Virtual Reality Modeling from a Sequence of Range Images

Heung-yeung Shum Katsushi Ikeuchi Raj Reddy

The Robotics Institute
Carnegie Mellon University
Pittsburgh, PA 15213

Abstract

Virtual reality object modeling from a sequence of range images has been formulated as a problem of *principal component analysis with missing data (PCAMD)*, which can be generalized as a weighted least square (WLS) minimization problem. An efficient algorithm has been devised to solve the problem of PCAMD. After all visible P regions appeared over the whole sequence of F views are segmented and tracked, a $3F \times P$ *normal measurement matrix* of surface normals and an $F \times P$ *distance measurement matrix* of normal distances to the origin are constructed respectively. These two measurement matrices, with possibly many missing elements due to occlusion and mismatching, enable us to formulate multiple view merging as a combination of two WLS problems. By combining information at both the signal level and the algebraic level, a *modified Jarvis' march* algorithm is proposed to recover the spatial connectivity among all the reconstructed surface patches. Experiments using synthetic data and real range images show that our approach is robust against noise and mismatch because it produces statistically optimal object model by making use of redundancy from multiple views. A toy house model from a sequence of real range images is presented.

1 Introduction

Virtual Reality (VR) has fascinated most people through convincing real-time graphical illusion. However, creating models to inhabit virtual world is still the bottleneck and one of the main obstacles to making use of the technology in useful applications. Currently, most object models are constructed by human operators [2]. It would be much better to have a system that can automatically build models of real objects and real scenes that it observes. If we can develop a reliable technique to generate realistic virtual reality models by observing real objects from multiple views, we can reduce the effort and cost of model construction, and we can significantly broaden the application areas of virtual reality.

Observation-based modeling systems usually work with a sequence of images of the object(s), where the sequence spans a smoothly varying change in the positions of the sensor and/or object(s). Much work has been done in object modeling from a sequence of range images [1][4][5][9][12][16][21]. Most work assumed that transformation between successive views is either known or can be recovered, so that all data can be transformed to a fixed coordinate system. Inferring scene geometry and camera motion from a sequence of intensity image is also possible in principle [3][6][18][19][20]. Most existing algorithms in so-called "structure from motion" seem to be more useful for determining camera motion than for building 3D object models because the recovered object shape is defined by a collection of 3D points whose connectivity is not explicitly known.

Most previous systems have attempted to apply inter-frame motion estimates to successive pairs of views in a sequential manner [3][5][12]. Whenever a new view is introduced, it is matched with the previous view, and the transformation between these two successive views has to be recovered before the object model is updated. This sequential method does not work well in practice because local motion estimates are subject to noise and missing data. Local mismatching errors accumulate and propagate along the sequence, yielding inaccurately reconstructed object models.

Rather than sequentially integrating successive pairs of view, we should instead search for the statistically optimal object model that is most consistent with all the views. Although every single view provides only partial information of the object it is likely that any part of the object will be observed a number of times along the sequence. Object modeling from this sequence of views can be formulated as an overdetermined minimization problem because significant redundancy exists among all the views.

We propose to build boundary surface representation (B-rep) [12] object models from a sequence of range images. There are two key differences between our approach and other traditional approaches. First, our approach is to recover bounding surfaces and transformations simultaneously by employing *principal component analysis with missing data (PCAMD)*. Statistically optimal surface patches can be reconstructed from a sequence of segmented and tracked range images. Second, by combining information on both the algebraic level (surface descriptions) and the signal level (data points), the problem of spatial connectivity among all

reconstructed surface patches is also recovered so that a B-rep model can be reconstructed.

This paper is organized as follows. In section 2 we formulate the problem of multiple view merging as one of principal component analysis when data is missing. In section 3 we discuss of how to track a sequence of segmented range images. The spatial connectivity among different surface patches is solved using a modified Jarvis' march algorithm in section 4. We demonstrate the robustness and effectiveness of our approach by applying it to synthetic data and real range images in Section 5. Final comments and conclusions are presented in Section 6.

2 Multiple view merging

2.1 A motivational example

Suppose that our task is to make a model for a dodecahedron (*12*-faced polyhedra) from a sequence of segmented range images. Assume that we have tracked *12* faces over *4* nonsingular views. The segmented range images provide trajectories of plane coordinates $\{\mathbf{p}_p^f \mid f=1,...,4,\ p=1,...,12\}$, where $\mathbf{p} = (\mathbf{v}^T, d)^T$ represents a planar equation with surface normal $\mathbf{v}$ and normal distance to the origin d. Then we may form a 16×12 measurement matrix as follows:

$$W = \begin{bmatrix} \mathbf{p}_1^1 & \mathbf{p}_2^1 & \mathbf{p}_3^1 & \mathbf{p}_4^1 & \mathbf{p}_5^1 & \mathbf{p}_6^1 & * & * & * & * & * & * \\ \mathbf{p}_1^2 & \mathbf{p}_2^2 & \mathbf{p}_3^2 & \mathbf{p}_4^2 & * & * & \mathbf{p}_7^2 & \mathbf{p}_8^2 & * & * & * & * \\ \mathbf{p}_1^3 & \mathbf{p}_2^3 & * & * & * & \mathbf{p}_6^3 & * & \mathbf{p}_8^3 & \mathbf{p}_9^3 & \mathbf{p}_{10}^3 & * & * \\ * & * & * & * & * & * & \mathbf{p}_7^4 & \mathbf{p}_8^4 & \mathbf{p}_9^4 & \mathbf{p}_{10}^4 & \mathbf{p}_{11}^4 & \mathbf{p}_{12}^4 \end{bmatrix}$$

where every * indicates an unobservable face since there are only six visible faces from each nonsingular view. Our modeling task is now to recover the poses of all the *12* faces in a fixed coordinate system.

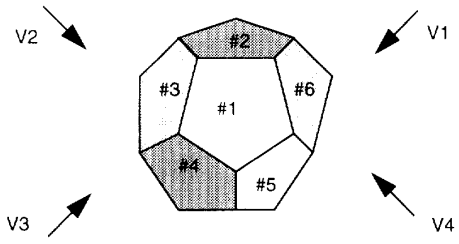

Figure 1 **Distinct views of a dodecahedron**

If the measurement matrix were complete, our task would be to average all those *12* faces over *4* views assuming data is noisy. In the absence of noise, any set of *12* faces from one of *4* views will do. The standard way to solve this problem is to apply singular value decomposition (SVD) [10] to this measurement matrix, whose rank is at most *4*. The measurement matrix can subsequently be factorized, with proper normalization, into a left matrix Q of transformation parameters and a right matrix P of plane coordinates

$$W = Q\ P$$

where

$$P = \begin{bmatrix} \mathbf{p}_1 & \mathbf{p}_2 & \cdots & \mathbf{p}_{12} \end{bmatrix},\ Q = \begin{bmatrix} Q^1 \\ \cdots \\ Q^4 \end{bmatrix}$$

and Q^f is the transformation of fth view with respect to the fixed world coordinate system, and $\mathbf{p}_p$ is the pth plane equation in the same world coordinate system.

Unfortunately, the measurement matrix is often incomplete in practice; it is not unusual for a large portion of the matrix to be unobservable. As we have seen in the above example, half of the measurement matrix is unknown. When the percentage of missing data is very small, it is possible to replace the missing elements with the mean or an extreme value; this is a common strategy in multivariate statistics [7]. However, such an approach is no longer valid when a significant portion of the measurement matrix is unknown.

One common practice in modeling from a sequence of images is to use extrapolation. For example, we can recover the transformation between view *1* and view *2* if there are at least three matched planar surfaces that are non-parallel [8]. Then we extrapolate the invisible planar surfaces in view *1* from its corresponding surfaces in view *2* which are visible using the transformation recovered. The same extrapolation step is similarly applied to invisible surfaces in view *1*. By repeating this process we can in principle extrapolate the locations of all invisible surfaces from visible surfaces [12]. A final step could be added to fine-tune the result by factorizing the extrapolated measurement matrix using SVD. A similar extrapolation approach "propagation method" [19] is used in motion and shape recovery from multiple intensity images.

The major problem with the extrapolation method is that once the estimated transformation is incorrect at any step, the extrapolated results will be erroneous. In sequential modeling errors accumulate and propagate all the way. The fine-tuning process at the last step would not improve the result dramatically since the extrapolated measurement matrix is inaccurate. To obviate this problem, we make use of more rigorous mathematical tools developed in computational statistics that caters for missing data without resorting to error sensitive extrapolation. We will demonstrate the formulation in this section and apply it to multiple view merging in the next section.

2.2 PCAMD

The problem of object modeling from a sequence of views shown in the previous section can be formulated as a problem of principal component analysis with missing data (PCAMD), which has been studied in computational statistics. Suppose that an $F \times P$ measurement matrix W consists of P individuals from an F-variate normal distribution with mean $\bar{\mu}$ and covariance Σ. Let the rank of W be r. If the data is complete and the measurement matrix filled, the problem of principal component analysis is to determine $\tilde{U}$, S, and $\tilde{V}$ such that

$$\left\| W - \mathbf{e}\mu^T - \tilde{U}\tilde{S}\tilde{V}^T \right\|$$

is minimized, where $\tilde{U}$, and $\tilde{V}$ are $F \times r$ and $P \times r$ matrices with orthogonal columns, $\tilde{S} = diag(\sigma_i)$ is an $r \times r$ diagonal matrix, μ is the maximum likelihood approximation of the mean vector and $\mathbf{e}^T = (1, \ldots, 1)$ is an F-tuple vector with all ones. The solution to this problem is essentially SVD of the centered (or registered) data matrix $W - e\mu^T$.

If data is incomplete, we have the minimization problem:

$$min \quad \phi = \frac{1}{2}\sum_{I} (W_{f,p} - \mu_p - \mathbf{u}_{f.}^T \mathbf{v}_{p.})^2 \qquad \text{(EQ 1)}$$

$$I = \{ (f,p) : \ W_{f,p} \ is \ observed \}$$

where $\mathbf{u}_{f.}$ and $\mathbf{v}_{p.}$ are column vector notations defined by

$$\begin{bmatrix} \mathbf{u}_{1.}^T \\ \ldots \\ \mathbf{u}_{F.}^T \end{bmatrix} = \tilde{U}\tilde{S}^{\frac{1}{2}} \text{ and } \begin{bmatrix} \mathbf{v}_{1.}^T \\ \ldots \\ \mathbf{v}_{P.}^T \end{bmatrix} = \tilde{V}\tilde{S}^{\frac{1}{2}}$$

Ruhe [14] proposed a minimization method to analyze one component model when observations are missing. One component model decomposes an $F \times P$ measurement matrix into an $F \times 1$ left matrix and a $1 \times P$ right matrix. Wiberg [22] extended Ruhe's method to the more general case of arbitrary component model. Wiberg [22] solved this problem by an iterative method using the bilinear property of the above problem. We have modified Wiberg's formulation and proposed a weighted least squares (WLS) approach to generalize this problem [15]. This approach has been successfully applied to merge planar patches tracked over multiple views.

2.3 Merging multiple planar patches

Suppose we have tracked P planar regions over F frames. We then have trajectories of plane coordinates $\{ (\mathbf{v}_{fp}, d_{fp}) \mid f = 1, \ldots, F, p = 1, \ldots, P \}$ where $\mathbf{v}_{fp}$ is the surface normal of the pth patch in the fth frame, and d_{fp} is the associated normal distance to the origin. To facilitate the decomposability of rotation and translation, instead of forming a $4F \times P$ measurement matrix as in section 2.1, we form surface normals V_{fp} into a $3F \times P$ matrix $W^{(v)}$ and distances d_{fp} into an $F \times P$ matrix $W^{(d)}$. $W^{(v)}$ and $W^{(d)}$ are called the *normal measurement matrix* and *distance measurement matrix* respectively.

It can be easily shown that $W^{(v)}$ has at most rank 3 and $W^{(d)}$ has at most rank 4 when noise-free, therefore, $W^{(v)}$ and $W^{(d)}$ are highly rank-deficient. We decompose $W^{(v)}$ into

$$W^{(v)} = R \ V$$

where

$$R = \begin{bmatrix} R^{(1)} \\ \ldots \\ R^{(F)} \end{bmatrix}$$

is the rotation matrix of each view with respect to the world coordinate system, and $V = [\mathbf{v}_1, \ldots, \mathbf{v}_P]$ is the surface normal matrix in the world coordinate system. Since R is a $3F \times 3$ matrix and V is an $3 \times P$ matrix, the rank of $W^{(v)}$ is at most 3.

Similarly, we can decompose $W^{(d)}$ into

$$W^{(d)} = T \ M$$

where

$$M = \begin{bmatrix} \mathbf{v}_1 \\ d_1 \end{bmatrix} \ldots \begin{bmatrix} \mathbf{v}_P \\ d_P \end{bmatrix}, \ T = \begin{bmatrix} [\mathbf{t}_1 R_1 \ 1] \\ \ldots \\ [\mathbf{t}_F R_F \ 1] \end{bmatrix},$$

and $\mathbf{t}_f$ and R_f are the translation vector and rotation matrix of view f with respect to a fixed world coordinate system.

We can also decompose $W^{(d)}$ into

$$W^{(d)} = \begin{bmatrix} t_1 R_1 \\ \ldots \\ t_F R_F \end{bmatrix} [v_1 \ \ldots \ v_P] + \begin{bmatrix} 1 \\ \ldots \\ 1 \end{bmatrix} [d_1 \ \ldots \ d_P]$$

When all elements in the two measurement matrices are known, we need to solve two least-squares problems. However, since only part of the planar regions are visible in each view, we end up with two WLS problems instead. The first least squares problem, labeled as WLS-1, is

$$min \sum_{f=1,\ldots,F, p=1,\ldots,P} (\gamma_{f,p}^{(v)} (W_{f,p}^{(v)} - [RV]_{f,p}))^2 \qquad \text{(EQ 2)}$$

and the second one, denoted as WLS-2, is

$$min \sum_{f=1,\ldots,F, p=1,\ldots,P} (\gamma_{f,p}^{(d)} (W_{f,p}^{(d)} - [TM]_{f,p}))^2 \qquad \text{(EQ 3)}$$

where $\gamma_{f,p} = 0$ if surface p is invisible in frame f, and $\gamma_{f,p} = 1$ otherwise. All weights can be any number between zero and one, depending on the significance or confidence of each measurement. A similar WLS formulation is also used in [18].

3 Surface Patch tracking

In this section, we briefly overview each module of our surface patch tracking system: range image segmentation, adjacency graph building, and two-view matching.

3.1 Range image segmentation

There are many different techniques for range image segmentation. By and large they can be divided into feature-based and primitive-based approaches, although statistics-based approaches have also been introduced recently. Feature-based approaches yield precise segmentation but are sensitive to noise in practice. For example, Gaussian and mean curvatures can be used to label different regions before region growing; however, this process is quite sensitive to noise. Primitive-based approaches are more robust to noise but constrained by the number of primitives. The higher the degree of surface polynomial, the more difficult and the less robust the segmentation is likely to be.

We have used the primitive-based region growing segmentation method of [8]. The regions are established via region growing from seed points, i.e., the seed points are chosen from points which are closest to their approximating primitives, and then merged with their neighbors until the best-fit errors become unacceptable.

3.2 Adjacency graph

Once we have successfully segmented the range data for each view, the range image associated with view i can be represented as a set of planar regions $I_i = \{v_{ij}, d_{ij}, c_{ij}\}$, where v_{ij} and d_{ij} are the normal and distance of the jth segment planar surface respectively, and c_{ij} is the centroid of jth segmented region.

From each view of the 3D object, we build an adjacency graph where every node in the graph represents a visible planar region and each arc connects two adjacent nodes. The adjacency graph is updated whenever this view is matched with another. Eventually we have adjacency information among all visible planar regions after tracking all of them for the whole sequence. From the adjacency graph, all the object vertices can be located; thus, 3D object model is obtained. However, augmenting adjacency graph is difficult for concave objects because of occlusion. A better way of establishing spatial connectivity among all surfaces is discussed in section 4.

We have implemented the planar surface patch tracking system which employs an algorithm to generate the adjacency graph. This algorithm makes use of range data because there is significant change in range data across an occluding edge.

3.3 Matching two views

Given two adjacent segmented images I_1 and I_2, we would like to find correspondence between different regions in two views, i.e., we want to find a mapping ϕ: $I_1 \rightarrow I_2$ such that a certain distance measurement $d(I_1, I_2)$ is minimized.

Two questions arise in matching two views of planar regions. The first is how to make correspondence between two views; the second is how to recover the transformation between them. Our solution to the first problem is to use adjacency information between two segmented patches and between segmented surface normals. If displacement between two views is relatively small, there should be only linear shape change [11] within the same aspect, corresponding segmented regions are of similar size (number of points), centroid, and surface normals. When a new aspect appears, which signals a nonlinear shape change, there would be significant change in these parameters. There may not always be solutions to the second problem because we need at least two corresponding non-parallel faces to determine rotation and three to determine translation. In practice, we can make the assumption that we always have two non-parallel corresponding faces in two adjacent views.

In fact, solving the second problem can be of help to the first problem because we can then make use of the hypothesis-and-test approach. We iteratively select two pairs of non-parallel faces from the two images to be matched, estimate the corresponding rotation matrix, and then attempt to match the rest of the faces. The estimated transformation matrix is only used to help building the adjacency graph, while the precise transformation is robustly recovered from our WLS method.

Multiple view tracking is done by sequentially matching two adjacent views. Whenever a new view is added, the adjacency graph and the weight matrix are automatically modified. Because of the problems associated with updating adjacency graph, subsequent to surface patch tracking and multiple view merging, we use a better method to establish the spatial connectivity among surfaces in next section.

4 Spatial Connectivity

Once we have extracted the equations of planar surfaces of the object, we then need to establish spatial connectivity relationship among these surfaces. Other than augmenting adjacency graph whenever a new view is introduced, we present a new method of recovering surface connectivity among reconstructed surface patches. We show that the problem of spatial connectivity of boundary surfaces can be reduced to that of supporting lines of a simple polygon.

4.1 Half-space intersection and union

We assume that every planar patch P of an object model is a simple polygon. Every (infinite) plane divides the space into two parts, inside and outside, with surface normal pointing towards the external side of the object. Given an unbounded planar surface, if we intersect all other planar surfaces on it, we obtain supporting lines as illustrated in Fig.2. Each supporting line is directed so that the interior of P lies locally to its right. The right half-plane created by such a directed supporting line c is called the supporting half-plane, and is characterized as supporting the polygon; however, a concave P might not all lie in the right half-plane as indicated in Fig.2.

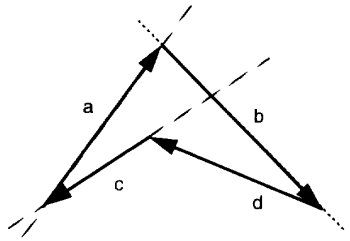

Fig 2 **A simple polygon and its supporting lines (stippled and solid lines)**

For each point x in the plane, if we know which side of each supporting line x lies on, we know if x is inside P. Therefore, the polygon P (and its interior) can be represented as a boolean formula whose atoms are those supporting lines. In other words, a simple polygon can be represented by intersection and union of its supporting line. For example, a boolean formula for the polygon in Fig.2 can be $c\bar{a} \oplus a\bar{b} \oplus b\bar{d} \oplus \bar{d}c$.

4.2 Modified Jarvis' march

The problem of establishing spatial connectivity of supporting lines can be formulated as a modified convex hull-like problem which involves only vertices. This problem can also be regarded as one of cell decomposition which involves data points. We propose a modified Jarvis' march algorithm to reconstruct simple polygons from supporting lines and valid data points. The algorithm to recover spatial connectivity among 3D surfaces is discussed in section 4.3.

Definition 1

A point is defined as valid in a simple polygon if there exist sufficient valid data points around its neighborhood.

Lemma 1

The intersection point P of two supporting lines is valid in a simple polygon if and only if the intersection of two corresponding half-planes is valid locally at P.

Proof:

When the intersection of two half-planes is valid locally at P, the intersection point of these two supporting lines is valid by its definition.

Assume that the intersection point of two supporting lines is valid. Since two lines divide the plane into four regions, there must exist at least one such region among four around the intersection point that is a valid cell of the simple polygon. Therefore, the intersection of two half-planes is valid locally at P. (Q.E.D.)

The Lemma 1 leads to a modified Jarvis' march algorithm of reconstructing simple polygon from supporting lines and valid data points.

To construct the simple polygon from all supporting lines and valid data points, we first precompute all intersection points which are candidates of vertices of the simple polygon. If we march successive vertices with the least turning angle, we obtain their convex hull; this is referred to as Jarvis' march algorithm [13]. The kernel of the simple polygon, if it exists, can also be found by intersecting all half-spaces. Using Lemma 1, however, enables us to find the correct simple polygon by marching all points whose local neighborhood is valid. We call this algorithm the "modified Jarvis' march".

Assume that we have first found the lowest left point p_1 of the set of vertex candidates, which is certainly a convex hull vertex, but not necessarily a vertex for our simple polygon (unless it is valid locally). For example, in Fig.3, p_1 is not a simple vertex because $p_5p_1p_2$ is not a valid triangle cell (valid cells are shaded areas which show valid data points). Since $p_6p_2p_3$ is a valid triangle cell, we start our algorithm from p_2.

A data structure is defined for each intersection point P as follows

```
typedef {
    intersect-point left, right, up, down;
    intersect-point previous, next;
} intersect-point  P;
```

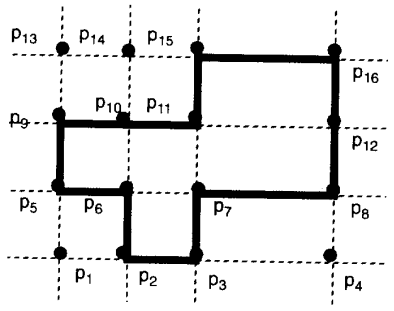

Fig 3 **Modified Jarvis' march and cell decomposition. Shaded area represents valid data points.**

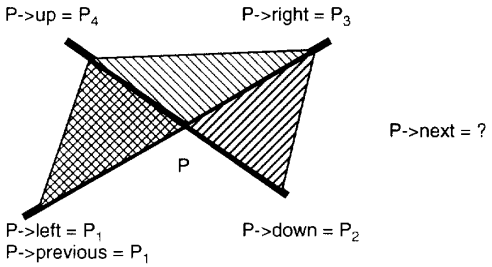

***Fig 4* Data structure of intersection point**

Fig.4 shows the relationship among the members of the data structure. Assume that an intersection point is intersected by only two supporting lines.

After the starting vertex is found, we march for the next vertex as illustrated in Fig.4. If there are sufficient data points in cell PP_2P_3, next valid vertex is P_2. If P_2 is not valid, we check if P_3 is valid. If P_3 is also invalid, P_4 must be valid, or an error will occur. The march ends when the next vertex is the starting vertex. The modified Jarvis' march (MJM) algorithm is given as follows:

Algorithm MJM

Step 1. initialize starting vertex

START->previous = NULL,
P = START->next,
P->previous = START;

Step 2. march

P->left = P_1; P->down = P_2; P->right = P_3; P->up = P_4;
if cell PP_2P_3 valid, P->next = P_2 (case 1)
else if cell PP_3P_4 valid, P->next = P_3 (case 2)
else if cell PP_4P_1 valid, P->next = P_4 (case 3)
else error occurs;

Step 3. terminate

if P->next = START.

A postprocessing step may be necessary to remove points which belong to case 2 in step 2 of the march algorithm. These points are on the same line with its previous point and its next point. For example, in Fig.3, p_{12} can be removed because p_8 and p_{16} make it redundant.

As can be seen from the above algorithm and Fig.3 as well as Fig.4, the problem of single polygon reconstruction from supporting lines and valid data points is one of cell decomposition. As we march around all supporting lines, a boolean formula of the simple polygon can be readily formulated.

4.3 3D spatial connectivity

So far we have discussed the problem of recovering the connectivity of supporting lines of a simple polygon. The approach uses information at both signal level (real data points) and algebraic level (line equations). The same hybrid approach can be applied to the problem of spatial connectivity of planar surfaces in 3D.

Indeed, the problem of connectivity of planar surfaces in 3D can be reduced to a set of problems of connectivity in 2D. Assume that we have recovered a set of N face equations and transformation among different views (e.g., from PCAMD). All valid data points from multiple views can be merged in the same world coordinate system. For each face F_i, if we intersect all other *N-1* faces F_j ($j = 1, \ldots, N-1, j \neq i$) with F_i and project all these lines onto F_i, we get *M (=N-1)* supporting lines on face F_i. We also project nearby 3D points onto this face F_i. We assume that no two supporting lines are parallel (or a normal threshold d can be set such that $v_i v_j \geq d$). For any of the M supporting lines, we intersect it with the rest *M-1* lines, we get all possible candidates for vertices of the valid simple polygon which is the model of face F_i, as illustrated in Fig.5. The modified Jarvis' march algorithm can be then applied to each of the N faces accordingly. By connecting all polygons recovered, we get the entire 3D object model boundary. A simple algorithm can be accordingly constructed to establish 3D spatial connectivity then.

Fig. 5 shows an example. It is a face of a toy house model. The complete house model is reconstructed and presented in next section. Fig.5(a) shows intersections of supporting lines and nearby data points projected on this face, while Fig.5(b) superimposes a reconstructed simple polygon model of this face on Fig.5(a).

***Fig 5* Reconstruction of connectivity. The tiny dots are projected nearby data points. Intersections of supporting lines are large dots. Vertices of reconstructed simple polygon are small squares.**

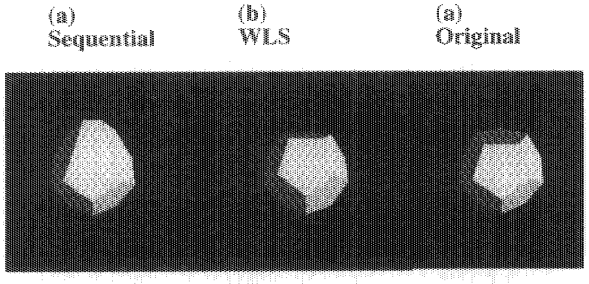

Fig 6 **Recovered and original dodecahedron models**

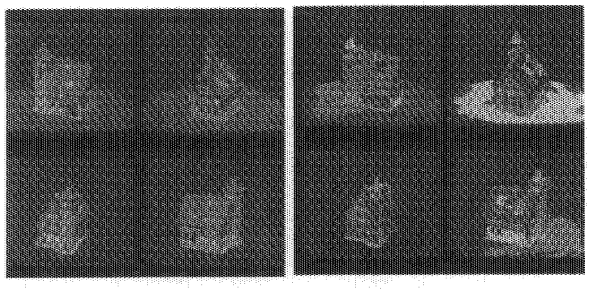

Fig 7 **A sequence of images of a toy house**

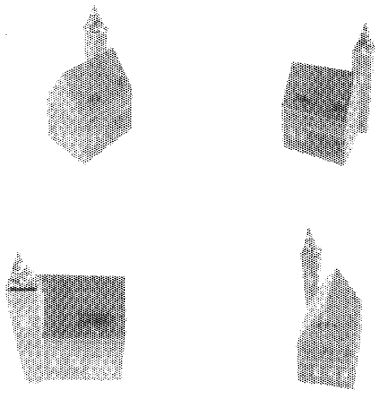

Fig 8 **Four views of display of recovered house model**

5 Experiments

In this section, we present results of applying our algorithm on synthetic data and on real range image sequence of objects. We demonstrate the robustness of our approach using synthetic data, and present the recovered model from real range images.

5.1 Synthetic data

Our synthetic data set consists of a set of 12 planes as in the case of the dodecahedron in the last section. A dodecahedron with 4 different views is shown in Fig.1 in section 2.

We study the effectiveness of our approach when data is corrupted by noise and mismatching occurs. Our synthetic data consists of a set of 12 surface patches randomly distributed around all faces of a dodecahedron. Correspondence is assumed to be known. Only the first WLS problem is studied because of the similarity between those two WLS problems.

When an observed surface normal is wrong in one particular view, the conventional sequential reconstruction method results in an erroneous surface normal and transformation. The errors propagate as new views are introduced regardless of the number of views in which this surface is visible. However, our WLS approach gives much smaller reconstruction error on this observed surface normal by distributing the errors in all views. There are 12 observations of this surface normal in 16 views and its first observation is off by an angle between 0^o and 40^o. The reconstructed models of sequential method and WLS method are shown in Fig. 6 along with the original model, for the case of 40^o angle deviation of one surface normal in the first view. Fig. 6a shows a badly-skewed model, which is the worst case from the sequential method since the error was introduced in the first frame. Fig. 6b shows the reconstructed model by the WLS method while the original dodecahedron model is presented in Fig. 6c. More detailed studies are carried out in [15].

5.2 Real range image sequence

We have applied our algorithm to a sequence of range images of a toy house. Figures 7(a) and 7(b) show four frames of the sequence of 12 views and their corresponding segmentation results. Segmentation is not perfect in several views. Figure 8 shows the result of our system, two shaded and texture mapped views of recovered toy house model.

6 Concluding remarks

A virtual reality object modeling system using multiple range images has been described in this paper. The boundary representation object model is reconstructed by integrating information from different views. One significant contribution of this work is the application of principal component analysis with missing data to object modeling from a sequence of views. An inherent problem in multiple view integration is that the information observed from each view is incomplete and noisy. Based on Wiberg's formulation, we have generalized principal component analysis with missing data as a WLS minimization problem and presented an efficient algorithm, PCAMD, to solve it. Spatial connectivity among different surface patches is also recovered using reconstructed surface descriptions and range data.

"One general principle in computer vision is, if surface information is not enough to determine each surface locally, use global constraints that constrain relative configuration of the surfaces so that the total degrees of freedom decrease." [17]. The object modeling technique presented in this paper is an example of this principle, where the algebraic structure of surface equations from multiple views is used as the global constraint. The reconstructed object model is statistically optimal because it is most consistent with all of the views. By observing and employing different forms of input redundancy, our approach can be easily extended to other vision problems such as shape and motion from a sequence of intensity images.

Acknowledgments

We thank Sing Bing Kang for his many valuable comments which have significantly improved the quality of this paper. We would also like to thank David Chan for helping segmentation of range images, and Mark Wheeler for proofreading.

This research was supported in part by the Avionics Laboratory, Wright Research and Development Center, Aeronautical Systems Division (AFSC), U.S. Air Force, Wright-Patterson AFB, Ohio 45433-6543 under Contract F33615-90-C-1465, ARPA Order No. 7597, and in part by NSF under Contract CDA-9121797.

References

[1] N. Ahuja and J. Veenstra. Generating Octrees from Object Silhouettes in Orthographic Views. IEEE Trans. PAMI Vol. 11 No. 2, pp. 137-149, 1989.

[2] F. Arman and J.K. Aggarwal. Model-based Object Recognition in Dense-Range Images - A Review. ACM Computing Surveys. Vol. 25, No. 1, pp. 5-43, 1993.

[3] A. Azarbayejani, B. Horowitz, and A. Pentland. Recursive Estimation of Structure and Motion using Relative Orientation Constraints. Proc. IEEE CVPR, pp. 294-299, 1993.

[4] B. Bhanu. Representation and Shape Matching of 3-D Objects. IEEE Trans. PAMI Vol. 6, pp. 340-351, 1984.

[5] Y. Chen and G. Medioni. Object Modeling by Registration of Multiple Range Images. Proc. IEEE Int. Conf. R & A, pp. 2724-2729, April 1991.

[6] C. Debrunner and N. Ahuja. Motion and Structure Factorization and Segmentation of Long Multiple Motion Image Sequences. Proc. ECCV, pp. 217-221, 1992.

[7] Y. Dodge. Analysis of Experiments with Missing Data. Wiley. 1985.

[8] O.D. Faugeras and M. Hebert. The Representation, Recognition, and Localization of 3-D Objects. Int. J. Robotics Research. Vol. 5, No. 3, pp. 27-52, 1986.

[9] F.P. Ferrie and M.D. Levine. Integrating Information from Multiple Views. Proc. IEEE Workshop on Computer Vision. pp. 117-122, 1987.

[10] G.H. Golub and C.F. Van Loan. Matrix Computation. 2nd Edition. The John Hopkins University Press. 1989.

[11] K. Ikeuchi. Generating an Interpretation Tree from a CAD Model for 3D-Object Reconstruction in Bin-Picking. Int. J. Computer Vision. pp. 145-165, 1987.

[12] B. Parvin and G. Medioni. B-rep from Unregistered Multiple Range Images. Proc. IEEE Int. Conf. R & A, pp. 1602-1607, May 1992.

[13] F.P. Preparata and M.I. Shamos. Computational Geometry. Springer-Verlag. 1988.

[14] A. Ruhe. Numerical Computation of Principal components when Several Observations are Missing. Tech Rep. UMINF-48-74, Dept. Information Processing, Umea Univ., Umea, Sweden, 1974.

[15] H. Shum, K. Ikeuchi, and R. Reddy. Principal Component Analysis with Missing D and Its Application to Object Modeling. To appear in Proc. CVPR 94, Seattle, June, 1994.

[16] M. Soucy and D. Laurendeau. Multi-Resolution Surface Modeling from Multiple Range Views. Proc. IEEE CVPR, pp. 348-353, 1992.

[17] K. Sugihara. Machine Interpretation of Line Drawings. The MIT Press, 1986.

[18] R. Szeliski and S.B. Kang. Recovering 3D Shape and Motion from Image Streams using Non-linear Least Squares. DEC CRL 93/3, 1993.

[19] C. Tomasi and T. Kanade. Shape and Motion from Image Streams under Orthography: A Factorization Method. Int. J. of Computer Vision, 9:2, pp. 137-154, 1992.

[20] S. Ullman. The Interpretation of Visual Motion. The MIT Press. 1979.

[21] B.C. Vemuri and J.K. Aggarwal. 3-D Model Construction from Multiple Views Using Range and Intensity Data. Proc. CVPR, pp. 435-437, 1986.

[22] T. Wiberg. Computation of Principal Components when Data are Missing. Proc. Second Symp. Computational Statistics, pp. 229-236, Berlin, 1976.