Mass Vector Controlled View Estimation for Automatic 3D Object Modeling

Xiaobu Yuan
Department of Computer Science
Memorial University of Newfoundland
St. John's, Newfoundland, Canada A1C 5S7

Abstract

The representation of three-dimensional(3D) objects is acquired by accumulating features extracted from sensors at different viewing directions. How many and what views are necessary to build up 3D object models is a topic that leads to automatic modeling. This paper first investigates surface visibility; and then, after introducing mass vector chains(MVC), discusses the relationship between MVC and spatial closure of object surfaces. As a result, a new method is proposed to estimate viewing directions of unprocessed surfaces for automatic 3D object modeling. Experimental results are given at the end.

1 Introduction

3D computer vision studies techniques that enable a computer system to understand its environment from various sensory inputs[9]. Cognitive abilities of a vision system are normally investigated in the field of object recognition. However, its fundamental visual capabilities that acquire symbolic 3D representations from sensory inputs come from the research of object reconstruction[1].

In the last decade, there has been considerable discussion on the theoretical as well as practical aspects of reconstruction systems. Significant results have been achieved from active investigation on optical devices[5], spatial feature extraction techniques[1, 11], and modeling methods[2, 3, 4, 7, 8]. However, few attempts were devoted to create an automatic modeling system, i.e., a reconstruction system that builds up object models without human intervention. The major problem is to decide by the system itself the number and direction of necessary views from which all the spatial features of scene objects can be obtained.

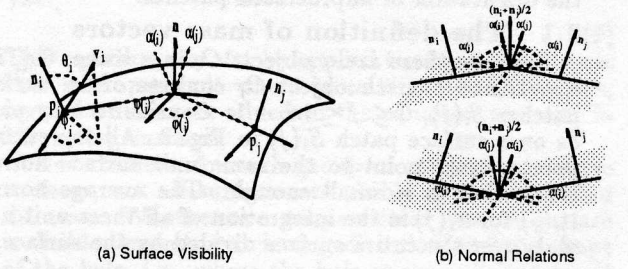
In principle, a modeling system includes both feature extraction that retrieves symbols from input data, and feature integration that accumulates these symbols into consistent object descriptions. For a regular convex object, all its boundary surfaces should be visible in two opposite viewing directions. Unfortunately, scene objects usually exhibit irregular boundaries of convex and concave planar or curve surfaces. At any viewing direction, even a same object could have different appearances because of random rotations and/or translations in the space. It is impossible to preset two or more viewing directions and guarantee that all the boundary surfaces of an object are visible in at least one observation. Therefore, except the basic functions to extract and accumulate features from images, a generalized reconstruction system expects a self-controlled modeling mechanism so that the system itself is able to check the spatial closure of object models and to optimize the sensory direction for unprocessed features.

2 Surface Visibility

In surface modeling methods, 3D objects are described by boundary surfaces[7, 8]. As surface patches are extracted from different viewing directions and used to form spatially closed boundaries, their visibility determines the number and direction of necessary views.

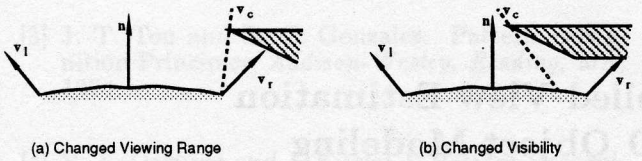
A surface patch is composed by a set of surface points. It is visible in a particular viewing direction only if all its surface points are fully exposed to that direction. When its normal is defined to be the surface normal at that point, a surface point p_i is visible as far as the angle θ_i between its normal $\mathbf{n}_i$ and the viewing vector $\mathbf{v}_i$ is less than 90°(Fig. 1(a)). Therefore, with respect to $\mathbf{n}_i$, the visible angle θ_i for p_i ranges from 0° to 90° as φ_i sweeps from 0° to 360°.

For any two surface points p_i and p_j, however, the range of visible angle is a conjunction of the visible ranges for both points. The normal at each surface point differs with the average visible direction $(\mathbf{n}_i + \mathbf{n}_j)/2$ by an angle as $\alpha(i)$ and $\alpha(j)$ (Fig. 1(b)). For convexly related points, the new visible vector $\mathbf{v}$ for the two surface points covers the ranges as $(\mathbf{n}_i + \mathbf{n}_j)/2$ moves to $90° - \alpha(j)$ along the direction to p_i, and to $90° - \alpha(i)$ for p_j. On the other hand, for concavely



(a) Surface Visibility (b) Normal Relations

Figure 1: Surface Visibility



(a) Changed Viewing Range (b) Changed Visibility

Figure 2: Visibility under Occlusion

related points, v covers the ranges to $90° - \alpha(i)$ along the direction to p_i, and to $90° - \alpha(j)$ for p_j.

Generally, in respect to the average normal direction n, the visible range $\mathbf{V}(S)$ for a surface patch S is the integration of visible viewing vectors v for $\theta(\varphi)$ in all φ directions, i.e., $\mathbf{v}(\mathbf{n}, \theta(\varphi))$. Along each φ, $\theta(\varphi)$ is the range between n and the surface normal with the biggest normal difference in the direction of φ for concave points or opposite to φ for convex points. Therefore, $\mathbf{V}(S)$ can be expressed as,

$$\mathbf{V}(S) = \int_0^{2\pi}\int_0^{\pi/2-\alpha(\varphi)} \mathbf{v}(\mathbf{n}, \theta(\varphi)) d\theta d\varphi. \quad (1)$$

Of course, under occlusion, surface visibility is usually different. The visible range of a surface patch becomes smaller if occlusion surfaces block part of its visible viewing direction. Suppose at a certain φ there is a surface on top of the surface patch(Fig. 2(a)). The original visible viewing direction ranges from n to $\mathbf{v}_r$. If the occluded visible direction is specified by $\mathbf{v}_c$, the surface has no effect to the visibility of the surface patch when $\mathbf{v}_c$ is outside the visible range $\mathbf{v}(\mathbf{n}, \theta(\varphi))$, i.e., beyond $\mathbf{v}_r$ in the figure. Otherwise, $\mathbf{v}_c$ narrows down the new visible range. Therefore, the new range is determined by

$$\theta'(\varphi) = min\{\angle(\mathbf{n}, \mathbf{v}_c), (\mathbf{n}, \mathbf{v}_r)\}. \quad (2)$$

As a special case, no occlusion exists if $\theta'(\varphi) = \theta(\varphi)$.

Sometimes, the occluded visible range may be so small that the surface will only be partially visible. This happens when part of the surface are invisible in any direction within the visible range $\mathbf{V}(S)$ because of the occluding surface, as illustrated in Fig. 2(b). It is called changed visibility since not only the visible range but also the visibility of the surface is changed.

3 Mass Vector Chains

While surface visibility reflects the shape of an object, processed surface patches can be used to estimate the orientation of unprocessed patches.

3.1 The definition of mass vectors

Suppose there are n objects O_i in a scene, $0 \le i \le n-1$, and the ith object O_i consists of m_i surface patches $S_i(j)$, $0 \le j \le m_i - 1$. Consider a tiny piece δs on a surface patch $S_i(j)$ in Fig. 3. All the surface points on δs point to the same unit surface normal $\mathbf{n}(s)$ when it is small enough. The average normal $\mathbf{n}_i(j)$ for $S_i(j)$ is the integration of all these unit normals over the entire surface divided by the surface,

$$\mathbf{n}_i(j) = \frac{1}{S_i(j)} \iint_{S_i(j)} \mathbf{n}(s)\, ds. \quad (3)$$

Let the plane perpendicular to $\mathbf{n}_i(j)$ be $P_i(j)$. Surface patch $S_i(j)$ becomes a planar region $R_i(j)$ when projected onto $P_i(j)$. Since a surface piece δs appears smaller when the observation direction is not $\mathbf{n}(s)$ but $\mathbf{n}_i(j)$, it maps onto plane $P_i(j)$ as δR,

$$\delta \mathrm{R} = (\mathbf{n}(s) \cdot \mathbf{n}_i(j))\, \delta s.$$

As a result, the projected region $R_i(j)$ becomes

$$R_i(j) = \iint_{S_i(j)} (\mathbf{n}(s) \cdot \mathbf{n}_i(j))\, ds. \quad (4)$$

A mass vector chain for object O_i is a series of weighted vectors. In this series, a vector $\vec{\mathbf{V}}_i(j)$ is defined for each individual surface patch $S_i(j)$ of O_i. This vector points to the average normal direction $\mathbf{n}_i(j)$ of the surface patch $S_i(j)$; and its weight is the projected region $R_i(j)$ on plane $P_i(j)$, which is perpendicular to $\mathbf{n}_i(j)$.

$$\vec{\mathbf{V}}_i(j) = \mathbf{n}_i(j)\, R_i(j) \quad (5)$$

For the surface patch, $\mathbf{n}_i(j)$ is its average visible direction and $R_i(j)$ is the surface size when viewed in that direction.

3.2 MVC and boundary closure

For a convex polyhedron, it has been proved[6] that, if the weight of a surface normal is the polygon size, the weighted normals for all the polygon surfaces form a closed chain when placed end to end. The relation was established by checking two sets of projected areas from surfaces visible in two opposite directions. If condition 'visible' is changed to 'facing', the proof is then extended to any ordinary concave object, i.e., its total Gaussian mass of boundary surface patches must be zero,

$$\iint_S G(\mathbf{n}(s))\mathbf{n}(s)\, ds = \oiint_S \mathbf{n}(s)\, ds = 0$$

where $G(\mathbf{n}(s))$ is the mass of extended Gaussian Image with a same $\mathbf{n}(s)$. Therefore, the total mass of extended Gaussian image for the boundary surface of object O_i with m_i surface patches $S_i(j)$ can be expressed as

$$\oiint_S \mathbf{n}(s)\, ds = \sum_{j=0}^{m_i-1} \iint_{S_i(j)} \mathbf{n}(s)\, ds = 0. \quad (6)$$

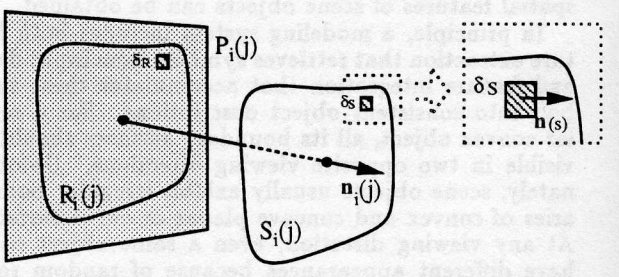


Figure 3: The Definition of Mass Vector

On the other hand, the total mass vector for object O_i is the sum of mass vectors for all the surface patches $S_i(j)$ defined in Eq. 5, i.e. $\sum_{j=0}^{m_i-1} \vec{\mathbf{V}}_i(j)$. Using Eq. 3 to Eq. 6, the total mass vector can be derived.

$$\begin{aligned}
\sum_{j=0}^{m_i-1} \vec{\mathbf{V}}_i(j) &= \sum_{j=0}^{m_i-1} \mathbf{n}_i(j)\, R_i(j) \\
&= \sum_{j=0}^{m_i-1} \mathbf{n}_i(j) \iint_{S_i(j)} (\mathbf{n}(s) \cdot \mathbf{n}_i(j))\, ds \\
&= \sum_{j=0}^{m_i-1} \mathbf{n}_i(j) [\mathbf{n}_i(j) \cdot \iint_{S_i(j)} \mathbf{n}(s)\, ds] \\
&= \sum_{j=0}^{m_i-1} [\frac{1}{S_i(j)} \iint_{S_i(j)} \mathbf{n}(s)\, ds] [S_i(j)(\mathbf{n}_i(j) \cdot \mathbf{n}_i(j))] \\
&= \sum_{j=0}^{m_i-1} \iint_{S_i(j)} \mathbf{n}(s)\, ds = 0. \qquad (7)
\end{aligned}$$

As a result, for any 3D object, the set of surface patches composes a closed surface boundary only when its mass vectors form a closed chain too.

3.3 Estimation of unprocessed surfaces

According to Eq. 7, the sum of mass vectors for a closed object model is a zero vector, i.e., $\sum_{j=0}^{m_i-1} \vec{\mathbf{V}}_i(j) = 0$. If, during reconstruction, the total of all mass vectors for a building model is not zero but equals to a vector $\vec{\mathbf{V}}_{gap}$, there must be some unprocessed surface patches whose mass vectors sum to be the negative of $\vec{\mathbf{V}}_{gap}$. Suppose the number for processed surface patches is m_i', then, from equation

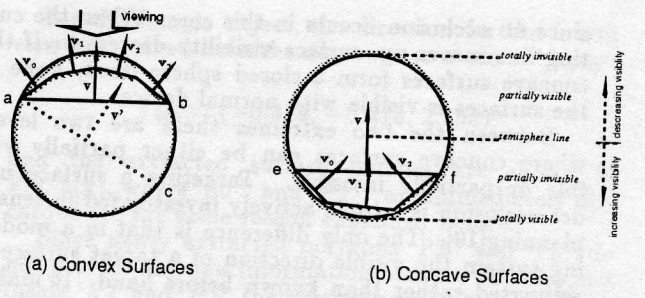
$$\sum_{j=0}^{m_i'-1} \vec{\mathbf{V}}_i(j) + \sum_{j=m_i'}^{m_i-1} \vec{\mathbf{V}}_i(j) = 0,$$

it is easy to get the result,

$$\sum_{j=m_i'}^{m_i-1} \vec{\mathbf{V}}_i(j) = -\sum_{j=0}^{m_i'-1} \vec{\mathbf{V}}_i(j) = -\vec{\mathbf{V}}_{gap}. \qquad (8)$$

As defined to be the average normal vector $\mathbf{n}_i(j)$, each mass vector $\vec{\mathbf{V}}_i(j)$ is actually the average visible direction of that surface patch. Therefore, $\vec{\mathbf{V}}_{gap}$ provides an estimated direction from which these unprocessed surface patches could be observed.

This can be further explained by the two spheres in Fig. 4. Since a convex object has a unique extended Gaussian image expression[6], a Gaussian sphere covering the object stands for its boundary condition, or a circle in two-dimensional case. Suppose from a certain viewing direction, four surface patches are extracted(Fig. 4(a)), the total mass vector is $\mathbf{v} = \mathbf{v}_0 + \mathbf{v}_1 + \mathbf{v}_2 + \mathbf{v}_3$, which stands for the observed small arc from a to b. It is obvious that the average visible direction for the unprocessed big arc from b to a is in the opposite direction of $\mathbf{v}$. Even if more surface patches are processed after several views, the negative of the updated total mass vector, which is $\mathbf{v}'$ in the figure, still points to the visible direction of unprocessed portion, the curve from c to a.



(a) Convex Surfaces (b) Concave Surfaces

Figure 4: View Estimation

For objects with concave surfaces, a similar Gaussian sphere can also be used to examine the boundary condition for the concave part where only concave surfaces are concerned. Depending on the shape of the cave, the sphere can be cut to open at any location. For instance, in Fig. 4(b), the cave is so shaped that the circle opens at e and f. Concave surfaces usually do not form a closed mass vector chain by themselves. When a virtual surface patch is introduced to represent the cutting surface, however, the total mass vector for the concave surfaces plus the virtual surface patch must be zero since they form a close surface. Suppose there are m_h concave patches in a hole, the following relation exits,

$$\sum_{j=0}^{m_h-1} \vec{\mathbf{V}}_i(j) + \vec{\mathbf{V}}_i(hole) = 0,$$

where $\vec{\mathbf{V}}_i(hole)$ is the mass vector for the virtual surface patch. As a result, an equation similar to Eq. 8 is obtained to estimate the visible direction for unprocessed concave surfaces,

$$\sum_{j=m_h'}^{m_h-1} \vec{\mathbf{V}}_i(j) = -(\sum_{j=0}^{m_h'-1} \vec{\mathbf{V}}_i(j) + \vec{\mathbf{V}}_i(hole)) = -\vec{\mathbf{V}}_{hgap}. \qquad (9)$$

This equation can also be applied to objects with more than one open hole where more virtual mass vectors, instead of one, are added to the equation to count in all cutting surfaces.

Nevertheless, for each group of connected concave surfaces, not all surfaces are visible in the estimated direction. Whether or not a particular concave surface is visible depends on viewing direction and the shape of the hole, i.e., where the hole opens and how big it is. Shown in Fig. 4(b), there are five levels of different visibility. At the bottom is the line "totally visible"

since no occlusion occurs in this case. When the cutting line moves up, surface visibility decreases. If the concave surfaces form a closed sphere itself, none of the surfaces is visible with normal devices.

Between the two extremes there are two levels where concave surfaces can be either partially visible or partially invisible. Targeting a surface under occlusion is a topic actively investigated in sensor planning[10]. The only difference is that in a modeling system the visible direction of a target surface is estimated rather than known before hand. It makes the problem even more difficult. But whenever its occluded visibility is confirmed by Eq. 2 and visible direction estimated with MVC, the surfaces can be made visible for the modeling system by using corresponding techniques.

3.4 Adjustment for concave surfaces

With the relation given in Eq. 8, the visible direction of unprocessed surfaces becomes predictable. When concave surfaces are involved, however, a two-sphere method has to be used that combines Eq. 8 for convex surface and Eq. 9 for concave surfaces.

Let a convex sphere stand for the regular part of an object and a concave sphere for the concave surfaces. Since the two types of surfaces form the boundary of the object together, the virtual surface patch represented by $\vec{\mathbf{V}}_i(hole)$ in Eq. 9 can be dealt as a virtual surface patch that makes the object boundary close without the concave surfaces. Therefore, after taking off the virtual surface from Eq. 8, and taking off $\vec{\mathbf{V}}_i(hole)$ from Eq. 9, the sum of mass vectors for the object becomes

$$\sum_{j=0}^{m_i-1} \vec{\mathbf{V}}_{vex}(j) + \sum_{j=0}^{m_h-1} \vec{\mathbf{V}}_{cave}(j) = 0. \qquad (10)$$

Consequently, Eq. 10 can be used to predicate unprocessed surfaces by checking the vector sum of processed surfaces, the same function as provided by Eq. 8 and Eq. 9. Even though the estimated viewing direction reflects the group effect for both convex and concave surfaces, partially visible concave surfaces can be processed with sensor planning techniques when convex surfaces are finished. It is explained further in the following section where a simplified version of Eq. 9 is applied to the sample object for view adjustment.

4 View Estimation for Automatic Modeling

A 3D reconstruction system has the ability to automatically build up object models only when the direction of unprocessed surfaces can be estimated by itself. Therefore, view controlling plays a key role in automatic modeling. With MVC introduced in the previous sections, viewing directions can be predicated by simply checking the closure of corresponding MVC. Following discussion uses experimental examples to describe how MVC is used to predict the direction of unprocessed surfaces with a sample object of convex and concave surfaces. Details are available in [12].

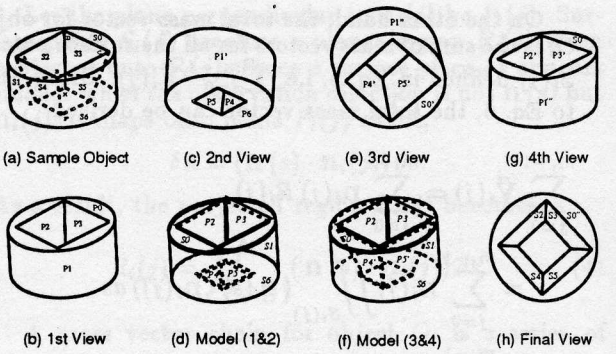


Figure 5: Automatic Modeling

Consider the object in Fig. 5(a). It is a cylinder cut by two planes and has a hole piercing through both its top and bottom surfaces. Its graphical surface model can be represented by its seven boundary surfaces,

$$M_g = \{\text{S0, S1, S2, S3, S4, S5, S6}\}. \qquad (11)$$

The four polygons 'S2', 'S3', 'S4' and 'S5' specify the hole, and they are sets of ordered vertices whose order indicates the polygon orientation.

$$\text{S2} = \langle V_A, V_E, V_H, V_D\rangle \quad \text{S3} = \langle V_D, V_H, V_G, V_C\rangle$$
$$\text{S4} = \langle V_A, V_B, V_F, V_E\rangle \quad \text{S5} = \langle V_B, V_C, V_G, V_F\rangle$$

The eight vertices are,

$$\begin{aligned} V_{A-H} = \{&(-1.7,\ 0.879,\ 0.0),(\ 0.0,\ 1.0,\ 1.75),\\ &(\ 1.798,\ 1.127,\ 0.0),(\ 0.0,\ 1.0,-1.75),\\ &(-1.0,-1.0,0.0),(\ 0.0,-1.0,\ 1.0),\\ &(\ 1.0,-1.0,\ 0.0),(\ 0.0,-1.0,-1.0)\}. \end{aligned}$$

And the other three curve surfaces are given below,

$$\begin{aligned} \text{S0} &= \{(x,\ y,\ z) \mid x^2+z^2=a^2,\\ &\qquad y = 1.0 + 0.071x\} - \text{S0}',\\ \text{S1} &= \{(x,\ y,\ z) \mid x^2+z^2=a^2,\ \text{S6} \le y \le \text{S0}\},\\ \text{S6} &= \{(x,\ y,\ z) \mid x^2+z^2=a^2,\ y=-1.0)\} - \text{S6}'. \end{aligned}$$

4.1 Closure check and view estimation

At a viewing direction, not all the seven surfaces are visible. When a sensory device is placed so that Fig. 5(b) is observed, only four surface patches are extracted, i.e., 'P0', 'P1', 'P2', and 'P3'.

After repositioning the sensor to a second view, which is set to be opposite to first viewpoint, four more surface patches are extracted(Fig. 5(c)). Since the newly obtained surface patch 'P1'' can be merged with 'P1' into the cylinder side 'S1', the updated partial model has seven items as shown in Fig. 5(d),

$$M_{ptl2} = \{\text{S0, S1, P2, P3, P4, P5, S6}\}. \qquad (12)$$

With two opposite views, all the boundary surfaces of a convex object should have been obtained. For instance, suppose the sample object has no hole, then its

boundary is composed by three surface primitives 'S0', 'S1', and 'S6', and they all exist in the partial model 'M_{ptl2}' after the second view. Checking the mass vector chain, whose weight are the areas of one circle '$S6$' and two ellipsis '$S0$' and '$S1'$' on the projection planes.

$$\begin{aligned} |S0| &= 2.0 \times 2.005\pi = 12.598, \\ |S1'| &= 2.0 \times 0.28\pi = 1.759, \\ |S6| &= 2.0^2\pi = 12.566. \end{aligned}$$

After taking surface normals into consideration, the sum of mass vectors becomes,

$$\begin{aligned} \vec{V}_{chain} &= |S0|\mathbf{n}_0 + |S1|\mathbf{n}_1 + |S6|\mathbf{n}_6 \\ &= (0.0, 0.0, 0.0) = 0. \end{aligned}$$

The zero vector $\vec{V}_{chain}$ shows that this object is closed.

For the sample object, however, the weighted vector chain for 'M_{ptl2}' is not zero. Because of the hole, two cutting planes are deducted from the top and bottom surfaces respectively, and four more polygons are in account. The real mass vector chain is given below,

$$\begin{aligned} \vec{V}_{M_{ptl2}} &= (|S0| - |S0'|)\mathbf{n}_0 + |S1|\mathbf{n}_1 + (|S6| - S6')\mathbf{n}_6 \\ &\quad + |P2|\mathbf{n}_2 + |P3|\mathbf{n}_3 + |P4|\mathbf{n}_4 + |P5|\mathbf{n}_5 \\ &= |S0'|\mathbf{n}_0 + |P2|\mathbf{n}_2 + |P3|\mathbf{n}_3 + |P4|\mathbf{n}_4 + \\ &\quad |P5|\mathbf{n}_5 + |S6'|\mathbf{n}_6 = (0.414, -2.878, 0.0). \end{aligned}$$

Since the nonzero vector $\vec{V}_{M_{ptl2}}$ predicates the orientation of unprocessed surfaces, the third viewing direction should be opposite to $\vec{V}_{M_{ptl2}}$, i.e.,

$$\vec{V}_{3rd} = (-0.142, 0.989, 0.0)$$

as a unit vector. After repositioning the sensor to this direction, four surface patches 'S0'', 'P1''', 'P4'', and 'P5'' are extracted from the third view in Fig. 5(e).

Again, the partial model is updated with the new information from the third view. The identical patches 'S0'' and 'P1''' that are either the same as '$S0$' or a part of '$S1$' are no longer in concern. However, polygons 'P4'' and 'P5'' are merged with 'P4' and 'P5' respectively. The updated partial model is shown in Fig. 5(f) and described as,

$$M_{ptl3} = \{\text{S0, S1, P2, P3, P4', P5', S6}\}. \quad (13)$$

4.2 View adjustment

Every time after a new viewpoint is processed, the reconstruction system checks the mass vector chain to see if the building model is closed in space. The vector connecting the tail to the head of the chain is the next viewing direction. Unfortunately, this is not always true especially when the direct viewing direction, the average normal, of the unprocessed surfaces are blocked by other surfaces. In this case, the viewing direction needs to be adjusted.

For the sample object, the direction for the fourth view is obtained from the mass vector chain of 'M_{ptl3}',

$$\vec{V}_{4th} = (-0.319, -0.218, 2.565)$$

and the extracted surface patches are shown in Fig. 5(g), where 'S0''' and 'P1''''' are immediately discarded for the same reason explained before.

Since newly extracted polygons 'P2'' and 'P3'' do not provide any new information because they are just inside '$P2$' and '$P3$', the partial model after the fourth view is the same as the one for the third view,

$$M_{ptl4} = M_{ptl3} = \{\text{S0, S1, P2, P3, P4', P5', S6}\}. \quad (14)$$

The unchanged model leads to the unchanged fifth viewing direction,

$$\vec{V}_{5th} = \vec{V}_{4th} = (-0.319, -0.218, 2.565).$$

If following the normal reconstruction procedure, all the succeeding viewpoints will be the same and the view estimation is in a dead loop.

At this time, surface visibility should be examined to check surface relationship. Since no surface patch occluding any of their visible range, the three out boundary surfaces S0, S1, and S6 are convex surfaces. As had been shown in Eq. 13 that they form a closed chain when added with the top and bottom virtual cutting planes. Therefore, left in the mass vector chain(Eq. 10) are the four hole polygons P2, P3, P4', and P5', plus two virtual planes $S0'_h$ and $S6'_h$.

Checking with Eq. 2, their blocked visible ranges indicate that the four polygons are the occluded surfaces that make the mass vector chain unclosed. Further investigation on their connectivity with shared vertices confirms that they form a connected concave surface. Therefore, Eq. 9 can be applied to adjust the fifth viewing direction. By calculating the group normal, the adjusted viewing direction can be obtained,

$$\begin{aligned} \vec{V}_{5th} &= (-1.0) \cdot (\mathbf{n}_2 + \mathbf{n}_3 + \mathbf{n}'_4 + \mathbf{n}'_5)/4.0 \\ &= (-0.044, 0.866, 0.498). \end{aligned}$$

Visible in the viewpoint specified by $\vec{V}_{5th}$ are five surfaces 'S0''', 'S2', 'S3', 'S4', and 'S5' (Fig. 5(h)). After substituting '$P2$', '$P3$', '$P4'$', and '$P5'$' with their overlapped surface primitives '$S2$', '$S3$', '$S4$', and '$S5$', the final model is acquired, and its zero mass vector chain indicates the closure of the model. The model building procedure terminates with

$$M_{fnl} = M_g = \{\text{S0, S1, S2, S3, S4, S5, S6}\}. \quad (15)$$

4.3 The Algorithm

Shown in Fig. 6 is the algorithm. As a boundary model, an object is composed by its surface primitives. When an image is obtained from the input device, it is segmented to extract surface patches and obtain surface descriptions. These surface patches are then used to build up a partial model for the first view, or

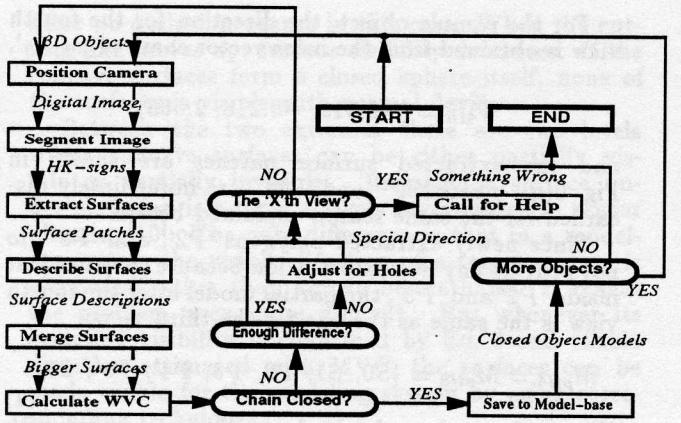


Figure 6: The Algorithm

alternatively, they are merged with processed surfaces to update the partial model.

Meanwhile, the mass vectors of obtained surface patches of the partial model are summed together to check the boundary closure of the model after the first two opposite views. A connected MVC indicates the closure of the current object model, and the system continues on to process other objects. Otherwise, the sensory device is repositioned to obtain unprocessed surfaces whose orientation is suggested by the tail-to-head vector of the mass vector chain.

To avoid the dead loop caused by occlusion, every updated partial model is compared with the previous partial model to see how much new information is obtained. An adjusted viewing direction for hole surfaces is necessary when the two partial models have little difference. View adjustment is made by finding out occluded concave surfaces with Eq. 1 and Eq. 2, and processing them as a group. However, if the same adjustment repeats twice, there must be something wrong that needs the help from the operator. It happens either in case the object has some unobservable surfaces or the system can not determine the right viewpoint, e.g., the surfaces in a partially visible hole (Fig. 4) are not totally visible under any particular direction.

View adjustment is also necessary when object surfaces observed in the first two opposite viewing directions happen to form a closed one/two-dimensional MVC. For instance, if a cube is observed in the two directions that the system extracts only two opposite sides, the MVC for this object model is one-dimensionally closed because the two surface are equal in size but opposite in direction even though the object model is not closed in space yet. As a result, a simple closure check of the vector chain is not enough to decide if an object model is finished. Adjustments to other dimensions are required if the MVC is closed in a dimension lower than three.

5 Conclusion

In this paper, an automatic modeling mechanism has been proposed. Using the relationship of mass vector chains with surface visibility and spatial closure of boundary surfaces, a reconstruction system is able to predicate the viewing direction of unprocessed surfaces, and even make adjustment for occluded surfaces. A simple object is used to explain the fundamental operations of automatic modeling. More objects had been tested in [12], and further research is being conducted to investigate its usage in practical applications such as automatic mechanical assembling.

6 Acknowledgment

I would like to express my thanks to Dr. Wayne Davis, my former supervisor, for his inspiration and help during my Ph.D study.

References

[1] J. Aloimonos and C. M. Brown. Robust computation of intrinsic images from multiple cues. In *Advances in Computer Vision*, volume 1, chapter 1.2, pages 115–164. Lawrence Erlbaum Associates, Hillside, N.J., 1988.

[2] Paul. J. Besl. *Surfaces in Range Image Understanding.* Springer-Verlap, New York, 1988.

[3] Ting-Jun Fan. *Describing and Recognizing 3D Objects Using Surface Properties.* Springer-Verlag, 1990.

[4] Robert B. Fisher. *From Surfaces to Objects.* John Wiley & Sons, 1989.

[5] W. B. Green. *Digital Image Processing: A Systems Approach.* Van Nostrand Reinhold, New York, second edition, 1989.

[6] B. K. Horn. Extended Gaussian images. *Proc. IEEE*, 72:1656–1678, December 1984.

[7] J. Lansdown. Methods of presentation of 3D structures. In *Geometric Modeling and Computer Graphics: Techniques and Applications*, pages 223–232. Technical Press, 1987.

[8] M. Sabin. Geometric modeling. In *Geometric Modeling and Computer Graphics: Techniques and Applications*, pages 50–74. Technical Press, 1987.

[9] Y. Shirai. *Three-Dimensional Computer Vision.* Springer-Verlag, 1987.

[10] Roger Y. Tsai and Kostantinos Tarabanis. Occlusion-free sensor placement planning. In *Machine Vision for Three-Dimensional Scenes*, pages 301–339. Academic Press, Inc., 1990.

[11] J. D. Warren and S. Lodha. Free-form quadric surface patches. In *Curves and Surfaces in Computer Vision and Graphics*, volume 1251, pages 30–40, 1990.

[12] Xiaobu Yuan. *3D Reconstruction as An Automatic Modeling System.* Ph.D thesis, Department of Computing Science, University of Alberta, Canada, 1992.