MULTI-RESOLUTION SPACE CARVING USING LEVEL SET METHODS

Gregory G. Slabaugh, Ronald W. Schafer

Georgia Institute of Technology
Center for Signal and Image Processing
Atlanta, GA 30332

Mat C. Hans

Hewlett-Packard Laboratories
Client & Media Systems Lab
Palo Alto, CA 94304

ABSTRACT

We present a multi-resolution space carving algorithm that reconstructs a 3D model of visual scene photographed by a calibrated digital camera placed at multiple viewpoints. Our approach employs a level set framework for reconstructing the scene. Unlike most standard space carving approaches, our level set approach produces a smooth reconstruction composed of manifold surfaces. Our method outputs a polygonal model, instead of a collection of voxels. We texture-map the reconstructed geometry using the photographs, and then render the model to produce photo-realistic new views of the scene.

1. INTRODUCTION

Digitizing real-world 3D objects is a challenging task. A number of 3D scanning approaches, such as shape from shading, laser range scanning, structured light, etc. have been described in the literature. Many of these techniques require specialized hardware that is expensive and/or difficult to use. We propose in this paper a 3D photography method that uses photographs (also called *reference views*) taken with inexpensive and simple to use digital cameras.

Our method is related to space carving approaches [1, 2, 3] recently proposed. These methods produce a voxel-based 3D model of the scene by removing voxels that are not consistent with the reference views. The voxels are processed in a similar fashion to the way a sculptor would chip away at a block of marble to reveal a shape. Upon completion of the space carving algorithm, the remaining voxels that are consistent with the reference views form a thin-shelled surface that, when texture-mapped, can be used to render photo-realistic views of the scene from new viewpoints.

While space carving is effective, it has some limitations. The reconstructed surface can be ragged and irregular, especially where there are cusps [4] due to homogeneous colors. Our level set approach mitigates this problem by including a curvature flow term, which produces a smoother reconstruction. Reconstructions produced with space carving can have small unwanted holes and extraneous voxels floating in space. Since our method represents the reconstruction implicitly, it produces a reconstruction composed of watertight surfaces. Small, extraneous geometry has high curvature, which our method penalizes.

When determining if a voxel should be carved, typical implementations rely upon a threshold parameter, and stop carving as soon as the reconstructed surface is “good enough”. Consequently, the algorithm typically stops short [5] of finding a better reconstructed surface, resulting in a model that is fatter than the true object. In contrast, our approach allows the reconstruction to proceed after a standard space carving algorithm would have terminated. This results in a thinner reconstruction, and helps mitigate the problem of extraneous floating geometry. We extract the zero level set using the marching cubes method, producing a polygonal model renderable with standard graphics hardware.

Our algorithm employs a level set approach [6] for surface evolution. An initial surface is embedded as the zero level set of a volumetric function $\psi(x, y, z, t)$. This surface then moves along its inwardly pointing normal, with a speed based on a measure of how well a point locally on the surface accounts for the reference views. Level set theory provides an accurate and stable numerical scheme that solves the partial differential equations (PDEs) that characterize the motion for the surface. Topological changes are naturally accommodated in this framework.

The goal of our approach is to deform the initial surface until it forms a shape that, when texture-mapped and projected to the camera viewpoints, reproduces the photographs. Such an approach is common to several stereo methods that use level set methods [7, 8, 9]. Unlike [9], the speed that our surface evolves is based on matching colors across views. Unlike [7], our method is a multi-resolution approach. When the resolution is increased, we dilate the surface, as done in [10], and then re-execute the surface evolution at a higher resolution. Our method uses photo-consistency [1] instead of cross-correlation [7, 8] to control the speed of evolution.

2. APPROACH

2.1. Photo-consistency

Our method identifies surfaces in the scene using photo-consistency [1]. To be photo-consistent, a point in 3D space must satisfy two criteria. First, the point cannot project to a background region in any photograph. For some scenes, the photographs are segmentable into foreground / background regions. Foreground pixels indicate where the objects being reconstructed project in a photograph - everything else is background. A 3D point in space that projects to background cannot be part of the scene being reconstructed.

Second, for a 3D point to be photo-consistent, the light exiting the point (i.e. radiance) in the direction of each photograph must be equal to the observed color in the photograph, when the point is visible in the reference view. For simplicity, one often assumes that the scene is Lambertian, although this is not strictly necessary. Under this assumption, a point on a scene surface will project to a similar color in each photograph. Therefore, to determine photo-consistency, one can simply project a 3D point into each image, and then match colors across the photographs that have visibility of the point. Due to image noise, errors in camera calibration, etc., the colors will likely not exactly match even for a point on the true surface being reconstructed.

In our algorithm, we model points as voxels, i.e. small cubes. When determining photo-consistency, we project a voxel into each reference view, and collect the set of visible pixels in the photograph to which the voxel projects. Visibility can be established using item buffers [2] or ray tracing [8]. Conventionally, a space carving approach then computes the standard deviation of all the pixels in the projection. This standard deviation is a measure of the color mismatch between views. If the standard deviation is above a user-defined threshold, the voxel is declared inconsistent. Rather than making a binary decision (consistent or inconsistent) on a voxel, we instead use the standard deviation in our evolution speed, described below. This will allow the surface to continue to propagate through consistent voxels, clearing up floating extraneous geometry and getting the evolving surface closer to the true geometry of the scene.

2.2. Surface Evolution

We embed an initial surface as the zero level set of a volumetric function $\psi(x, y, z, t = 0)$. The exact shape of the initial surface is not very important, as long as it contains the scene being reconstructed. This requirement is necessary since we shrink the surface during reconstruction.

The surface S evolves over time guided by the equation

$$\frac{\partial S}{\partial t} = \alpha_0 \phi \mathbf{N} + \alpha_1 \phi H \mathbf{N}, \tag{1}$$

where ϕ is a measure of color mismatch, $\mathbf{N}$ is the inwardly pointing surface normal, and H is the mean curvature. This flow is comprised of two terms: a weighted normal flow and a weighted curvature flow. The constants α_0 and α_1 can be used to weight the two flows relative to each other. In our application we set α_0 and α_1 both to 0.5. The weighted curvature keeps the reconstruction smooth, while the weighted normal flow allows the evolving surface to better fit the scene being reconstructed.

The weighting factor ϕ is data-dependent, and proportional to the standard deviation used in the photo-consistency measure described in Section 2.1. We scale ϕ so that it is within the range [0, 1]. The weighting factor is large when a voxel modeling the surface is inconsistent. This will allow the surface to readily propagate along its inwardly pointing normal through regions of space that do not contain object surfaces being reconstructed. The ϕ term becomes small when a voxel modeling the surface is consistent. Here, the surface propagation slows. At places where the surface is fully consistent, $\phi = 0$ and the surface evolution stops altogether.

This flow is implemented in the level set framework using the equation

$$\frac{\partial \psi}{\partial t} = (\alpha_0 + \alpha_1 H)\, \phi \, |\nabla \psi| \, , \tag{2}$$

with

$$H = \nabla \cdot \frac{\nabla \psi}{|\nabla \psi|}. \tag{3}$$

The temporal derivative is approximated using the first forward difference; all other derivatives are approximated using central differences. [1]

During evolution, we keep track of which voxels are on the zero level set. We call these *surface voxels*. We evolve the surface in the fashion described above until the number of surface voxels does not change for X iterations of Equation 2, where X is a user-specified constant. In our implementation, we set $X = 10$.

2.3. Multi-Resolution

For efficiency, our algorithm works in a coarse-to-fine fashion. We first perform a reconstruction at resolution R using large voxels. At a lower resolution, we are able to carve away a large part of space that would require a lot of computation at a higher resolution. Once the reconstruction at resolution R is complete, we dilate the surface. After the dilation, we tessellate each voxel into eight sub-voxels, which increases the resolution to $R + 1$. We then re-execute the algorithm at the higher resolution. This process continues until a desired resolution is obtained.

[1] In implementing Equation 2, we use a time step that satisfies CFL conditions to ensure stability.

Dilation of the surface prior to the resolution increase is necessary to prevent the evolving surface S from passing though fine details that cannot be properly modeled at a lower resolution. For example, consider a large voxel that contains a small patch of the true 3D surface T being reconstructed. If the voxel is projected into the reference views, the majority of the pixels in the voxel's projection will not represent T. Such a voxel would be inconsistent, and the zero level set could propagate through the voxel. At coarse resolutions this can cause some parts of T to go undetected. A simple solution, presented in [10], is to dilate S before the resolution increase. Doing so provides a mechanism for S to back up, and then re-evaluate the skipped part of T at a higher resolution where it can properly be reconstructed.

3. RESULTS

We executed our algorithm on a set of images of a broccoli stalk photographed from 17 different viewpoints using an HP PhotoSmart C200 digital camera. The broccoli stalk was placed on checkered paper so that the camera placements could be calibrated. We calibrated each image using Tsai's method [11]. The reprojection error in each image was approximately two pixels for a 576 x 436 pixel image. Three of the reference views are shown in Figure 1.

We performed a multi-resolution reconstruction on this data set. We started with a box-shaped initial surface in a coarse 21x17x23 volume. We executed the algorithm for three resolution increases, resulting in a 168x136x184 volume. Figure 2 shows the initial evolution of the zero level set from the initial surface. Once the geometry is reconstructed, we extract a polygonal representation of the surface by executing the marching cubes algorithm. Next, we texture-map the polygonal model using the photographs of the scene. For this, we consider the rays between the triangle center and each reference view that has visibility of the triangle. We then compute the angle θ between each ray and the triangle surface normal. We apply pixels onto the triangle from the reference view that has the smallest θ. New photo-realistic views of the reconstructed broccoli stalk are shown in Figure 3.

4. CONCLUSION

We have presented a multi-resolution space carving algorithm implemented with level set methods. Using a set of photographs taken with calibrated cameras, our approach generates a smooth, texture-mapped 3D polygonal model that can be rendered using standard graphics hardware to produce new views of the scene.

Our approach does not necessarily find the most photo-consistent surface. A possible extension to this work would be to design a flow that stops at maxima of the photo-consistency function. Also, we are interested in exploring ways to automatically adapt α_0 and α_1 during reconstruction.

We thank Anthony Yezzi for helpful discussions regarding level set methods, and HP for support of this research.

5. REFERENCES

[1] K. Kutulakos and S. Seitz, "A theory of shape by space carving," *Int. J. Computer Vision*, vol. 38, no. 3, pp. 199–218, 2000.

[2] W. B. Culbertson, T. Malzbender, and G. Slabaugh, "Generalized voxel coloring," in *ICCV Workshop, Vision Algorithms Theory and Practice*. 1999, pp. 100–115, Springer-Verlag LNCS 1883.

[3] P. Eisert, E. Steinbach, and B. Girod, "Multi-hypothesis, volumetric reconstruction of 3-d objects from multiple calibrated camera views," in *ICASSP*, 1999, vol. 6, pp. 3509–3512.

[4] S. Seitz and Charles R. Dyer, "Photorealistic scene reconstruction by voxel coloring," *Int. J. Computer Vision*, vol. 35, no. 2, pp. 151–173, 1999.

[5] G. Slabaugh, W. B. Culbertson, T. Malzbender, and R. Schafer, "Improved voxel coloring via volumetric optimization," Tech. Rep. 3, Center for Signal and Image Processing, Georgia Tech., 2000.

[6] J. Sethian, *Level Set Methods and Fast Marching Methods*, Cambridge University Press, second edition edition, 1999.

[7] O. Faugeras and R. Keriven, "Variational principles, surface evolution, pde's, level set methods and the stereo problem," *IEEE Transactions on Image Processing*, vol. 7, no. 3, pp. 336–344, 1998.

[8] A. Colosimo, A. Sarti, and S. Tubaro, "Image-based object modeling: A multiresolution level-set approach," in *ICIP*, 2001, pp. 181–184.

[9] A. Yezzi and S. Soatto, "Stereoscopic segementation," in *ICCV*, 2001, vol. 1, pp. 59–66.

[10] A. Prock and C. Dyer, "Towards real-time voxel coloring," in *Image Understanding Workshop*, 1998, pp. 315–321.

[11] R. Tsai, "A versatile camera calibration technique for high-accuracy 3d machine vision metrology using off-the-shelf tv cameras and lenses," *IEEE Transactions on Robotics and Automation*, vol. 3, no. 4, pp. 323–344, 1987.

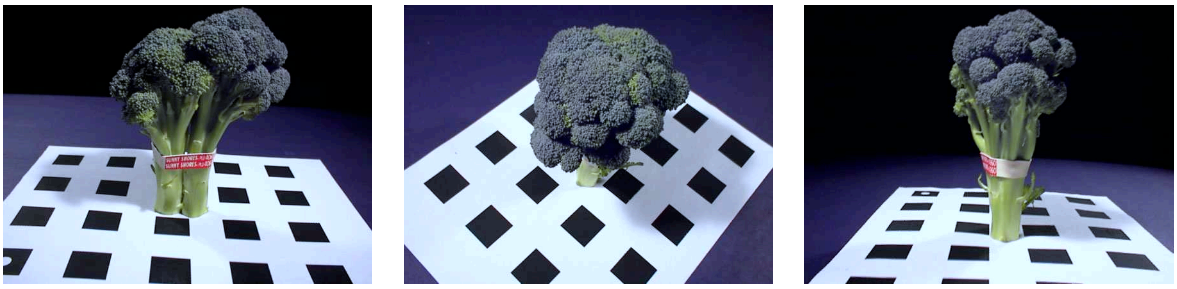


Fig. 1. Three of seventeen reference views of a broccoli stalk.

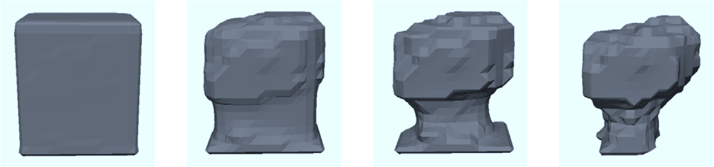
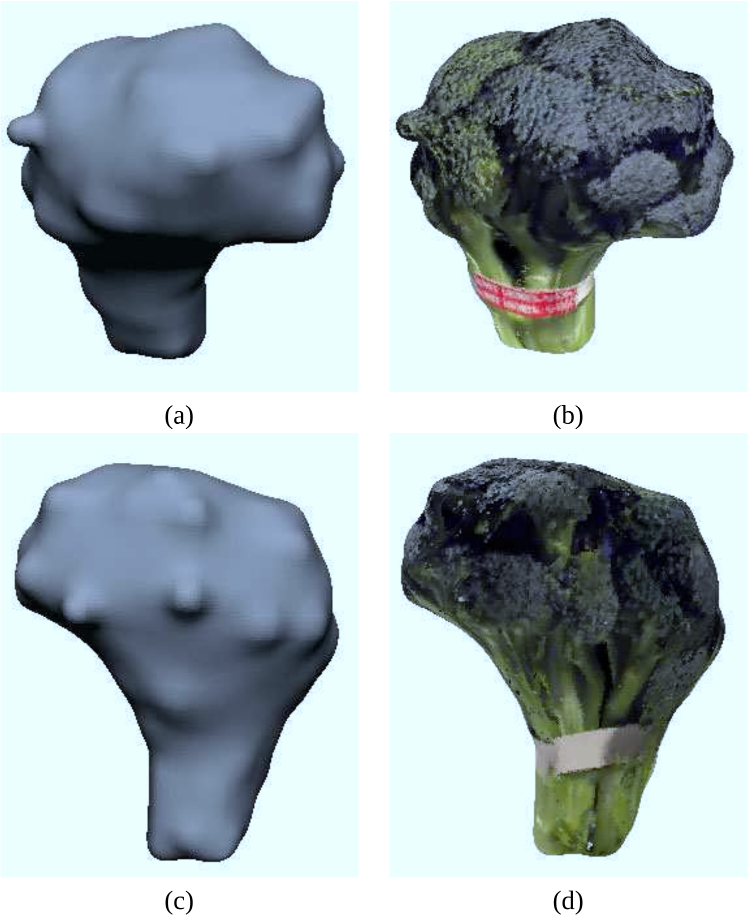


Fig. 2. Initial evolution of the zero level set at the lowest resolution.



(a)

(b)

(c)

(d)

Fig. 3. New synthesized views of the broccoli stalk. Untextured surfaces are shown in (a) and (c), textured surfaces are shown in (b) and (d).