AUTOMATIC 3D MODEL RECONSTRUCTION USING VOXEL CODING AND POSE INTEGRATION

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

Automatic reconstruction of a complete 3D model of a complex object is presented. The complete 3D model is reconstructed by integrating two 3D models which are reconstructed from different poses of the object. For each pose of the object, a 3D model is reconstructed by combining stereo image analysis, shape from silhouettes, and a volumetric integration technique. Stereo image analysis and shape from silhouettes techniques complement each other to reconstruct an accurate and noise-resistant 3D model. For a reliable volumetric integration of multiple partial shapes, a voxel coding technique is introduced. Voxel coding technique facilitates a selection of consistent partial shapes for shape integration. In order to reconstruct all visible surfaces of a complex object with concavities and holes, two 3D models from different poses of the object are reconstructed and integrated to obtain the complete 3D model. A voxel coding technique is again used during pose integration. Experimental results on a real object demonstrate that our approach has advantages and is effective.

1. INTRODUCTION

Complete 3D model reconstruction of a complex object is a difficult problem in computer vision [1, 2, 3, 10]. A complex object has concavities and holes which are visible from some directions of view or for some pose of the object but not others. Therefore, automatic and reliable reconstruction of complete 3D model requires combining many complementary techniques at different stages. In this paper we present an approach that combines several such techniques at two stages, first during 3D model reconstruction for a given pose, and then during pose integration.

Our approach uses a low-cost digital camera mounted on a translation stage for stereo image acquisition and a rotation stage on which the object of interest is placed as in [4]. The rotation stage is turned to change the direction of view of the object for a given pose. First, partial 3D shapes of the object are computed for different horizontal directions of view. Then these shapes are registered and then integrated to obtain a full 3D shape (360 degree view) for the horizontal or upright pose. In this step, we use a volumetric integration technique and the marching cubes algorithm to find the isosurface of the object [6]. This 3D shape obtained after this first stage will be incomplete or erroneous for complex objects that have concavities or holes which are visible only from the vertical (e.g. top and bottom) views. Therefore the pose of the object is changed by manually turning and placing the object on its side. In this new side (vertical) pose, the original top and bottom of the object become visible when the rotation stage is turned. Another full 3D shape is reconstructed for this side/vertical pose using steps similar to that for the upright/horizontal pose. The two full 3D shapes for the horizontal and vertical poses are then registered and integrated to obtain a complete (4π steradian view) 3D shape of the object. Texture map is then extracted and added to obtain a complete photo-realistic 3D model of the object.

Hilton et al. [5] reconstruct a volumetric implicit surface representation from multiple range images. Curless and Levoy [3] also use a volumetric integration technique to generate an implicit surface representation. All of them assume that the partial shapes are noise-free. Recently, Wheeler et al. [10] investigate a volumetric integration technique using a consensus-surface algorithm for multiple partial shapes which have erroneous points.

Our vision system uses a digital still camera and a stereo vision technique for initial partial 3D shape recovery. In order to remove and reduce the effects of erroneous points on the recovered implicit surface, we combine complementary techniques– *shape from silhouettes* technique, stereo vision, and *voxel coding* technique [8]. *Shape from silhouettes* technique can be easily used to remove erroneous points outside of the *visual hull*. *Voxel coding* classifies voxel space into multiple binary coded regions based on signed distances from all views. Reasonable number of shortest signed distances are picked according to the voxel code and a weighted average is computed. Voxel coding also facilitates integration of multi-pose 3D models to reconstruct a more accurate and complete 3D model.

2. PARTIAL SHAPE RECONSTRUCTION AND REGISTRATION

2.1. Partial shape reconstruction

In order to acquire partial shapes of an object, we use a digital still camera, Olympus C-3030 Zoom, and a translation stage to move the camera in the horizontal direction to implement a parallel stereo geometry. We also use a rotation stage to change the viewing direction of the object placed on it. One pair of stereo image is taken at every viewing direction. A stereo matching technique based on multi-resolution stereo pyramid is then applied to the image pairs to acquire range images from different viewing directions. A correlation based stereo matching technique is used to find the stereo correspondences on the horizontal epipolar lines.

Supported in part by Olympus Corporation.

Object's silhouettes are acquired by a *blue screen* segmentation technique, and they are used for finding stereo correspondence only in the object region and for volume intersections. Calibration of our vision system is done by Tsai's calibration technique [9]. To introduce contrast on the object's surface, we project a random dot pattern on the object using a slide projector. Figure 1 shows a picture of our vision system named as SVIS-2.

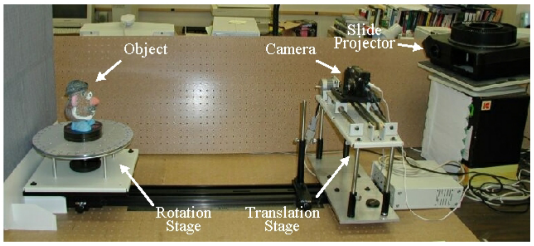


Fig. 1. SVIS-2 vision system

2.2. Multi-view registration

Multiple partial shapes are registered to a common coordinate system using initial calibration parameters. Initial registration is refined again by using a 3D shape registration technique, which is one of the *point-to-tangent plane* approaches [1, 2]. Registration step reduces mean distance between control points of two overlapping shapes. We minimize the mean distance error to make the result of implicit surface function unbiased.

In order to register multiple partial shapes simultaneously, we pick one camera coordinate system as a reference frame of a network of multiple views. And we register all partial shapes together using a multi-view registration technique proposed by Bergevin et al.[1]. Since we use a rotation stage to rotate the object, we adopt a circular network of views. Multi-view registration technique minimizes registration errors in all overlapping shapes simultaneously.

3. MULTI-SHAPE INTEGRATION

3.1. Surface representation

Implicit surface function consists of signed distances from a grid of voxels to the object's surface. When there is a voxel p in 3D space, the signed distance $f(p)$ from the voxel to the object's surface is computed by taking the dot product of two vectors,

$$f(p) = (d(p) - p) \cdot \hat{n}, \quad (1)$$

where $d(p)$ is a vector from the origin of the camera coordinate system to a point on the object's surface, which is on the same projection line of the vector p, and $\hat{n}$ is a unit normal vector of the object surface at $d(p)$.

For integration of partial shapes, the signed distance from a voxel to the object's surface is estimated by averaging weighted signed distances to all overlapping shapes. If a vision system has N multiple views, N signed distances from a voxel to all partial shapes are measured and the weighted signed distance is estimated by averaging them. For a voxel point p, the weight of depth measure for ith partial shape is defined as $w_i(p) = \hat{p} \cdot \hat{n}_i$, where $\hat{p}$ is the normal vector of p. And the signed distance of the voxel is

$$F(p) = \frac{\sum w_i(p) f_i(p)}{\sum w_i(p)} \quad (2)$$

,where $f_i(p)$ is the signed distance from the voxel p to the ith partial shape.

In order to compute the signed distance using only overlapping shapes from the voxel, it is necessary to take some short distances by thresholding the magnitude of the signed distances. However, if there are some errors on the partial shapes, computation of the signed distance may not be an accurate distance to the 3D model. We use *shape from silhouettes* technique to remove erroneous voxels outside of the object's visual hull.

3.2. Volume intersection

Volume intersection technique using silhouettes of an object is a very useful technique for removing erroneous voxels, because silhouettes are often dominant image features and reliable. If every view point is outside the convex hull of an object **O**, intersections of all cones of views generate the object's visual hull **VH(O)**. In order to reconstruct **VH(O)** as close as possible to the object **O**, usually a lot of silhouettes are neded. However, since we use a limited number of silhouettes, the reconstructed visual hull contains not only the object **O** but also some volume $\bar{\mathbf{O}}$ which is outside the object (but inside the visual hull):

$$\bar{\mathbf{O}}\big|_{\mathbf{VH}} = \{\mathbf{p}|\mathbf{p} \in \mathbf{VH}(\mathbf{O}) \cap \bar{\mathbf{O}}\} \quad (3)$$

where, $\bar{\mathbf{O}}$ is outside of the object **O**. In our integration step, volume intersection is mainly used for removing erroneous surface due to stereo mismatchings, which lie outside of **VH(O)**.

Some erroneous points inside of **VH(O)** can not be removed by volume intersection and they may introduce a distortion in the final 3D model. Therefore, we use a voxel coding technique to reliably select overlapping shapes and to accurately estimate the signed distance from a voxel.

3.3. Voxel coding

Integration of partial shapes begins with finding a seed voxel as a starting point of the marching cubes algorithm [6]. From the seed voxel, the algorithm computes the weighted signed distance of every voxel on such cubes that intersect the object's surface. Because *shape from silhouettes* technique considers voxels outside of **VH(O)** to have the positive sign, it only needs to compute the signed distance of voxels inside of **VH(O)**.

It is necessary to select only reliable partial shapes to compute the signed distance of a voxel in an overlapping region **L**. For this purpose we introduce a *voxel coding* technique to select partial shapes which are consistent and reliable with the system geometry. Let's define the maximum number of overlapping shapes and the corresponding views. Because we use a rotation stage which rotates an object by $360/N$ angle where N is the number of different views, we set the number of maximum overlapping shapes to $N_o = N/2$.

Each voxel which is inside of **VH(O)** is coded by a N bit binary code. We assign one bit to each view. If the signed distance of a voxel to the i-th surface or shape S_i is negative, we assign 0 to the corresponding bit, otherwise we assign 1. Therefore, 0 of ith bit means the voxel is behind S_i and 1 means it is between S_i and the ith view point V_i.

Let us consider an 8 multi-view system, for example. Then there are 8 partial shapes from $\mathbf{S_0}$ to $\mathbf{S_7}$, and a voxel is coded with an 8 bit binary code. If a voxel is inside of partial shapes from all viewing directions, then the code becomes 0000000 in binary. If the voxel has positive sign only for the shape $\mathbf{S_0}$, then it is coded as 00000001. Figure 2 shows a graphical diagram of voxel coding for 8 view points. As shown in the figure, **VH(O)** is divided into 8 regions from $\mathbf{R_0}$ to $\mathbf{R_7}$ according to N_0 overlapping shapes.

If a voxel p is in region $\mathbf{R_0}$, for example, its voxel code

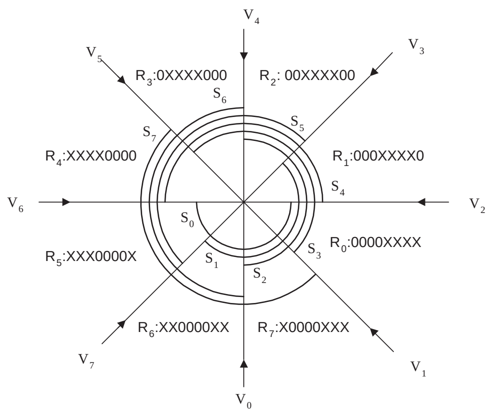


Fig. 2. A graphical diagram of voxel coding

VC(p) always has a form of 0000XXXX as a binary representation, where X is a *don't care* bit. If the voxel has negative sign from all overlapping shapes, its voxel code is 00000000 and it is inside of the object **O**. If it has positive sign from all overlapping shapes, the code is 00001111. The position of this voxel is outside of **O**, but still inside of **VH(O)**, in other word $p \in \bar{O}\big|_{VH}$. Else, there are 14 other combinations of the code.

Suppose there are N view points and **VC** is coded as a N bit binary code $\{b_{N-1}b_{N-2}\cdots b_2b_1b_0\}$. Then, based on the vision system geometry, a *consistent* voxel code must satisfy the following conditions:

1. $b_i \ \& \ b_{i+N_0} \neq 1, \ \ for \ \ i = 0 \cdots N_0 - 1.$
2. $if \ \ b_i \otimes b_{i+N_0} = 1,$
   $then \ \ |f_i(p)| \ < \ |f_{i+N_0}(p)| \ \ if \ \ b_i = 1.$
   $|f_i(p)| \ > \ |f_{i+N_0}(p)| \ \ if \ \ b_{i+N_0} = 1.$

where, $\&$ is a bit AND operator, $\otimes$ is a bit exclusive OR operator, and $f_i(p)$ is the signed distance of a voxel p.

Selecting reliable signed distance from a voxel is based on the **VC(p)**. Let's define reliable signed distance $d_i(p)$, where $i = 0 \cdots N_0 - 1$. Since two distances $f_i(p)$ and $f_{i+N_0}(p)$ are measured always from two opposite viewing directions, selecting one of them as a signed distance is very likely reliable. Therefore, distance $d_i(p)$ is selected from one of them according to the status of the two bits b_i and b_{i+N_0}. Here is an algorithm for computing the weighted signed distance $D(p)$ of a voxel p.

```
for i = 0 to N_0 - 1
    if (b_i && b_{i+N_0}) d_i ← inconsistent;
    if (b_i || b_{i+N_0})
        if (b_i) d_i ← f_i;
        else d_i ← f_{i+N_0};
    else
        if (|f_i| < |f_{i+N_0}|) d_i ← f_i;
        else d_i ← f_{i+N_0};
for i = 0 to N_0 - 1
    if ((d_i is consistent) && (d_i < TH))
        W(p) += wd_i(p);
        D(p) += wd_i(p)d_i(p);
    D(p) = D(p) / W(p)
```

where, $wd_i(p)$ is the weight value corresponding to the distance $d_i(p)$ and TH is the threshold of the magnitude of signed distance.

4. POSE INTEGRATION

We obtain a full 360 degree view 3D model as described above for two poses, first an upright (horizontal) pose and then a side (vertical) pose, as explained earlier. These two 3D models are then registered to a common coordinate system and then integrated to obtain a final complete 3D model. Registration of the two 3D models is done using a technique similar to the multi-shape registration. However, because we place the object on the rotation stage in an arbitrary pose, it is generally very difficult to find an initial estimate between two poses.

For registring the two 3D pose models, we adopt a *pose estimation* technique to estimate an initial registration parameters[7]. This technique uses properties of the Gaussian image of the model, mainly moments of the image. It computes a central inertia tensor and the principal axes which define an orthogonal coordinate system. The pricipal axes are identified by the eigenvectors of the inertia tensor. We match the eigenvectors corresponding to each principal axis for the two models to estimate the initial registration. The pose estimation technique gives an approximate registration parameters for the two models. These initial estimates are refined using the same algorithm used in the multi-shape registration step.

After pose registration, two models are integrated into a complete 3D model by incorporating the voxel coding technique. Suppose there is a concavity on the object, and the first pose is not able to observe the concavity, but the second pose is. If a voxel p is inside of unseen surface region as shown in Figure 3, the voxel is coded as 0x00 in hexadecimal from the fisrt pose. But, it is coded as one of the others from the second pose. However, even if the concavity is seen from the second pose, not every view point in the second pose can see the concavity. Therefore, the voxel code is actually one of possible code except a **VC(p)** such that $p \in \bar{O}\big|_{VH}$.

An algorithm integrating two 3D models into a complete 3D

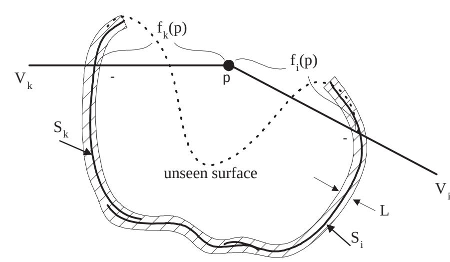


Fig. 3. Voxel coding in concave region

model based on voxel codes from two poses is summarized as follows:

$D_{avg}(p) = \frac{\sum W_i(p)D_i(p)}{\sum W_i(p)};$
$D_{shr}(p) = min\{D_1(p), D_2(p)\};$
$if \ \ ((p \in L_1) \ \&\& \ (p \in L_2))$

$$
\begin{array}{l}
\quad if\ ((VC_1 = \mathtt{0x00})\&\&(VC_2 = \mathtt{0x00}))\ \ D_f \leftarrow D_{avg};\\
\quad else\ if\ \ ((VC_1 = \mathtt{0x00})\&\&(p \notin \bar{O}\big|_{VH_2}))\ \ D_f \leftarrow D_2;\\
\quad else\ if\ \ ((VC_2 = \mathtt{0x00})\&\&(p \notin \bar{O}\big|_{VH_1}))\ \ D_f \leftarrow D_1;\\
\quad else\ D_f \leftarrow D_{avg};\\
else\ if\ \ ((p \in VH_1(O))\ \&\&\ (p \in VH_2(O))\\
\quad if\ ((VC_1 = \mathtt{0x00})\&\&(VC_2 = \mathtt{0x00}))\ \ D_f \leftarrow D_{shr}\\
\quad else\ if\ \ (VC_1 = \mathtt{0x00})\ \ D_f \leftarrow D_2;\\
\quad else\ if\ \ (VC_2 = \mathtt{0x00})\ \ D_f \leftarrow D_1;\\
\quad else\ \ D_f \leftarrow D_{shr};\\
else\ \ D_f \leftarrow D_{outside};
\end{array}
$$

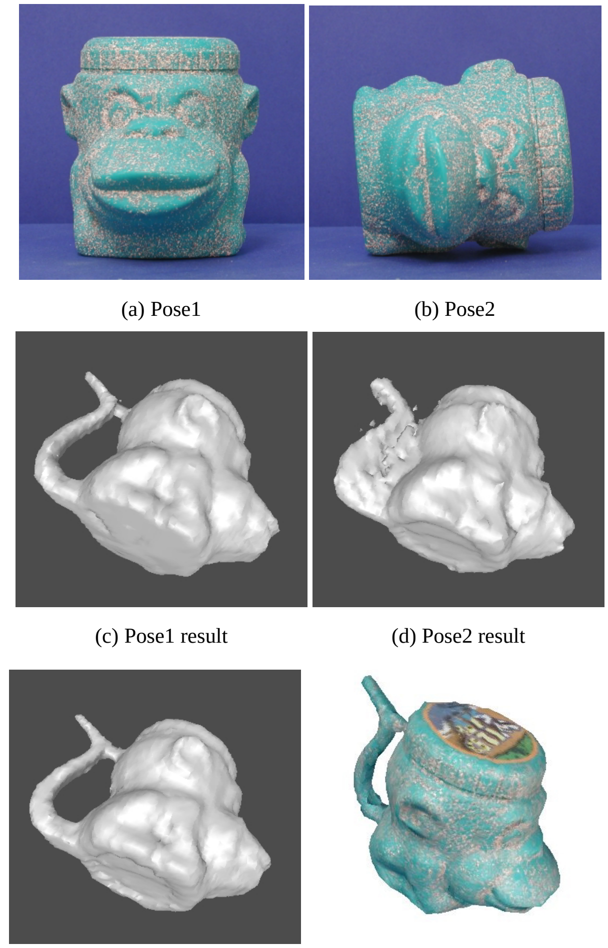
where, D_f is the final signed distance, L_i, and VC_i are the overlapping region and the voxel code at the ith pose respectively. And VH_i is the visual hull of the ith pose.

5. EXPERIMENTAL RESULTS

The 3D model reconstruction technique described above is tested on a real object shown in Figure 4. We sprayed a random dots pattern on the object's surface to enhance contrast. We took 8 stereo image pairs at 45 degree intervals for each pose. The two resulting 3D pose models and the final 3D model obtained by integrating the two poses are shown in Figure 4(c,d,e). The advantage of pose integration can be seen in these results. The final 3D model correctly depicts a hole and a concavity like the two original pose models. A novel view of the texture mapped model is also shown in Figure 4(f). Texture of each triangle on the surface is selected from an input image with the best-view criteria. Texture blocking between triangles with different view points is interpolated by the barycentric coordinate system.

6. CONCLUSIONS

Complete 3D model reconstruction based on volumetric integration of multiple partial shapes from two different poses is presented. Multiple partial shapes are acquired by incorporating a stereo vision technique and registered to a common coordinate system. We combine *shape from silhouettes* technique and *voxel coding* technique to remove erroneous data points due to stereo mismatchings. In order to recover all visible surfaces of the object, we acquire and reconstruct two 3D models of the object and integrate them together based on voxel codes from two poses. Pose integration improves the results by reconstructing a compelte 3D model closer to the original object.



(a) Pose1 (b) Pose2

(c) Pose1 result (d) Pose2 result

(e) Pose integration (f) Novel view of the object

Fig. 4. Test object and integration results

7. REFERENCES

[1] R. Bergevin, M. Soucy, H. Gagnon, and D. Laurendeau, "Toward a general multi-view registration technique," *IEEE Trans. on PAMI*, Vol. 18, No. 5, May. 1996.

[2] Y. Chen and G. Medioni, "Object modeling by registration of multiple range images," *Image and Vision Computing*, Vol. 10, No. 3, pp. 145-155, 1992.

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

[4] A.W. Fitzgibbon, G. Cross, and A. Zisserman, "Automatic 3D model construction for turn-table sequences", *European Workshop SMILE'98*, pp. 155-170, 1998.

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

[6] W.E. Lorensen and H.E. Cline, "Marching cubes: A high resolution 3D surface construction algorithm," *Computer Graphics*, Vol. 21, No. 4, July 1987.

[7] M.Novotni and R.Klein, "Geometric approach to 3D object comparison," *Proc. of Shape Modelling International(SMI2001)*, Genova, Italy, 2001.

[8] S.Y. Park and M. Subbarao, "Registration and integration of two pose models for complete 3D model reconstruction," *British Machine Vision Conference*, Submitted, 2002.

[9] 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*, Vol. 3, No. 4, Aug. 1987.

[10] M.D. Wheeler, Y. Sato, and K. Ikenuchi, "Consensus surfaces for modeling 3D objects from multiple range images", *IEEE Conference on Computer Vision*, 1998.