Chapter 4

Spacetime analysis: implementation and results

The previous chapter provides a theoretical foundation for accurate optical triangulation through spacetime analysis. In this chapter, we demonstrate that the method can be adapted to work with an existing commercial laser triangulation scanner (Section 4.1). We develop an efficient algorithm for performing the spacetime analysis and show how increased resolution can be obtained by interpolating the imaged light reflections (Section 4.2). Finally, we show the results of applying the spacetime analysis to some real objects (Section 4.3).

4.1 Hardware

We have implemented the spacetime analysis presented in the previous chapter using a commercial laser triangulation scanner and a real-time digital video recorder. The optical triangulation system we use is a Cyberware MS 3030 platform scanner (shown in Figure 1.5). This scanner collects range data by casting a laser stripe onto the object and by observing reflections with a CCD camera positioned at an angle of 30° with respect to the plane of the laser. The platform can either translate or rotate an object through the field of view of the triangulation optics. Figure 1.2b illustrates the principle of operation. The laser width (the distance between the beam center and the e^{-2} point) varies from 0.8 mm to 1.0 mm over the field of view which is approximately 30 cm in depth and 30 cm in height. Each CCD pixel

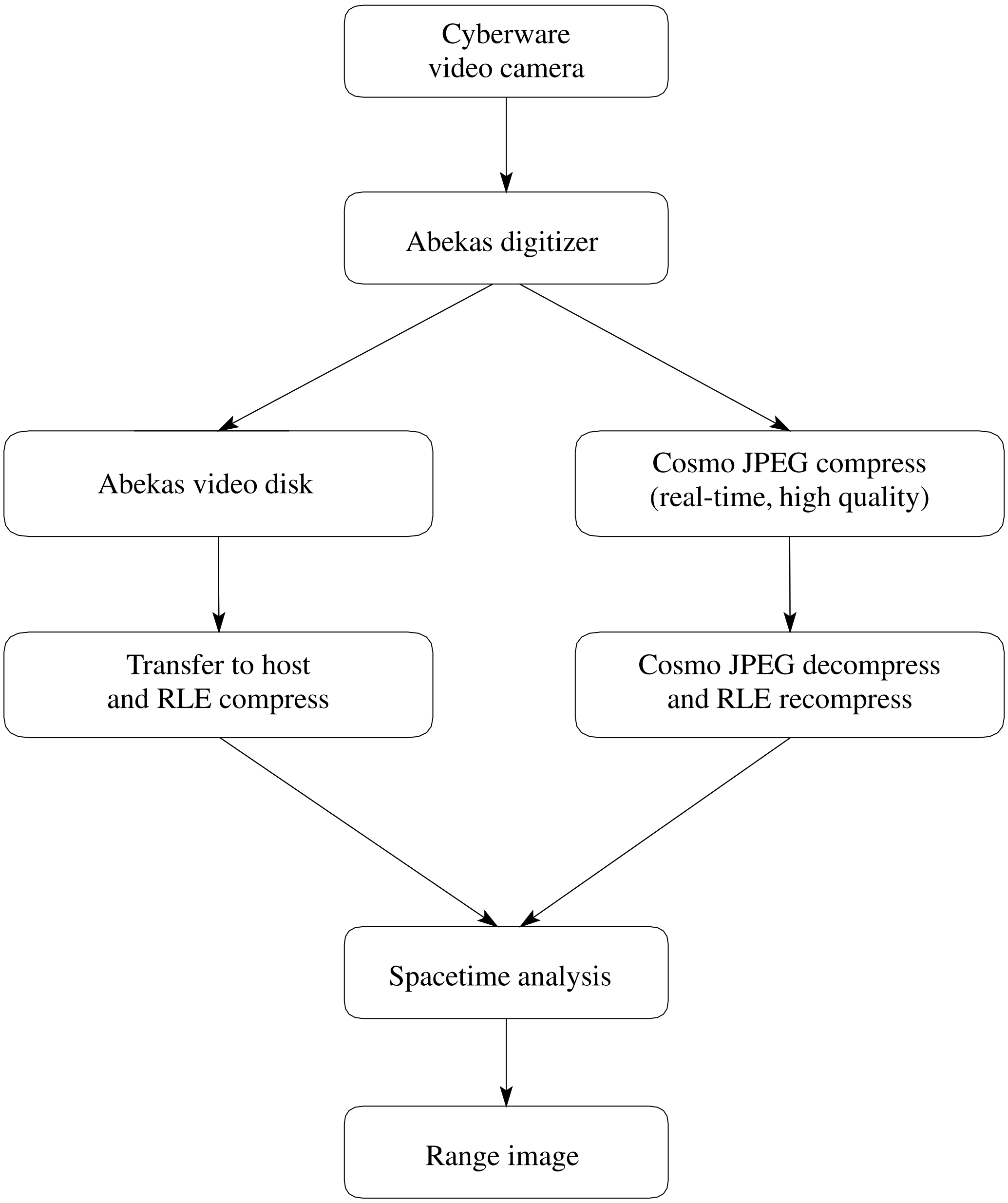


Figure 4.1: From range video to range image. To perform the spacetime analysis, we digitize the Cyberware video with an Abekas digitizer. We can then either store the digitized video to disk and copy it to the host computer, or perform a real-time JPEG compression and fast decompression through an SGI Cosmo board. The JPEG compression approach has proven to be more time efficient, but the results in this chapter were originally obtained with the video disk option. In either case, the frames are run-length encoded and then converted to a range image through the spacetime analysis.

images a portion of the laser plane about 0.5 mm square. Although the Cyberware scanner performs a form of peak detection in real time, we require the actual video frames of the camera for our analysis. Figure 4.1 illustrates the possible routes for processing the range video. We can digitize these frames with an Abekas A20 video digitizer, which can acquire 486 by 720 pixel frames at 30 Hz. The captured frames have approximately the same resolution as the Cyberware CCD camera, though the scanlines are first converted to analog video before being resampled and digitized. The digitized frames can then be stored in real time with an Abekas A60 digital video disk. Alternatively, we can run the digitized signal through a commercially available JPEG compression board. Because each frame is an image of the narrow laser sheet intersecting the surface, it is mostly blank. As a result, we can JPEG compress the frame with a high quality factor, but still obtain high compression. The JPEG compression efficiently encodes the blocks of empty space without sacrificing the quality of the imaged light reflections. The key advantage of this approach is that the range video compresses well enough to be stored on the host computer's main memory, as opposed to a video disk which limits the speed of recovery of the recorded frames. The results in this chapter were obtained with the Abekas digital video disk, but the results described in Chapter 8 are based on range data acquired with the help of JPEG compression.

4.2 The spacetime algorithm

Using the principles of the previous chapter, we can devise a procedure for extracting range data from spacetime images:

1. Perform the range scan and capture the spacetime images.

2. Rotate the spacetime images by θ.

3. Find the statistics of the Gaussians in the rotated coordinates.

4. Rotate the means back to the original coordinates.

Figure 4.2 illustrates this new method graphically. In order to implement step 1 of this algorithm, we require a sequence of CCD images. Most commercial optical triangulation

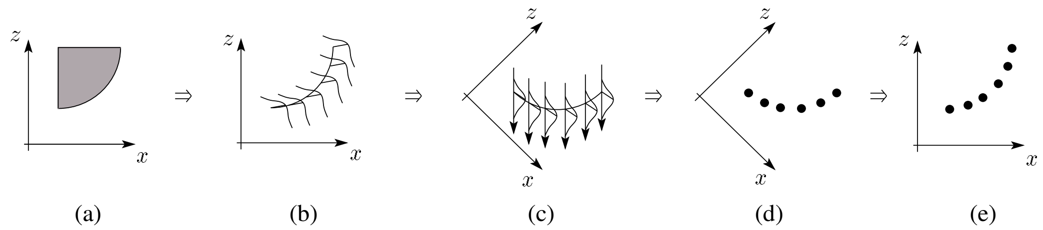


Figure 4.2: Method for optical triangulation range imaging with spacetime analysis. (a) The original geometry. (b) The resulting spacetime image when scanning with the illuminant in the $+z$ direction. (c) Rotation of the spacetime image followed by evaluation of Gaussian statistics. (d) The means of the Gaussians. (e) Rotation back to world coordinates.

systems discard each CCD image after using it (e.g. to compute a stripe of the range map). As described in the previous section, we have assembled the necessary hardware to record the CCD frames. In the previous chapter, we discussed a one dimensional sensor scenario and indicated that perspective imaging could be treated as locally orthographic. For a two dimensional sensor, we can imagine the horizontal scanlines as separate one dimensional sensors with varying vertical (y) offsets. Each scanline generates a spacetime image, and by stacking the spacetime images one atop another, we define a spacetime *volume*. Complications arise when we have perspective projection in the vertical direction, as the image of a point sweeping through the Gaussian illuminant can cross scanlines (i.e., a point is not constrained to lie in a single spacetime image plane). We have not yet implemented the general spacetime ranging method that accounts for this effect, so we restrict most of our analysis to the spacetime volume near the vertical centerline of the CCD where range points will not cross scanlines while passing through the illuminant (i.e., we can visualize the spacetime volume as a stack of spacetime images). Nevertheless, toward the end of this chapter, we demonstrate some results for larger objects that show improvement even though they are not restricted to the center of the field of view in the vertical direction. This improvement is probably due to the fact that even at the extreme positions of the field of view a point does not traverse more than half a scanline while passing through the illuminant.

In step 2, we rotate the spacetime images so that Gaussians are vertically aligned. In a practical system with different sampling rates in x and z, the correct rotation angle can be computed as:

$$\tan\theta_{ST} = \frac{\tau_z}{\tau_x}\tan\theta \tag{4.1}$$

where θ_{ST} is the new rotation angle, τ_x and τ_z are the sample spacings in x and z respectively, and θ is the triangulation angle. In order to determine the rotation angle, θ_{ST}, for a given scanning rate and region of the field of view of our Cyberware scanner, we first determined the local triangulation angle and the sample spacings in depth, z, and lateral position, x. Equation 4.1 then yields the desired angle. When performing the rotation, we avoid aliasing artifacts which would lead to range errors by employing high-quality filters. For this purpose, we use a bicubic interpolant.

After performing the rotation of each spacetime image, we compute the statistics of the Gaussians along each rotated spacetime image raster[1]. Our method of choice for computing these statistics is a least squares fit of a parabola to the log of the data; i.e, given a Gaussian of the form:

$$G(x) = a\exp\left[-\frac{2(x-b)^2}{w^2}\right]$$

where a is the amplitude and b is the desired center, we can take the log of the Gaussian:

$$\log[G(x)] = \log[a] - \frac{2(x-b)^2}{w^2} = \left(-\frac{2}{w^2}\right)x^2 + \left(\frac{4b}{w^2}\right)x + \left(\log[a] - \frac{2b^2}{w^2}\right)$$

and fit a parabola to the result. The linear coefficient contains the center, b.

We have also experimented with fitting the data directly to Gaussians using the Levenberg-Marquardt non-linear least squares algorithm [Press et al. 1986], but the results have been substantially the same as the log-parabola fits.

The Gaussian statistics consist of a mean, which corresponds to a range point, as well as a width and a peak amplitude, both of which indicate the reliability of the data. Widths that are far from the expected width and peak amplitudes near the noise floor of the sensor imply unreliable data which may be down-weighted or discarded during later processing (e.g.,

[1] Actually, we do not expect to find perfect Gaussians on the rotated image rasters. The filtering and sampling processes described in Section 3.5 alter the ideal spacetime image. Nevertheless, fitting Gaussians is a reasonable approximation that has yielded excellent results in practice

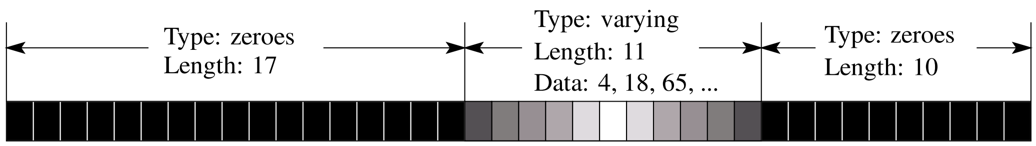


Figure 4.3: Run-length encoded spacetime scanline. Each scanline is composed of runs of zeroes and runs of varying intensity data.

when combining multiple range meshes). For the purposes of this thesis, we discard unreliable data and rely on future scans to fill in reliable data used in the surface reconstruction process.

Finally, in step 4, we rotate the range points back into the global coordinate system.

4.2.1 Fast rotation of the spacetime image

In order to process the spacetime images, we first compress the frames into a run length encoded (RLE) representation as shown in Figure 4.3. The time evolution of a single scanline is thus a run length encoded spacetime image which we must rotate in order to extract range data. The naive approach to rotating each image would be to expand it into uncompressed form, and then visit the pixels of the target image while walking through the rotated source slice, i.e., while "gathering" data from the source slice. Alternatively, we can process the source image's runs, and only "send" data to the source image in the regions of runs with varying data [Lacroute 1995]. Figure 4.4 illustrates this principle.

Note that when performing the traditional gather approach, we step through the target image and resample the source image using a reconstruction filter. As a result, a neighborhood of source pixels will influence the value of each target pixel, as shown in Figure 4.5c. We can also look at this in another way: each source pixel will influence a neighborhood of target pixels. Thus, when performing the send operation, each pixel adds weighted copies of itself to a neighborhood of pixels in the target image. This send operation is illustrated in Figure 4.5d.

The RLE send method we have described so far will significantly reduce the computational expense, because it visits only the interesting pixels in the source image. Nevertheless, it still creates an uncompressed target image whose pixels must all be visited in the

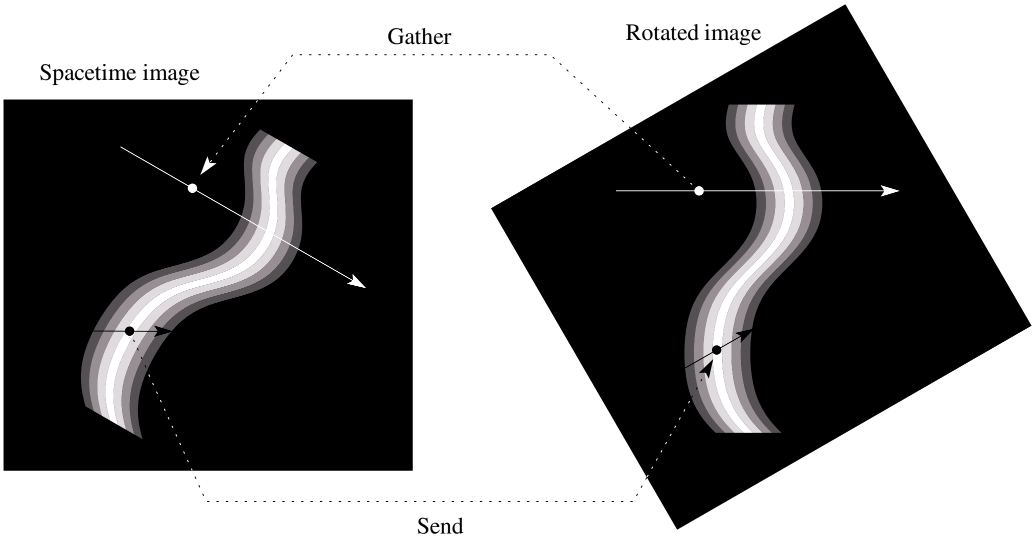


Figure 4.4: Fast RLE rotation. The naive approach to rotating an image would be to expand the RLE source image, and then traverse all of the pixels of the target image while "gathering" data from (resampling) the source image. A much more efficient approach is to traverse the RLE structure in the source image and "send" only non-zero data to the target image.

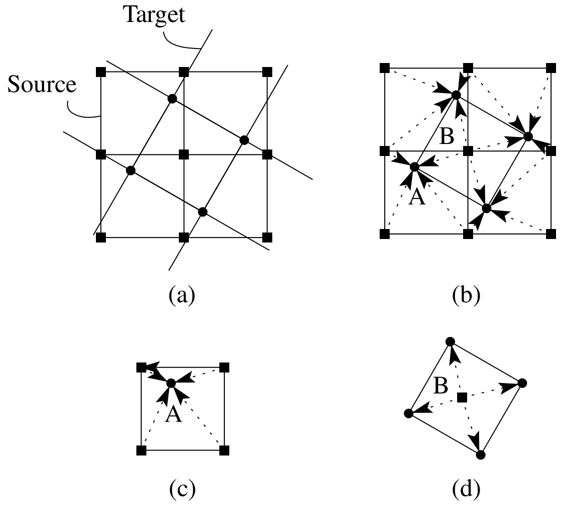


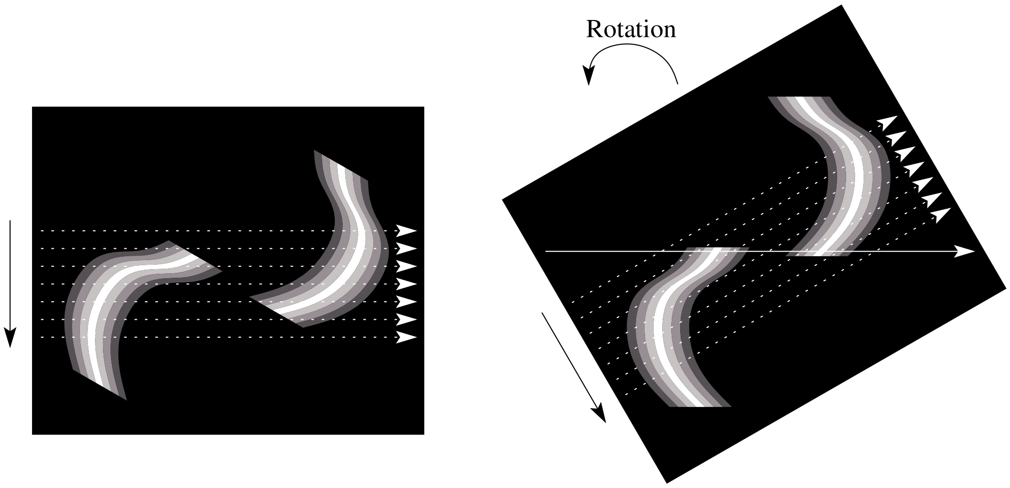
Figure 4.5: Reconstruction: send vs. gather. (a) The target image is rotated with respect to the source image. (b) We can find the values at the target pixels by interpolating among the source pixels. (c) "Gather" interpolation for a single target image pixel, A. (d) "Send" interpolation for a single source pixel. The example in this figure illustrates the case for a bilinear filter with a 2x2 support; extension to filters with larger supports is straightforward.

next step of the spacetime algorithm. Ideally, we would like to create an RLE target image directly. We can achieve this goal with the following procedure:

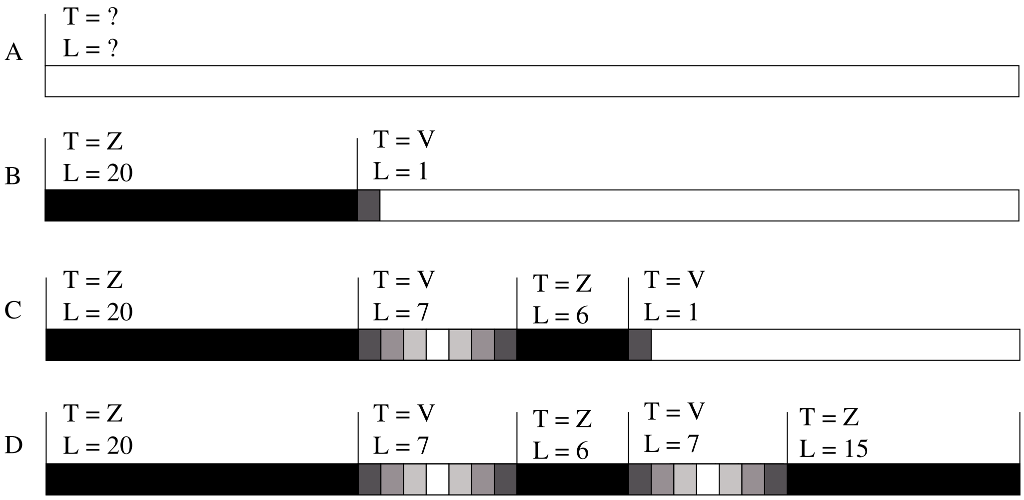
1. Allocate space for each scanline in the target image.
2. Stream through each source scanline, ignoring runs of zeroes. When sending varying data:
   - If the data being sent is not within the current target run being constructed:
     - Mark the current target run as finished.
     - Declare a run of zeroes between the finished run and the new target position.
     - Start a new run of varying pixels.
   - Else,
     - Deposit the pixel value into the existing run, increasing its length as needed.
3. Mark the last run on each scanline as being a run of zeroes.

Figure 4.6 shows a scanline as it is being constructed.

This RLE rotation algorithm delivers a dramatic performance improvement. If we consider a spacetime volume of n voxels on a side, then the brute force rotation would execute in time proportional to the size of the volume, i.e., $O(n^3)$. On the other hand, the fast algorithm described here requires time proportional to the part of the volume with non-zero data. The sensor samples the surface onto a grid of size roughly n x n, which is then "blurred" into the spacetime volume by the spacetime impulse response (Equation 3.11). The width of this impulse response function is constant, so the overall size of the spacetime surface embedded in the volume is $O(n^2)$. Thus, the fast rotation algorithm runs in $O(n^2)$ time, substantially faster then the brute force method. In practice, we observe a typical speed-up of 50:1 over the brute force approach.



(a)



(b)

Figure 4.6: Building a rotated RLE scanline. (a) The source image is traversed in scanline order (dotted white lines) from top to bottom (solid black arrow to the left), and the non-zero values are deposited in the target image. (b) The scanline indicated by the solid white arrow in the target image is constructed in steps. "T" is the data type, and "L" is the run length. In A, The scanline is initialized to have no data or length. In B, when the first source scanline intersects the target scanline, the first run is declared to be zeroes (T = Zero or "Z") and its length is computed. In addition, a varying run (T = Varying or "V") is begun. As successive source scanlines intersect the target scanline, more varying data is added to the current run, which grows as needed. At C, more varying data is laid down, but it is offset from the first varying run. Thus, a new run of zeroes is declared with its length, and a new varying run is begun. Eventually, the source scanline sweep finishes, and the remainder of the scanline is declared to be zeroes, as shown in D.

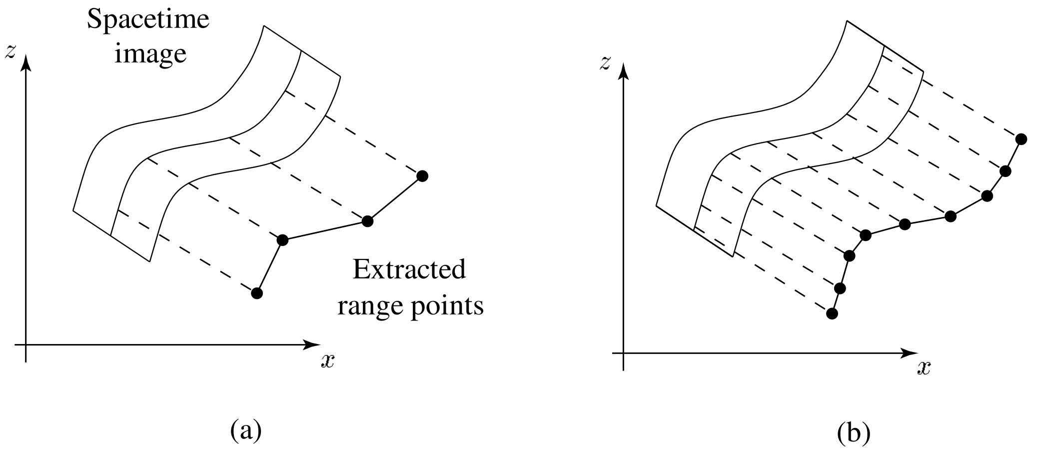
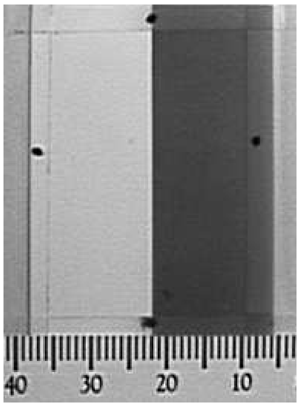


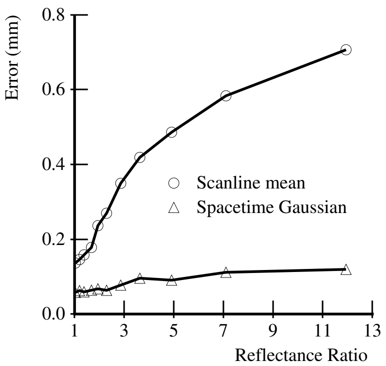
Figure 4.7: Interpolation of spacetime image data. (a) Extraction of range data from a spacetime image followed by linear interpolation of the range points. (b) Same as (a) but more interpolation in the spacetime image rather than the among the range points.

4.2.2 Interpolating the spacetime volume

Traditionally, researchers have extracted range data at sampling rates corresponding to one range point per sensor scanline per unit time. Interpolation of shape between range points has consisted of fitting primitives (e.g., linear interpolants like triangles) to the range points. Instead, we can regard the spacetime volume as the primary source of information we have about an object. After performing a scan, we have a sampled representation of the spacetime volume, which we can then reconstruct to generate a continuous function. In the previous chapter, we described this reconstruction for a spacetime image; for the spacetime volume, we have added a vertical dimension, but the principle remains the same. This reconstructed function then acts as our range "oracle," which we can query for range data at a resampling rate of our choosing. Figure 4.7 illustrates this idea for a single spacetime image. In practice, we can magnify the sampled spacetime volume prior to applying the range imaging steps described above. The result is a range grid with a higher sampling density directly derived from the imaged light reflections.



(a)



(b)

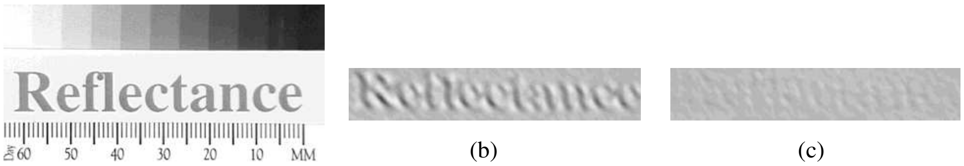
Figure 4.8: Measured error due to reflectance step. (a) Printed cards with reflectance discontinuities were carefully taped to a machined planar surface. The black dots were used to distinguish the taped boundary from the target surface. (b) A plot of range errors shows that the spacetime method yields a 60-80% reduction in range error for steps varying from 1:1 to 12:1.

4.3 Results

We have performed a variety of tests to evaluate the performance of traditional triangulation and spacetime analysis. These tests include experiments with reflectance variations, simple shape variations, errors due to speckle, and scanning complex objects.

4.3.1 Reflectance correction

To evaluate the tolerance of the spacetime method to changes in reflectance, we performed two experiments, one quantitative and the other qualitative. For the first experiment, we produced reflectance steps by printing alternating white and gray stripes with a color printer. By varying the reflectances of the solid gray stripes, we obtained reflectance ratios (white:gray) of about 1:1 to 10:1. We scanned them at an angle of 30^o (roughly facing the sensor) and extracted range data using both the traditional method and the spacetime method. To measure deviations from planarity, we first computed the parameters of the plane by deleting those range points lying in the neighborhood of the reflectance discontinuity. This step avoids permitting erroneous data to influence the estimated orientation of the plane. When computing range errors, we used all range data, including points near the reflectance discontinuity.



(a) (b) (c)

Figure 4.9: Reflectance card. (a) Photograph of a planar card with the word "Reflectance" printed on it. (b) and (c) Shaded renderings of the range data generated by traditional mean analysis and spacetime analysis, respectively. With traditional mean analysis, the planar card appears embossed with the lettering, indicating a confusion between reflectance and shape. The spacetime analysis yields a nearly planar surface, showing a disambiguation of reflectance and shape.

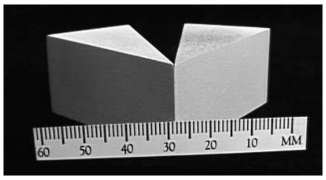
Figure 4.8 shows a plot of maximum deviations from planarity. The spacetime method has clearly improved over the old method. The reduction in range errors varies from 65% for a reflectance change of 2:1 up to 85% for a reflectance change of 12:1.

For qualitative comparison, we produced a planar sheet with the word "Reflectance" printed on it. Figure 4.9 shows the results. The old method yields a surface with the characters well-embossed into the geometry, whereas the spacetime method yields a much more planar surface indicating successful decoupling of geometry and reflectance.

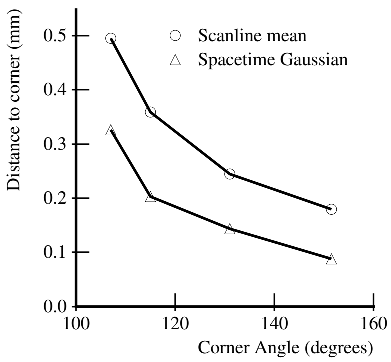
4.3.2 Shape correction

We conducted several experiments to evaluate the effects of shape variation on range acquisition. In the first experiment, we generated corners of varying angles by abutting sharp edges of machined aluminum wedges which were painted white. Figure 4.10 shows the range errors that result for traditional and spacetime methods. Again, we see an increase in accuracy, though not as great as in the reflectance case.

We also scanned two 4 mm strips of paper at an angle of 30^o (roughly facing the sensor) to examine the effects of depth discontinuities. Figure 4.11b shows the "edge curl" observed with the old method, while the spacetime method in Figure 4.11c shows a significant reduction of this artifact under spacetime analysis. We have found that the spacetime method reduces the length of the edge curl from an average of 1.1 mm to an average of approximately 0.35 mm.

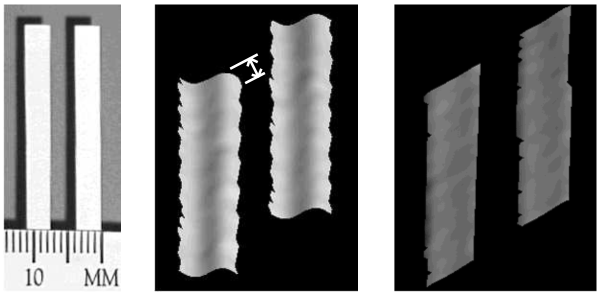


(a)



(b)

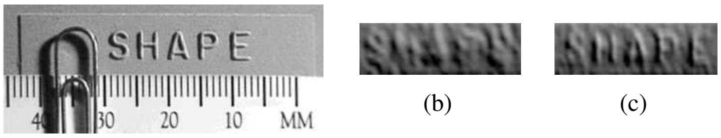
Figure 4.10: Measured error due to corner. (a) Two machined wedges with razor sharp corners were placed at varying angles to each other to study the effect of corner angle on range accuracy. (b) A plot of range errors shows that the spacetime method yields a 35-50% reduction in range error for corners of angles varying from 110^o to 150^o.



(a) (b) (c)

Figure 4.11: Depth discontinuities and edge curl. (a) Photograph of two strips of paper. (b) and (c) Shaded renderings of the range data generated by traditional mean analysis and spacetime analysis, respectively. With traditional mean analysis, the edges of the strips exhibit large edge curl artifacts (1.1 mm between the hash marks in (b)). The curl is nearly eliminated when using the spacetime analysis.

Finally, we impressed the word "SHAPE" onto a plastic ribbon using a commonly available label maker. In Figure 4.9, we wanted the word "Reflectance" to disappear because it represented changes in reflectance rather than in geometry. In Figure 4.12, we want the word "SHAPE" to stay because it represents real geometry. Furthermore, we wish to resolve it as highly as possible. Figure 4.12 shows the result. Using traditional mean analysis, the word is barely visible. Using spacetime analysis, the word becomes legible.



(a) (b) (c)

Figure 4.12: Shape ribbon. (a) Photograph of a surface with raised lettering (letters are approx. 0.3 mm high). (b) and (c) Shaded renderings of the range data generated by traditional mean analysis and spacetime analysis, respectively. Using mean pulse analysis, the lettering is hardly discernible, while the spacetime analysis yields a legible copy of the original shape.

4.3.3 Speckle

To evaluate the influence of laser speckle, we scanned planar surfaces of varying roughness under coherent and incoherent light. First, we verified the existence of the laser speckle. Recording the light reflections from the surfaces under incoherent (single filament, non-diffuse) illumination showed negligible variation ($< 5\%$) in image intensity, while the images of laser reflections showed significant variations ($> 20\%$) in peak intensity. These facts, coupled with the observation that surfaces with roughness finer than the resolution of the pixels yielded variations at a scale larger than a pixel, lead us to believe that we are observing laser speckle.

Next, we performed range scans on the planar surfaces and generated range points using the traditional and spacetime methods. After fitting planes to range points, we found that both methods resulted in range errors with a standard deviation of about 0.06 mm and a distribution that is very nearly Gaussian (see Figure 4.13). As expected, both traditional and spacetime triangulation are susceptible to errors due to laser speckle. In Section 2.2.4, we computed the range error for a perfectly rough surface to be 0.25 mm. Since the surface being measured is unlikely to be perfectly rough, it is not surprising that the measured error is less.

4.3.4 Complex objects

To show how the spacetime analysis performs with more complex objects occupying a larger portion of the field of view, we scanned two models. Figure 4.14 shows the results for a model tractor. Figure 4.14b is a rendering of the data generated by the Cyberware scanner

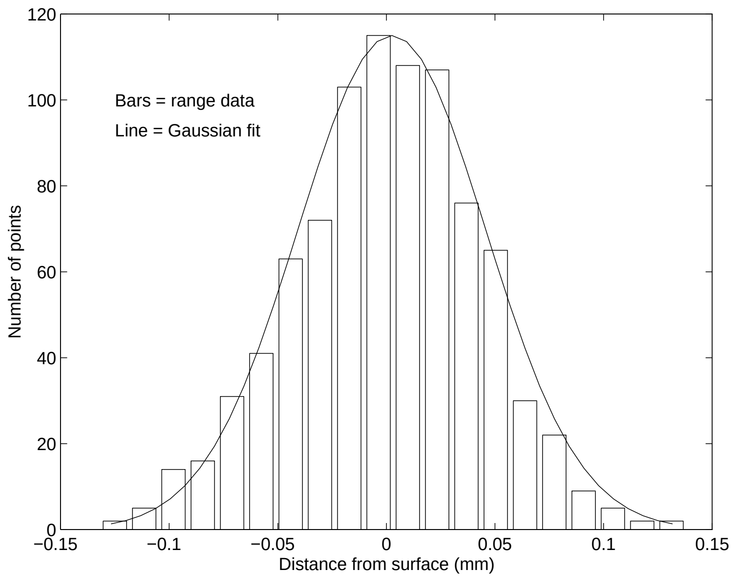


Figure 4.13: Distribution of range errors over a planar target. We extracted the range data using spacetime analysis and fit a plane through the resulting data. The bar graph represents a histogram of distances from the planar fit, while the curve corresponds to a Gaussian approximation to the histogram.

hardware and is particularly noisy. This added noisiness results from the method of pulse analysis performed by the hardware, a method similar to peak detection. Peak detection is especially susceptible to speckle noise, because it extracts a range point based on a single value or small neighborhood of values on a noisy curve. Mean analysis tends to average out the speckle noise, resulting in smoother range data as shown in Figure 4.14c. Figure 4.14d shows our spacetime results and Figure 4.14e shows the spacetime results with 3X interpolation and resampling of the spacetime volume as described in Section 4.2. Note the sharper definition of features on the body of the tractor.

Figure 4.15 shows the result for a model of an alien creature. As with the tractor image, the Cyberware output is noisier than the traditional mean analysis results. Note, however, that the traditional mean analysis appears to smooth the geometric detail more than the Cyberware method. Both the Cyberware and the mean analysis suffer from edge curl. By contrast, the spacetime method reduces edge curl, exhibits less noise, and attains comparable or greater detail than either of the other methods.

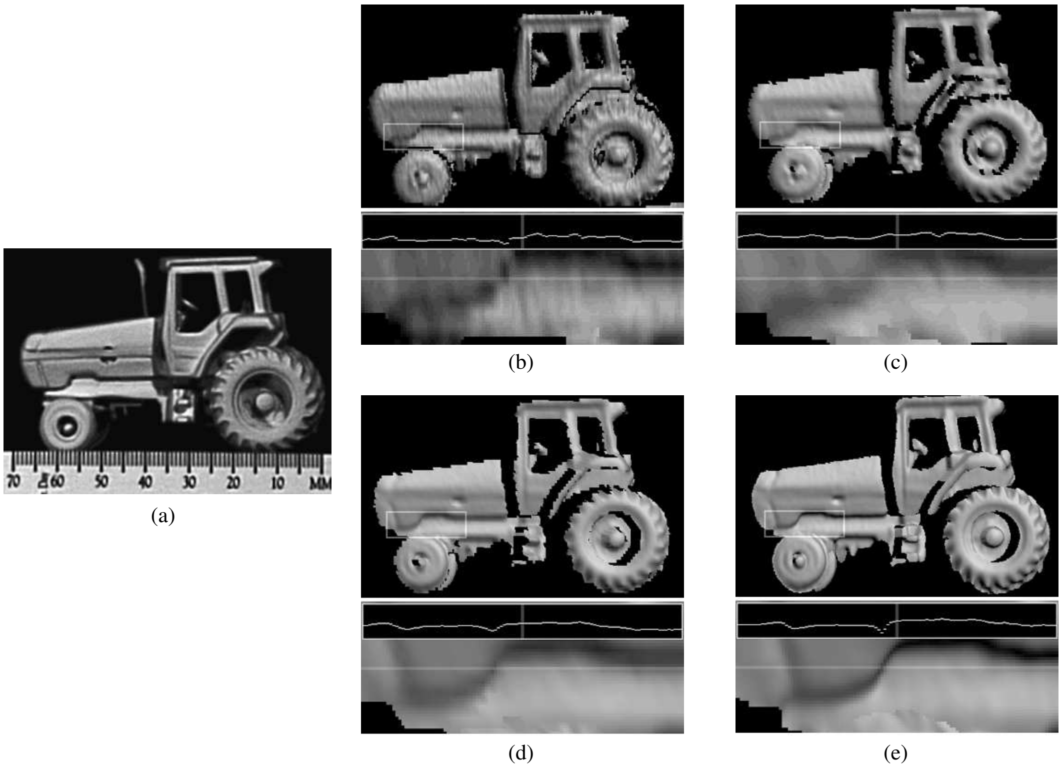


Figure 4.14: Model tractor. (a) Photograph of original model and shaded renderings of range data generated by (b) the Cyberware scanner hardware, (c) mean pulse analysis, (d) our spacetime analysis, and (e) the spacetime analysis with 3X interpolation of the spacetime volume before fitting the Gaussians. Below each of the renderings is a blow-up of one section of the tractor body (indicated by rectangle on rendering) with a plot of one row of pixel intensities. Because of the scanner hardware's peak detection method, it is the most susceptible to speckle noise as is evident in the rendering and in the "bumpy" intensity plot. Mean analysis yields a decrease in noise and spacetime analysis yields even less noise and more detail. The 3X interpolated spacetime analysis reveals the most detail of all.

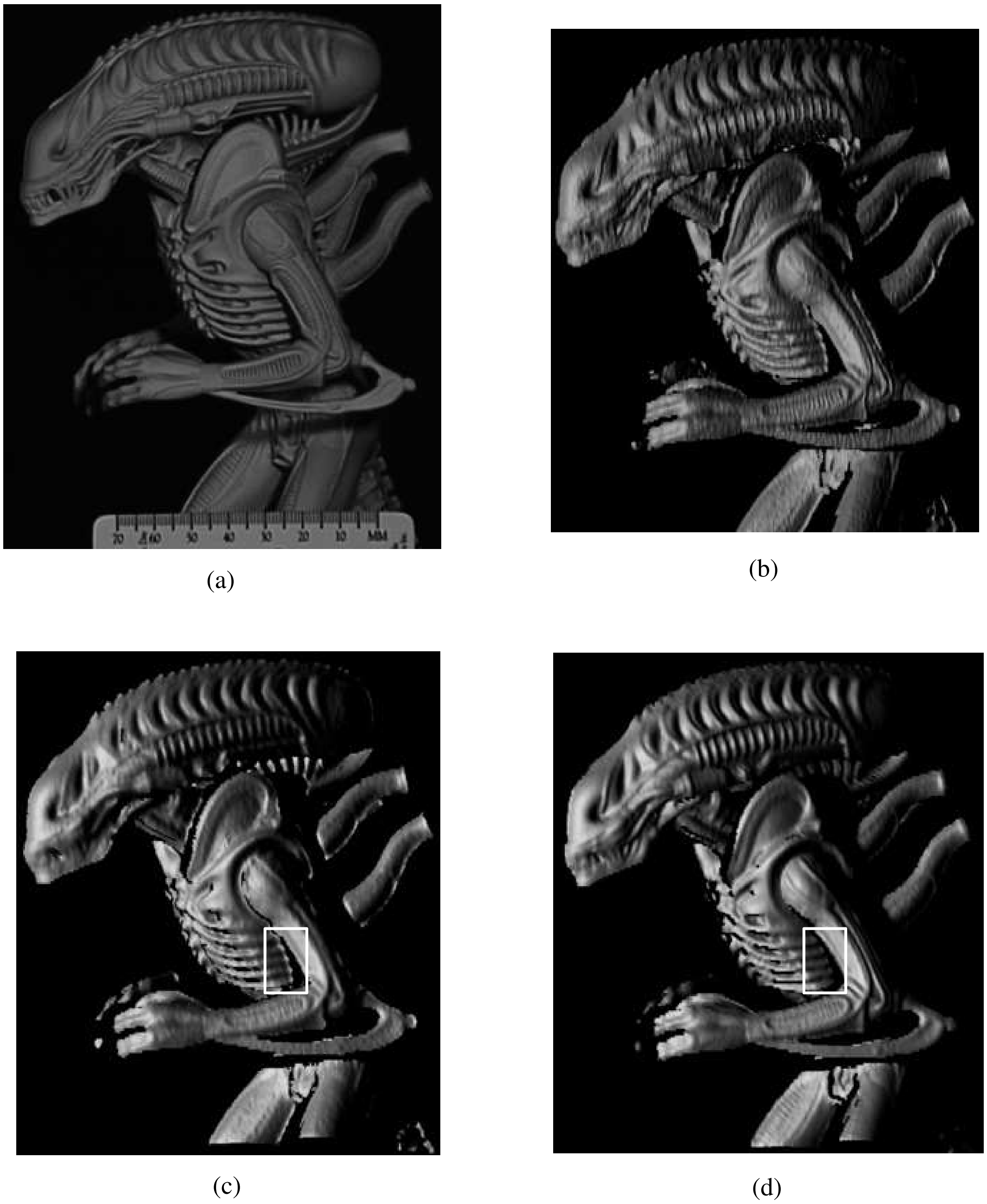


Figure 4.15: Model alien. (a) Photograph of original model and shaded renderings of range data generated by (b) the Cyberware scanner hardware, (c) mean pulse analysis, and (d) our spacetime analysis. The highlighted box in (c) and (d) shows the reduction of edge curl along the depth discontinuity at the alien's torso where the CCD line of sight was occluded. Note that only the spacetime method resolves the ridges in the right leg.

4.3.5 Remaining sources of error

The results we presented in this section clearly show that the spacetime analysis yields more accurate range data, but the results are imperfect due to system limitations. These limitations include:

- CCD noise
- Finite sensor resolution
- Optical blurring and electronic filtering
- Quantization errors
- Calibration errors
- Surface-surface inter-reflections

In addition, we observed some electronic artifacts in our Cyberware scanner that influenced our results. We expect, however, that any measures taken to reduce the effects of the limiting factors described above will lead to higher accuracy. By contrast, if one uses traditional methods of range extraction, then increasing sensor resolution and reducing the effects of filtering alone will *not* significantly increase tolerance to reflectance and shape changes when applying the traditional methods of range extraction.