Scanning and Reconstruction for Dynamic Surfaces

University of Virginia Computer Science Technical Report CS-2006-25

Kevin Dale, Ewen Cheslack-Postava, Greg Humphreys,
and David P. Luebke

Abstract

We present a novel 3D scanning system with the potential for interactive acquisition and visualization of dynamic scenes. Our system uses a spatio-temporally adaptive sampling strategy, and can take advantage of multiple simultaneous scanning devices operating at different resolutions. We also employ a level set framework for reconstructing potentially dynamic scenes from multiple concurrent streams of range data. In our framework, implicit surfaces are reconstructed periodically from new samples on a coarse grid, creating a sequence of reconstructions from disjoint sample sets that is used to estimate motion in the scene. A high-resolution reconstruction proceeds alongside, where the surface is evolved by a convective flow that guides it towards the sample set. We employ a spatially-varying distance metric based on our motion estimate that adaptively controls the contribution of older samples to the final reconstruction.

1 Introduction

Our increasing ability to accurately measure real world objects and phenomena has led to significant recent advances in computer graphics. However, the acquisition and reconstruction process for 3D geometry is still very time-consuming and computationally intensive, often generating gigabytes of data and taking many hours to complete [16]. Most acquisition systems employ a uniform sampling strategy, which can lead to artifacts near silhouette or crease edges. This assumption also leads to redundant data in regions of the scene that are changing slowly, which needlessly increases the required processing time. Furthermore, if the scene being scanned changes during scanning, most existing systems will produce incorrect and unpredictable results. Laser scanning systems produce highly detailed models, but are far too slow to capture dynamic scenes, while structured light approaches are fast but the fidelity of their output is limited by the resolution of the camera and structured light source.

In this paper, we present a novel 3D scanning system to address these issues. Throughout the system we stress adaptive allocation of scanning resources, including sample placement, CPU usage, and reconstruction effort. Our system consists of a prototype scanner, adaptive scan control algorithms, and spatio-temporally adaptive reconstruction software. The scanner is based on a time-of-flight laser rangefinder capable of sampling up to 50 KHz, coupled with a fast, low-resolution depth camera.

We believe the combination of a low resolution camera, either color or depth, and a highly accurate laser rangefinder provides a good compromise within the space of scanners. Structured light scanners lie at one end of the spectrum, providing fast, dense sampling but are limited in precision by the camera and structured light source. Laser rangefinders lie on the other end of the spectrum and produce dense sampling along a given path but cannot cover the entire field of view as quickly as structured light scanners. Somewhere in the middle lie stereo vision cameras which will often not produce a sampling density at the resolution of the capture device but are very fast and cheap, and fixed arrays of time of flight

rangefinders which produce low resolution depth images that are consistently dense. While any sufficiently fast and densly sampling device could be used to augment the laser rangefinder in our system, we find the fixed array of time of flight rangefinders to be an attractive choice because of the consistent sample density and accuracy of the device. The low resolution depth camera is used both to guide sampling and to initialize the reconstruction and the rangefinder is used to generate very dense, accurate samples to use as the destination for the reconstructed surface evolution.

Our system is robust to noise and can capture dynamic scenes in real time by making efficient and adaptive use of available scanning resources. Rangefinder-based devices typically can capture depth information at very fine resolution but must compromise between uniform spatial and temporal coverage. However our device, driven by an adaptive scanning technique, is capable of capturing geometry at rates much closer to that of structured light scanning. Furthermore, our scanning and reconstruction algorithms are easily parallelized, and even support multiple simultaneous scanning devices.

To efficiently capture dynamic scenes, our algorithms detect and focus on regions of spatial and temporal change. In particular, we build on the frameless sampling and reconstruction techniques introduced by Dayal et al. [10]. We have extended their adaptive sample-point selection ideas to choose entire sampling *paths* for the laser to follow in order to maximize new information to be gathered. In this way, scanning resources can quickly find small features and object edges to yield extremely high fidelity object reconstructions, while simultaneously tracking moving objects and providing a real-time reconstruction of a dynamic environment.

To address the issue of computational complexity and interactivity, we exploit temporal coherence with a reconstruction algorithm that is inherently parallelizable and handles both rigid and deforming objects. The basic idea is that reconstruction filters are shaped in *three* dimensions based on an estimate of the local gradient. In this way, old samples are given less weight in areas with a large temporal gradient, since they are likely to be less representative of the current surface. If the temporal gradient is low, old samples can be reused, so static portions of the scene quickly converge to a very detailed model. These adaptive reconstruction filters are combined with a level-set approach which is robust to noise and outliers, and produces very high quality 3D models.

The remainder of the paper is organized as follows. Section 2 reviews related work. Sections 3 and 4 describe our sampling and reconstruction algorithms, respectively. Section 5 shows our results. We conclude in Section 6 and and also describe areas of future research.

2 Related Work

Surface acquisition and reconstruction is a well-studied field; here we refer mainly to the most relevant work to our own.

2.1 3D Scanning Techniques

Rusinkeiwicz et al. [22] present a system for acquiring 3D models in real-time. The user holds an object in front of a 60 Hz structured-light rangefinder and the display is constantly updated showing the most recent version of the scanned model. Because of the immediate feedback, the user is able to quickly identify holes in the model and rotate the model accordingly to acquire missing regions.

Blais et al. describe a method to recursively refine a model obtained from sparse range data using the iterated closest points algorithm. [6, 7]. Their system also tracks the object to center scanning on the object, resulting in an accurate, high resolution 3D model of the moving object.

The DeltaSphere-3000 Scene Digitizer [FIXME: cite] is a commercial 3D scanner. It can scan a very wide field of view (360° horizontal, 290° vertical), but is relatively slow, typically spending 12 minutes to perform a high resolution 360° scan. The DeltaSphere treats reconstruction as a post-process, generating triangle meshes after the scan has completed.

Koninckx et al. describe an adaptively coded structured light approach that is able to efficiently

scan deforming surfaces by exploiting temporal coherence [13]. Our system has similar goals but can produce much higher fidelity models due to the resolution of our laser rangefinder.

2.2 Surface Reconstruction

Reconstruction of static surfaces has been implemented in a number of fashions. Explicit schemes have been used that triangulate data based on Voronoi criteria [3, 11]. Implicit shemes have also proven effective; a common formulation of the problem is to find a scalar-valued function with an isocontour that approximates the surface, from either unorganized points [12] or aligned range images [9, 26]. [12, 26] present strategies that compute the most likely surface given a set of range maps. Zhao et al. [28] present a similiar variational formulation of surface reconstruction from unorganized points in which they introduce a fast tagging algorithm to construct an initial approximation of a minimal surface, speeding reconstruction by an order of magnitude.

2.3 Level Set Methods

The approaches taken in [26, 28] are based on the level set methods of [18] for stable and efficient deformation of implicit surfaces. Fast marching methods provide an efficient, non-iterative means to track propagating fronts but are only applicable to monotonically advancing fronts [23]. General level set techniques are required for surfaces that do not meet this criteria, such as those in this work. For iterative level set methods, the narrow band [1] and sparse field [26] approaches provide efficient solutions by limiting computation to only those cells close to the target level set, effectively reducing computational complexity by an order of magnitude.

Techniques based on level set methods for segmentation and tracking in video are similar to the techniques described in this paper. The *geodesic active contour model* iteratively refines a goal curve by minimizing an energy function; this process is equivalent to finding the geodesic curve that best matches specified image characteristics [20]. Methods in [5] track segmentations of interest across frames of video by coupling their evolution to an image-based deformation.

3 Adaptive Sampling

Most previous scanning systems, especially those based on lasers, offer only uniform sampling strategies. Our system samples the target adaptively, focusing effort on spatial edges such as silhouettes and geometric detail, as well as temporal edges such as occlusion events and deformations. We find these events using a low resolution camera (either depth or color), and our sampling algorithm guides the scanner towards these regions when appropriate.

The scanner effectively restricts sample distributions to fall along piecewise-smooth curves in the scanner's field of view. Although it would be possible to generate discrete sampling patterns, this would be inefficient because physical scanners scanner spend the majority of their time repositioning the mirrors for the next sample and very little time actually sampling. For this reason, we cast the laser sampling problem as a path planning problem. Specifically, we seek the path $\gamma(t) : [0, \infty) \rightarrow \Re^2$ from the starting mirror orientation $\gamma(0)$ that minimizes the following cost function:

$$C(\gamma) = \int_0^\infty \frac{D(\gamma(t), t)}{I(\gamma(t), t)}\, dt,$$

where $D(\gamma(t), t)$ is a sample density estimate and $I(\gamma(t), t)$ is an importance measure. The calculation and update of I and D are discussed in detail in Section 3.2. Note that this problem is ill-posed because even with global information the optimal path must minimize the cost for all times in the future. Our algorithm is a greedy approximation to the optimal path.

3.1 Gradient Descent Path Planning

We use a multi-resolution gradient descent method to greedily select the new direction to move the laser. The base importance map I and density estimate D are both maintained as image pyramids [27] without

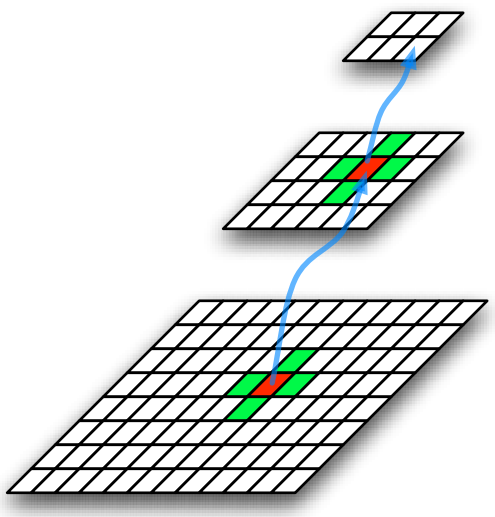

Figure 1: The gradient descent algorithm operates by examining the neighborhood surrounding the grid cell containing the current laser position at the finest level of detail. If the importance of that cell is above a threshold, the search stops and the laser is guided towards the neighbor cell with the highest importance. Otherwise, the process is repeated at the next coarser level. If no motion is suggested by the importance maps, the laser falls back on a raster-scan pattern.

the topmost level (e.g., the top level in our representation is a 2×2 image; see Section 3.2). Beginning at the bottommost cell containing the laser location, we ascend the pyramid looking for a neighboring grid cell that would yield a path γ with low cost $C(\gamma)$. A low path cost indicates that the cell is insufficiently sampled given its current importance estimate. If the cost is below a user specified minimum then we stop ascending the pyramid and choose to move the laser towards that grid cell. If no neighboring grid cell satisfies this condition at the current level, we continue to the next level of the pyramid. If the entire pyramid is traversed and no direction has been chosen, we fall back on a raster scan pattern.

To produce a smooth curve, we interpolate the chosen next direction with the current direction. This is beneficial because the physical limitations of scanner hardware make paths with high curvature very expensive, as the mirrors must decelerate to follow a sharp turn. A smoother curve will therefore allow the scanner to cover the field of view more quickly.

The primary benefit of using a multi-resolution method is that the scanner will focus on spatially coherent, highly important regions, but will also perform broad scans as distant regions become increasingly important. Furthermore, it guarantees that broad scans will be performed periodically. If a region is neglected, its associated sampling density will drop and the resulting low cost will draw the scanner away from other regions. In addition, a multiresolution technique allows the scanner to intelligently allocate available samples by spending more time locally when other areas are relatively unimportant, resulting in increasingly refined reconstructions.

3.2 Updating Importance and Density

A single user-specified parameter $\alpha \in [0, 1]$ controls the relative importance of spatial and temporal change. Specifically, the importance function $I(\theta, \phi, t)$ is defined as

$$I = \alpha T + (1 - \alpha)S + \epsilon,$$

where $S(\theta, \phi, t)$ and $T(\theta, \phi, t)$ are measures of spatial and temporal incoherence, respectively. ϵ is a small constant used to ensure the importance function is non-zero everywhere.

For $\alpha = 0$, the scanner is responsive solely to spatial incoherence; the scanner will focus only on edges and will be slow to respond to moving objects. For $\alpha = 1$, the scanner is responsive only to temporal changes; the scanner would focus almost entirely on the dynamic regions of the scene, or, in the case of a static scene, would degrade to a uniform raster sampling pattern. In practice, we have found that the value $\alpha = .7$ provides an adequate balance between spatial and temporal adaptivity, although it would likely be fruitful to dynamically update α as the scanner learns about the varying ratios of spatial and temporal change in the scene being acquired.

With each image captured from the camera we update S and T, and in turn I. The spatial importance measure S is a simple Canny edge detection filter [8]. The temporal importance measure T is calculated using a motion history computed over a predetermined length of time (we use 1 second). At each frame, pixels with significant temporal change are found by performing binary thresholding on the absolute difference of the current and previous video frame. The motion history image contains either the most recent time at which a significant color difference was detected at each pixel or zero if the most recent color difference is not within the tracked interval. This motion history image is then scaled to fit the correct range of values for importance so that pixels for which a change was detected will have the highest temporal importance measure and pixels which have not changed recently will have the smallest temporal importance measure. Once new S and T images have been computed, the base importance image and then the entire importance pyramid are updated.

There are two steps to tracking the sample density estimation D. The first step occurs every time a new goal position is selected. The path between the current position and the goal position is traversed at the base density image level. For each pixel intersected by the selected path, we visit each level of the pyramid and increment the density of the cell at the level which contains the pixel, adjusting the increase in density appropriately based on the current level. Although this is only approximate (because the laser path might be updated before or after it reaches its goal), it avoids latency problems that would arise in computing the true density. This way, the path planner can avoid getting stuck in local regions that will not be updated for a long time.

The second update to the density estimation occurs each time a camera frame is collected. This update handles static decay of samples. As samples grow older our confidence in them decreases, so older samples contribute less weight than newer samples. The static decay rate, measured in samples/pixel/second, accounts for this effect. We approximate this decay by reducing the density values in the density map by a constant decay rate R, scaled by the elapsed time Δt since the last update:

$$D = D - R\Delta t.$$

Finally, we must account for regions of temporal change. In regions where there was motion, the previous samples are no longer valid. When we detect such a region, we set the corresponding density estimates to zero, using T from the above importance map calcuation as a binary mask. After the the density is fully estimated, its associated mipmap pyramid is regenerated.

4 Reconstruction

Our reconstruction algorithm extends the level set-based reconstruction technique of [28] to dynamic surfaces. Iterative level set methods are inherently parallelizable, as demonstrated by implementations in [4, 15]. While we have not parallelized our technique, this possibility motivated our selection of a level set-based reconstructor over other methods.

The high level idea is to use fresh samples to reconstruct a coarse approximation of the current scene. Differencing successive coarse approximations provides a motion estimation over the domain, which is then used to determine the contribution of future samples to the high-fidelity reconstruction. To do this, we employ an adaptive, non-euclidean distance metric that allows old samples to contribute to static regions of the scene, while pruning them from dynamic regions. The remainder of this section describes this algorithm in detail.

4.1 Level Set Methods and Static Reconstruction

Here, we briefly review level set methods and the relevant details from [28], upon which our technique is based. The level set method [18] provides a set of tools for deforming implicit surfaces. Specifically, a deformable surface Γ is modeled as the k-level set of a time-varying scalar function $\phi(\boldsymbol{x}, t)$ in 3D:

$$\Gamma(t) = \{\boldsymbol{x} | \phi(\boldsymbol{x}, t) = k\},$$

for $k \in \Re$, $\boldsymbol{x} \in \Re^3$, and $t \in \Re^+$. In [28], for a general data set S, they compute the distance function to S, $d(\boldsymbol{x}) = dist(\boldsymbol{x}, S)$, over the domain. A non-iterative tagging algorithm finds an initial approximation of the surface, at which point the embedding is evolved via the convection equation

$$\frac{\partial \phi}{\partial t} = \nabla(d(\boldsymbol{x})) \cdot \nabla\phi.$$

To smooth the reconstructed surface, evolution proceeds according to their minimal surface model,

$$\frac{\partial \phi}{\partial t} = |\nabla\phi| \left[\nabla d \cdot \frac{\nabla\phi}{|\nabla\phi|} + d\nabla \cdot \frac{\nabla\phi}{|\nabla\phi|} \right],$$

which contains an additional mean curvature term that smooths the reconstruction in proportion to local sample density. They employ the narrow band algorithm [1] in their iterative solver, which restricts computation to a band of cells around the zero level set.

The level set approach to surface reconstruction and evolution automatically handles topological changes such as merging and pinching, which are difficult to handle with explicit surface representations. Moreover, it handles the relatively sparse, non-uniform, and somewhat noisy sample sets generated by our scanner with relative ease [28]. When evolving the surface, this method computes geometric information such as normals and curvature at each iteration, yet computation at a given cell depends only on that cell's local neighborhood. So the SIMD nature of these techniques can be exploited by parallel implementations [4,14]; this idea is discussed more fully in Section 6. Additionally, the narrow band approach reduces both the storage and computational complexity of reconstruction over an N^3 grid to $O(N^2)$, making it comparable to explicit surface representations in this regard.

4.2 Dynamic Targets

For each set of incoming low-resolution depth maps, we perform static reconstruction on the sample points, similarly to the method discussed above, with two exceptions. Like [28], we use a signed distance function for our embedding ϕ, as it improves accuracy and efficiency of numerical computations. The convection model does not maintain this property, however, and requires periodic reinitialization to restore ϕ to a signed distance function–i.e., $|\nabla\phi| = 1$. An iterative approach to reinitialization solves the reinitialization equation [25]:

$$\phi_t + S(\phi)(|\nabla\phi| - 1) = 0.$$

$S(\phi)$ is the numerically smeared sign function from [21]:

$$S(\phi) = \frac{\phi}{\sqrt{\phi^2 + |\nabla\phi|^2(\Delta x)^2}}$$

To avoid periodically reinitializing, we add a reinitialization term to the convection equation itself, yielding

$$\frac{\partial \phi}{\partial t} = \nabla(d(\boldsymbol{x})) \cdot \nabla\phi - S(\phi)(|\nabla\phi| - 1)$$

Numerical error during reinitialization generally causes the zero level set to move [19]. This additional term produces the same problem here, and while there are more sophisticated reinitialization techniques to avoid this, we have found the impact to be minimal. We additionally omit the final minimal surface step, whose mean curvature term requires a prohibitively small timestep. We found that the convection model alone produces sufficient results in our tests.

The low-resolution depth maps alone do not provide a sufficient sampling of the target; however, they provide enough information to capture low-frequency geometric features. Most importantly, we can approximate motion of the target surface by differencing successive low-resolution static reconstructions. On the zero level set of ϕ at world-time τ, this differencing, $\Delta\phi\Delta\tau$, approximates the motion in the normal direction, and is conveniently defined across all level sets.

After the differencing step, we update the unsigned distance function $d(\boldsymbol{x})$ for the full reconstruction with both the low-resolution depth maps and with new samples from the range scanners. During this step,

Figure 2: The sample static scene, the associated low resolution depth image, and the sample distribution using our adaptive technique. Note that important regions, i.e. depth discontinuities, are sampled at a higher rate than geometrically uninteresting areas.

we employ the following distance metric based on the previous differencing step. For space-time points $a, b \in \Re^4$,

$$dist(a,b)^2 = (a_x - b_x)^2 + (a_y - b_y)^2 + (a_z - b_z)^2 + k|\frac{\Delta\phi(a)}{\Delta\tau}\frac{\Delta\phi(b)}{\Delta\tau}|(a_t - b_t)^2,$$

for a user-defined scale factor k. We store d for the high-resolution sample distance field in a regular grid, and additionally store the range sample that is closest to the given cel In particular, we update d by placing each new sample into its corresponding grid cell, placing these sample-cell pairs into a queue, and computing distance based on the above metric. For each updated cell-sample pair, the new sample is tested against the cell's six neighboring cells in a similar fashion. This process continues until the queue is exhausted.

The convection model assumes a euclidean metric, and additionally requires that samples are bounded by the initial surface in order to converge on a reasonable reconstruction [28]. We perform two additional steps so that we can use the convection model. First, we evolve ϕ in the normal direction for some distance δ, so that the zero level set contains the previous interior region and the new samples.

Then, we perform an iterative reinitialization of d, iterating over only those cells in the current narrow band. This restores d to an unsigned distance field for the purpose of reconstruction; however, each cell still contains the position and time of the closest sample for use in the next d update. Once this is complete, we evolve ϕ via the convection model to obtain the current reconstruction at world-time τ.

5 Results

Although we built a prototype scanner, it is necessary to use multiple scanners to capture and reconstruct dynamic scenes using our reconstruction technique. We chose to simulate multiple scanners in software with objects represented by highly detailed meshes. The simulated scanner consists of a laser rangefinder with performance roughly identical to that of the hardware as well as a low resolution depth camera, providing a compromise between spatial and temporal resolution. We examine the two parts of our system with two scenarios: a static target and a dynamic target.

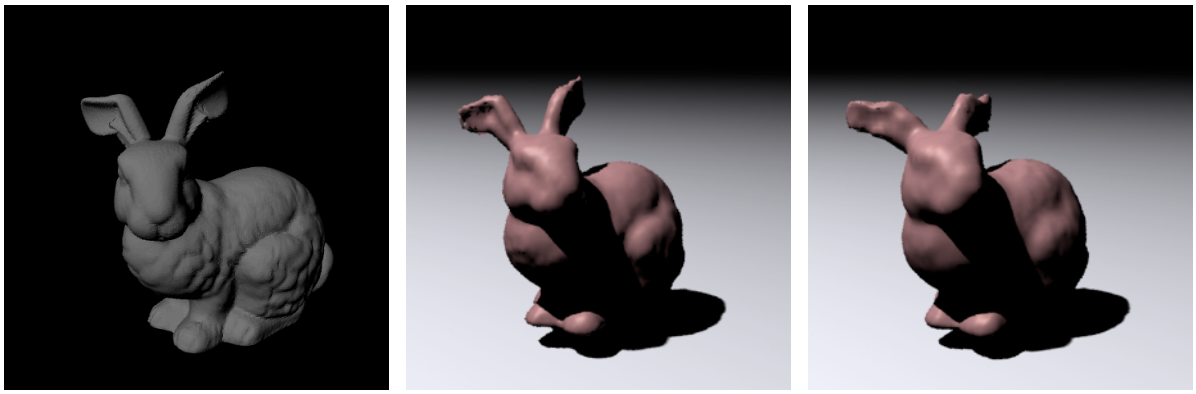

Figure 3: A frame from the original animation and two sample frame results sampled and reconstructed using our system.

5.1 Static target

Here we show the effectiveness of our adaptive sampling technique by scanning a static object. We scan a sample scene, a room from Monticello, using a standard raster sample pattern and our adaptive sampling approach. The raster pattern produces uniformly distributed samples as we would expect from any standard laser rangefinder scanner. We run the scanners for the same amount of time to generate the same number of samples. The results of a 3 second scan using our adaptive method is shown in Figure 2.

Note that the raster pattern wastes samples in areas with no geometric detail whereas our system intelligently dedicates more samples to geometrically important regions and therefore collects more useful information using fewer samples.

5.2 Dynamic target

To test a dynamic target we use a Stanford bunny with animated ears. This represents a difficult case: a balance must be struck between static areas and dynamic areas, both with high geometric detail. The static geometry must be scanned enough to improve reconstruction detail, but the dynamic region must be scanned often enough to maintain at least a rough approximation of the surface.

We arranged four virtual scanners around the bunny in physically plausible positions and orientations. Each scanner roughly matches the specifications of our prototype scanner.

The simulated scanners were arranged about the bunny such that no scanner was covering the bottom of the bunny. This is analogous to the object being placed on a surface for scanning, obscuring the base of the object. This leaves a large region unsampled and thus a large hole in the set of samples. We add a constraint such that the plane upon which the bunny rests blocks the evolution of the surface in the area beneath the bunny. Results from this simulation can be seen in Figure 3.

Our sampling density must meet a number of criteria in order to resolve fine features of the surface. In a simple static reconstruction, sampling densities must satisfy conditions similar to those in [2,28]. However, we must extend these ideas into the temporal dimension as well since we are capturing dynamic surfaces. Because we rely on a video signal from an on-board camera, temporal aliasing can occur if there are motion features that cannot be resolved at the camera's refresh rate.

6 Conclusions and Future Work

We have presented an spatially and temporally adaptive 3D scanning system. The system has a combination of the best aspects of previous scanning solutions: high resolution output characteristic of previous laser rangefinder-based systems and the speed of structured light solutions.

Although the results presented in this paper use simulated range data, we have built a prototype scanner that is capable of following the paths computed by our sampling algorithms (Figure 4). We have not yet integrated this device fully into our scanning system; this remains for future work.

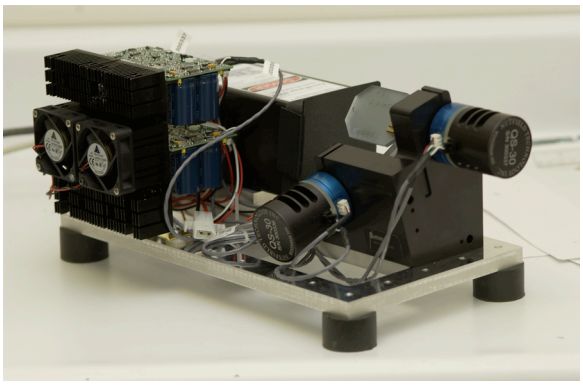

Figure 4: Our prototype scanner is composed of an infrared 780nm laser rangefinder capable of producing range samples at 50 KHZ. A pair of galvonometer-mounted mirrors control sampling direction.

Our scanning system is inherently parallelizable. An arbitrary number of physical scanners can be used, each operating independently based on its own importance map. The reconstructor can also be parallelized and accelerated using programmable graphics hardware; fast distance field computation and a narrow-band level-set solver have been implemented on the GPU [15, 24]. Level set solvers have also been efficiently parallelized on multiprocessor systems [4]. Proper implementation of a scalable system will require attention to load-balancing, routing samples to appropriate solvers, and minimizing the need for inter-block communication in the solvers. This parallel system would also benefit from the loose coupling between sampling and reconstruction in our system.

Although the narrow band level set method provides an efficient method for surface evolution, other implementations may prove more suitable for our adaptive sampling approach. For example, the octree representation used by Losasso et al. [17] aligns well with our spatially adaptive approach. Another possibility would be to represent the surface embedding with a spatially adaptive tetrahedral mesh. However, extending these methods to parallel implementations is an unsolved problem.

Our reconstruction technique could be enhanced by various computer vision techniques such as image segmentation and motion tracking. For example, certain optimizations could be made if we could classify the motion. If we could determine an object is undergoing only rigid body transformations then it might be possible to find the transformation directly and arrive immediately at a converged solution.

Another possible extension of our techniques would be defining importance as a function over the space being scanned rather than the fields of view of the scanners. This requires collaboration between the scanners but would also allow us to evaluate the next best view and dynamically allocate scanners to the positions and orientations they are most needed.

References

[1] Adalsteinsson, D., and Sethian, J. A. A fast level set method for propagating interfaces. *J. Comput. Phys. 118*, 2 (1995), 269–277.

[2] Amenta, N., Bern, M., and Eppstein, D. The crust and the β-skeleton: Combinatorial curve reconstruction. *Graphical models and image processing: GMIP 60*, 2 (???? 1998), 125–??

[3] Amenta, N., Bern, M., and Kamvysselis, M. A new voronoi-based surface reconstruction algorithm. In *SIGGRAPH '98: Proceedings of the 25th annual conference on Computer graphics and interactive techniques* (New York, NY, USA, 1998), ACM Press, pp. 415–421.

[4] Awate, S. P., and Whitaker, R. T. An interactive parallel multiprocessor level-set solver with dynamic load balancing. Tech. rep., University of Utah School of Computing, 2004.

[5] Bertalmío, M., Sapiro, G., and Randall, G. Morphing active contours. *IEEE Trans. Pattern Anal. Mach. Intell. 22*, 7 (2000), 733–737.

[6] Blais, F., Picard, M., and Godin, G. Recursive model optimization using icp and free moving 3d data acqusisition. In *Proceedings of 3DIM* (October 2003), pp. 251–258.

[7] Blais, F., Picard, M., and Godin, G. Accurate 3d acquisition of freely moving objects. In *Proceedings of the 2nd International Symposium on 3D Data Processing, Visualization, and Transmission* (Sept. 2004), vol. 0, pp. 422–429.

[8] Canny, J. A computational approach to edge detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence 8*, 6 (1986), 679–698.

[9] Curless, B., and Levoy, M. A volumetric method for building complex models from range images. In *Proceedigs of SIGGRAPH '96* (New York, NY, USA, 1996), ACM Press, pp. 303–312.

[10] Dayal, A., Wooley, C., Watson, B., and Luebke, D. Adaptive frameless rendering. In *Proceedings of the 2005 Eurographics Symposium on Rendering* (June 2005), pp. 265–275.

[11] Edelsbrunner, H. Shape reconstruction with delaunay complex. In *LATIN '98: Proceedings of the Third Latin American Symposium on Theoretical Informatics* (London, UK, 1998), Springer-Verlag, pp. 119–132.

[12] Hoppe, H., DeRose, T., Duchamp, T., McDonald, J., and Stuetzle, W. Surface reconstruction from unorganized points. In *Proceedings of SIGGRAPH '92* (New York, NY, USA, 1992), ACM Press, pp. 71–78.

[13] Koninckx, T. P., Griesser, A., and Gool, L. V. Real-time range scanning of deformable surfaces by adaptively coded structured light. In *Fourth International Conference on 3-D Digital Imaging and Modeling (3DIM03)* (October 2003), S. Kawada, Ed., IEEE Computer Society, pp. 293–302.

[14] Lefohn, A., Cates, J. E., and Whitaker, R. T. Interactive, gpu-based level sets for 3d segmentation. In *Proceedings of Medical Image Computing and Computer Assisted Intervention (MICCAI) 2003* (2003), pp. 564–572.

[15] Lefohn, A., Kniss, J., Hansen, C., and Whitaker, R. A streaming narrow-band algorithm: Interactive deformation and visualization of level sets. *IEEE Transactions on Visualization and Computer Graphics 10*, 4 (2004), 422–433.

[16] Levoy, M., Pulli, K., Curless, B., Rusinkiewicz, S., Koller, D., Pereira, L., Ginzton, M., Anderson, S., Davis, J., Ginsberg, J., Shade, J., and Fulk, D. The digital michelangelo project: 3d scanning of large statues. In *Proceedings of SIGGRAPH 2000* (New York, NY, USA, 2000), ACM Press/Addison-Wesley Publishing Co., pp. 131–144.

[17] Losasso, F., Fedkiw, R., and Osher, S. Spatially adaptive techniques for level set methods and incompressible flow.

[18] Osher, S., and Fedkiw, R. Fronts propogating with curvature dependent speed, algorithms based on a hamilton-jacobi formulation. *Journal of Computational Physics 79* (1988), 12–49.

[19] Osher, S., and Fedkiw, R. *Level Set Methods and Dynamic Implicit Surfaces*, first ed. Springer-Verlag, 2001.

[20] Paragios, N., and Deriche, R. Geodesic active contours and level sets for the detection and tracking of moving objects. *IEEE Transactions on Pattern Analysis and Machine Intelligence 22*, 3 (Mar. 2000), 266–280.

[21] Peng, D., Merriman, B., Zhao, H., Osher, S., and Kang, M. A pde based fast local level set method, 1999.

[22] Rusinkiewicz, S., Hall-Holt, O., and Levoy, M. Real-time 3d model acquisition. In *ACM Transactions on Graphics (Proceedings of SIGGRAPH)* (July 2002), vol. 21, pp. 438–446.

[23] Sethian, J. A. A fast marching level set method for monotonically advancing fronts. *PNAS 93*, 4 (1996), 1591–1595.

[24] Sud, A., Otaduy, M. A., and Manocha, D. Difi: Fast 3d distance field computation using graphics hardware. In *Computer Graphics Forum (Proceedings of EUROGRAPHICS)* (2004), vol. 23, pp. 557–566.

[25] Sussman, M., Smereka, P., and Osher, S. A level set approach for computing solutions to incompressible two-phase flow. *J. Comput. Phys. 114*, 1 (1994), 146–159.

[26] Whitaker, R. A level-set approach to 3d reconstruction from range data. *International Journal of Computer Vision 29*, 3 (1998), 203–231.

[27] Williams, L. Pyramidal parametrics. In *Proceedings of SIGGRAPH '83* (New York, NY, USA, 1983), ACM Press, pp. 1–11.

[28] Zhao, H.-K., Osher, S., and Fedkiw, R. Fast surface reconstruction using the level set method. In *Proceedings of IEEE Worshop on Variational and Level Set Methods in Computer Vision* (July 2001), pp. 194–201.