Pose Estimation and Integration for Complete 3D Model Reconstruction

Soon-Yong Park and Murali Subbarao
Department of Electrical and Computer Engineering
State University of New York at Stony Brook
Stony Brook, NY11794-2350, USA
{parksy,murali}@ece.sunysb.edu

Abstract

An automatic 3D model reconstruction technique is presented to acquire complete 3D models of real objects. The technique is based on novel approaches to pose estimation and integration. Two different poses of an object are used because a single pose often hides some surfaces from a range sensor. The presence of hidden surfaces makes the 3D model reconstructed from any single pose a partial model. Two such partial 3D models are reconstructed for two different poses of the object using a multi-view 3D modeling technique. The two partial 3D models are then registered. Coarse registration is facilitated by a novel pose estimation technique between two models. The pose is estimated by matching a stable tangent plane (STP) of each pose model with the base tangent plane (BTP) which is invariant for a vision system. The partial models are then integrated to a complete 3D model based on voxel classification defined in multi-view integration. Texture mapping is done to obtain a photo-realistic reconstruction of the object.

1. Introduction

In recent years, there are considerable investigations on the topic of complete 3D model reconstruction from multiple views of an object. One of the approaches is merging multiple range images into a complete 3D model [4, 3]. Most investigations on 3D model reconstruction are limited to using a single pose of an object. However, for many real objects, using a single pose may yield only a partial 3D model because some hidden surfaces of the object from the sensor. A 3D model of such hidden surfaces could be reconstructed by placing the object in a different suitable pose. In order to obtain a complete 3D model, the two partial 3D models for the two different poses need to be registered and integrated. However, registration and integration of multiple partial 3D models is a very difficult problem. For this reason, only a few researchers have considered this problem.

P. Allen and R. Yang [1] stitch a bottom surface of an object by matching edge features of the object's 3D model with an edge image of the bottom. After reconstructing a 3D model, they acquire a partial shape of the bottom surface and stitch the shape and texture of the bottom to the 3D model using a feature matching technique. K. Wong and R. Cipolla [11] and W. Niem [7] employ shape-from-silhouettes techniques for 3D modeling. But they manually register the top and the bottom surfaces of the object. D. Huber [5] also presents a 3D reconstruction technique using an unconstrained registration of n-view partial shapes. He register the partial shapes using *Spin* images and a graph searching technique.

A schematic diagram of our 3D modeling system is shown in Figure 1. We register multiple range images into a common coordinate system based on the system calibration parameters. After refining the registration, we use the *Marching Cubes*(MC) algorithm to polygonize volumetric space into triangle meshes [2, 6]. We classify voxel space according to the signed distance of a voxel for accurate reconstruction of the implicit surface of the object [8]. Registration of two pose models consists of two steps, coarse registration and its refinement. We use a novel pose estimation technique of two 3D models to determine coarse registration parameters [9]. The pose estimation technique finds a stable tangent plane (STP) on a 3D model which can be transformed to the base tangent plane (BTP) of the other model and *vice versa*.

A novel pose integration technique for two 3D models is presented to reconstruct an accurate and complete 3D model. The novelty of our technique is integrating two iso-surfaces rather than multi-view range images. Because there may be occlusions either the first or the second pose, the integration technique merges two models by combining the signed distance and the classification of a voxel. Texture mapping for photo-realistic 3D model is also presented on real objects.

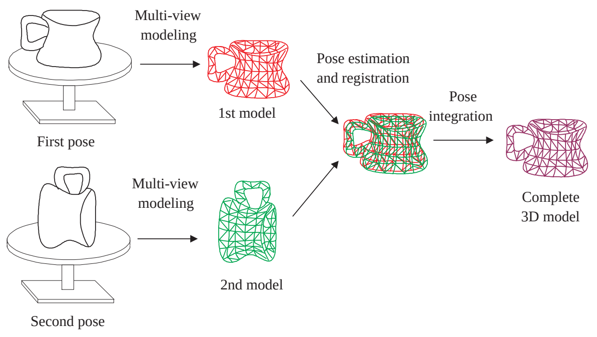


Figure 1. Schematic diagram of 3D modeling

2. Reconstruction of a single pose-model

2.1. Multi-view modeling

We use a high-resolution digital stereo camera to acquire range image of objects. In order to change viewing direction, we use a rotation stage to rotate the object. Object's silhouettes are also segmented by a *blue screen* technique. In order to refine calibration parameters of the system, a point-to-plane registration technique is employed to minimize transformation errors between all overlapping shapes [2].

Given multiple range images of an object, we estimate the iso-surface of the object and convert it to a mesh model using the Marching Cubes(MC) algorithm [6]. However, because our vision system uses a stereo camera for range image acquisition, there are inherent mismatching errors on range images. In order to accurately integrate partial shapes, we classify the voxel space into multiple regions based on signed distances $d(p)$ of a voxel p as follows:

- If at least two distances $|d_i(p)|$ are shorter than a threshold d_{TH}, where $i = 0, \ldots, N_0 - 1$, a voxel p is in an overlapping region, and $p \in P_{overlap}$.
- If for all $i, d_i(p) > 0$ and $|d_i(p)| > d_{TH}$, then the voxel is outside the object, and $p \in P_{outside}$.
- If for all $i, d_i(p) < 0$ and $|d_i(p)| > d_{TH}$, then the voxel is inside the object, and $p \in P_{inside}$.
- Otherwise, the voxel is in non-overlapping area, $p \in P_{nonoverlap}$,

where N_0 is the number of overlapping shapes, and it is $N/2$ in this paper.

The implicit distance $D(p)$ of the voxel p is a function of signed distances $d_i(p)$ to all overlapping shapes, which is

$$D(p) = f(d_0(p), \cdots, d_{N_o}(p)). \quad (1)$$

For more information about computing the signed distance $D(p)$, see the reference [8]. In order to remove erroneous data points outside the visual hull, we also combine *shape-from-silhouettes* technique.

3. Pose estimation and registration

3.1. Tangent plane matching

In order to integrate two 3D models, it is necessary to register two models into a common coordinate system. Therefore, we estimate the pose between two models for coarse registration. In this paper, we employ a novel pose estimation technique of two 3D models [9]. This technique finds a stable tangent plane (STP) on a 3D model which can be matched to the base tangent plane (BTP) of the other model. When we place a rigid object on the flat (planar) top of a turning table, the object rests with its outer surface touching the table top. The planar table top will be a tangent plane of the object's surface. We call the planar table top the BTP of the turntable. The BTP is invariant with respect to the object's pose and the world coordinate system.

Suppose an object is placed on the turntable with two different poses, $Pose1$ and $Pose2$ as shown in Figure 2. Then there is a unique tangent plane $T1$ ($\hat{n}_{T1}$) in the first pose which matches the BTP B ($\hat{n}_B$) in the second pose. Similarly, there is also a unique plane $T2$ ($\hat{n}_{T2}$) in $Pose2$, which matches B ($\hat{n}_B$) in $Pose1$. Because $\hat{n}_B$ is a common and invariant vector in the vision system, we can estimate a rotation matrix using $\hat{n}_{T1}$ and $\hat{n}_{T2}$. Translation between two models is estimated by Center of Mass (CoM) of the models.

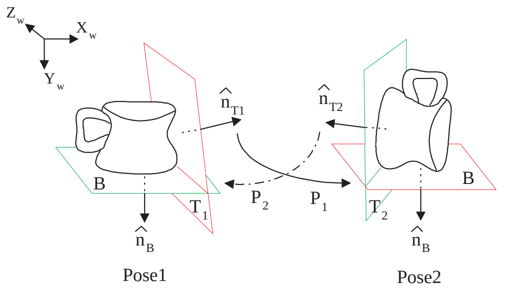


Figure 2. A tangent plane of the first pose $\hat{n}_{T1}$ uniquely matches with the BTP of the second pose $\hat{n}_B$, and *vise versa*.

3.2. Finding stable tangent plane (STP)

We find all tangent planes using Extended Gaussian Image (EGI) of the 3D model. For each face of the tessellated sphere of EGI, we determine whether the tangent plane corresponding to the face is a candidate for matching with the

BTP of the other pose. After determining an initial tangent plane $T = Ax + By + Cz + D$, we refine the plane using its supporting vertices which are close to the plane. Each refined tangent plane T consists of supporting vertices and has its own coordinate system. See [9] for more details.

There will be a finite, but large number of tangent planes for each 3D mesh model. Let a set of tangent planes of $Pose1$ be $\mathcal{T}_{\mathbf{1}}$, and $\mathcal{T}_{\mathbf{2}}$ be another tangent plane set of $Pose2$. In order to reduce computation time for finding matching tangent planes from all possible combinations of two sets, we remove some local or unstable tangent planes by employing three geometric constraints.

The first constraint is *Base plane constraint*. The BTP is a global tangent plane in the sense that it will not intersect the object's volume anywhere. Therefore we check all vertices to determine if a tangent plane intersects the volume. If the dot product of any vertex v with the plane normal $\hat{n}_T$ is greater than the parameter $D+\delta_I$ of the plane, we remove the plane.

Next constraint is *Stability constraint*. The BTP of the turntable is horizontal and the object is in a stable pose. Therefore, given the CoM of a 3D model, its projection to a STP, CoM_T is always inside the convex hull of the projections of all supporting vertices. The object will be unstable and fall over if CoM_T is outside the convex hull. The last constraint is *Height constraint*. If two pose models are correctly registered, their heights will be very similar. We reject tangent plane when the height difference is greater than a threshold δ_H. Figure 3 shows an example of finding STPs.

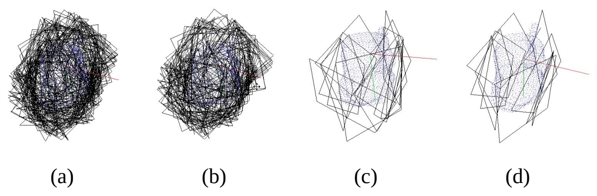


Figure 3. Example of finding STP (a)Initial TPs (186 planes) (b) After baseplane constraint (δ_I = 3mm, 141 planes) (c) After stability test (18 planes) (d) After height comparison (δ_H = 3mm, 9 planes)

3.3. Matching tangent planes

Rejection of unstable and local tangent planes significantly reduces the number of tangent planes. The last step in pose estimation is finding two tangent planes, one from each pose, which registers two 3D models with a minimum pose error. For every STP in $\mathcal{T}_{\mathbf{1}}$, we derive a transformation matrix Q_{ij} using every STP in $\mathcal{T}_{\mathbf{2}}$, measure the pose error, and find two STPs which yield the best-matching. Let a STP in $\mathcal{T}_{\mathbf{1}}$ be T_1 and another STP in $\mathcal{T}_{\mathbf{2}}$ be T_2. The transfomration matrix of the second pose (vertex set $\mathcal{V}_2$) to the first pose (vertex set $\mathcal{V}_1$) is estimated by using T_2 and T_1. The transformation matrix first aligns T_2 with the BTP B and T_1 with B' by rotating the model along the Y axis

$$B' = {T_2}^{-1} B, \tag{2}$$
$$T_1 = {R_y}^{-1} B'. \tag{3}$$

The coordinate system of B is the same as the world (or common) coordinate system. A rotation matrix R_y aligns B' with the coordinate system T_1. It is a rotation along the Y axis and computed by

$$R_y = \begin{pmatrix} cos\theta & 0 & -sin\theta & 0 \\ 0 & 1 & 0 & 0 \\ sin\theta & 0 & cos\theta & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$
$$where, \begin{pmatrix} cos\theta \\ sin\theta \end{pmatrix} =$$
$$\begin{pmatrix} B'_x + B'_z & B'_z - B'_x \\ B'_x - B'_z & B'_x + B'_z \end{pmatrix}^{-1} \begin{pmatrix} T_{1x} + T_{1z} \\ T_{1x} - T_{1z} \end{pmatrix}. \tag{4}$$

Let the translation from the origin of the common coordinate system to each CoM be M_1 and M_2. Then the pose transformation matrix Q_{21} from $\mathcal{V}_2$ to $\mathcal{V}_1$ is computed by

$$\mathcal{V}'_2 = M_1 {R_y}^{-1} {T_2}^{-1} {M_2}^{-1} \mathcal{V}_2 = Q_{21} \mathcal{V}_2. \tag{5}$$

The best matching pose transformation Q_{21} is estimated by minimizing a cost function between two models,

$$\min_{\{T_1 \in \mathcal{T}_{\mathbf{1}}, T_2 \in \mathcal{T}_{\mathbf{2}}\}} \{\sum \|\mathcal{V}_1 - Q_{21}\mathcal{V}_2\|^2\}. \tag{6}$$

3.4. Pose registration

Based on the estimated pose Q_{21}, we register and refine the second pose (range image set) to the first pose. Refinement algorithm is similar with that of multi-view registration [2]. However, because partial surfaces within a pose are already registered in multi-view registration, we refine the pose between two range image sets.

4. Pose integration

After the pose registration, two models are integrated into a complete 3D model. Pose integration computes final signed distance of the model $D_f(p)$, which is a function of signed distances to each pose,

$$D_f(p) = f(D_1(p), D_2(p)), \tag{7}$$
$$where, D_i(p) = f(d_0^i(p), \cdots, d_{{N_o}^i}^i(p)) \;\; for \;\; i = 1, 2,$$

and $D_i(p)$ is a weighted signed distance of a voxel p in pose i, $d^i_j(p)$ is the signed distance in pose i and view j, and ${N_o}^i$ is the number of overlapping shapes in pose i. Computing $D_f(p)$ of the voxel p is based on average of signed distance $D_i(p)$. If the voxel p is seen from any view in both poses, $D_f(p)$ can be computed by weighted average. However, if there is any occlusion or concavity in either pose, $D_f(p)$ should be estimated according to the class of the voxel in both poses.

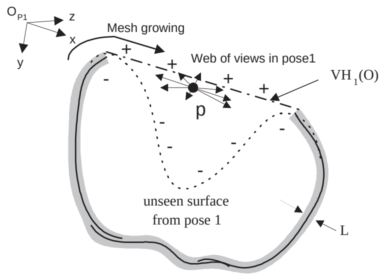
Suppose there is a concavity on the object's surface as shown in Figure 4. As shown in Figure 4(a), O_{p1} is the common coordinate system of the first pose and all view points are nearly on the XZ plane of the coordinate system. See that no view in the first pose can observe the concavity. But in Figure 4(b), some of the views in the second pose can see the concavity. If a voxel p is inside of unseen surface region as shown in Figure 4(a), voxel is classified to $p \in P_{inside}$ from the first pose. And a continuation approach of the MC algorithm closes a mesh model by following the visual hull $VH_1(O)$ of the multi-view frustum as shown in the figure. It means there is a voxel p which is close to the visual hull and classified to P_{inside} in the first pose. However, because it is seen from the second pose, there is no possibility that the MC algorithm marches on the same voxel p. Instead, the algorithm must follow the object's surface, such as a voxel $p \in P_{overlap}$ as in Figure 4(b).

We average two weighted signed distances $D_i(p)$, when $p \in P^i_{overlap}$ for j =1 and 2. Otherwise, we heuristically select one of the distances or the shorter one as follows.

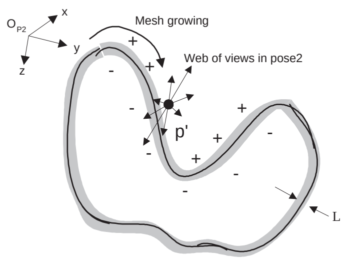
- $D_f(p) = \frac{\sum W_i(p)D_i(p)}{\sum W_i(p)}$, if $p \in P^1_{overlap}$ and $p \in P^2_{overlap}$.
- $D_f(p) = min\{D_1(p), D_2(p)\}$, if $p \in P^1_{inside}$ and $p \in P^2_{inside}$.
- $D_f(p) = D_1(p)$, if $p \in P^2_{inside}$, or $D_f(p) = D_2(p)$, if $p \in P^1_{inside}$.

However, in some situations, there is a voxel which is in $P^i_{nonoverlapp}$ at both poses. A voxel in a concave region is sometimes in this class. Consider an example shown in Figure 5. In the figure, the signed distance from voxel p to two pose models are $D_1(p)$ and $D_2(p)$ and $p \in P^1_{nonoverlap}$ and $p \in P^2_{nonoverlap}$. As shown in the figure, $D_1(p)$ has minus sign, even the voxel is outside the object, because it is not visible from $pose1$. If $|D_1(p)| < |D_2(p)|$, the MC algorithm may choose the shorter one, then the voxel is considered to be inside the object. Typically, this kind of errors happen near the visual hull.

Rather than selecting the shorter distance of two poses, we compare visibility of a voxel from views in each pose, if the voxel is in $P_{nonoverlap}$ in both poses. If the voxel has many positive signs from the view points, that means it has high visibility (Our implicit representation has (+)



(a) First pose with an unseen concavity



(b) Second pose

Figure 4. Signed distance in concave region: (a) MC algorithm follows the visual hull $VH_1(O)$ to close mesh model. (b) MC algorithm follows the actual surface of the object.

sign when a voxel is outside an object). The visibility of the voxel can be considered to be the number of positive distances to each pose. It can be easily decided by counting the number of positive distances of $d^i_j(p)$ in each pose. Therefore, we define more conditions for computing $D_f(p)$ in $P_{nonoverlap}$. If a voxel $p \in P^1_{nonoverlap}$ and $p \in P^2_{nonoverlap}$,

- $D_f(p) = D_1(p)$ if $count^+(d^1_j(p)) > count^+(d^2_j(p))$.
- $D_f(p) = D_2(p)$ if $count^+(d^2_j(p)) > count^+(d^1_j(p))$.

where, $count^+()$ is a function counting the number of positive distances in $d^i_j(p)$, where $j = 0, \cdots, {N_o}^i$.

5. Experimental Results

Our vision system is implemented under a Pentium III 1GHz computer. We have tested our modeling technique on two complex objects. Each object is placed on the rotation stage and 8 stereo image pairs are taken for each pose. A stereo image pair has two 1280×960 color images. And range image size is 320×240. In oder to introduce contrast

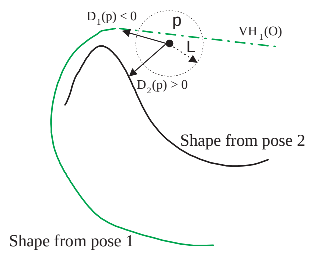
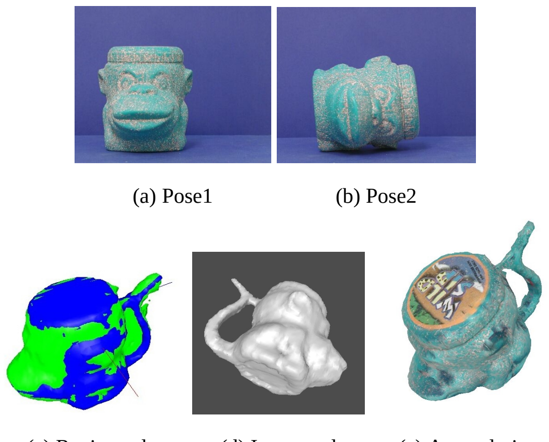


Figure 5. An error on mesh growing

on the object's surface, we project a random dot pattern using a slide projector.

Figure 6(a) and 6(b) show two pose images of an object "Monkey". Pose registration and integration results are shown in 6(c) and (d). Figure 6(e) shows a texture-mapped novel view of the object. The second object is a small toy, "Pokemon". Texture mapping result in Figure 7(e) shows a photo-realistic reconstruction of the object.



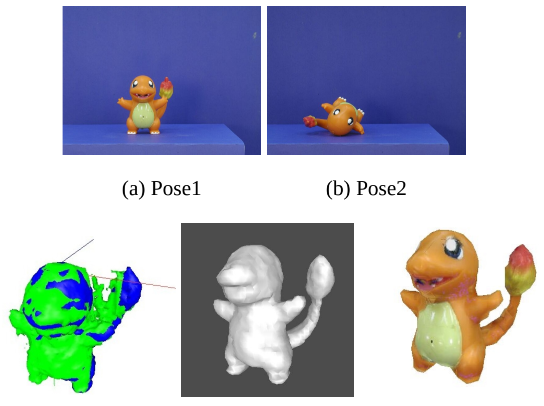
(a) Pose1 (b) Pose2

(c) Registered (d) Integrated (e) A novel view

Figure 6. Results of "Monkey" object

6. Conclusions

Pose estimation and integration techniques are presented for an automatic and complete 3D model reconstruction. In order to reconstruct all visible surfaces of the object, we introduce a novel technique of merging two 3D pose-models. Two models are constructed from two different poses of the object. Pose between two 3D models is estimated by matching global and stable tangent planes (STP) with the base tangent plane (BTP). The two 3D models are then pose-refined and integrated to a complete 3D model. Texture mapped 3D models of real objects are presented.



(a) Pose1 (b) Pose2

(c) Registered (d) Integrated (e) A novel view

Figure 7. Reconstruction of "Pokemon"

References

[1] P. Allen and R. Yang, "Registering, integrating, and building CAD models from range data," *IEEE Int. Conf. on Robotics and Automation*, 3115-3120, May 1998.

[2] R. Bergevin, M. Soucy, H. Gagnon, and D. Laurendeau, "Toward a general multi-view registration technique," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 18(5), May 1996.

[3] B. Curless and M. Levoy, "A volumetric method for building complex models from range images," In *Proceedings of SIGGRAPH*, 303-312, 1996.

[4] A. Hilton, A.J. Stoddart, J. Illingworth, and T. Windeatt, "Reliable surface reconstruction from multiple range images," *Proceedings. of the ECCV'96*, 117-126, Springer-Verlag, 1996.

[5] D. F. Huber, "Automatic 3D Modeling Using Range Images Obtained from Unknown Viewpoints," *Proceedings of the Third Internaional Conference on 3-D Digital Imaging and Modeling*, IEEE Computer Society, May, 2001, pp.153-160.

[6] W.E. Lorensen and H.E. Cline, "Marching cubes: A high resolution 3D surface construction algorithm," *Computer Graphics*, 21(4), July 1987.

[7] W. Niem, "Automatic reconstruction of 3D objects using a mobile camera," *Image and Vision Computing*, 17:125-134, 1999.

[8] S. Y. Park and M. Subbarao, "Automatic 3D model reconstruction using voxel coding and pose integration" *IEEE International Conference on Image Processing*, Sep. 2002.

[9] S. Y. Park and M. Subbarao, "Pose estimation of two-pose 3D models using the base tangent plane and stability constraints," *Vision, Modeling, and Visualization* Nov. 2002.

[10] R.Y. Tsai, "A versatile camera calibration technique for high-accuracy 3D machine vision metrology using off-the-shelf TV camera and lenses," *IEEE Journal of Robotics and Automation*, 3(4), August 1987.

[11] K. Wong and R. Cipolla, "Structure and motion from silhouettes," In *Proc. 8th IEEE International Conference on Computer Vision (ICCV01)*, 2:217-222, Vancouver, Canada, July 2001.