3D Reconstruction of Single Picture

Zhao Ting[1, 2], David Dagan Feng[1, 3], and Tan Zheng[2]

[1]Center for Multimedia Signal Processing (CSMP)
Dept of Electronic & Information Engineering, Hong Kong Polytechnic University, Hong Kong
[2]Dept of Electronic & Information Engineering, Xi'an Jiao Tong University, China
[3]School of Information Technologies, University of Sydney, Australia

tzhao@it.usyd.edu.au

Abstract

This paper presents a novel approach for creating curvilinear, texture mapped, 3D scene models from a single painting or photograph with no prior internal knowledge about the shape. The new technique takes input as a sparse set of user-specified constraints, and generates a well-behaved 3D surface satisfying the parameters. As each constraint is specified, the system recalculates and displays the reconstruction in real time. In contrast to previous work in single view reconstruction, our technique enables high quality reconstructions of curved surfaces. A key feature of the approach is a novel hierarchical transformation technique for accelerating convergence on a non-uniform, piecewise continuous grid. The technique is interactive and updates the model in real time as constraints are added, allowing fast reconstruction of photorealistic scene models. The approach is shown to yield high quality results.

Keywords: Single picture, curvilinear modality, photorealistic, constraints, hierarchical

1. Introduction

The topic of 3D reconstruction from a single image is a long-standing issue in computer vision literatures. Traditional approaches for solving such problems often isolate a particular cue, such as shading [1], texture [2], or focus [3]. As these techniques make strong assumptions on shape, reflectance, or exposure, they tend to produce acceptable results for only a restricted class of images. More recent work by a number of researchers has shown that moderate user-interaction is highly effective in creating 3D models from a single view [4, 5, 6, 7]. A limitation of these approaches is that they are limited to scenes composed of planes or other simple primitives and do not permit modeling of curvilinear form scenes which are more complex. In order to solve this problem, a different approach is brought forward by using domain knowledge. For example, Blanz and Vetter [8] have obtained remarkable reconstructions of human faces from a single view using a database of head models. However, they don't have a general approach for curvilinear surface. In order to resolve this problem, many researchers fix their eyes on the human visual system. For example, Koenderink and his colleagues explored the depth perception abilities of the human visual system by having several hand-annotate images with relative distance or surface normal information [9].

Although it is also based on the principles put forth in the Koenderink's work, our modeling technique is much more efficient, works from sparse constraints, and incorporates discontinuities and other types of constraints in a general-purpose optimization framework. In this paper, our technique is proposed to treat the scene as an intensity-coded depth image and use traditional image editing techniques to sculpt the depth image [10, 11, 12]. We cast the single-view modeling problem as a constrained variational optimization problem. Building upon previous work in hierarchical surface modeling [13, 14, 15], the scene is modeled as a piecewise continuous surface represented on a quad-tree-based adaptive grid and is computed using a novel hierarchical transformation technique. The advantages of our approach are: A general constraint mechanism, Adaptive resolution and Real-time performance.

In order to better illustrate our technique, this paper is structured as follows. Section 2 formulates single-view modeling as a constrained optimization problem in a high dimensional space. In order to solve this large scale optimization problem efficiently with adaptive resolution, a novel hierarchical transformation technique is introduced in Section 3. Section 4 presents experimental results, Section 5 concludes.

2. A variational structure for modeling

The subset of a scene that is visible from a single image may be modeled as a piecewise continuous surface. In our approach, this surface is reconstructed from a set of user specified constraints, such as point positions, normals, contours, and regions. The problem of computing the best surface that satisfies these constraints is cast as a constrained optimization problem.

In this paper, the scene is represented as a piecewise continuous function, $f(x, y)$, referred to as the depth map. Samples of f are represented on a discrete grid, $g_{i,j} = f(id, jd)$, where the i and j samples correspond to pixel coordinates of the input image, and d is the distance between adjacent samples, assumed to be the same in x and y. Denote g as the vector whose components are $g_{i,j}$. A set of four adjacent sample $A = (i, j)$ $B = (i+1, j)$ $C = (i+1, j+1)$ and $D = (i, j+1)$ define the corners of a grid *cell*. Note that its vertices listed in counter-clock-wise order specify a cell.

The technique presented in this paper reconstructs the smoothest surface that satisfies a set of user-specified constraints. A natural measure of surface smoothness is

the thin plate functional [16]:

$$Q_0(g) = \frac{1}{2d^2}\sum_{i,j}[\alpha_{i,j}\left(g_{i+1,j} - 2g_{i,j} + g_{i-1,j}\right)^2 + 2\beta_{i,j} \quad (1)$$

$$\left(g_{i+1,j+1} - g_{i,j+1} - g_{i+1,j} + g_{i,j}\right)^2 + \gamma_{i,j}\left(g_{i,j+1} - 2g_{i,j} + g_{i,j-1}\right)^2$$

where $\alpha_{i,j}$, $\beta_{i,j}$ and $\gamma_{i,j}$ are weights that take on values of 0 or 1 and are used to define discontinuities.

Our technique supports five types of constraints: point constraints, depth discontinuities, creases, planar region constraints, and fairing curve constraints. Point constraints specify the position or the surface normal of any point on the surface. Surface discontinuity constraints identify tears in the surface, and crease constraints specify curves across which surface normals are not continuous. Planar region constraints determine surface patches that lie on the same plane. Fairing curve constraints allow users to control the smoothness of the surface along any curve in the image.

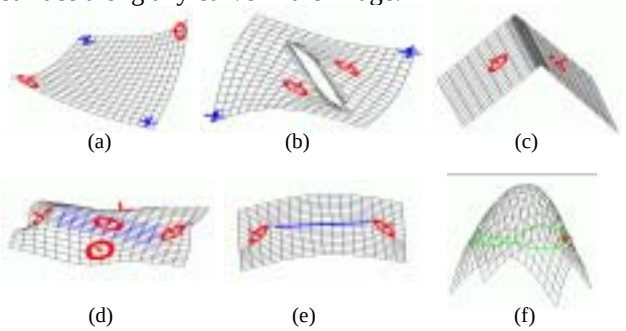

Figure 1: Modeling constraints (a) The effects of position (blue) and surface normal constraints (red) (b) A depth discontinuity constraint creates a tear (c) A crease constraint (green) (d) The blue region is a planar region constraint (e) A fairing curve minimizing curvature (f) A fairing curve minimizing torsion makes the surface bend smoothly given a single normal constraint–this type of constraint is useful for modeling silhouettes.

A point constraint sets the depth and/or the surface normal of any point in the input image [9]. A position constraint is specified by clicking at a point in the image to define the (sub-pixel) position (x_0, y_0), and then dragging up or down to specify the depth value. A surface normal is specified by rendering a figure representing the projection of a disk sitting on the surface with a short line pointing in the direction of the surface normal (Figure 1(a)).

A depth discontinuity is a curve across which surface depth is not continuous, creating a tear in the surface. A crease is a curve across which the surface normal is not continuous while the surface depth *is* continuous. Depth discontinuities and creases are introduced to model important features in real-world imagery. For example, mountain ridges can be modeled as creases and silhouettes of objects can be modeled as depth discontinuities. Examples of depth discontinuity and crease constraints are shown in Figures 1(b) and (c) respectively. And an example of a planar region constraint is shown in Figure 1(d).

It is often very useful for users to control the smoothness of the surface both along and across a specific curve. For example, surface depth is made to vary slowly *along* a curve in Figure 1(e), and the surface gradient is made to vary slowly *across* a curve in Figure 1(f). Fairing curves provide better control of the shape of the surface along salient contours such as silhouettes, and are achieved as follows.

Suppose that a user specifies a curve $\tau(l)=(x(l),y(l))^T$ in the image. To maximize the smoothness along the curve, we define a function $Q_d(\tau)$. In the same way, to make the surface gradient across $\tau(l)$ have small variation, the function $Q_S(\tau)$ is defined. The resulting equations are added, with weights ς_τ and η_τ, into Eq. (1), resulting in a modified surface smoothness objective function $Q(g)$:

$$Q_C(\tau) = \varsigma_\tau Q_d(\tau) + \eta_\tau Q_S(\tau)$$
$$Q(g) = Q_0(g) + \sum_\tau Q_C(\tau) \quad (2)$$

We call $\varsigma_\tau Q_d(\tau)$ the *curvature* term and $\eta_\tau Q_S(\tau)$ the *torsion* term. Note that $Q(g)$ is a quadratic form. Based on the surface objective function and constraints presented above, finding the smoothest surface that satisfies these constraints may be formulated as a linearly constrained quadratic optimization. Point constraints and planar region constraints introduce a set of linear equations, for the depth map g, expressed as $Ag = b$. Surface discontinuity and crease constraints define weights α, β and γ and fairing curve constraints introduce $\sum_\tau Q_C(\tau)$ in Eq. (2). $Q(g)$ is a quadratic form and can be expressed as $g^T Hg$, where H is the Hessian matrix. Consequently, our linearly constrained quadratic optimization is defined by

$$g^* = \arg\min_g \{Q(g) = g^T Hg\} \quad \text{Subject to} \quad Ag = b \quad (3)$$

Lagrange multiplier method is used to convert this problem into the augmented linear system

$$\begin{bmatrix} H & A^T \\ A & 0 \end{bmatrix}\begin{bmatrix} g \\ \lambda \end{bmatrix} = \begin{bmatrix} 0 \\ b \end{bmatrix} \quad (4)$$

The Hessian matrix H is a diagonally-banded sparse matrix. For a grid of size N by N, H is of size N^2 by N^2, with band width of $O(N)$ and about 13 non-zero elements per row. Direct methods, such as LU Decomposition, are of $O(N^4)$ complexity, and are therefore do not scale well for large grid sizes. Iterative methods are more applicable. We use the Minimum Residue method [17], designed for symmetric non-positive-definite systems. However, the linear system arising from Eq. (3) is often poorly conditioned, resulting in slow convergence of the iterative solver. To address this problem, a hierarchical basis preconditioning approach with adaptive resolution is presented in the next section.

3. Hierarchical transformation

The reason for the slow convergence of the MR method is that it takes much iteration to propagate a constraint to its neighborhood, due to the sparseness of H. Multigrid techniques [18] have been applied to this type of problem, however, they are tricky to implement and require a fairly smooth solution to be effective [14]. Szeliski [14] and Gortler et al. [13] use hierarchical basis functions to accelerate the solution of linear systems like Eq. (4).We review some approaches [13,14,18] next, to provide a foundation for our work which builds upon it.

In the hierarchical approach, a regular grid is represented with a pyramid of coefficients [19], where the number of coefficients is equal to the original number of grid points. The coarse level coefficients in the pyramid determine a low resolution surface sampling and fine level coefficients determine surface details, represented as displacements relative to the interpolation of the low resolution sampling. To convert from coefficients to depth values, the algorithm starts from the coarsest level, doubles the resolution by linearly interpolating the values of current level, adds in the displacement values defined by the coefficients in the next finer level, moves to the next finer level, and repeats the procedure until the finest resolution is obtained. the process can be written:

procedure CoefToDepth(coef)
  for $l = L - 1$ *down to 1*
    for every grid point P *in level* l
      $depthP = coefP + \sum_{Q \in N_P} wP_lQ * depthQ$
  return depth
end CoefToDepth

Where L is the number of levels in hierarchy, $coefP$ is the hierarchical coefficient for P, N_P is the set of grid points in Figure 2. The depth at the center point I is interpolated from the midpoints E, F, G, and H, which are in turn interpolated along each edge of the cell. level $l-1$ used in interpolation for P in level l, and wP_lQ is a weight that will be described later. Level 0 consists of a single cell, with coefficients defined to be the depth values at the corners of the cell.

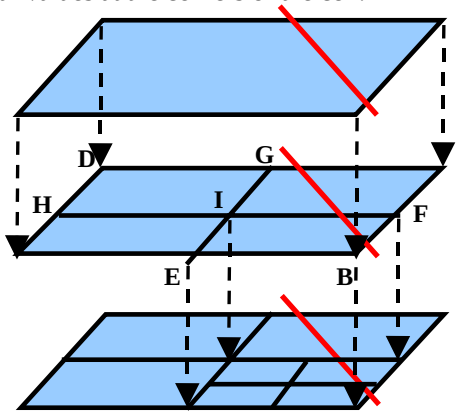

Figure 2: A cell is the primitive for 2D hierarchical transformation. The depth at the center point I is interpolated from the midpoints E, F, G, and H, which are in turn interpolated along each edge of the cell.

In previous works, the weights wP_lQ were defined to average all the points in N_P, resulting in a simple averaging operation for computing P from N_P. This approach implicitly assumes local smoothness within the region defined by N_P, resulting in poor convergence in the presence of discontinuities. In practice, this choice of weights causes the artifact that modifying the surface on one side of a discontinuity boundary disturbs the shape on the other side during the iterative convergence process. As a result, it takes longer to converge to a solution, and results in unnatural convergence behavior. The latter artifact is a problem in an incremental solver where the evolving surface is displayed for user consumption, as is done in our implementation. To address this problem, we next introduce a new interpolation rule to handle discontinuities between the grid points in N_P. The basic unit in the 2D hierarchical transformation technique is the cell shown in Figure 2, where the depth for corners A, B, C, and D has already been computed and the task is to transform coefficients at E, F, G, H, and I to depth values at these points. With the same notation as in the procedure $N_F = \{B, C\}$, $N_G = \{C, D\}$, $N_H = \{D, A\}$ and $N_I = \{E, F, G, H\}$. g_E, g_F, g_G, and g_H are first interpolated from A, B, C and D along edges, and then offset by their respective coefficients $\bar{g}_E$, $\bar{g}_F$, $\bar{g}_G$ and $\bar{g}_H$. Second, g_I is interpolated from g_E, g_F, g_G, and g_H and offset by its coefficient, $\bar{g}_I$. The two interpolation steps above use continuity-based interpolation with weights defined as

$$wP_lQ = \begin{cases} \dfrac{eP_lQ}{\sum_{Q \in N_P} eP_lQ} & \text{if } \sum_{Q \in N_P} eP_lQ > 0 \\ 0 & \text{otherwise} \end{cases}$$

$$\text{where } eP_lQ = \begin{cases} 1 & \text{if edge } P-Q \text{ is continuous} \\ 0 & \text{otherwise} \end{cases}$$

In the absence of discontinuities, the proposed continuity-based weighting scheme is the same as simple averaging schemes used in previous work [14, 15]. In the presence of discontinuities, only locally continuous coarse level grid points are used in the interpolation. The new scheme prevents interference across discontinuity boundaries and consequently accelerates the convergence of the Minimum Residue algorithm. To summarize our approach in brief, instead of solving Eq. (4) directly, we solve the hierarchical coefficients g of the grid point instead. The conversion from $\bar{g}$ to g is implemented by the procedure, CoefToDepth, with continuity-based weighting. The procedure implements a linear transformation and can be described by a matrix S [6]. Substituting $g = S\bar{g}$ into Eq. (3) and applying the Lagrange Multiplier method yields the transformed linear system [3]:

$$\begin{bmatrix} S^T HS & S^T A^T \\ AS & 0 \end{bmatrix} \begin{bmatrix} \bar{g} \\ \lambda \end{bmatrix} = \begin{bmatrix} 0 \\ b \end{bmatrix}$$

The matrix $S^T HS$ is shown to be better conditioned [6], resulting in faster convergence. The number of floating point operations of the procedure CoeToDepth and its adjoint [14] is approximately $4N^2$ for a grid size of $N \times N$. Considering that there are around 13 non-zero elements per row in H, the overhead introduced by S in multiplying $S^T HS$ with a vector is about 30%. Given the considerable reduction in number of iterations, the total run time is generally much lower using a hierarchical technique, even with this overhead.

4. Experimental results

We have implemented the approach described in this paper and applied it to create reconstructions of a wide variety of objects. Smooth objects without position discontinuities are especially easy to reconstruct using our approach. As a case in point, the image in the first row of Figure 3 requires only isolated normals and

creases to generate a compelling model, and can be created quite rapidly (about 20 minutes, including time to specify constraints) using our interactive system. The first row of Figure 3 shows the input image, quad-tree grid with constraints, a view of the quadtree from a novel viewpoint, and a texture mapped rendering of the same view. This model has 144 constraints in all, 3396 grid points, and required 25 seconds to converge completely on a 1.5GHz Pentium 4 processor, using our hierarchical transformation technique with continuity-based weighting. The system is designed so that new constraints may be added interactively at any time during the modeling process–the user does not have to wait until full convergence to specify more constraints. The third row of Figure 3 shows a single-view reconstruction of The Great Wall of China. This model has 135 constraints, 2566 grid points, and required 40 seconds to converge completely.

An interesting application of single view modeling techniques is to reconstruct 3D models from paintings. In contrast to other techniques [12, 17, 10, 11], our approach does not make strong assumptions about geometry, making it amenable to impressionist and other non-photorealistic works. Here we show a reconstruction created from a picture of David. This model has 204 constraints, 3481 grid points, and required 35 seconds to converge. This was the most complex model we tried, requiring roughly 1 hour to design. For comparison, it takes 70 seconds to converge without using the hierarchical transformation and 3 minutes using the hierarchical transformation without continuity-based weighting, i.e., an inappropriately weighted hierarchical method can perform significantly worse than not using a hierarchy at all. Note, however, that there is significant room for optimization in our implementation; we expect that the timings for both hierarchical methods could be improved by a factor of 1.5 or 2.

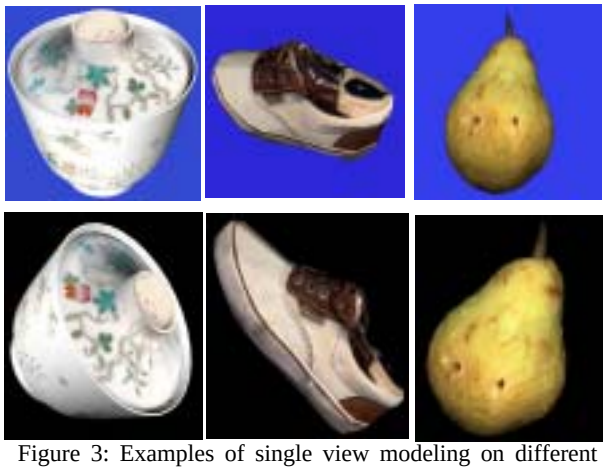

Figure 3: Examples of single view modeling on different scenes. The top one is the original image, the bottom is textured rendering in novel viewpoint.

5. Conclusion

In this paper, it was demonstrated that a reasonable amount of user interaction is sufficient to create high-quality 3D scene reconstructions from a single image, without placing strong assumptions on either the shape or reflectance properties of the scene. To justify this argument, an algorithm was presented that takes as input a sparse set of user-specified constraints, including surface positions, normals, silhouettes, and creases, and generates a well-behaved 3D surface satisfying the constraints. As each constraint is specified, the system recalculates and displays the reconstruction in real time. A technical contribution is a novel hierarchical transformation technique that explicitly models discontinuities and computes surfaces at interactive rates. The approach was shown to yield very good results on a variety of images.

6. Acknowledgement

This Research is supported by the HK- RGC, and ARC, SVT grant.

References

[1] B. K. P. Horn, “Height and gradient from shading”, Int’l J. of Computer Vision, vol. 5, no. 1, pp. 37-75, 1990.

[2] B. J. Super, A. C. Bovik, “Shape from texture using local spectral moments”, *IEEE Trans on PAMI*, vol. 17, no. 4, pp.333-343, 1995.

[3] S. K. Nayar, Y. Nakagawa, “Shape from focus”, *IEEE Trans. on PAMI*, vol.16, no.8, pp. 824-831, 1994.

[4] Y. Horry, K. Anjyo, K. Arai, “Tour into the picture: using a spidery mesh interface to make animation from a single image”, *ACM SIGGRAPH Proceedings*, pp. 225-232, 1997.

[5] A. Criminisi, I. Reid, and A. Zisserman, “Single view metrology”, *Int’l Conf. on Computer Vision*, pp.434-442, 1999.

[6] H.-Y. Shum, M. Han, and R. Szeliski. “Interactive construction of 3D models from panoramic mosaics”, *IEEE Conf. on CVPR*, pp. 427-433, 1998.

[7] P. Debevec, C. Taylor, and J. Malik, “Facade: modeling and rendering architecture from photographs”, *ACM SIGGRAPH Proceedings*, pp. 11-20, 1996.

[8] V. Blanz, T. Vetter, ”A morphable model for the synthesis of 3D faces”, *ACM SIGGRAPH Proceedings*, pp. 187–194, 1999.

[9] J. J. Koenderink, “Pictorial Relief”, *Phil. Trans. of the Roy. Soc.: Math., Phys, and Engineering Sciences*, 356(1740), pp. 1071-1086, 1998.

[10] B. M. Oh, M. Chen, J. Dorsey, and F. Durand, “Image-based modeling and photo editing”, *ACM SIGGRAPH Proceedings*, pp. 433-442, 2001.

[11] L. Williams, “3D paint”, *Proceedings of the Symposium on Interactive 3D Graphics Computer Graphics*, pp. 225-233, 1990.

[12] S.B. Kang, “Depth painting for image-based rendering applications”, Tech. Rep. CRL, Compaq Computer Corporation, Cambridge Research Lab., Dec. 1998.

[13] S. Gortler and M. Cohen, “Variational modeling with wavelets”, *TR-456-94*, Dept of Computer Science, Princeton Univ, 1994.

[14] R. Szeliski, “Fast surface interpolation using hierarchical basis functions”, *IEEE Trans. on PAMI*, vol. 12, no. 6, pp. 513-528, 1990.

[15] R. Szeliski, H.-Y. Shum, “Motion estimation with quadtree splines”, *IEEE Trans. on PAMI*, vol. 18, no. 12, pp. 1199-1209, 1996.

[16] D. Terzopoulos, “Regularization of inverse visual problems involving discontinuites”, *IEEE Trans. on PAMI*, vol. 8, no. 4, pp. 413-424, 1986.

[17] W. H. Press, B. P. Flannery, S. A. Teukolsky, and W. T. Vetterling, *Numerical recipes in C*, Cambridge University Press, pp. 83-89, 1988.

[18] D. Terzopoulos, “Image analysis using multigrid relaxation methods”, *IEEE Trans. on PAMI*, vol. 8, no. 2, pp. 129-139,1986.

[19] P. J. Burt, E. H. Adelson, “The Laplacian pyramid as a compact image code”, *IEEE Trans. on Comm.* vol. 31, no. 4, pp. 532-540, 1983.