

# Recovery of Dynamic Scene Structure from Multiple Image Sequences

Peter W. Rander, P.J. Narayanan, and Takeo Kanade

Robotics Institute  
Carnegie Mellon University  
Pittsburgh, PA 15221, USA

## Abstract

*Despite significant progress in automatic recovery of static scene structure from range images, little effort has been made toward extending these approaches to dynamic scenes. This disparity is in large part due to the lack of range sensors with the high sampling rates needed to accurately capture dynamic scenes. We have developed a system that overcomes this problem by exploiting video cameras, which easily capture images of dynamic scenes, and image-based stereo, which estimates scene structure based on correspondences among the images from different cameras. Our system uses a synchronized multi-camera recording system to capture live video of the scene and a software implementation of image-based stereo to compute range images off-line. By combining this system with multi-image fusion, we created a novel system for dynamic structure recovery, with many applications including telepresence, training, and entertainment. Development of this system has also revealed the potential use of fusion as both a multi-view and multi-resolution integration process for stereo.*

## 1.0 Introduction

Recovering 3D structure of static scenes has been the focus of much research [2][4][6][7][17][19][21]. Good solutions to this problem have applications in a number of domains, especially reverse engineering for virtual reality and manufacturing. Dynamic scene structure recovery, on the other hand, has received little attention from the research community, despite broad usefulness in traditional telepresence and virtual reality, as well as in such new areas as Virtualized Reality [10][11] and Multiple-Perspective Interactive Video [8]. This lack of work on the dynamic problem is due in large part to the lack of range sensors capable of meeting the high demands of the dynamic scenes.

In static structure recovery, multiple range images are fused into a single 3D model. Usually the range images are collected with a single range sensor, either by moving the sensor around the scene or by moving the scene around the sensor. This approach eliminates any interaction between sensors at different viewpoints, allowing the use of active

sensors such as laser range finders. In addition, exposure time and sampling rate are unimportant because no motion occurs during sampling. In fact, many existing range sensors require several seconds to capture a single range image.

In dynamic structure recovery, multiple range image sequences are fused into a sequence of 3D models. Because the scene changes, multiple viewpoints require multiple sensors operating simultaneously. Using a collection of active sensors is not possible because the "illumination" (the active signal transmitted) from one sensor would interfere with that of another sensor. In addition, the exposure time must be short to reduce blur and the sampling rate must be high to avoid missing parts of the scene motion -- that is, to avoid temporal aliasing. Existing sensors are simply too slow to achieve these requirements for many interesting scenes.

Dynamic scene range imaging, then, requires sensors which have high sampling rates and which are capable of co-existing with other sensors. One way of satisfying these constraints is to apply image-based stereo to images acquired from simple video cameras. A video camera's sampling rate is high enough to capture significant motion (for NTSC video, 30Hz) and since video cameras are passive sensors, they easily co-exist with one another in a common environment. Recent efforts have, in fact, demonstrated the utility of this approach: Kanade et. al. [9] have developed a real-time stereo machine that provides depth images with 240x256 pixels at 30 Hz while searching up to 60 depth levels. This stereo machine performs both image capture and stereo computation in dedicated hardware at video rate. We apply the same concept in our system, but separate real-time image recording from off-line stereo computation. This approach allows us to use inexpensive recording equipment for image capture and general purpose computers for stereo processing.

Given sequences of range images from many perspectives, one then must fuse the images to form 3D models that accurately represent the 3D scene structure at each time instant. In general, this would require pose estimation for each sensor position (assuming fixed sensor location) as well as temporal synchronization of the sequences. We eliminate both problems, though, by exploiting the information required for stereo processing. Stereo requires camera pose

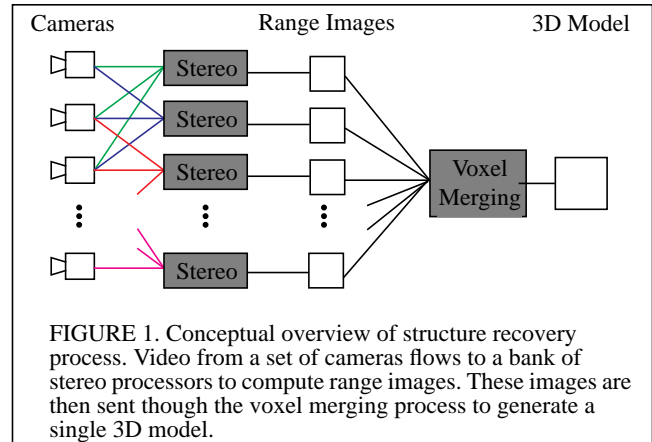
to estimate depth and requires synchronized intensity images so that the pixel disparities between images are due entirely to differences in camera pose.

With these added constraints, we approximate dynamic scenes with a sequence of static scenes, reducing the dynamic structure recovery problem to static structure recovery at each time sample. For this problem, we apply an algorithm very similar to Curless and Levoy [2], integrating multiple range images into a single 3D grid of voxels, then extracting a 3D triangle mesh surface description from the voxel space. If desired, the triangle mesh can be decimated to achieve significantly smaller meshes with little loss in structural accuracy. Also, by reprojecting the intensity images for that time instant onto the recovered model, it is possible to create a realistic textured model of the scene.

This approach can be considered a two-stage fusion process, since stereo itself is a low-level fusion process. In this context, fusion of multiple stereo views aims to increase precision and to reduce ambiguity -- the same reasons to use multi-camera stereo rather than two-camera (binocular) stereo [15]. In stereo, it is common to use multi-resolution processing to improve speed and reliability, but fusing the multi-resolution information in stereo has proven to be problematic. It is possible, though, to delay this multi-resolution fusion until the second-level fusion for multiple viewpoints, which may be desirable because the other viewpoints may help resolve ambiguity and localize more precisely. We discuss this possibility more completely in Section 5.0.

There are a few alternative approaches that attempt to bypass the range image construction phase, mapping more directly from intensity images to full 3D models. Khalili [12] proposes an intensity-based voxel stereo algorithm in which each voxel is projected into all images to acquire a set of intensity samples. Assuming the voxel is actually occupied, the variance of these samples should be low, since they are all images of the same point. By thresholding the variance, the occupancy of each voxel is determined. As Khalili admits, though, occlusion is expected to be a significant problem with this kind of approach, because occluded regions would unknowingly receive conflicting intensity samples from the occluding surface. In addition, this approach assumes that the intensity observed in the image is independent of the viewpoint. Physically, this would require uniformly illuminated, perfectly-lambertian scenes and identical cameras, neither of which is likely to exist in the real world. Transforming from intensity to edges, as Collins [1] does, may overcome some of these difficulties since edges are less dependent on intensity variations from viewpoint changes and imaging hardware inconsistencies. Since edges do not provide dense structure, though, significantly more effort must go into constructing the model from the edges, which itself is a non-trivial issue.

In summary, then, we propose an architecture as shown in Figure 1, with many cameras feeding images to a bank of



stereo processes which generate range images corresponding to each intensity image. These range images are then fed to a voxel merging process, which reconstructs the scene at each time instant. The following sections discuss each of the main steps in this system. First, we describe the acquisition of synchronized intensity image sequences and the stereo algorithm for computing depth. Next we describe the fusion of multiple range images from different viewpoints into one 3D model. We then present experiments using the system and evaluate the potential benefits of using multi-image fusion to address common breakdowns in stereo.

## 2.0 Intensity and Range Image Acquisition

Although it would be simplest to generate range images using hardware like that of the stereo machine [9], practical limitations (especially cost) prevent us from directly implementing this strategy. To work around these constraints, we developed a separate system that performs synchronized multi-camera video capture in real-time but computes range images off-line, in a software implementation of the stereo algorithm used on the stereo machine.

### 2.1 Synchronized Multi-Camera Video Capture

The most direct way to capture synchronized video from many cameras is to store digitized versions of synchronized video signals. Examination of this process reveals, however, the difficulty in actually implementing this approach. Consider that a single monochrome video camera generates about 10 MBytes of data each second. A 50-camera system would require 0.5 GBytes/sec bandwidth and more than 1 TeraByte of storage for an hour of recording time. Clearly, this approach is impractical using current technology.

The approach we take (discussed much more completely in [14]) is to split the image capture problem into two stages: synchronous real-time recording and off-line digitization. In the recording system, shown schematically in Figure 2, video cameras are externally synchronized. The video signals from

the cameras are time-stamped with the industry-standard Vertical Interval Time Code (VITC) [18] before being recorded by consumer-grade VCRs. The digitization system (Figure 3) contains a computer-controllable VCR, a digitizer, and a host computer. The computer instructs the VCR to play a videotape while the digitizer captures as many frames as it can. For most systems, the bandwidth from digitizer to disk will be too low to capture every frame of the video stream in one pass. To recover missed frames, the computer instructs the VCR to rewind and play again while the digitizer grabs as many new frames as possible. This process continues until all the frames in the video sequence have been captured. To identify each frame, the digitizer captures the VITC as scan lines in the image. The computer interprets the time code at video rate to determine whether or not the current frame has already been captured. In this way, each pass through the tape captures only new frames, and the rewind-play cycle needs to be repeated very few times (determined predominantly by the memory size and speed of the digitizer and host computer).

While limited to applications that allow off-line processing, this approach has three significant strengths. First, the system is trivially scalable in channels because of the simplicity of the interconnections. Second, the system has large storage at low cost since video is stored on standard video tape. Third, the system is inexpensive, both in up-front costs as well as in per-channel costs. The base cost covers a computer, a digitizer, and computer-controllable VCR, while each channel requires a camera, VITC inserter, and VCR -- all available commercially.

## 2.2 Image to Depth Conversion

To compute range images from the multi-camera images collected with our video capture system, we use the Multi-Baseline Stereo (MBS) technique [15], extended to handle non-parallel cameras. The choice of the MBS algorithm was motivated primarily by two factors. First, MBS recovers dense depth maps — that is, a depth estimate for every pixel in the intensity images — which is useful for surface reconstruction. Second, MBS takes advantage of multiple cameras to increase precision and reduce errors in depth estimation. The algorithm does break down in three situations, however. First, as a window-based method, the MBS algorithm is fundamentally limited by the available image texture; textureless regions in the images generate unreliable (and frequently incorrect) stereo data. Second, and again characteristic of window-based methods, MBS does not reliably localize depth discontinuities as observed in the images. This error can make surfaces appear to be wider or narrower than they should be, as shown in Figure 4. While some efforts have been made to improve performance near these discontinuities (e.g., [16]), this problem is still an open research issue. Third, precision of the depth estimates is limited by the sampling resolution and noise of the images, which limits the resolu-

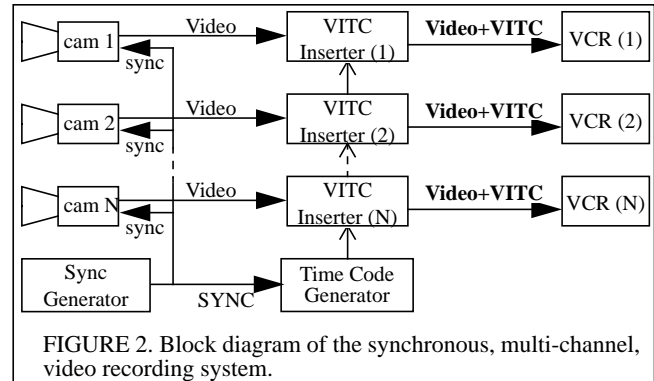


FIGURE 2. Block diagram of the synchronous, multi-channel, video recording system.

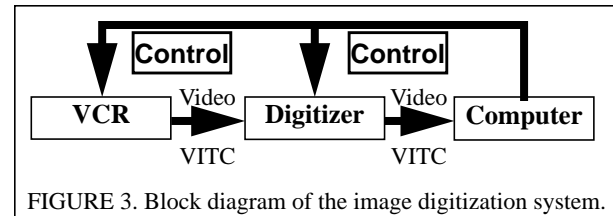


FIGURE 3. Block diagram of the image digitization system.

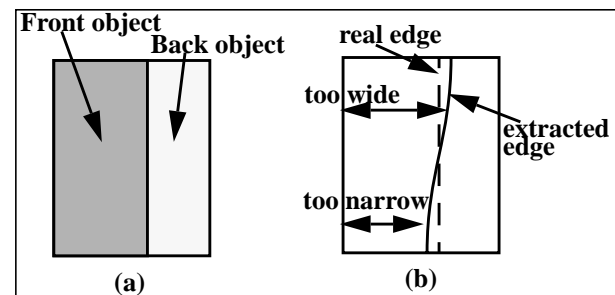
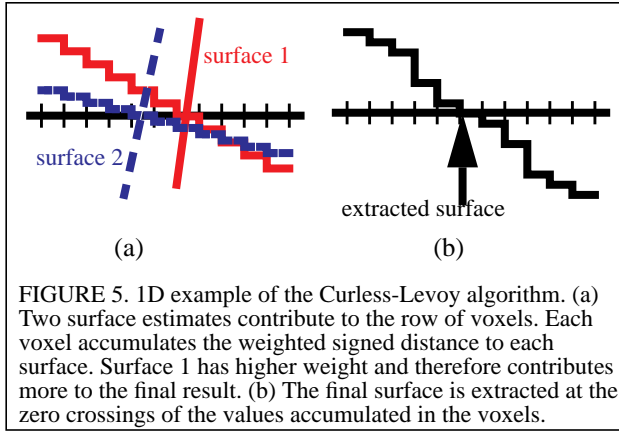


FIGURE 4. Example of poor depth edge (discontinuity) localization in stereo. (a) A view of the scene. (b) The real and extracted edges from stereo. The poor localization can make surfaces appear wider or narrower in the depth maps than they really are in the world.

tion with which a region of one image can be aligned with a region in another image.

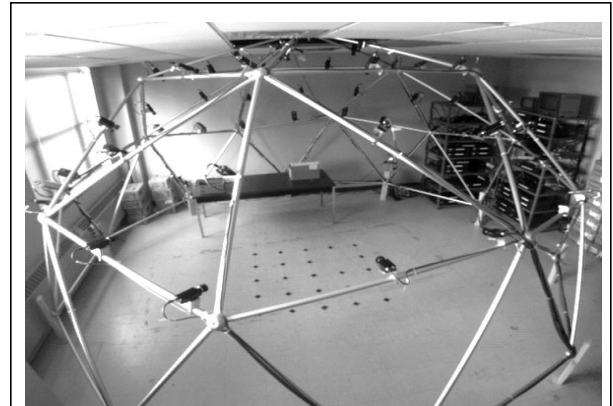
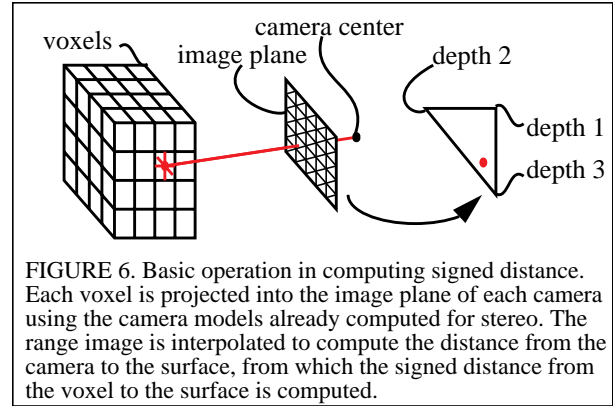
Before we can apply MBS, we must first calibrate each camera in the multi-camera system. We use an approach from Tsai [20], implemented by Reg Willson, that calibrates an 11-parameter camera model to each camera. This model accounts for camera pose in a world coordinate system (3D rotation and translation), scaling (focal length and aspect ratio), shifting of the image (image center), and lens distortion (a single coefficient for radial lens distortion). Rather than calibrate all parameters of each camera at the same time, we separate the process into two steps, one for intrinsic parameters (focal length, aspect ratio, image center, and lens distortion) and the second for extrinsic parameters (rotation and translation). In the first step, each camera images a calibration object with known locations of points in a 3D volume. We then extract the image projections of these 3D



points. The calibration algorithm adapts the camera model to align the image points with the projections of the corresponding 3D points. The second step is to position the cameras in the recording environment and perform the final calibration step, determining the camera pose relative to a coordinate system common to all the cameras. We calibrate using the portion of the lab floor visible to all cameras, having laid out marks on the floor with known separation to provide 3D calibration points as before.

### 3.0 Fusion of Multiple Range Images

The next step is to integrate all the images into one 3D model of the scene at each time sample. Because fusion uses the noisy range images from stereo, the fusion algorithm must be robust, even with systematic errors (common with stereo) rather than random ones. Based on its resilience to noise, simplicity of design, and non-iterative operation, we use an algorithm very similar to the voxel-space merging strategy of Curless and Levoy [2], which itself bears similarity to several other algorithms [6][7]. Their algorithm accumulates, at each voxel, the signed distance to the surfaces in the range images, weighted by any reliability estimates for the sample. Each range image is converted into a set of surfaces by tessellating the image, constructing triangles among the nearest neighbors of each pixel in the image. If neighboring pixels have a large difference in depth, no tessellation occurs between the pixels. A weight may also be attached to each range estimate, allowing easy incorporation of range estimate reliability into the fusion process. Next, each voxel in an object-centered 3D space is projected onto the tessellated surface. The weighted, signed distance to the surface is accumulated at each voxel in the vicinity of the tessellated surface (see Figure 5(a)). After accumulating across all range images, the voxels implicitly represent the surface by the zero crossings of their values. By finding these zero crossings, the surface is extracted (see Figure 5(b)). This process, implicit surface (or isosurface) extraction, is well studied and



has standard solutions such as the marching cubes algorithm [13][22], which generates 3D triangle mesh representations of the implicit surfaces. Note that one of the most important aspects of the approaches of Curless and Levoy (as well as those of [6][7]) is framing the surface recovery problem as one of extracting an isosurface rather than one of finding peaks or ridges. Unlike isosurface extraction, ridge finding is a problem without robust solutions [3].

Our approach is similar to that of Curless and Levoy, but with one noteworthy change. They limit the extent of each tessellated surface to the voxels near it, while we allow the algorithm to adjust all voxels in front of the surface as viewed from the sensor generating this surface. For voxels far in front of the surface, we limit the weighted, signed distance contribution of each viewpoint so that this single view does not overwhelm all others in the fusion process. This modification gives significant improvement in the ability of the algorithm to reject the numerous outliers in our range images, while not significantly degrading the recovered shape. As a result of this change, we touch many more voxels than the straightforward approach of Curless and Levoy, which would reduce the effectiveness of some of their coding

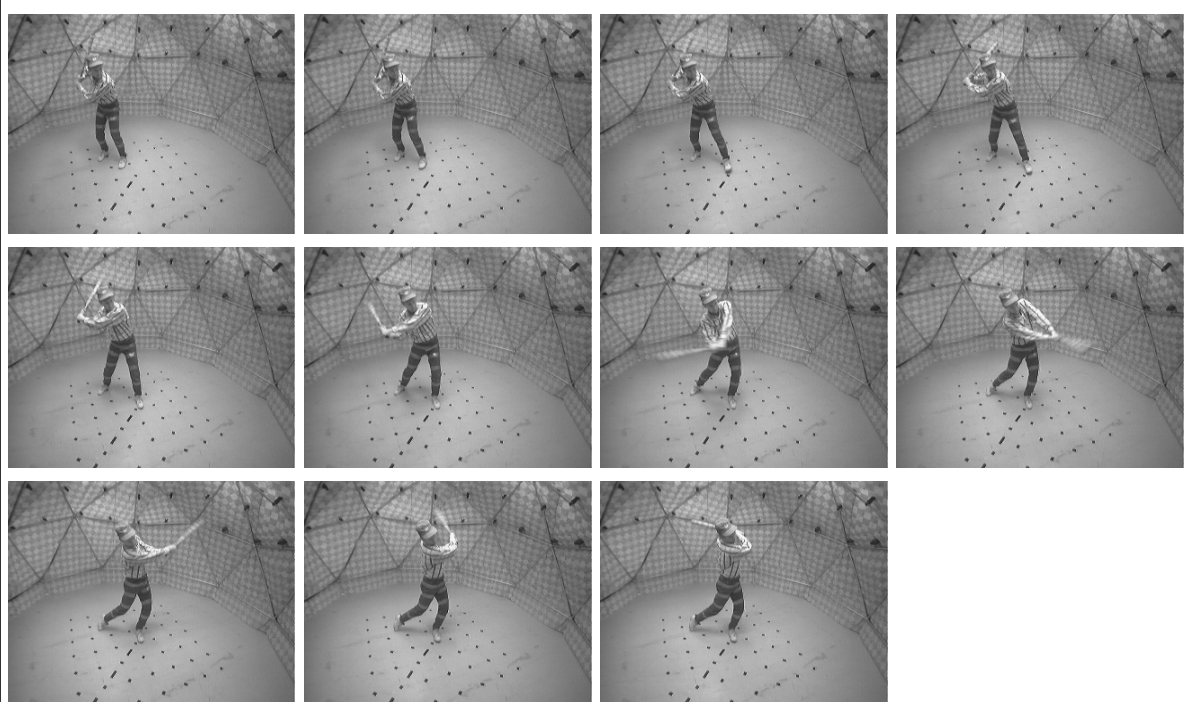


FIGURE 8. One camera's view of the test sequence. One person swings a baseball bat one time.

techniques such as run-length encoding, shear-warp factorization, and binary depth trees; we therefore do not use them. Instead, we forward map (i.e., project) each voxel into each image using the known camera models and interpolate depth in the images, as shown in Figure 6.

## 4.0 Experiments

### 4.1 Hardware Setup and Calibration

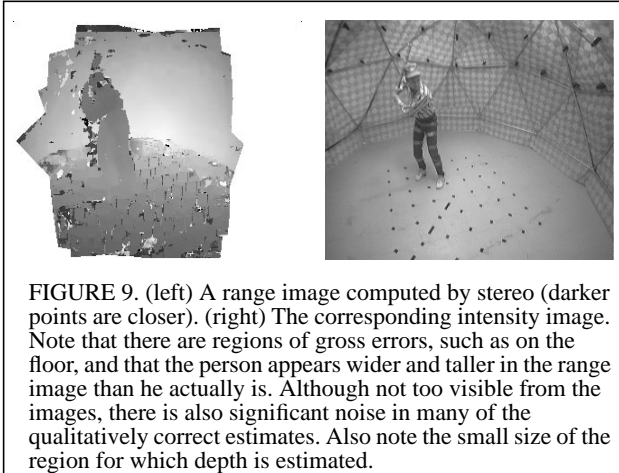
We tested our system on an 11-frame image sequence captured from 51 cameras mounted on a geodesic dome (shown in Figure 7), 5 meters in diameter. The scene contained one person swinging a baseball bat, as shown in the sequence of images in Figure 8. Frames were captured at a rate of 6 per second. (All 30 images per second could have been captured using our system, but we limited the initial tests to the slower sampling for practical considerations.) The sampling interval was 1/60th second, corresponding to one field of an NTSC video frame. Each camera underwent a 2-stage calibration process, as discussed in Section 2.2. Each camera had fixed gain and shutter set to open for the full sampling interval (one field time, or 1/60 second). The lenses were set to manual focus and iris control, adjusted to bring the scene into reasonable focus and intensity. The images from each camera have approximately 90 degrees horizontal by 70 degrees vertical field of view.

### 4.2 Image Capture

To capture the images, we used the synchronized multi-camera recording system discussed earlier, recording NTSC video onto S-VHS VCRs. For this test, we determined that we only needed images at approximately 6 Hz, so we digitized the necessary frames, at 490x640 resolution, using the approach of repeatedly playing the videotape until the necessary frames were captured. Because NTSC video is interlaced, the even and odd rows of each NTSC frame are sampled one field time apart (1/60 second). The change in dynamic scenes during that time may be significant (e.g. Figure 10) for stereo processing, so we separated each video frame into the two fields, each with resolution 245x640, and discarded the second field. We then rescaled the first field back to 245x320. These are the images shown in Figure 8.

### 4.3 Range Computation

With all of the images collected, we next computed range images using the MBS algorithm. We grouped each camera with all of its immediate neighbors to create clusters of 4 to 7 cameras. For each image in each sequence, we used these clusters to compute range images at 245x320 resolution, with depth resolution ranging from 155-264 levels, depending on the actual geometric configuration of each cluster. The depth search range began at 750 mm and ended at 5000 mm from each reference camera, measured along its optical axis. A typical range image is shown in Figure 9.



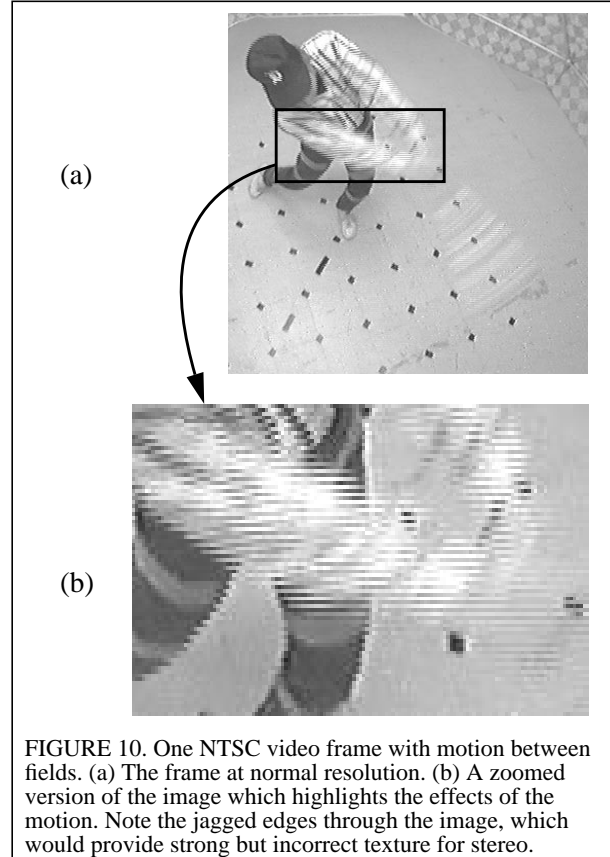
#### 4.4 Range Image Fusion

The range images from stereo were then merged using the voxel fusion process. Although the algorithm can easily exploit reliability estimates, we eliminated all weights for simplicity, treating all samples equally. The merging process was run over the same 3D volume for each time frame, a space 6000mm x 6000mm x 6000mm (which includes all of the sensor positions in addition to the scene). The process used 300x300x300 voxels to represent the space, corresponding to a voxel size of 20mm x 20mm x 20mm.

Note that the voxels in our approach are treated independently, allowing us to process individual voxels or groups of voxels separately. This freedom allows us to group the voxels and the range images for effective use of available memory and of parallel hardware. By allocating only a few planes of voxels at a time, for example, we can greatly reduce memory usage during fusion. By distributing voxel planes over a number of processors -- or completely independent computers -- we easily achieve substantial parallel speedups. In this initial set of experiments, we decomposed several large voxel spaces into sets of planes and process the sets on independent computers. This simple strategy provides nearly linear speedups with the number of computers used.

#### 4.5 Surface Extraction

We next extracted the 3D object surfaces by applying an implementation [22] of the Marching Cubes algorithm [13]. We decimated the resulting mesh using a mesh simplification program written by Andrew Johnson. The sequence of meshes is shown in Figure 11, where we have removed the background so that the person is more clearly visible. The player is clearly extracted from each time instant, although the bat is recovered poorly because the voxel size (2cm on a side) is nearly as large as the bat. This behavior highlights the need for voxels to be smaller than the smallest feature to be extracted from the scene.



#### 5.0 Evaluation of Fusion and Stereo

MBS (and many other stereo algorithms) have difficulty with three tasks: handling textureless image regions, localizing depth discontinuities, and generating precise range estimates. From the perspective of stereo, each problem is somewhat independent of the others, but in the context of fusion, the latter two actually converge to the same problem. That is, fusion sees both poor localization of depth discontinuities and poor precision of range estimates as a problem of imprecise surface boundary localization.

Conceptually, the fusion algorithm should do little to improve handling of textureless regions, because the range estimates for these regions are frequently far from correct. In practice, fusion improved the surface reconstruction of many surfaces with little texture because enough of the range images were correct to overcome the numerous images with large errors. Localization of boundaries, on the other hand, was expected to improve, at least when the range images contained only random noise. For systematic errors, we expected fusion to help little, because the additive contribution of the different viewpoints might not average out to the correct surface.

In our case, the fusion algorithm had to deal with systematic errors because stereo makes consistent error in local-



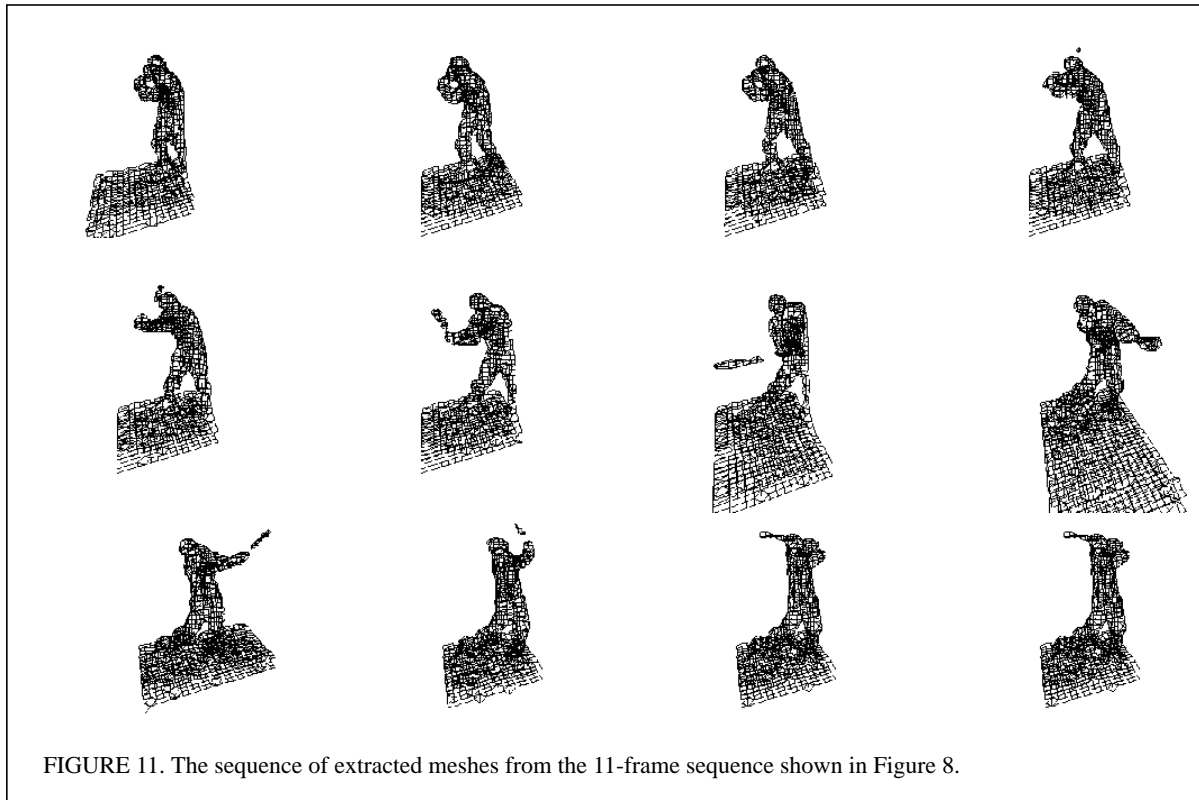


FIGURE 11. The sequence of extracted meshes from the 11-frame sequence shown in Figure 8.

izing depth discontinuities, as shown in Figure 9. This error tends to “fatten” objects, i.e. to increase the width of objects, or to “thin” them, i.e. to decrease their width. This phenomenon (either fattening or thinning) increases linearly with the size of the stereo window and the affected region tends to be oriented in a plane parallel to the image plane. To our surprise, fusion reduces the effects of this error, even when the stereo window size changes. We believe that while no single view localized the true edge of the foreground object, nearly all of the views did contain the edge. Because the false edge extensions are oriented approximately parallel to the image planes, they do not align and therefore destructively interfere with the creation of implicit surfaces in these regions. This process is shown in Figure 12. Note that the destructive interference occurs at any length of the false edges, and therefore is relatively independent of the stereo window size.

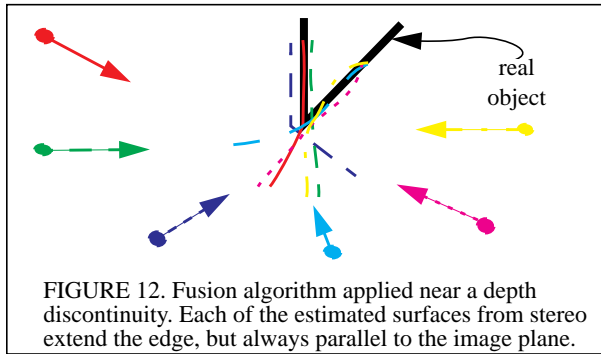
Based on these results, we believe that range image fusion may be a natural location at which to merge hierarchical stereo data. It is not uncommon, for example, to run stereo with a number of window sizes or different image resolutions and then to merge the multiple estimates to improve robustness. In a multi-camera system such as ours, it may be better to compute the range images at each window size or resolution without merging the results at the level of stereo, passing each of the range images along and let the fusion process resolve the ambiguity.

## 6.0 Conclusions

We have presented a system for off-line analysis of dynamic scenes and shown how it can be used to recover scene structure of dynamic scenes. Our multi-camera hardware system performs video-rate image capture, while our off-line digitization provides access to this imagery. A software implementation of the MBS algorithm uses these images to compute range data, which is then fused using an object-centered voxel merging strategy that generates a 3D triangle mesh representation of the surfaces in the scene.

By implementing this system and analyzing its behavior, we have also discovered the ability of the fusion algorithm to greatly reduce the poor depth discontinuity localization of stereo. This discovery suggests a novel method of fusing multi-resolution stereo estimates that holds great promise for resolving a long-standing problem of stereo methods. While this approach requires more evaluation to be considered reliable, we believe that such studies will demonstrate the utility of this approach.

**Acknowledgments:** We would like to thank Andrew Johnson for the use of his mesh decimation program and Reg Willson for the use of his calibration program.



## 7.0 References

- [1] R. Collins, A Space-Sweep Approach to True Multi-Image Matching, *Proceedings of IEEE CVPR'96*, to appear.
- [2] B. Curless and M. Levoy, A Volumetric Method for Building Complex Models from Range Images, *Computer Graphics SIGGRAPH '96*, August 1996.
- [3] D. Eberly, R. Gardner, B. Morse, S. Pizer, and C. Scharlach, Ridges for Image Analysis, *Journal of Mathematical Imaging and Vision*, 4(4):353-373, Dec. 1994.
- [4] P. Fua, Reconstructing Complex Surfaces from Multiple Stereo Views, *Proceedings of International Conference on Computer Vision (ICCV'95)*, 1078-1085, June, 1995.
- [5] H. Fuchs, G. Bishop, K. Arthur, L. McMillan, R. Bajcsy, S.W. Lee, H. Farid, and T. Kanade. Virtual Space Teleconferencing using a Sea of Cameras, In *Proceedings of the First International Symposium on Medical Robotics and Computer Assisted Surgery*, pp.161-167, 1994
- [6] A. Hilton, On Reliable Surface Reconstruction From Multiple Range Images, *Technical Report VSSP-TR-5/95*, University of Surrey, Oct. 1995.
- [7] H. Hoppe, T. DeRose, T. Duchamp, M. Halstead, H. Jin, J. McDonald, J. Schweitzer, and W. Stuetzle, Piecewise Smooth Surface Reconstruction, *Computer Graphics SIGGRAPH'94*, 295-302, 1994.
- [8] R. Jain and K. Wakimoto, Multiple perspective interactive video, In *Proceedings of IEEE Conference on Multimedia Systems*, May 1995.
- [9] T. Kanade, H. Kano, S. Kimura, A. Yoshida, K. Oda, Development of a Video-Rate Stereo Machine, *Proceedings of International Robotics and Systems Conference (IROS'95)*, Aug. 1995.
- [10] T. Kanade, P. J. Narayanan, and P. W. Rander. Virtualized Reality: Being Mobile in a Visual Scene, *International Conference on Artificial Reality and Tele-Existence and Conference on Virtual Reality Software and Technology*, Japan, Nov 1995.
- [11] T. Kanade, P. J. Narayanan, and P. W. Rander. Virtualized Reality: Concept and Early Results, *IEEE Workshop on the Representation of Visual Scenes*, Boston, June, 1995.
- [12] P. Khalili, Forming a Three-Dimensional Environment Model for Autonomous Navigation Using a Sequence of Images, *University of Michigan Ph.D. thesis*, Oct. 1994.
- [13] W. Lorensen and H. Cline, Marching Cubes: a High Resolution 3D surface Construction Algorithm, *Computer Graphics SIGGRAPH'87*, 163-170, July 1987
- [14] P. J. Narayanan, P. Rander, and T. Kanade, Synchronizing and Capturing Every Frame from Multiple Cameras, *Robotics Institute Technical Report, CMU-RI-TR-95-25*, 1995.
- [15] M. Okutomi and T. Kanade. A multiple-baseline stereo, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 15(4):353-363, 1993.
- [16] K. Satoh and Y. Ohta. Passive Depth Acquisition for 3D Image Displays. In *IEICE Transactions on Information and Systems*, E77-D(9), Sep. 1994.
- [17] H.Y. Shum, M. Hebert, K. Ikeuchi and R. Reddy, An integral approach to free-form object modeling, to appear in *IEEE Transactions on Pattern Analysis and Machine Intelligence*. Also appeared in *Proceedings of International Conference on Computer Vision (ICCV'95)*, 870-875, Boston, USA, June, 1995.
- [18] Society of Motion Picture and Television Engineers. American National Standard for Television -- Time and Control Code, *SMPTE Journal*, June 1986.
- [19] M. Soucy and D. Laurendeau, Multi-Resolution Surface Modelling from Multiple Range Views, *Proceedings of IEEE CVPR'92*, 348-353, 1992.
- [20] R. Tsai, A versatile camera calibration technique for high-accuracy 3D machine vision metrology using off-the-shelf tv cameras and lenses, *IEEE Journal of Robotics and Automation*, RA-3(4):323-344, 1987.
- [21] G. Turk and M. Levoy, Zippered Polygon Meshes from Range Images, *Computer Graphics SIGGRAPH'94*, 311-318, 1994.
- [22] J. Bloomenthal, An Implicit Surface Polygonizer, *Graphics Gems IV*, ed. P. Heckbert, 324-349, 1994 (<ftp://ftp-graphics.stanford.edu/pub/Graphics/Graphics-Gems/GemsIV/GGemsIV.tar.Z>).