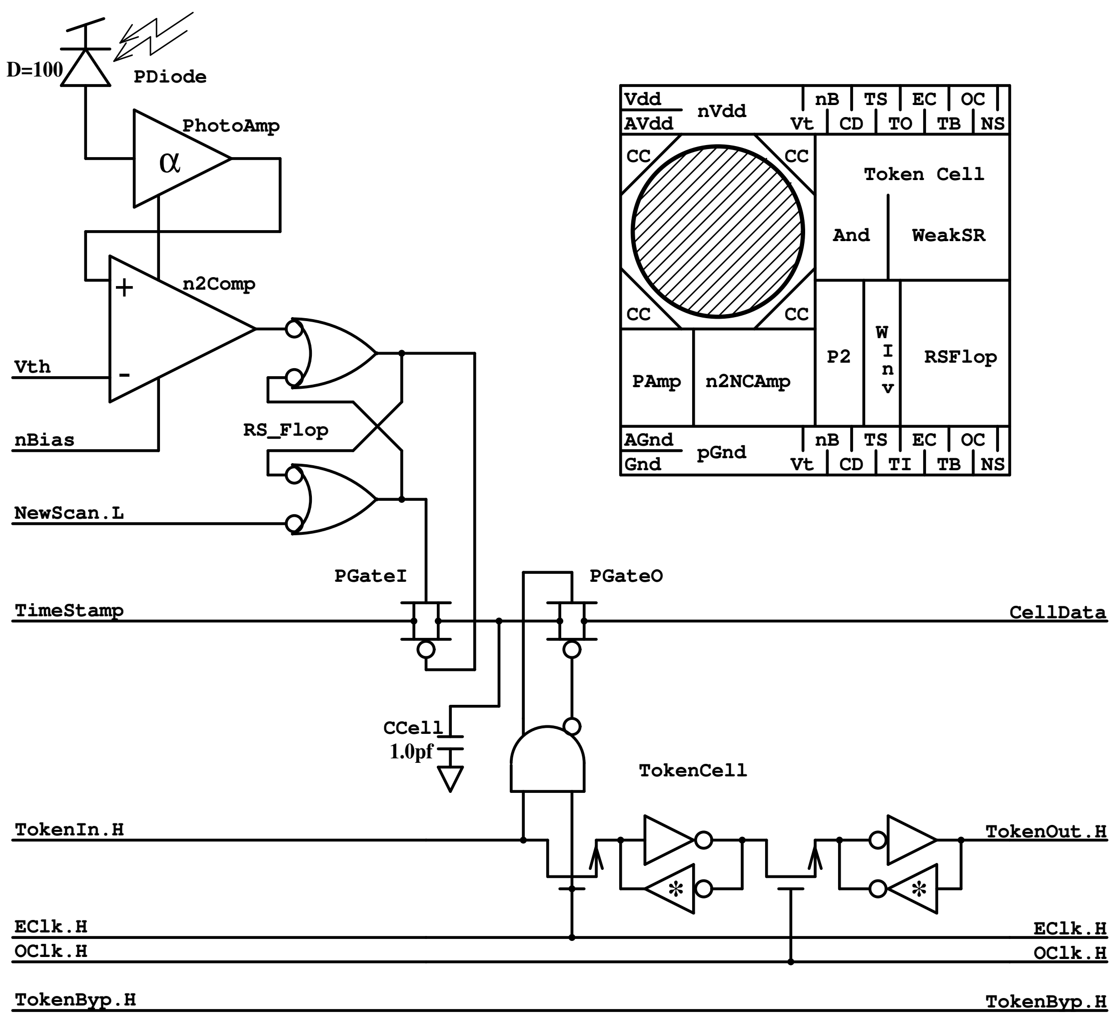


Figure A-2. Sensing element circuitry

Appendix B. Sensor Interface Details

The chip interface consists of two boards — the *master board*, which controls the system operation and communicates with the host computer, and the *slave board*, which contains delicate analog circuitry. The entire system is controlled by an microcoded engine implemented on the master board.

B.1. Circuitry Overview

Figure B-1 and Figure B-2 outline the interface. The master board is installed in the host computer, and the sensor chip and its optics are mounted on the slave board. Two 10-bit half-duplex parallel buses connect these two boards, which can transfer data up to 10 megabytes per second in each direction.

Commands issued from the host cause execution of microcode on the master board (Figure B-1). The microcode is held in memory "**`uCode`**" and sequenced via program counter "**`uCCnt`**". Microcode memory is 32 bits wide. The upper eight bits are used as an input for a system clock prescaler (**`PSCnt`**), used to change the system cycle time on a clock by clock basis. The next eight bits are for control use, such as selecting an internal register. The lower twenty bits are multiplexed and sent to the slave board on every clock cycle. Data, such as range data or intensity data read from the sensor chip, appears on receiving buffers(**`RxDBuf`**). A memory (**`CellD`**) is provided for the range and intensity data storage and is capable of storing 511 continuous range data frames or 255 pairs of range/intensity image frames. The master interface uses a VME bus as its host interface. Microcode is loaded from the host into the microcode memory and the acquired range and/or intensity data is read into the host at the completion of a measurement.

Multiplexed 20-bit data arrives at receiving buffer (**`RxDBuf`**) on the slave board (Figure B-2). The lower sixteen bits are used in three different ways — as specified by upper four bits. During the acquisition phase, the data sent is the time-stamp ramp signal from at **`RampBuf`**, converted to a voltage by the 12-bit digital-to-analog converter (**`RampDAC`**). During the readout phase, the value specifies the current column and the row being read.

The amplifiers which buffer the range and intensity data have variable gain and offset under control of the microcode engine. During the readout phase, range and intensity data are digitized using the analog-to-digital converters "**`TSADC`**" and "**`IADC`**". A conversion time of 800 ns is required so the output bus is fast enough to transfer both data values in time-multiplexed fashion. The data is stored in buffer "**`TADCBuf`**" or "**`IADCBuf`**" after conversion and sent out to the master board through the transmitting buffers "**`TxDBuf`**".

B.2. Microcode Format and Execution

In operation, a micro instruction is read from the microcode memory every clock cycle. As shown in Figure B-3, each instruction consists of three fields — the *prescale* field (upper eight bits), the *control* field (next eight bits), and *data* field (lower sixteen bits). The prescale field is loaded into the system clock prescaler and determines the execution time

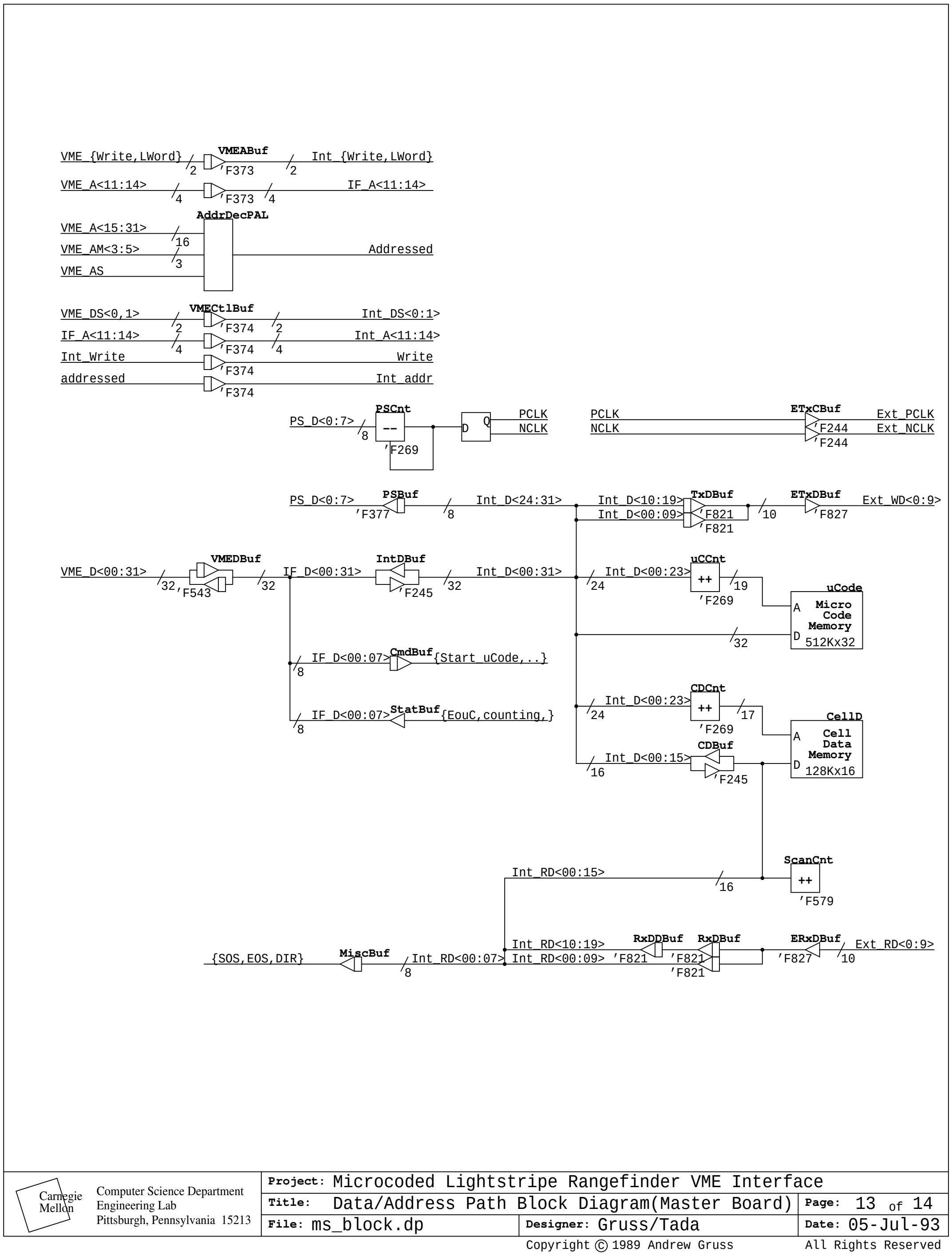


Figure B-1. Chip interface (master board).

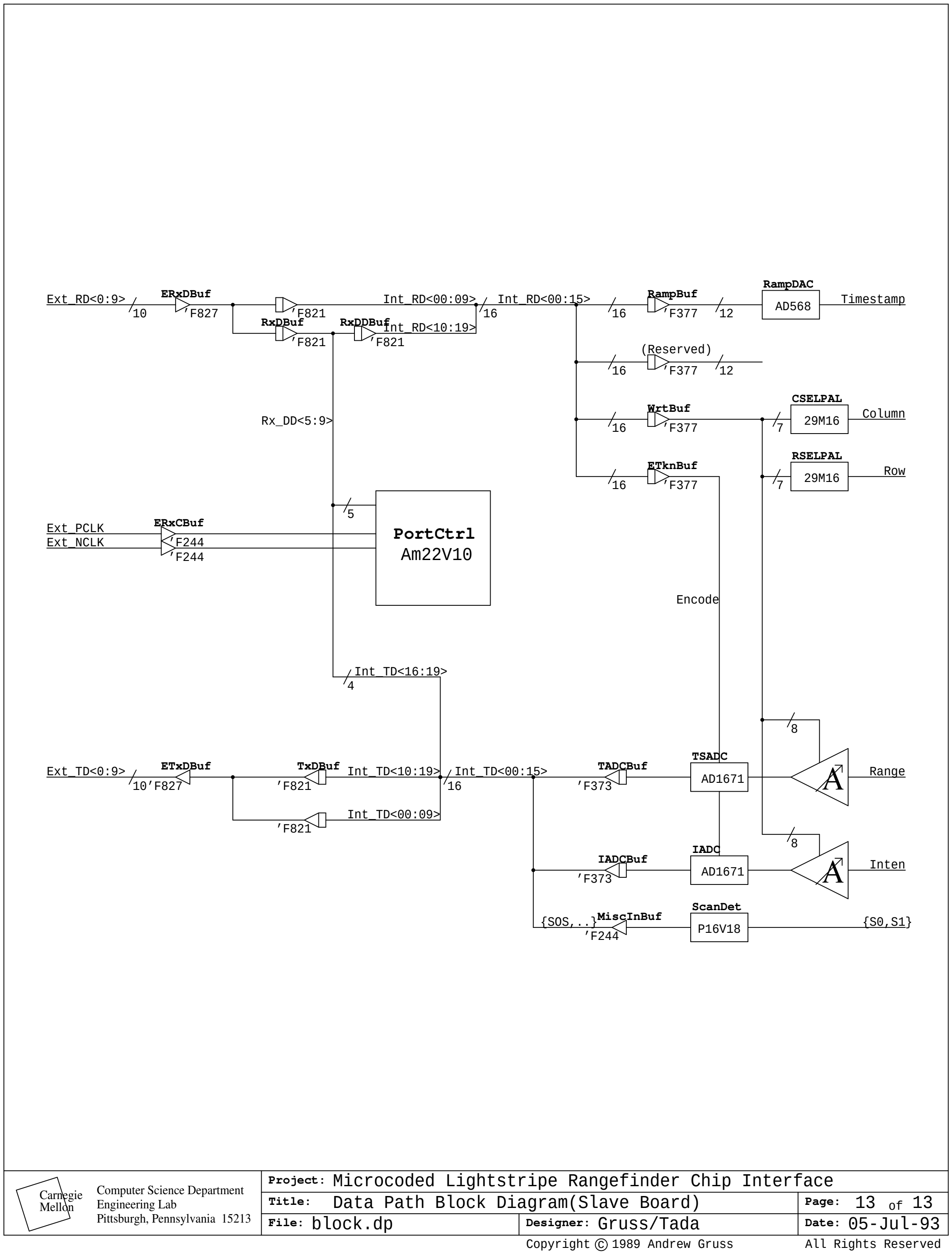


Figure B-2. Chip interface (slave board).

Bit Assignment

Bit	31 … 24	23 … 16	15 … 0
	Prescale	**Control**	**Data**

Port Select

Write operation

[D17 D16]	[D15]	Write Port	Data Field
[0 0]	[0]	WrtBuf	Row/Col Select, Gain/Offset Set
	[1]	Internal Control Port	Control Word
[0 1]		RampBuf	Time-Stamp Value (12-bit)
[1 0]		Not Used	
[1 1]		ETknBuf	Start A/D Conversion

Read Operation

[D19 D18]	Read Port
[0 0]	Not Used
[0 1]	MiscIn
[1 0]	Time-Stamp A/D
[1 1]	Intensity A/D

Figure B-3. Microcode format.

of the next instruction. As the system clock is 16 megahertz, the prescaled execution time is calculated as follows —

$$T_p = \left(\frac{1000ns}{16}\right) \times (C_p + 1) = 62.5ns \times (C_p + 1)$$

where C_p is the value of prescale field from the *previous* instruction. Prescaling in this way provides flexibility to efficiently cope with quick tasks like switching gates on the chip, as well as relatively slow events, like an analog-to-digital conversion.

The control field describes how the data field should be interpreted during execution. The lower twenty bits of each instruction are sent to the slave board every cycle. Bits D17 and D16 specify which of the three buffers "**`RampBuf`**", **`WrtBuf`**" or "**`ETknBuf`**" are the destination of the data field (lower sixteen bits). However, if all three bits, D17, D16 and D15, are asserted data is not latched into any buffer on the slave board. Rather, the data field is used for internal control on the master board.

The slave board sends twenty bits of data to the master board during every instruction.

Data	Microcode Mnemonic	Function
D0	LUC_WAIT_SOS	Wait SOS
D1	LUC_WAIT_EOS	Wait EOS
D2	LUC_ENABLE_CNT	ScanCnt enable
D3	LUC_WRITE_CDATA	Assert CDMem WE
D4	LUC_END_UCODE	Assert End_of_uCode

- LUC_WAIT_SOS
  When D0 is asserted in microcode execution, the program counter (`uCCnt` in Figure B-1.) stops counting until SOS signal is detected. In operation, the acquisition phase is synchronized with the stripe scan using this control.
- LUC_WAIT_EOS
  If D1 is asserted in execution. `uCCnt` is stopped until EOS is detected. By this function, readout phase waits for the stripe sweep completion.
- LUC_ENABLE_CNT
  D2 enables `ScanCnt` to count up for scan-time measurement. The measurement result will be stored in `CellD` and read by the host afterward for scan-time correction of range measurement.
- LUC_WRITE_CDATA
  If D3 is asserted in execution, write-enable pulse to the memory (`CellD`) is generated. Thus range data sent from the slave board is stored in the memory.
- LUC_END_UCODE
  If D4 is asserted in execution, microcode execution will be stopped at the next step. The host can know the state of the engine by reading `EouC` bit in `StatBuf`.
- LUC_SCNT_OE
  If D5 is asserted in execution, `ScanCnt` output is enabled at that step.

Table B-1. Control word mnemonic.

The bits D19 and D18 select output from one of three sources “**`TADCBuf`**”, “**`IADCBuf`**” or “**MiscInBuf**”.

The function of data field changes as defined by the control field, such as a digital time-stamp ramp signal fed to a D/A converter, or an encoded row/column select signal, or a trigger to the A/D converters. This field is also used to control execution flow in the master board. The special meanings assigned to each bit are listed in Table B-1.

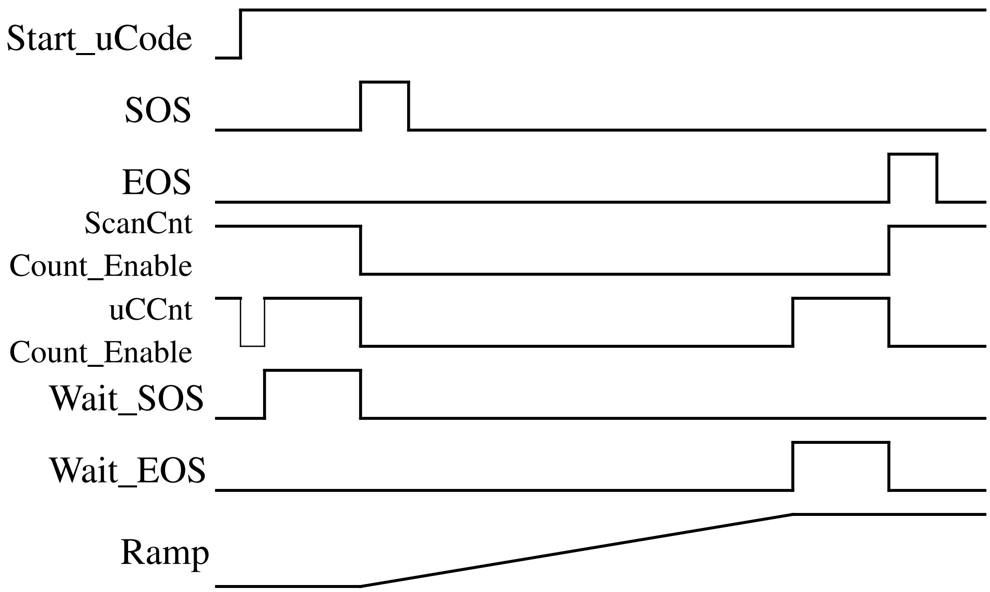


Figure B-4. Acquisition-phase timing diagram.

B.3. Control Algorithm

In operation, the range-sensor chip cycles between the acquisition and readout phases (Section 3.2). During the acquisition phase, range is recorded and held as a time-stamp voltage on a capacitor in each sensing cell. After the stripe has swept the scene, recorded data is read out immediately in the readout phase that follows.

B.3.1. Acquisition Phase

The host triggers the microcoded engine by asserting the "Start_uCode" bit in **`CmdBuf`** and the acquisition phase begins. The timing diagram is drawn in Figure B-4. is executed, which pauses the system until the SOS signal is detected. When SOS arrives, the microcode program counter and the scan-time counter (**`ScanCnt`**) are started. (The scan-time counter measures the duration between SOS and EOS signal.) Each microcode instruction now updates the time-stamp output at regular intervals. When the time-stamp ramp has completed, LUC_WAIT_EOS is executed. The program counter is paused and waits for the stripe to complete the scan. The program counter starts counting when EOS is detected and **`ScanCnt`** stops counting. If EOS is seen before the time-stamp ramp finishes, the microprogram continues without pausing (**`ScanCnt`** is stopped when EOS has arrived).

B.3.2. Readout Phase

The readout phase immediately follows the acquisition phase. As described in Section 2.2, sensing cells are addressed for readout using row and column select lines, specified by seven bits each in the data field that reads them. Circuitry on the slave board (`CSELPAL` and `RSELPAL`) interprets each 7-bits value into one of thirty-two row or column enable

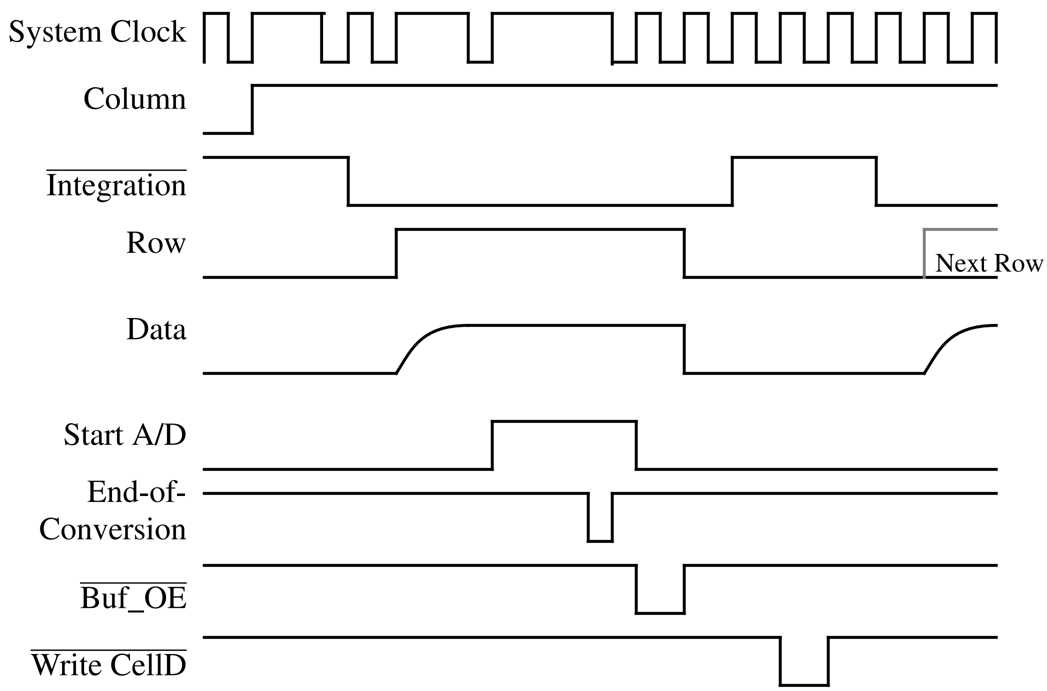


Figure B-5. Readout-phase timing diagram.

lines. In operation (Figure B-5), the column to be read is first enabled to precharge that column's readout path. Then cells in the column are read in tern by sequentially enabling each row. After all cells in that column have been read, the next column is enabled. In this manner, the entire 32×32 range image is read from the chip.

For each cell, an integration time is needed before to charge the capacitor in the output integration circuit. This integration time is adjusted by using the prescaling function. After enough integration time, the A/D converters are triggered by the next microcode instruction. The conversion time is also adjusted by prescaling the system clock. The end-of-conversion signal from A/D enables the "`TADCBuf`" and "`IADCBuf`" buffers to latch the result. After conversion, the data is transferred to the master board with a microinstruction. Three stages of latches are involved in the readout operation (i.e. — the data write command "LUC_WRITE_CDATA" is executed three instructions after the corresponding readout operation).

Appendix C. Pseudo-Inverse Least-Squares Fitting

To fit an $n \times 1$ observation matrix Y to a linear model of p parameters, the prediction is written as —

$$Y = XB + E$$

where X is an $n \times p$ variable matrix, B is $p \times 1$ parameter matrix whose values are to be approximated, and E represents the difference between the actual value and the prediction[7]. To minimize the prediction error, defined to be the sum of the elements of E squared $(E^T E)$, differentiate it with respect to B and set to 0. First, $E^T E$ is rewritten as follows —

$$\begin{aligned} E^T E &= (Y - XB)^T (Y - XB) \\ &= Y^T Y - B^T X^T Y - B^T X^T Y + B^T X^T XB \\ &= Y^T Y - 2B^T X^T Y + B^T X^T XB \end{aligned}$$

By differentiating it and setting the derivative to 0 —

$$0 = X^T XB - X^T Y$$

Thus following solution is obtained —

$$B = (X^T X)^{-1} X^T Y = X_P Y$$

where X_P is called the pseudo-inverse of X.