Visual Hull Construction, Alignment and Refinement Across Time

German K.M. Cheung
Robotics Institute
Carnegie Mellon University
Pittsburgh, PA 15213
german+@cs.cmu.edu

CMU-RI-TR-02-05

Thesis Proposal

December 2001

Abstract

Visual Hull (VH) construction is a popular method of shape estimation. The method, also known as Shape from Silhouette (SFS), approximates shape of an object from multiple silhouette images by constructing an upper bound of the shape called the Visual Hull. SFS is used in many applications such as non-invasive 3D object digitization, 3D object recognition and more recently human motion tracking and analysis. Though SFS is straightforward to implement and easy to use, it has several limitations. Most existing SFS methods are too slow for real-time applications and the estimated shape is sensitive to silhouette noise and camera calibration errors. Moreover, VH is only a conservative approximation of the actual shape of the object and the approximation can be very coarse when there are only a few cameras. In my thesis, I propose to investigate some of these shortcomings and suggest solutions to overcome them. First, a voxel-based real-time SFS algorithm called SPOT is proposed and its behavior under noisy silhouette images is analyzed. Secondly, the conservative nature of SFS is improved by incorporating silhouette images across time. The improvement is achieved by first estimating the rigid motions between visual hulls formed at different time instants (visual hull alignment) and then combining them (visual hull refinement) to get a tighter bound of the object's shape. The ambiguity issue of visual hull alignment is identified and addressed. This study is presented here in 2D. In the thesis I proposed to extend it to 3D objects. Thirdly, an algorithm that uses color consistency to resolve alignment ambiguity problem is proposed. This algorithm constructs entities called bounding edges of the VH and utilizes the Fundamental Theorem of Visual Hull to find points on the surface of the object. Using an idea similar to image alignment, these surface points are used to find the motion between two visual hulls. Once the rigid motion across time is known, all the silhouette images are treated as being captured at the same time instant and the shape of the object is refined. The algorithm is validated by both synthetic and real data. Finally the advantages and disadvantages of representing VH by three different ways : bounding cones intersection, voxels and bounding edges are discussed.

1. Introduction

3D shape estimation has always been a very important and active research topic in the field of computer vision. A large number of algorithms have been proposed in the last two decades. These range from estimating the shape of static rigid objects from a single calibrated image, to estimating structure and motion of non-rigidly moving objects from multiple images [36] [1] [28] [16] [41] [15] [33] [35] [37] [8] [22]. Among these, there is a category of methods called Visual Hull (VH) construction or Shape From Silhouettes (SFS). These methods approximate the shape of an object by using its silhouettes. Since its first introduction by Baumgart in [3], different variations of SFS methods, VH representations and applications have been proposed. For example, Aggarwal et. al. suggested using voxel as a representation of visual hulls [28] [21]. Potmesil used an octree data structure to speed up VH construction [34]. Pujari derived optimal positions and directions to take silhouette images for building 3D volume models [38]. Szeliski built a non-invasive 3D digitizer using a turntable and a single camera with SFS as the reconstruction method [39]. Laurentini studied the theoretical properties of VHs of 3D polyhedral objects [23] [24] and curved objects [26]. De Bonet and Viola extended the idea of voxel reconstruction to transparent objects by introducing the concept of Roxels [4]. Buehler et. al. used VH as the geometric basis for image-based rendering [10]. Cipolla et. al. recovered the camera positions and orientations from silhouettes under circular motions [45] [30]. Ponce et. al. studied the exact VH of objects with smooth surfaces. In the past five years, due to advances in extracting silhouette (background subtraction) [17] [14] from images and video sequences, a large number of researchers have applied SFS to human related applications [31] [20] [13] [5]. In other words, SFS has become a standard and popular method of shape estimation.

Estimating shape using SFS has many advantages. Silhouettes are readily and easily obtainable, especially in indoor environment where the cameras are static and there are few moving shadows. The implementation of most SFS methods is relatively straightforward, especially when compared to other shape estimation methods such as multi-baseline stereo [33] or space carving [22]. Visual hull constructed from SFS is upper bound of the object of interest. This inherently conservative property is particularly useful in applications such as obstacle avoidance in robot manipulation and visibility analysis in navigation where an upper bound on the shape of the object is preferred to a lower bound.

On the other hand, SFS suffers from a number of limitations and inadequacies. Existing SFS methods involve time-consuming testing steps which hinder their use in real-time applications. SFS is also sensitive to errors in silhouette extraction and camera calibration, making it less desirable with noisy images. Moreover, VH obtained from SFS is only an approximation of the actual object shape. The approximation can be very coarse when there are only a few cameras, posing a disadvantage for SFS in applications such as detailed shape acquisition and realistic re-rendering of objects. The goal of this proposal is to investigate some of these short-comings and suggest solutions for them.

In order to use SFS in real-time applications, the computational bottleneck of existing methods has to be identified and replaced by faster alternatives. The behavior of the alternatives under noisy silhouette images has to be studied in order to maintain reasonable shape estimates. I will do both by developing a fast SFS algorithm called **SPOT** (Sparse Pixel Occupancy Test) which takes into account the effect of noisy silhouette images. A real-time system for reconstructing 3D volumetric models of people performing arbitrary motion is built based on this fast algorithm.

The shape inexactness of existing SFS methods is due to two reasons : (1) the inherent conservative nature of SFS principle and (2) the approximate representations of VHs. The visual hull is the theoretical limit of shape information that can be deduced from a set of silhouette images and hence it will always be an approximation to the actual object. The discrete voxel representation used by most SFS methods is only a coarse approximation to the actual visual hull. This approximation further reduces the shape estimation accuracy of SFS.

The shape inaccuracy problem due to the theoretical limit of SFS can be solved by increasing the number of *distinct* silhouette images. This can be done in two ways : across space or across time. By across space, we mean increasing the number of physical cameras. This solution may not be feasible in many practical situations due to financial or physical limitations. For moving objects, an alternative way to increase the number of *effective* cameras is by combining silhouette information across time. In other words, if we estimate the motion of the object between several different time instants, we can then combine the information at these time instants to refine the shape. These two tasks are termed visual hull alignment and refinement.

It can be shown that using silhouette images (or VHs) alone for alignment is inherently ambiguous. Generally there exist an infinite number of *consistent* rigid motions between two visual hulls. I will provide a full account of the ambiguity in this "silhouettes-only" alignment problem. Necessary and sufficient conditions for consistently aligning two visual hulls will be derived. In this proposal, I will only consider 2D case. I propose to extend the analysis to 3D in my thesis. The conditions derived are not only useful in aligning visual hulls, they also have potential use in improving the robustness of SFS against camera calibration errors at a single time instant.

In order to get unambiguous alignment between two visual hulls, other information besides the silhouettes is needed. I will describe an algorithm which uses color consistency as a measure for estimating the alignment between two visual hulls. A simple procedure is then used to refine the visual hulls across time to give a tighter bound for the object. A novel visual hull representation called bounding edges will emerge as a by-product of the algorithm. I will point out the advantages and disadvantages of using bounding edges to represent VH compared to other representations.

The remainder of this proposal is organized as follows. In Section 2, I introduce the notion of visual hulls and present the most popular SFS method to compute them called voxel classification. My fast SFS algorithm (**SPOT**) and real-time human model reconstruction system is described in Section 3. In Section 4, I focus on deriving the color consistency algorithm for aligning two visual hulls, followed by the refinement procedure. Section 5 is devoted to the theoretical analysis of the visual hull alignment ambiguity and in Section 6, I discuss the representation issues of visual hulls. I conclude by providing a brief summary, related work, expected contributions and thesis schedule in Section 7.

2. Background

2.1. The Shape From Silhouette Scenario

Suppose there are K cameras positioned around a rigid 3D object $\mathcal{O}$. Let $\{\mathcal{S}_j^k;\ \ k = 1;\cdots\cdots, K\}$ be the set of silhouette images of the object $\mathcal{O}$ obtained from the K cameras at time t_j. It is assumed that the cameras are calibrated with $\mathbf{\Pi}^k() : I\!R^3 \to I\!R^2$ and $\mathcal{C}^k$ being the perspective projection function and the center of camera k respectively. Hence $\boldsymbol{p} = \mathbf{\Pi}^k(\boldsymbol{P})$ are the 2D image coordinates of a 3D point $\boldsymbol{P}$ in the k^{th} image. As a slight extension of this notation, $\mathbf{\Pi}^k(\boldsymbol{\mathcal{A}})$ represents the projection of a volume $\boldsymbol{\mathcal{A}}$ onto the image plane of camera k. An example scenario is depicted in Figure 1 with the head being the object surrounded by four cameras.

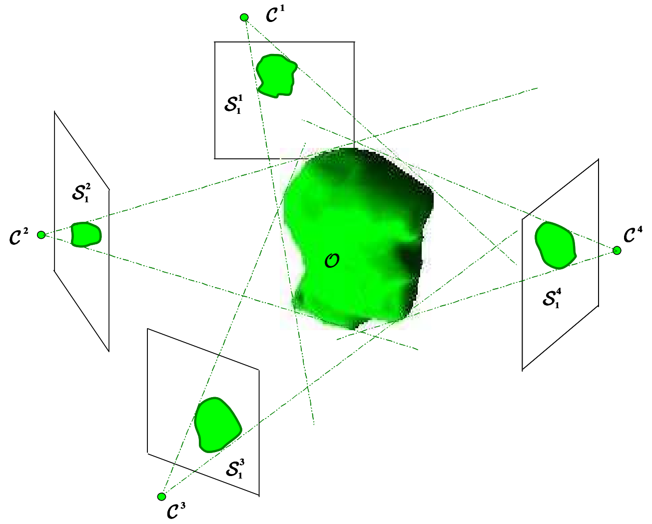


Figure 1. An example SFS scenario. An object $\mathcal{O}$ is surrounded by four cameras. The silhouette images at time t_1 and camera centers are represented by $\{\mathcal{S}_1^k\}$ and $\{\mathcal{C}^k\}$ respectively.

2.2. Visual Hull

Suppose we are given a set of K silhouette images $\{\mathcal{S}_j^k\}$ and projection functions $\{\mathbf{\Pi}^k\}$. A volume $\boldsymbol{\mathcal{A}}$ is said to *explain exactly* $\{\mathcal{S}_j^k\}$ if and only if its projection onto the k^{th} image plane coincides exactly with the silhouette image $\mathcal{S}_j^k$ for all $k \in \{1, \cdots\cdots, K\}$, i.e. $\mathbf{\Pi}^k(\boldsymbol{\mathcal{A}}) = \mathcal{S}_j^k$. If there exists at least one entity which explains the silhouette images exactly, we say the set of silhouette images is consistent, otherwise we call it inconsistent. Normally a set of silhouette images obtained from an object is consistent, unless there are camera calibration errors or silhouette image noise. Now we are ready to give a formal definition of the visual hull $\mathcal{H}_j$ with respect to $\{\mathcal{S}_j^k\}$.

Definition of Visual Hull
The visual hull $\mathcal{H}_j$ with respect to a set of consistent silhouette images $\{\mathcal{S}_j^k\}$ is defined to be the largest possible volume which *exactly explains* $\{\mathcal{S}_j^k\}$ for all $k = 1, \cdots\cdots, K$.

Generally for a consistent set of silhouette images $\{\mathcal{S}_j^k\}$, there exists an infinite number of volumes (including the object $\mathcal{O}$ itself) that *exactly explain* the silhouettes. Visual hull $\mathcal{H}_j$ is defined to be the largest one among these volumes. It can be shown that $\mathcal{H}_j$ equals the intersection of the bounding cones defined by the silhouettes and the camera centers [23]. The VH provides an upper bound of the object $\mathcal{O}$ and the bound gets tighter if we increase the number of distinct silhouette images. Asymptotically if we have an infinite number of every possible silhouette images of a convex object, the visual hull is exactly equal to the object.

2.3. Visual Hull Construction

2.3.1. Computing visual hulls by intersecting bounding cones

For a consistent set of silhouette images, the most direct way to construct a visual hull is by intersecting the bounding cones formed by the silhouette images $\{\mathcal{S}_j^k\}$ and the corresponding camera centers $\{\mathcal{C}^k\}$. The resulting visual hull consists of surface patches. For illustration purpose, an example in two-dimension is given in Figure 2(a) in which the visual hull is constructed by intersecting the 2D bounding wedges. Although simple and obvious in 2D, this direct approach is not used in practice for 3D. This is because there is no easy way of expressing the surface patches resulted from the cones intersection effectively by simple geometrical attributes. The computational complexity and numerical instability of intersecting surfaces with lines and planes in 3D are also reasons why the cones intersection method is not used. Two 3D bounding cones are illustrated in Figure 3 which shows that it is difficult to express, represent or even visualize the surface patches of the visual hull.

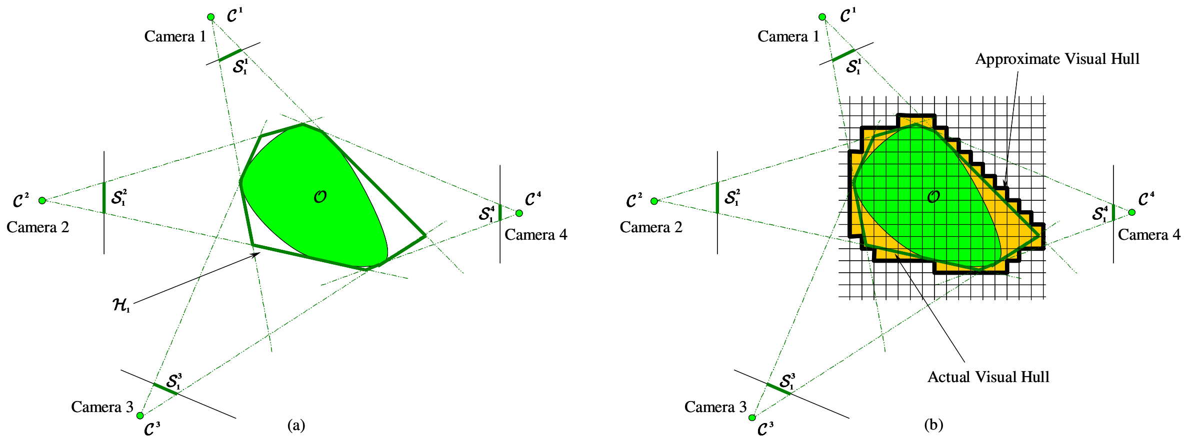


Figure 2. A two dimensional example of constructing visual hull $\mathcal{H}_1$ from silhouette set $\{\mathcal{S}_1^k\}$ and camera centers $\{\mathcal{C}^k\}$ (a) by direct intersection of bounding wedges, (b) by voxel-based approximation. The shaded region represents the approximate visual hull while the polygon denotes the true one. The former is significantly larger than the latter.

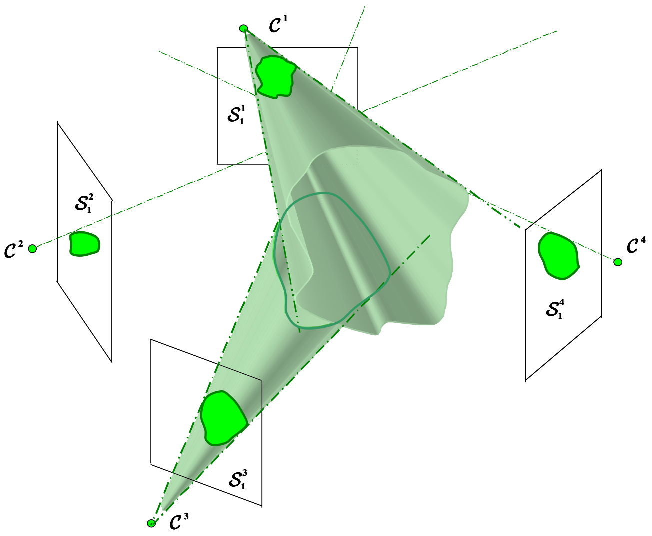


Figure 3. Example of intersecting two 3D bounding cones formed from two silhouette images. It can be seen that it is difficult to express, represent or even visualize the surface patches of the resultant visual hull. In 2D similar problem does not exist, see Figure 2(a).

2.3.2. Construction of approximate visual hull by voxel classification

Researchers have been proposing other effective ways to represent and construct 3D visual hulls instead of intersecting the bounding cones. Most of the methods suggested so far are discrete and voxel-based. A simple version of these methods is summarized as follows :

Algorithm 1 : Standard Voxel-based Shape from Silhouette

1. Divide the space of interest into $N \times N \times N$ discrete voxels $v_n, \;\; n = 1, \cdots\cdots N^3$.

2. Initialize all the N^3 voxels as *inside* voxels.

3. Perform :

   For $n = 1$ to N^3 **do** {
       For $k = 1$ to K **do** {
           (a) Project the voxel v_n with $\mathbf{\Pi}^k()$ onto the k^{th} image plane;
           (b) **If** the projected area $\mathbf{\Pi}^k(v_n)$ lies totally outside $\boldsymbol{\mathcal{S}}_j^k$
               then classify v_n as *outside* voxel;
       }
   }

4. The approximate visual hull $\hat{\boldsymbol{\mathcal{H}}}_j$ is represented by the union of all the *inside* voxels.

The basic idea is to divide the space of interest into discrete voxels and classify them into two categories :*inside* and *outside*. The union of all the *inside* voxels is an approximation of the visual hull. For a voxel to be classified as *inside*, its projection (step 3(a)) on each and every one of the K image planes has to be inside or partially overlap with the corresponding silhouette images. If the projection of the voxel is totally outside any of the silhouette images, it is classified as *outside*. Figure 2(b) gives a 2D example of this voxel-based method. The area of interest is divided into 16 by 16 squares. The convex polygon represents the true visual hull while the shaded region denotes the approximate visual hull obtained using the 2D version of Algorithm 1. The approximate 2D VH is significantly larger than the true one. This is one of the disadvantages of using discrete voxels to represent visual hulls.

3. Real-Time Construction of Visual Hulls From Noisy Silhouettes

3.1. Speed and Accuracy of Voxel-based Visual Hull Construction Method

Visual hull construction using the voxel representation is by far the most popular SFS method used by researchers because it is easily implementable and gives reasonable results in applications where approximate but complete shape information of the object is required. The speed of the voxel-based SFS method described in Algorithm 1 depends heavily on two procedures : (1) voxel projection (step 3(a)) and (2) silhouette overlap testing (step 3(b)). These two procedures, together with the quality of the silhouette images, also determine the accuracy of the estimated VH. Silhouette images are always noisy. Figure 4 shows a foreground image and its noisy silhouette image obtained using the real-time background subtraction method described in [11]. Although post-processing steps, such as morphological operations and connected component analysis, can be applied to clean up the silhouette, they are too slow to be used in real time applications. The aim of this section is to propose fast implementations of procedures (1) and (2) while understanding the effect of silhouette noise on the implementations. To characterize the quality of the silhouette images, hereafter let ξ be the probability that during silhouette extraction, a non-silhouette pixel is wrongly marked as a silhouette pixel. Likewise, let η represent the probability that a silhouette pixel is wrongly marked as a non-silhouette pixel. The noise is assumed to be independent between pixels. There are two ways of determining ξ and η : theoretically by assuming certain error probability distribution functions or experimentally by checking wrongly marked pixels in a large set of noisy silhouettes. For example, ξ and η are found experimentally to be 0.021 and 0.043 for the real-time background subtraction method described in [11].

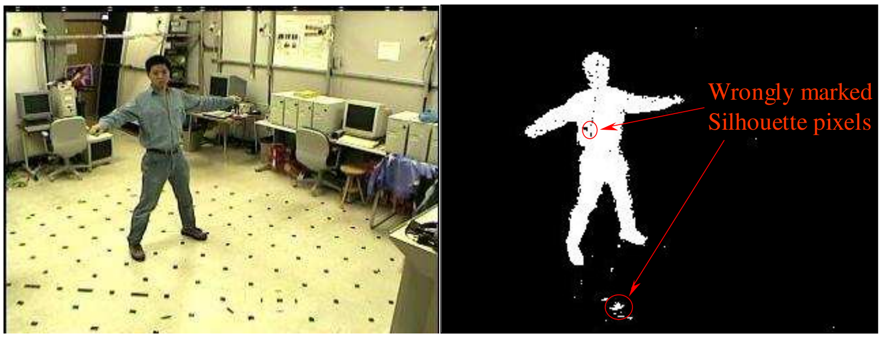


Figure 4. An example of a foreground image and its silhouette extracted by real-time background subtraction method. There are wrongly marked pixels in the silhouette image due to noisy original and background images.

One way of finding the projected voxel $\mathbf{\Pi}^k(v_n)$ is to project the eight vertices of v_n onto the k^{th} image plane and compute the convex hull of the projected points. Suppose on the average, there are Z pixels inside the convex hull. A straightforward way to implement silhouette overlapping is to check all the Z pixels and if at least Z_ε of them lie inside the silhouette image $\boldsymbol{\mathcal{S}}_j^k$, the voxel is said to be inside $\boldsymbol{\mathcal{S}}_j^k$. Due to silhouette noise, this implementation will cause voxel misclassification. There are two types of voxel misclassification : False Acceptance (FA) which means an *outside* voxel is misclassified as *inside* and

False Rejection (FR) which means an *inside* voxel is misclassified as *outside*. The probabilities $P(FR)$ and $P(FA)$ depend on ξ, η, Z and Z_ε. For voxels that are either totally outside or totally inside the silhouette (i.e. non-boundary voxels), $P(FR)$ and $P(FA)$ are found to be

I. False Acceptance :

$$P(FA) = \left[\sum_{i=Z_\varepsilon}^{Z} \binom{Z}{i} \xi^i (1-\xi)^{Z-i} \right]^K . \tag{1}$$

II. False Rejection :

$$P(FR) = p \sum_{j=0}^{K-1} (1-p)^j ; \qquad p = \sum_{i=Z-Z_\varepsilon+1}^{Z} \binom{Z}{i} \eta^i (1-\eta)^{Z-i} . \tag{2}$$

The exponential of K in equation (1) is due to the fact that an *outside* voxel has to be misclassified as *inside* in *all* the K images for a FA to happen. In equation (2), p is the probability that an *inside* voxel is classified as *outside* in *one* image. The summation in j comes from the reason that an inside voxel is misclassified as outside once it is misclassified in one image and the rest of the images are not tested. Note that these equations are applicable only to totally inside or totally outside voxels. With slight modifications, similar analysis can be applied to boundary voxels although they are not considered here as the number of them are relatively fewer than the number of totally inside or totally outside voxels.

For fixed Z, ξ and η, $P(FR)$ increases with Z_ε while $P(FA)$ decreases when Z_ε increases. Using the values ξ = 0.021 and η = 0.043 (the values determined experimentally), graphs of $\log(P(FR))$ and $\log(P(FA))$ versus Z_ε with different values of Z are shown in Figure 5 (a) and (b) respectively. The optimal Z_ε for a particular Z is chosen by considering the total error probability $P(FA) + P(FR)$. The graph of $\log(P(FA) + P(FR))$ against Z_ε with different Z is plotted in Figure 5(c). Clearly, different values of Z have different optimal Z_ε. For example the optimal Z_ε is 1 for $Z = 2$ while the optimal Z_ε is 7 when Z is 30. The graphs of optimal Z_ε and optimal $\log(P(FA) + P(FR))$ against Z are plotted in Figure 5(d)(e). These graphs are useful in determining the optimal silhouette testing rule and the corresponding error probabilities.

Besides serving as guidelines for choosing optimal Z_ε, the graphs in Figure 5(d)(e) also give us insight into deciding voxel size of the system. Intuitively the smaller the voxel size, the better the shape approximation (see Figure 2(b) as an example) because the discretization error is smaller. Niem verified this conjecture in [32] by analyzing effect of voxel size on shape reconstruction error *without considering noisy silhouettes*. However, my analysis (Figure 5(e)) shows that when there is noise in the silhouettes, the smaller the voxel size, the smaller the number of projected pixel Z which implies the larger the misclassification probabilities. Therefore to pick an optimal voxel size for a VH construction system, there is a tradeoff between lowering discretization errors (favoring smaller voxels) and minimizing misclassification probabilities (favoring larger voxels).

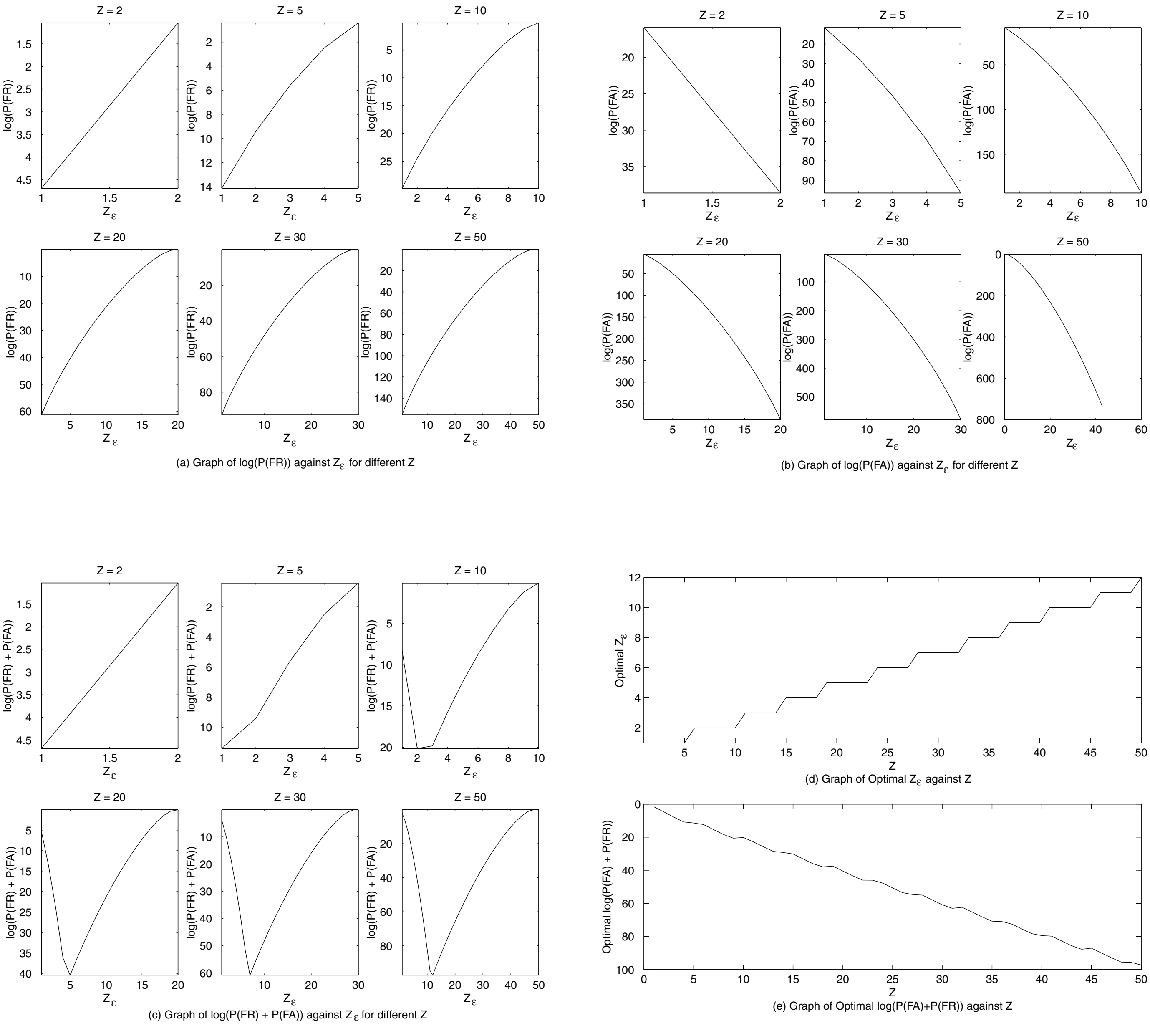


Figure 5. (a) $\log(P(FR))$ **vs.** Z_ε**, (b)** $\log(P(FA))$ **vs.** Z_ε**, (c)** $\log(P(FA)+P(FR))$ **vs.** Z_ε**, (d) Optimal** Z_ε **vs.** Z**, (e) Optimal** $\log(P(FA)+P(FR))$ **vs.** Z**.**

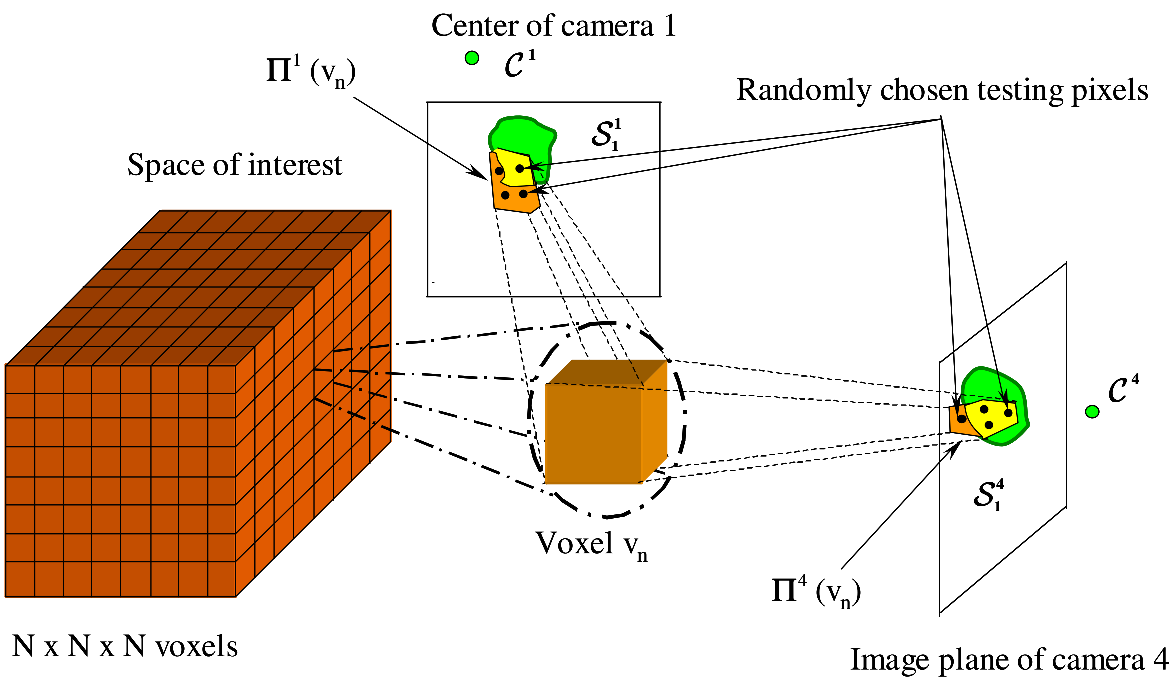


Figure 6. The modified voxel-based visual hull construction using SPOT.

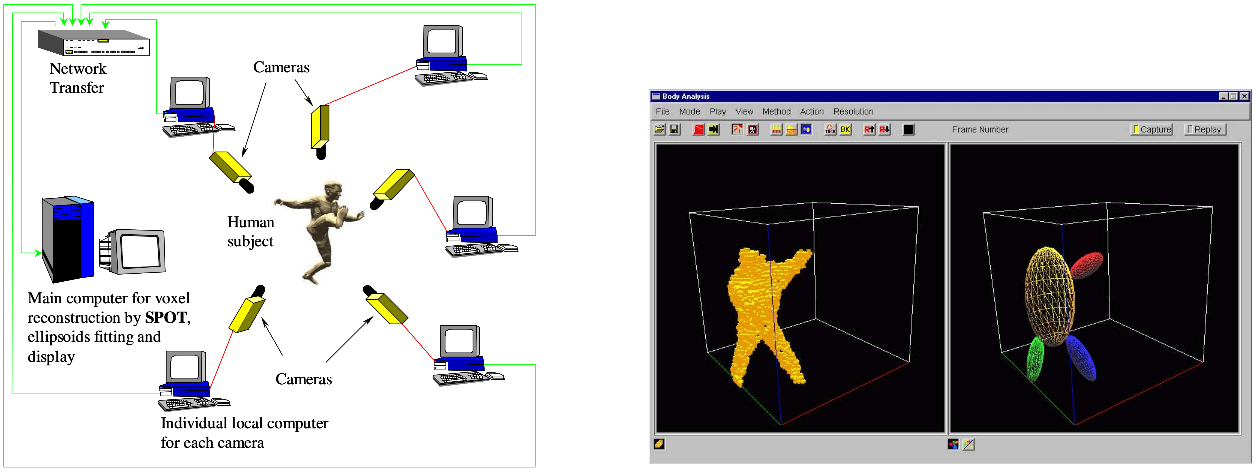
3.2. Fast Voxel-based Visual Hull Construction Algorithm - SPOT

To apply the straightforward implementations of procedures (1) voxel projection and (2) silhouette overlap testing, we need N^3Z amount of memory (assume we precompute the projected voxel convex hull and store the pixel locations in a lookup table) and N^3Z testing operations for a volume of $N \times N \times N$ voxels. The high memory requirement and number of testings make this straightforward approach too slow and expensive for real time applications. Here I propose faster and more efficient implementations. For each image and each voxel, instead of testing all Z pixels, Q random pixels are chosen within the Z-pixel convex hull. If at least Q_ε of these chosen pixels lie inside the silhouette image $\mathcal{S}^k_j$, the voxel is said to be inside $\mathcal{S}^k_j$. I named this approach **S**parse **P**ixel **O**ccupancy **T**est **(SPOT)** and the idea is illustrated in Figure 6.

There are two advantages of using **SPOT**. It is $\frac{Z}{Q}$ times faster and the memory requirement is $\frac{Z}{Q}$ times less than the straightforward approach. The downside is the misclassification probabilities also increase when we reduce the number of testing pixels. The question remains is how do we choose Q and Q_ε accordingly? Since we have assumed the pixels are independent of each other and the Q points are chosen randomly, the notion of testing Q out of Z pixels is theoretically equivalent to testing a projected area with Q pixels. In other words, equations (1), (2) and the graphs in Figure 5 are valid for analyzing **SPOT** if we replace Z by Q. The choice of Q is a compromise between speed, memory and accuracy. In real-time applications where speed is more important than quality, a low value of Q is used to trade accuracy for speed. For example, if we use $Q = 5$ for a 10-pixel projected voxel (i.e. $Z = 10$), we gain a factor of 2 in both speed and memory but the total misclassification probability $P(FA) + P(FR)$ also increase from 1.7e-9 to 1.1e-5 (see Figure 5(e)). Once Q is fixed, Q_ε is chosen according to Figure 5(d).

3.3. A Real-time System for Robust 3D Voxel Reconstruction of Human Motions

Using **SPOT**, I have built a real time system to construct 3D voxel models of human motions [11]. The system consists of five synchronized cameras placed evenly around the moving human subject. Each camera is connected to its individual local computer. The camera captures images continuously and a real-time shadow-removing background subtraction method (details in [11]) is applied in the camera's own computer to extract the silhouettes. The synchronized silhouette images are then transferred, via local network, from the five local computers to a main host computer where **SPOT**-modified SFS is used to build the voxel model of the person. Six ellipsoids (as analog to the head, body, two arms and two legs of a human) are fit to the voxel data to show that the system has great potential in real time human limb tracking applications. The fitting algorithm is described in more details in [11]. The system architecture is illustrated in Figure 7(a). Using a simple and user-friendly interface, the user can display and observe, in real time and from any view-point, the 3D models and the fit ellipsoids of the moving human body. Figure 7(b) is a snapshot of the interface with the voxel models being shown on the left and the fitted ellipsoids on the right.



(a) (b)

Figure 7. (a) The system architecture of the real-time human motion model reconstruction system. (b) Screen shot of the user interface.

The following summarizes various aspects and parameters of the system.

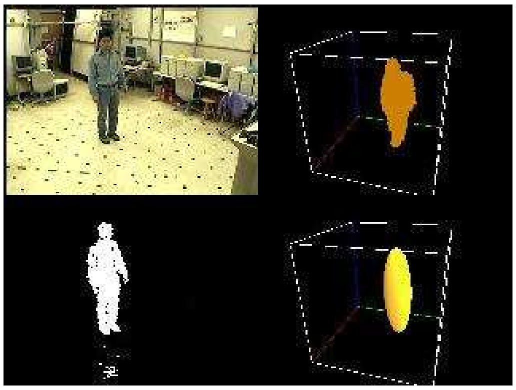
1. Five cameras are used and each of them is connected to a 266 MHz PC. They are calibrated by planar calibration patterns with the method described in [46]. The main computer is equipped with dual Pentium II 400 MHz processors. The computers are connected together by a 100Mb/s hub.
2. The space of interest is a cube of size 2m $\times$ 2m $\times$ 2m divided into 64 $\times$ 64 $\times$ 64 (i.e. N = 64) voxels with a resolution of about 3cm for each voxel.

3. The images are captured at a resolution of 320x240 pixels. The silhouette extraction method described in [11] gives us silhouette error probabilities of η = 0.043 (the probability that a silhouette pixel is wrongly marked as a non-silhouette pixel) and ξ = 0.021 (the probability that a non-silhouette pixel is wrongly marked as a silhouette pixel).

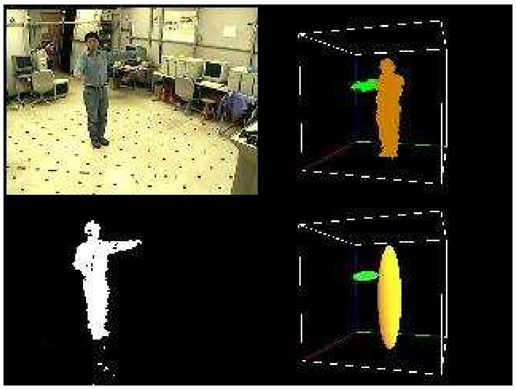
4. The average number of pixels in a projected voxel is 10 (i.e. $Z \approx$ 10). To obtain the desired speed, only two pixels (i.e. Q = 2) are chosen for each voxel in the silhouette overlap test. The optimal Q_ε is 1 according to Figure 5(d). This corresponds to a total misclassification probability $P(FA) + P(FR)$ of 0.0092, compared to a value of 1.78e-9 if we set $Q = 10$.

5. With the display function turned off, the system outputs results of higher than 15 frames per second.

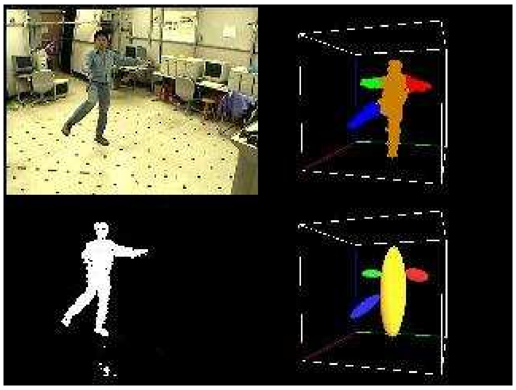
Figure 8 shows four representative frames extracted from a movie clip of the system made in real time. Within each frame, the upper left picture is the run-time image captured by one of the five cameras and the lower left picture depicts its silhouette generated by the background subtraction method. The upper right picture shows the reconstructed voxels of the person using **SPOT** while the lower right picture represents the fit ellipsoids. There are two frames delay between the run-time image/silhouette computation and the reconstructed voxels/ellipsoids due to the pipeline processing of the system.



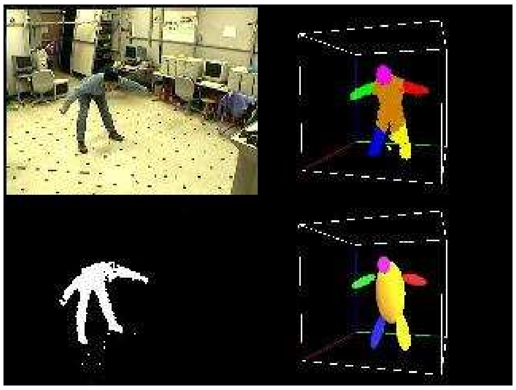
(a)



(b)



(c)



(d)

Figure 8. Four representative frames, (a) frame 59, (b) frame 109, (c) frame 139 (d). frame 299.

4. Visual Hull Alignment and Refinement

4.1. Ambiguity in Visual Hull Alignment

The only way to improve the shape estimate beyond the theoretical limit of SFS is to increase the number of distinct silhouettes, either across space or across time. Across space means increasing the number of physical cameras in the system. Across time means combining information of silhouette images across frames. The approach of increasing the number of physical cameras not only suffers from setup and financial limitations, it is also not as effective as integrating silhouette information across time. For example, for a system with K cameras, if we are able to combine the silhouette information across N frames, we have an effective number of NK cameras. This is equivalent to adding an additional $(N-1)K$ physical cameras to the system.

To combine silhouette images across time, the rigid motion of the object between frames is required. The motion can be known in advance [39], assumed nearly circular [29] [45] or completely arbitrary and unknown. In this proposal, I assume the motion of the object between frames is rigid but unknown. The problem of visual hull alignment is to estimate the unknown rigid motion from the silhouette images and the corresponding visual hulls as stated below.

Visual Hulls Alignment by Silhouette Images :
Suppose we are given two sets of silhouette images $\{\mathcal{S}_j^k; \quad k = 1; \cdots\cdots, K; \quad j = 1,2\}$ of an object $\mathcal{O}$ from K cameras at two different time instants t_1, t_2. Without loss of generality, assume the first set of images $\{\mathcal{S}_1^k\}$ are taken when the object is at position and orientation of $(\boldsymbol{I}, \underline{0})$ while the second image set $\{\mathcal{S}_2^k\}$ is taken when the object is at $(\boldsymbol{R}, \boldsymbol{t})$. The problem of visual hull alignment is to find $(\boldsymbol{R}, \boldsymbol{t})$ from the silhouette images $\{\mathcal{S}_j^k\}$ and the Visual Hulls $\mathcal{H}_j$ such that there exists an object $\mathcal{O}$ which exactly explains the silhouettes at both times t_j by $\mathcal{H}_j; \ \ j = 1,2$, and the relative position and orientation of $\mathcal{O}$ is related by $(\boldsymbol{R}, \boldsymbol{t})$ from t_1 to t_2.

The problem of visual hull alignment using silhouette images alone is inherently ambiguous. There may exist more than one set of $(\boldsymbol{R}, \boldsymbol{t})$ which satisfy the alignment criterion. A 2D example is shown in Figure 9. In the figure, both (a) and (b) have the same silhouette image sets (and hence the same visual hulls) at times t_1 and t_2. However, in (a), the silhouettes are formed by a curvy object with pure translation as motion between t_1 and t_2, while in (b), the silhouettes are caused by a polygonal object with motion of both rotation (200 degrees) and translation between t_1 and t_2. The motion ambiguity is closely related to the indeterminacy in shape. I will devote Section 5 to a full analysis of the theoretical aspects of the alignment issue. In the mean time, a practical solution is devised to break the ambiguity by using color information. In the coming discussions, I will present ways to incorporate color information into the framework of SFS. I will first introduce the Fundamental Theorem of Visual Hull which captures the essence of the SFS principle. Then I will define an entity called bounding edge. By considering color consistency along the bounding edges and among images across time, an algorithm is developed to align visual hulls at two time instants.

4.2. Fundamental Theorem of Visual Hull and Bounding Edges

Although SFS has been used by researchers for years, most of them concentrated on the property that the VH is an upper bound on the object, i.e. $\mathcal{O} \subseteq \mathcal{H}$. An equally important property that has been

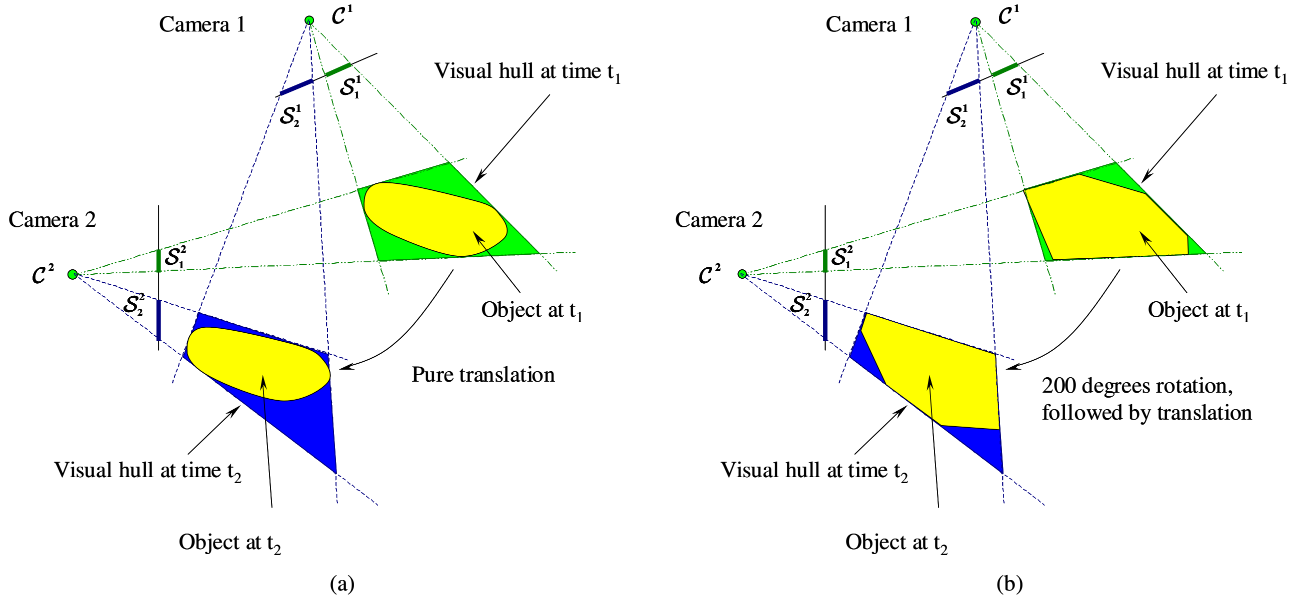


Figure 9. A 2D example showing the ambiguity issues of aligning visual hulls. Both cases in (a) and (b) have the same silhouette image sets at times t_1 and t_2 but they are formed from two different objects with different motion.

overlooked by most people is that VH actually touches the object at certain points. The essence of this property is captured by the following theorem.

Fundamental Theorem of Visual Hull (Single Silhouette Image)
The ray $\mathcal{R}$ joining the camera center and any boundary point of any silhouette touches the object (that forms the silhouette) *at at least one point*.

This theorem (FTVH), follows directly from how the silhouette is formed (Proof omitted). FTVH is a key source of information from the silhouette image about the object. In fact, the direct visual hull computation by intersecting bounding cones comes from this theorem. Since the above theorem refers to a single silhouette image, it is of little practical use as the ray $\mathcal{R}$ is of infinite length. To extend the theorem to multiple silhouette images, an entity called bounding edge is used and defined as follows.

Definition of Bounding Edge
Let $\mathcal{U}_j^{i,k}$ be a point on the boundary of the silhouette image $\mathcal{S}_j^k$. By projecting $\mathcal{U}_j^{i,k}$ into the 3D space through the camera center $\mathcal{C}^k$, we get a ray which we denote by $\mathcal{R}_j^{i,k}$. Now a bounding edge $\mathcal{E}_j^i$ is defined to be the part of $\mathcal{R}_j^{i,k}$ such that the projection of $\mathcal{E}_j^i$ on the l^{th} image plane lie completely inside $\mathcal{S}_j^l$ for all $l \in \{1, \cdots\cdots, K\}$. Mathematically the condition is expressed as

$$\mathcal{E}_j^i \subset \mathcal{R}_j^{i,k} \quad \text{and} \quad \mathbf{\Pi}^l(\mathcal{E}_j^i) \quad \subset \quad \mathcal{S}_j^l \quad \forall \quad l \in \{1, \cdots\cdots, K\}. \tag{3}$$

An example illustrating the definition of a bounding edge is shown in Figure 10(a). The bounding edge is obtained by first projecting the ray $\mathcal{R}_j^{i,k}$ onto the $K-1$ silhouette images $\mathcal{S}_j^l$, $l = 1, \cdots\cdots, K$; $l \neq k$, and then re-project the segments which overlapped with $\mathcal{S}_j^l$ back into the 3D space. The bounding edge

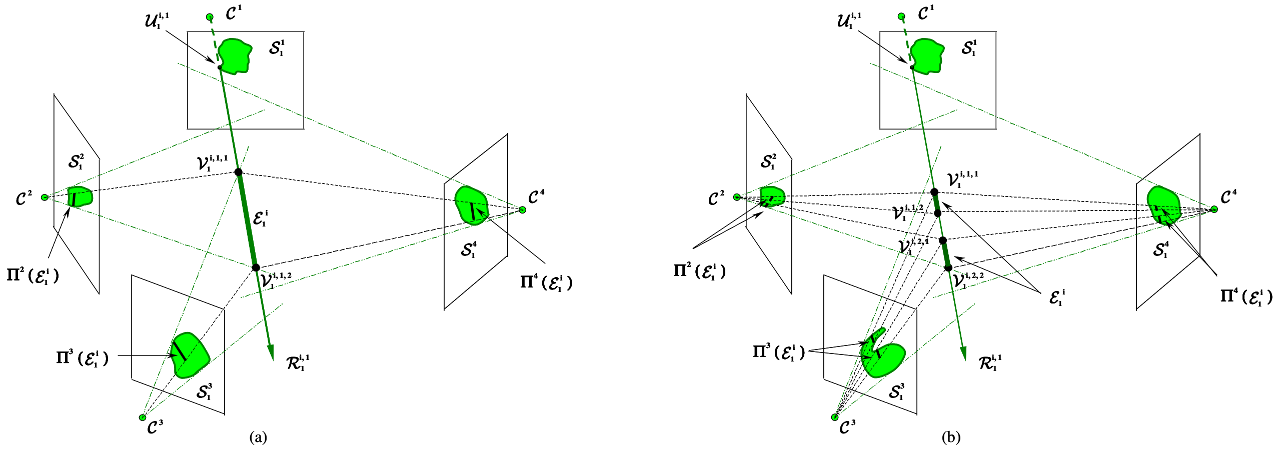


Figure 10. (a) The definition of bounding edge $\mathcal{E}_1^i$. It can be obtained easily by projecting the ray $\mathcal{R}_1^{i,1}$ onto all other silhouette images and re-project the portions which overlapped with the silhouettes back into the 3D space. (b) Situation where the bounding edge $\mathcal{E}_1^i$ consists of more than one segments when one or more silhouette images are not convex.

is the common intersecting portion of the $K-1$ segments. It is important to note that the bounding edge $\boldsymbol{\mathcal{E}}_j^i$ is *not necessarily* a continuous line. It may consist of several segments if one or several silhouette images are not convex. Figure 10(b) gives an example where $\boldsymbol{\mathcal{S}}_1^3$ is not convex. In this case, $\boldsymbol{\mathcal{E}}_j^i$ consists of two discontinuous segments. By sampling points on the boundaries of $\{\boldsymbol{\mathcal{S}}_j^k; \quad k = 1, \cdots\cdots, K\}$, we can construct a list of L_j bounding edges. Hereafter, a bounding edge $\boldsymbol{\mathcal{E}}_j^i$ is denoted by a set of *ordered* vertex pairs as follow :

$$\boldsymbol{\mathcal{E}}_j^i = \{\{\boldsymbol{\mathcal{V}}_j^{i,m,1}, \boldsymbol{\mathcal{V}}_j^{i,m,2}\}; \quad i = 1, \cdots\cdots, L_j; \quad m = 1, \cdots\cdots, M_j^i\} \tag{4}$$

where L_j is the number of bounding edges obtained from the set of silhouette images $\{\boldsymbol{\mathcal{S}}_j^k\}$ and M_j^i is the number of segments with which each $\boldsymbol{\mathcal{E}}_j^i$ is comprised of. The multiple silhouette image version of FTVH is stated below by using bounding edge. The theorem will be used shortly to incorporate color information into aligning visual hulls.

Fundamental Theorem of Visual Hull (Multiple Silhouette Images)

Each bounding edge $\boldsymbol{\mathcal{E}}_j^i$ defined above touches the object (that forms the silhouette images) *at at least one point*.

4.3. Visual Hull Alignment by Color Consistency

The idea of incorporating color information to estimate alignment between two visual hulls utilizes FTVH which states that each bounding edge touches the object at at least one point. The question is how are we going to locate this point? One answer is to use the color images from which the silhouette images are derived from. If we assume the object is Lambertian and all the cameras are color balanced, any point on the surface of the object will have the same projected color in the color images. In other words for any point on the surface of the object, its projected color variance of the visible cameras is zero. By searching along a bounding edge for the point with zero projected color variance, we can estimate the location where the edge touches the object. I will hereafter call these touching points the *colored surface* points of the object.

To express the idea mathematically, let $\{\mathcal{Y}_j^k; \quad k = 1, \cdots\cdots, K; \quad j = 1, 2\}$ be the color images from which the silhouette images $\{\mathcal{S}_j^k\}$ are derived from. Two sets of bounding edges, denoted by $\{\mathcal{E}_j^i, \{\mathcal{V}_j^{i,m,1}, \mathcal{V}_j^{i,m,2}\}; \quad j = 1, 2; \quad i = 1, \cdots\cdots, L_j; \quad m = 1, \cdots\cdots, M_j^i\}$ are constructed from $\{\mathcal{S}_j^k\}$. Let us consider a bounding edge $\mathcal{E}_j^i$ from the j^{th} visual hull. We parameterize a point $\mathcal{W}_j^i(m, w)$ on $\mathcal{E}_j^i$ by two parameters m and w, where $m \in \{1, \cdots\cdots, M_j^i\}$ and $0 \leq w \leq 1$ with

$$\mathcal{W}_j^i(m, w) = \mathcal{V}_j^{i,m,1} + w * (\mathcal{V}_j^{i,m,2} - \mathcal{V}_j^{i,m,1}) \, . \tag{5}$$

Let $\boldsymbol{c}^k(\mathcal{W}_j^i(m, w))$ be the projected color of the 3D point $\mathcal{W}_j^i(m, w)$ on the k^{th} color image. The projected color mean $\mu_j^i(m, w)$ and variance $\sigma_j^i(m, w)$ of the point $\mathcal{W}_j^i(m, w)$ are given as

$$\mu_j^i(m, w) = \sum_k \boldsymbol{c}^k(\mathcal{W}_j^i(m, w)); \qquad \sigma_j^i(m, w) = \sum_k [\boldsymbol{c}^k(\mathcal{W}_j^i(m, w)) - \mu_j^i(m, w)]^2 \, . \tag{6}$$

The projected color $\boldsymbol{c}^k(\mathcal{W}_j^i(m, w))$ from camera k is used in calculating the mean and variance *only if* $\mathcal{W}_j^i(m, w)$ is *visible* in that camera. The issue of how to conservatively determine the visibility of a 3D point with respect to a camera by using the silhouette images will be addressed shortly.

In practice, due to noise and inaccuracies in color balancing, instead of searching for point which has a zero projected color variance, we locate the point with the minimum variance. In other words, we set the colored surface point of the object on $\mathcal{E}_j^i$ to be $\mathcal{W}_j^i(\tilde{m}, \tilde{w})$ where $\tilde{m}$ and $\tilde{w}$ minimizes $\sigma_j^i(m, w)$ for $0 \leq w \leq 1; \quad m \in \{1, \cdots, \cdots, M_j^i\}$. There are two advantages of using this strategy. Since the bounding edge may touch the object at multiple points, the one with the lowest projected color variance is chosen. By choosing the point with the minimum variance, the problem of tweaking parameters or thresholds of any kind is also avoided. The need to adjust parameters or thresholds is always a problem in other shape reconstruction methods such as space carving [22] and stereo. Space carving relies heavily on a color variance threshold to remove non-object voxels and stereo matching result is sensitive to the search window size. Note that as discussed in [2], the strategy of choosing point with the minimum color variance, as well as stereo and space carving, may fail if the object is of constant color over an extended region. For simplicity, we denote $\mathcal{W}_j^i(\tilde{m}, \tilde{w})$, $\sigma_j^i(\tilde{m}, \tilde{w})$ and $\mu_j^i(\tilde{m}, \tilde{w})$ by $\tilde{\mathcal{W}}_j^i$, $\tilde{\sigma}_j^i$ and $\tilde{\mu}_j^i$ respectively.

With the acquisition of two sets of colored surface points of the object at two different time instants, aligning two visual hulls is similar to the 2D image alignment problem as stated in [40]. In our case, I am aligning 2D images ($\{\mathcal{Y}_2^k\}$) at time t_2 with a "3D image" (the colored surface points $\{\tilde{\mathcal{W}}_1^i\}$) at time t_1

through the projection functions $\{\mathbf{\Pi}^k\}$. The error measure used is the sum of color differences between the colored surface points at time t_1 and their projected colors from the color images at time t_2 and vice versa. Mathematically, let $\{\boldsymbol{\mathcal{Y}}_j^k, \boldsymbol{\mathcal{S}}_j^k, \tilde{\boldsymbol{\mathcal{W}}}_j^i, \tilde{\mu}_j^i, \tilde{\sigma}_j^i; \quad i = 1, \cdots\cdots, L_j; \quad k = 1, \cdots\cdots, K; \quad j = 1, 2\}$ be the two sets of data. To find the most color consistent alignment $(\boldsymbol{R}, \boldsymbol{t})$, consider the color error function

$$E = \sum_{i=1}^{L_2} E_{1,2}^i + \sum_{i=1}^{L_1} E_{2,1}^i \;, \tag{7}$$

$$E_{1,2}^i = \sum_k E_{1,2}^{i,k} = \sum_k [\boldsymbol{c}^k(\boldsymbol{R}^T \tilde{\boldsymbol{\mathcal{W}}}_2^i - \boldsymbol{R}^T \boldsymbol{t}) - \tilde{\mu}_2^i]^2 \;, \tag{8}$$

$$E_{2,1}^i = \sum_k E_{2,1}^{i,k} = \sum_k [\boldsymbol{c}^k(\boldsymbol{R} \tilde{\boldsymbol{\mathcal{W}}}_1^i + \boldsymbol{t}) - \tilde{\mu}_1^i]^2 \,. \tag{9}$$

$E_{2,1}^{i,k}$ represents the difference between mean color $\tilde{\mu}_1^i$ (obtained from the colored surface point $\tilde{\boldsymbol{\mathcal{W}}}_1^i$ at time t_1) and its projected color $\boldsymbol{c}^k(\boldsymbol{R}\tilde{\boldsymbol{\mathcal{W}}}_1^i + \boldsymbol{t})$ on camera k at time t_2. Note that at time t_2, the new position of $\tilde{\boldsymbol{\mathcal{W}}}_1^i$ is $\boldsymbol{R}\tilde{\boldsymbol{\mathcal{W}}}_1^i + \boldsymbol{t}$ due to the motion of the object. Likewise, $E_{1,2}^{i,k}$ is the difference between mean color $\tilde{\mu}_2^i$ of $\tilde{\boldsymbol{\mathcal{W}}}_2^i$ and its projected color $\boldsymbol{c}^k(\boldsymbol{R}^T\tilde{\boldsymbol{\mathcal{W}}}_2^i - \boldsymbol{R}^T\boldsymbol{t})$ on camera k at time t_1. Same as for calculating color consistency of points on bounding edge, the summations in equations (8) and (9) include the projected color of camera k *only if* the point of interest is visible in that camera.

If we parameterize $\boldsymbol{R}$ and $\boldsymbol{t}$ by $\mathbf{\Phi} = [\Phi_1, \Phi_2, \Phi_3, \Phi_4, \Phi_5, \Phi_6]^T$ where Φ_1, Φ_2, Φ_3 are the Euler's angles of $\boldsymbol{R}$ and Φ_4, Φ_5, Φ_6 are the x, y, z components of $\boldsymbol{t}$, the minimization of (7) can be solved by the iterative Levenberg-Marquardt algorithm [42]:

1. With an initial estimate $\hat{\mathbf{\Phi}}$, calculate the Hessian matrix $\boldsymbol{H} = \{h_{mn}\}$ with $m, n = 1, \cdots\cdots, 6$ where

   $$h_{mn} = \sum_{i=1}^{L_2} \sum_k \frac{\partial E_{1,2}^{i,k}}{\partial [\mathbf{\Phi}]_m} \frac{\partial E_{1,2}^{i,k}}{\partial [\mathbf{\Phi}]_n} + \sum_{i=1}^{L_1} \sum_k \frac{\partial E_{2,1}^{i,k}}{\partial [\mathbf{\Phi}]_m} \frac{\partial E_{2,1}^{i,k}}{\partial [\mathbf{\Phi}]_n} \;, \tag{10}$$

   and the difference vector $\boldsymbol{d} = \{d_m\}$ with $m = 1, \cdots\cdots, 6$

   $$d_m = -2[\sum_{i=1}^{L_2} \sum_k E_{1,2}^{i,k} \frac{\partial E_{1,2}^{i,k}}{\partial [\mathbf{\Phi}]_m} + \sum_{i=1}^{L_1} \sum_k E_{2,1}^{i,k} \frac{\partial E_{2,1}^{i,k}}{\partial [\mathbf{\Phi}]_m}] \;, \tag{11}$$

2. Update the parameter $\hat{\mathbf{\Phi}}$ by an amount $\delta\mathbf{\Phi} = (\boldsymbol{H} + \lambda \boldsymbol{I})^{-1}\boldsymbol{d}$, where λ is a time-varying stabilization parameter.

3. Go back to 1. until the estimate of $\hat{\mathbf{\Phi}}$ converges.

The process of visual hull alignment by color consistency is illustrated in Figure 11.

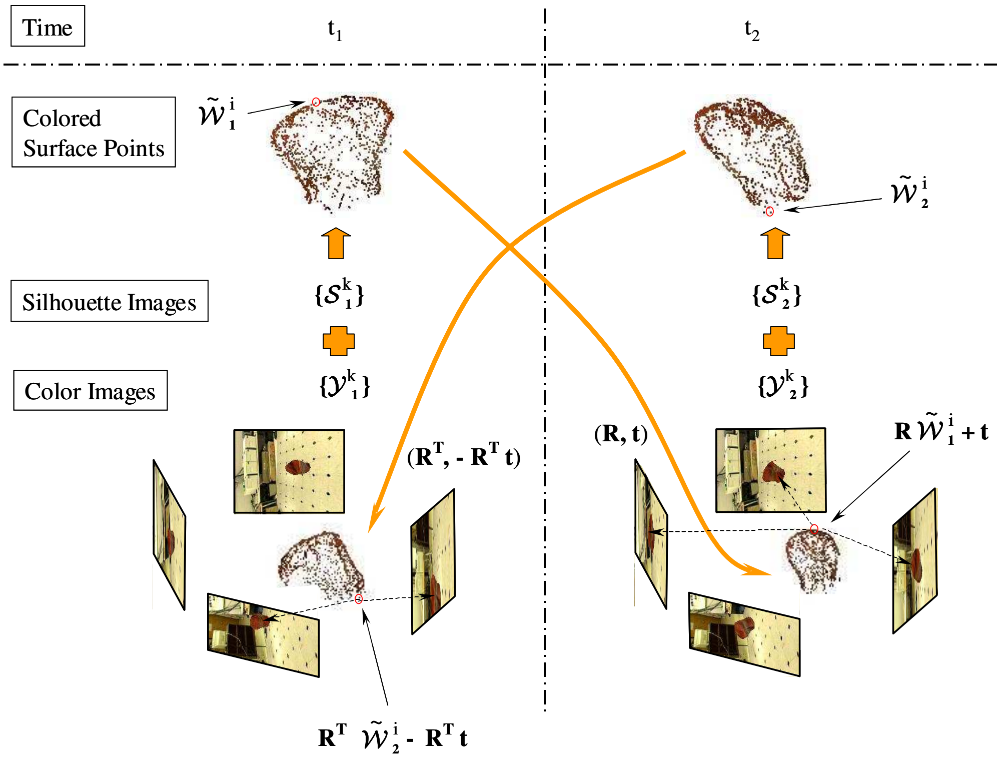


Figure 11. Figure illustrating the idea and process of visual alignment using color consistency.

4.3.1. Determining visibility for color consistency and alignment

To determine color consistency in equation (6), the visibility information of 3D point $\mathcal{W}_j^i(m,w)$ with respect to all the K cameras is required. Here, I present a way to determine the visibilities *conservatively* using the silhouette images only. Suppose we are given a 3D point $\boldsymbol{P}$ and a set of silhouette images $\{\mathcal{S}_j^k\}$ with camera centers $\{\mathcal{C}^k\}$ and projection function $\{\mathbf{\Pi}^k()\}$. If we assume all the cameras have infinite extension of image plane, then the following lemma holds for all cameras :

Lemma 1 : Let $\mathbf{\Pi}^l(\boldsymbol{P})$ and $\mathbf{\Pi}^l(\mathcal{C}^k)$ be the projections of $\boldsymbol{P}$ and the k^{th} camera center $\mathcal{C}^k$ on the image plane of camera l. If the 2D line segment joining $\mathbf{\Pi}^l(\boldsymbol{P})$ and $\mathbf{\Pi}^l(\mathcal{C}^k)$ does not intersect the silhouette image $\mathcal{S}_j^l$, *then* $\boldsymbol{P}$ is visible with respect to camera k.

Figure 12(a) gives examples where the points $\boldsymbol{P}_1$, $\boldsymbol{P}_2$ and $\boldsymbol{P}_3$ are visible with respect to camera 2. The converse of lemma 1 is *not* necessarily true : the visibility *cannot* be determined if the segment joining $\mathbf{\Pi}^l(\boldsymbol{P})$ and $\mathbf{\Pi}^l(\mathcal{C}^k)$ intersects the silhouette $\mathcal{S}_j^l$. One counter example is shown in Figure 12(a). Both points $\boldsymbol{P}_1$ and $\boldsymbol{P}_2$ project to the same 2D point $\boldsymbol{p}$ on the image plane of camera 1 and the segment joining $\boldsymbol{p}$ and $\mathbf{\Pi}^1(\mathcal{C}^4)$ intersects with $\mathcal{S}_1^1$. However, $\boldsymbol{P}_1$ and $\boldsymbol{P}_2$ have different visibilities with respect to camera 4 ($\boldsymbol{P}_2$ is visible while $\boldsymbol{P}_1$ is not).

Though conservative, there are three advantages of using Lemma 1 to determine visibility in our alignment algorithm. First of all, Lemma 1 uses information directly from the silhouette images, avoiding the need to estimate the shape of the object for visibility test. Secondly, recall that to construct a bounding edge $\mathcal{E}_j^i$, we start with the boundary point $\mathcal{U}_j^{i,k}$ of the k^{th} silhouette. Hence all the points on $\mathcal{E}_j^i$ project

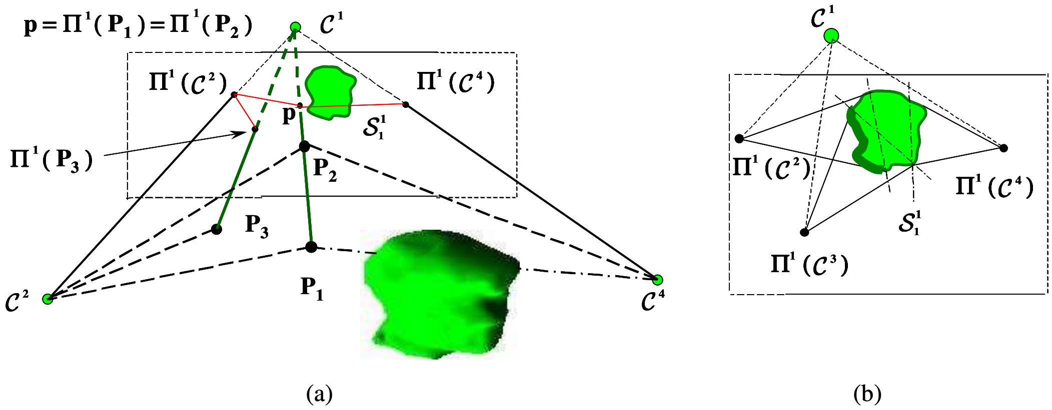


Figure 12. (a) Visibility of points with respect to cameras by Lemma 1. (b) Boundary points that can be used to construct bounding edges are marked by the thick boundary. Those boundary points are the ones which the resulting bounding edges can be seen by at least two other cameras (camera 2 and 3) besides camera 1.

to the same 2D point $\mathcal{U}_j^{i,k}$ on camera k which implies all points on the bounding edge $\mathcal{E}_j^i$ have the same set of conservative visible images. This property ensures the color consistencies of points on the same bounding edge are calculated from the same set of images. Accuracy in searching the optimal point $\tilde{\mathcal{W}}_j^i$ is increased because comparisons are made fairly among points on the same bounding edge. Finally Lemma 1 also provides a guideline to sample the silhouette boundary points for constructing bounding edges. To have meaningful color consistencies, the number of color images used in (6) has to be larger than 2 (or otherwise the projected color variances will always be 0). By Lemma 1, boundary points $\mathcal{U}_j^{i,k}$ are chosen such that the resulting $\mathcal{E}_j^i$ is seen by at least 2 other images (excluding the image $\mathcal{S}_j^k$ from which the boundary point is chosen from). An example is shown in Figure 12(b). Only points on part of the boundary of $\mathcal{S}_1^1$ (marked by thicker lines) are used to construct bounding edges because they are the points from which the resulting bounding edges can be seen by at least two other cameras (camera 2 and 3) besides camera 1.

4.3.2. Results for synthetic data

A synthetic data set was created to test the validity of the alignment algorithm. A textured wire-frame computer model resembling the human torso was used. The model was set to move under a known trajectory of motion with twenty two frames. At each time, images of six cameras ($K = 6$) with known camera parameters are rendered using OpenGL. Twenty two sets of data, represented by $\{\mathcal{S}_j^k, \mathcal{Y}_j^k; \quad k = 1, \cdots\cdots, 6; \quad j = 1, \cdots\cdots, 22\}$ are generated. The color consistency based alignment algorithm is applied to the data set and the estimated motion results are plotted in Figure 13. The black solid lines represent the ground truth and the red dashed lines denote the estimated parameters by my algorithm. It can be seen that the algorithm works very well with this synthetic data set.

To illustrate the superiority of using bounding edges and colored surface points over voxels in the alignment problem, voxel-based visual hulls are constructed by Algorithm 1 described in section 2.3.2.

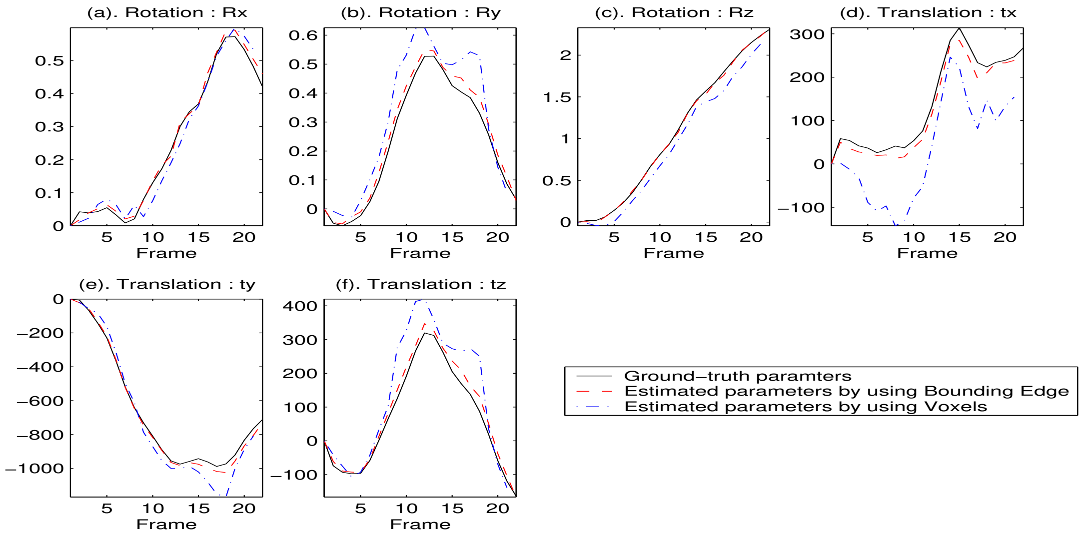


Figure 13. Graphs of ground-truth motion values (black solid lines) and estimated motion values (red dashed line for using bounding edges, blue dashed-dotted line for using voxels) across frames.

The surface voxels are extracted and colored by projecting them onto the color images. Two sets of colored surface voxels from two different time instants are used for alignment by replacing the colored surface points in equation (7) by the center of the colored surface voxels. The results of this voxel-based alignment on the synthetic data are shown by the blue dashed-dotted lines in Figure 13. The performance of alignment by using bounding edges and colored surface points (the red dashed lines) is much better than using the voxel representation. The superiority of using bounding edges over voxels lies in the fact that voxels do not take advantage of the important principle specified by FTVH while bounding edges make full use of FTVH to find the colored surface points. The colored surface points lie on the surface of the object and represent exact shape geometry of the object. On the contrary, the voxel centers only represent approximate geometric information about the object's shape. Aligning 2D images with projections of the geometrically exact 3D colored surface points is definitely more accurate than using geometrically approximate voxel centers.

4.4. Visual Hull Refinement

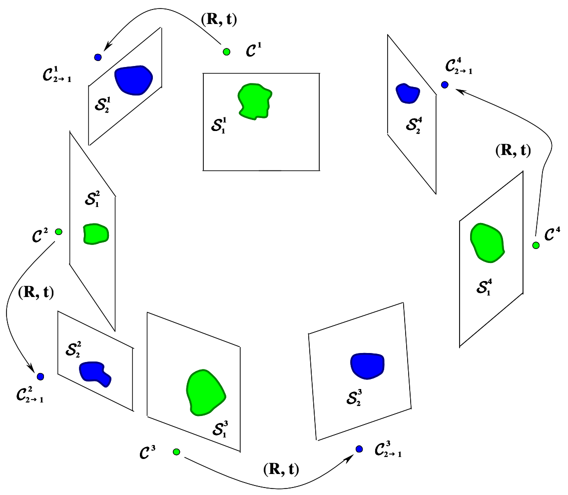


Figure 14. Incorporating $\{\mathcal{S}_2^k\}$ to $\{\mathcal{S}_1^k\}$ by translating and rotating the original cameras by $(\boldsymbol{R}, \boldsymbol{t})$.

After estimating the alignment for two sets of silhouette images $\{\mathcal{S}_j^k; \quad k = 1, \cdots\cdots, K; \quad j = 1, 2\}$, the result is used to refine the two visual hulls $\mathcal{H}_j$ to get a tighter upper bound of the object. Assume t_1 is used as the reference time. We incorporate $\{\mathcal{S}_2^k\}$ into $\{\mathcal{S}_1^k\}$ by considering the former as "new" images captured by additional cameras placed at positions and orientations transformed by $(\boldsymbol{R}, \boldsymbol{t})$ as shown in Figure 14. In other words, we set $\{\mathcal{S}_1^1, \cdots\cdots, \mathcal{S}_1^K, \mathcal{S}_2^1, \cdots\cdots \mathcal{S}_2^K\}$ as the silhouette images of the object at t_1 with corresponding projection functions $\mathbf{\Pi}^1, \cdots\cdots, \mathbf{\Pi}^K, \mathbf{\Pi}^1_{2\to1}, \cdots\cdots, \mathbf{\Pi}^K_{2\to1}$ where $\mathbf{\Pi}^k_{2\to1}$ is the perspective functions derived from $\mathbf{\Pi}^k$ by transforming the orientation and position of camera k by $(\boldsymbol{R}, \boldsymbol{t})$. As a result, the *effective* number of cameras at t_1 is increased to $2K$.

To study the performance of the refinement results, I use the standard voxel construction method (Algorithm 1). The visual hull at time t_j is constructed by using all the silhouettes from the previous frames $\{t_1, \cdots\cdots, t_{j-1}\}$. The refined visual hulls from the synthetic data set are shown in Figure 15 across time. To quantify the refinement results, the ground-truth wire-frame model used to render the input images is converted into a ground-truth voxel model (see Figure 15(b)). The number of extra and missing voxels between the refined shape and the ground-truth voxel models are calculated and plotted against the number of frames used in Figure 16(a) and (b) respectively. It can be seen that the number of extra voxels decreases as the number of frames used increases because a tighter visual hull is obtained with an increase in the number of distinct silhouette images. However, missing voxels increase as the number of frames used increases. This is due to alignment errors which remove correct voxels during construction. The graph of the ratio of total incorrect (missing and extra) to total voxels across frames is plotted in Figure 16(c). For this dataset, the ratio is the lowest when 11 frames (a total of 66 silhouette images) are used for refinement. For comparison, similar graphs for refinement by using the ground-truth motion values (the black solid curves) and by using the voxel-based alignment values (the blue dashed-dotted lines) are also shown in Figure 16.

(a) Original mesh model

(b) Ground-truth voxel model

(c) Voxel model at t_1
(6 images used)

(d) Refined voxel model at t_{11}
(66 images used)

(e) Refined voxel model at t_{21}
(126 images used)

Figure 15. Refinement results for the synthetic data set. (a) The original wire-mesh model, (b) ground-truth voxel model derived from the original mesh model, (c) voxel model at t_1 (6 images used), some of the surface voxels are black because they are not seen by any cameras, (d) refined voxel model t_{11} (66 images used), (e) refined voxel model at t_{21} (126 images used).

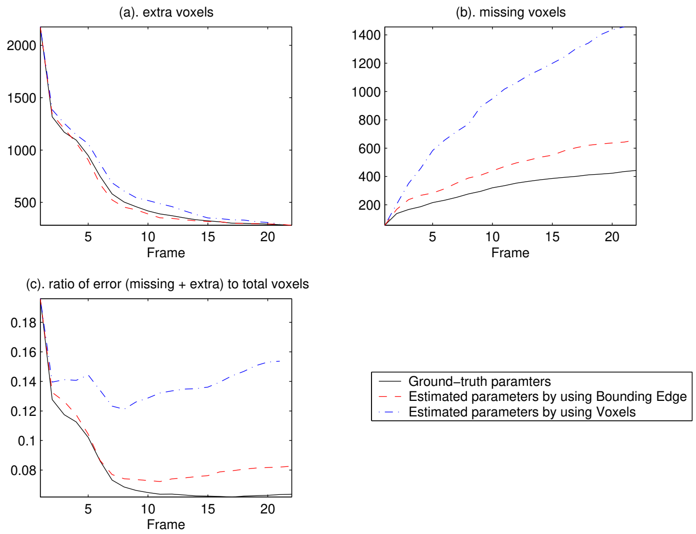


Figure 16. Graphs of refinement errors (missing and extra voxels) across time (frames). Black solid curves are results from using ground-truth motion values. Red dashed lines are results from using bounding edge based alignment motion values while the blue dashed-dotted lines are results from using voxel-based alignment motion values.

4.5. Results on Real Data

A real data set is captured to test my alignment and refinement algorithms. The object is a toy (Pooh) and six cameras ($K = 6$) are used. The cameras are calibrated using the method described in [46]. The toy is placed on a table and moved to new positions and orientations manually at each frame. A total of fifteen frames are captured. The input images of cameras 1 and 4 at times t_1, t_7 and t_{12} are shown in Figure 17(a). Figure 17(b) shows the estimated motion between frames and Figure 17(c) illustrates the refinement results at three time instants t_1, t_6 and $_{15}$. The improvement in shape is very significant from t_1 when 6 silhouette images are used to t_{15} when 90 silhouette images are used.

4.6. Proposed Further Work

4.6.1. A better refinement algorithm

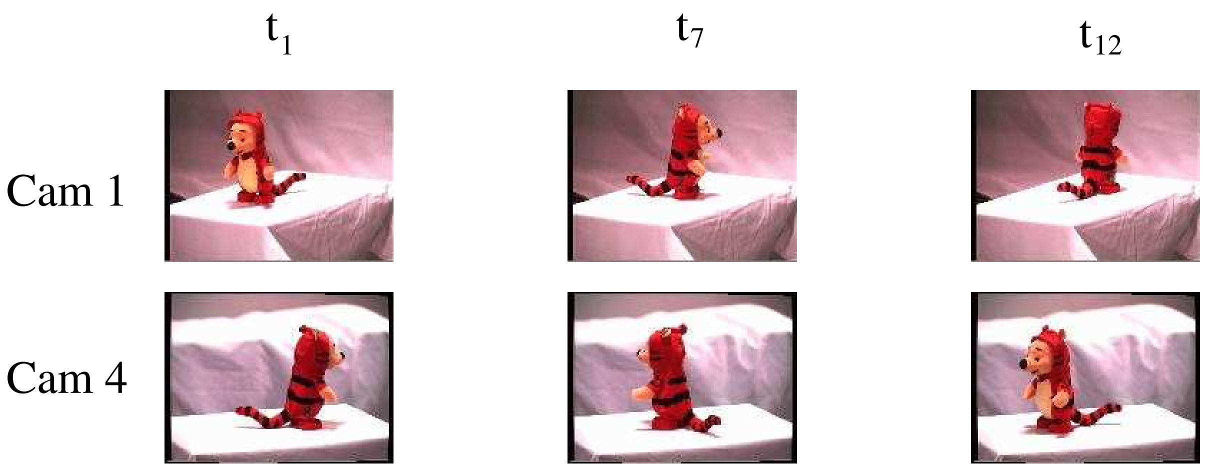
As a further step, I propose to characterize the effect of error in alignment parameters on shape refinement and derive a robust algorithm to deal with the errors. The new algorithm will be used not only in refinement across time but also in *robust* VH construction at single time instant with camera calibration errors.

4.6.2. Applications to refined human body shape acquisition and motion tracking

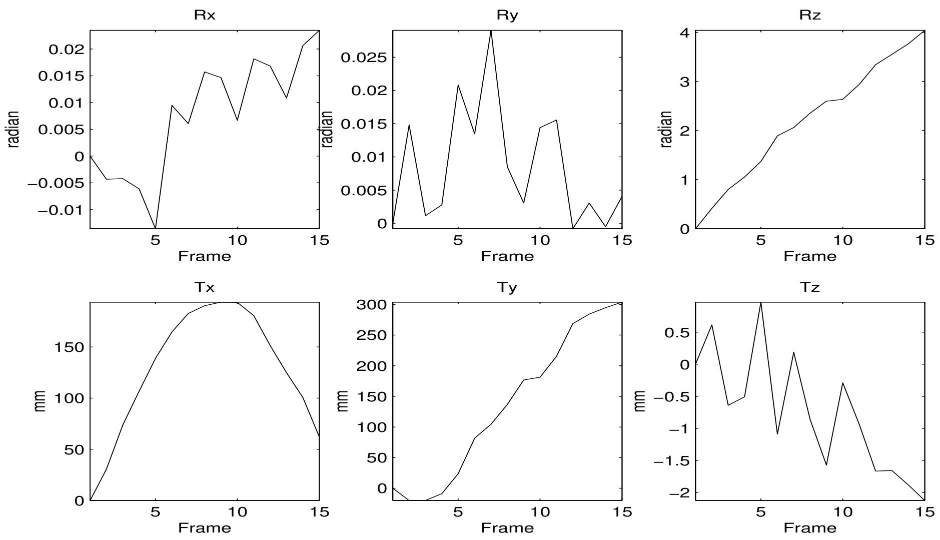
I propose to apply the alignment and refinement algorithms to acquire the shape of human body parts. The subject of interest is asked to perform simple motions while video sequences from multiple cameras are captured. The individual body part is segmented, aligned and refined across time. Refined voxel models of the body parts are built and reassembled to form a complete model of the human.

With a set of dense colored surface points of the body parts obtained from the shape acquisition process above, I propose to use a similar color consistency measure/image alignment paradigm (equation (7) and optimization procedures in Section 4.3) for tracking and locating the body parts in a motion sequence. Body joints information of the human can be applied to provide more constraints for the optimization.

A preliminary experiment has already been performed to test the idea of shape acquisition. A 6-camera 20-frame sequence of a person rotating his head horizontally is taken from the Virtualized Reality^{TM} Laboratory database. The input images of cameras 3 and 4 at times t_1, t_{10} and t_{19} are shown in Figure 18(a). Note that the cameras are not color balanced and the camera calibrations are not very precise in this data set. The head silhouettes across all the twenty frames are segmented and then aligned using my algorithm. Figure 18(b) shows the estimated motion which agrees with the actual motion of the head (rotating and moving horizontally). Figure 18(c) illustrates the refinement results at three time instants t_1, t_{10} and t_{20}. The algorithm performs well in improving the shape of the head over time in spite of the camera calibration errors and poorly color-balanced cameras.



(a)



(b)

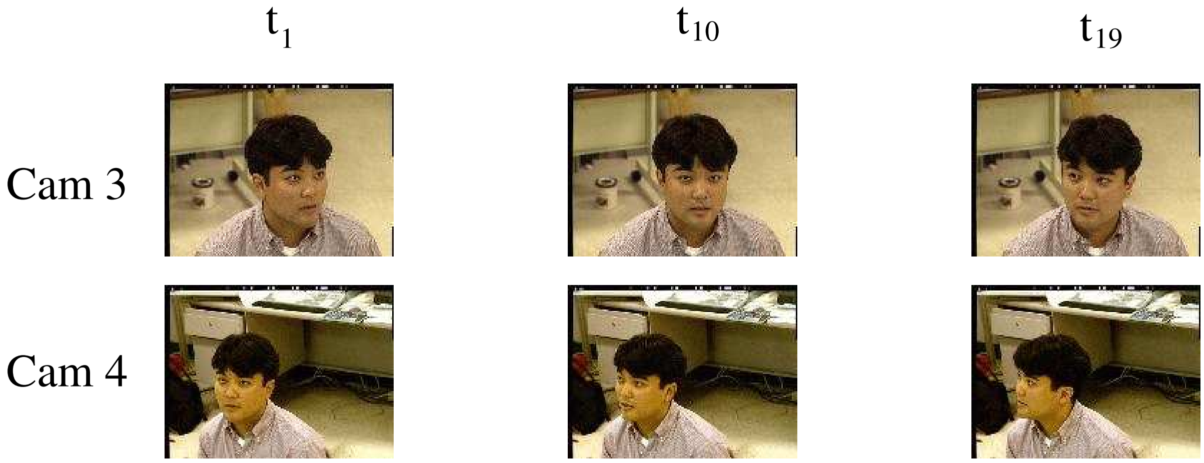
Voxel model at t_1 (6 silhouette images used)

Refined voxel model at t_6 (36 silhouette images used)

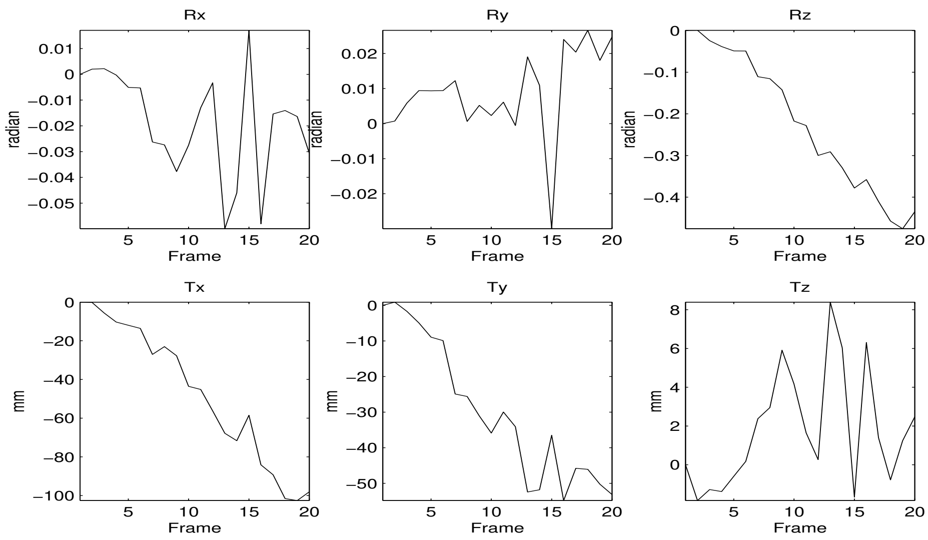
Refined voxel model at t_{15} (90 silhouette images used)

(c)

Figure 17. Pooh Data Set. (a) Example input images for the Pooh in Tigger Suit data set from cameras 1 and 4, (b) estimated motion parameters, (c) refined voxel models at times t_1 (6 silhouettes used), t_6 (36 silhouettes used) and t_{15} (90 silhouettes used).



(a)



(b)

Voxel model at t_1 (6 silhouette images used)

Refined voxel model at t_{10} (60 silhouette images used)

Refined voxel model at t_{20} (120 silhouette images used)

(c)

Figure 18. Head Sequence. (a) Example input images for the Head Sequence from cameras 3 and 4, (b) estimated motion parameters, (c) refined voxel models at times t_1 (6 silhouettes used), t_{10} (60 silhouettes used) and t_{20} (120 silhouettes used).

5. Theoretical Analysis of Visual Hull Alignment Ambiguity

In this section, I analyze the ambiguity issue of aligning two visual hulls using silhouette images only. The two dimensional case is discussed first. Geometrical constraints of aligning two 2D visual hulls are derived. Similar discussion is proposed to apply to 3D objects. The symbols we have been using for 3D objects will be used to represent the corresponding 2D entities.

5.1. Geometrical Constraints of Aligning Two 2D Visual Hulls

To establish the arguments, I will first introduce Lemma 2 which states the property of a 2D visual hull, followed by a two dimensional version of FTVH. The FTVH will lead to Lemma 3 which states the geometrical constraints for aligning two 2D VHs.

Lemma 2 : For a *closed* and *connected* 2D object, its visual hull from K silhouette images *must* be a convex polygon with at most $2K$ bounding edges. Conversely, any convex polygon with $M \geq 4$ edges can be thought of as a visual hull formed from K silhouettes of some closed and connected 2D object where $K = \lceil \frac{M}{2} \rceil$.

We have limited ourselves to *closed* and *connected* 2D objects for this lemma to be true. Now the two dimensional version of the FTVH can be stated as follows.

Two-dimensional Fundamental Theorem of Visual Hull
Each side of the 2D polygonal visual hull has to touch the object at at least one point.

The two dimensional version of FTVH is useful in deriving the geometrical constraints of aligning two 2D visual hulls. Given two 2D visual hulls $\mathcal{H}_1$ and $\mathcal{H}_2$, we said they are *aligned consistently with transformation* $(\boldsymbol{R}, \boldsymbol{t})$ if and only if we can find an object $\mathcal{O}$ such that $\mathcal{H}_1$ is the visual hull of $\mathcal{O}$ at orientation and position $(\boldsymbol{I}, \underline{0})$ and $\mathcal{H}_2$ is the visual hull of $\mathcal{O}$ at orientation and position $(\boldsymbol{R}, \boldsymbol{t})$. The definition of consistent alignment of two visual hulls is general in the sense that there is no limitation on the number of cameras used to form the visual hulls, as long as they satisfy Lemma 2. For the sake of presentation, let $\mathcal{E}_j^i$ represents the edges of $\mathcal{H}_j$, $\mathcal{T}_{(\boldsymbol{R},\boldsymbol{t})}(\mathcal{A})$ represents the entity after applying transformation of $(\boldsymbol{R}, \boldsymbol{t})$ to $\mathcal{A}$ and $\mathcal{T}^{-1}_{(\boldsymbol{R},\boldsymbol{t})}()$ denotes the inverse transformation.

Lemma 3 Given two 2D visual hulls $\mathcal{H}_1$ and $\mathcal{H}_2$, the necessary and sufficient condition for the two visual hulls to be aligned consistently with transformation $(\boldsymbol{R}, \boldsymbol{t})$ is given as follows : No edge of $\mathcal{T}_{(\boldsymbol{R},\boldsymbol{t})}(\mathcal{H}_1)$ lies completely outside $\mathcal{H}_2$ and no edge of $\mathcal{H}_2$ lies completely outside $\mathcal{T}_{(\boldsymbol{R},\boldsymbol{t})}(\mathcal{H}_1)$.

Figure 19(a)(b) show examples of two 2D visual hulls from the same object. In (c), the alignment is consistent and all edges from both visual hulls satisfy Lemma 3. In (d), the alignment is inconsistent and the edges $\mathcal{E}_1^1$, $\mathcal{E}_1^4$, $\mathcal{E}_1^5$, $\mathcal{T}^{-1}_{(\boldsymbol{R}',\boldsymbol{t}')}(\mathcal{E}_2^1)$, $\mathcal{T}^{-1}_{(\boldsymbol{R}',\boldsymbol{t}')}(\mathcal{E}_2^2)$, $\mathcal{T}^{-1}_{(\boldsymbol{R}',\boldsymbol{t}')}(\mathcal{E}_2^7)$ violate Lemma 3. Lemma 3 provides a good way to test if an alignment is consistent or not. In order to use it in practical situation and extend the results to 3D, we consider the following variation of Lemma 3 which we called Lemma 3.1

Lemma 3.1 $(\boldsymbol{R}, \boldsymbol{t})$ is a consistent alignment of two 2D visual hulls $\mathcal{H}_1$ and $\mathcal{H}_2$, constructed from silhouette sets $\{\mathcal{S}_1^k\}$ and $\{\mathcal{S}_2^k\}$ if and only if the following condition is satisfied : For each edge $\mathcal{E}_1^i$ of $\mathcal{T}_{(\boldsymbol{R},\boldsymbol{t})}(\mathcal{H}_1)$, there exists at least one point $\boldsymbol{P}$ on $\mathcal{E}_1^i$ so that the projection of $\boldsymbol{P}$ on the k^{th} image lies inside or on the silhouette $\mathcal{S}_2^k$ for all $k = 1, \cdots\cdots, K$ and vice versa.

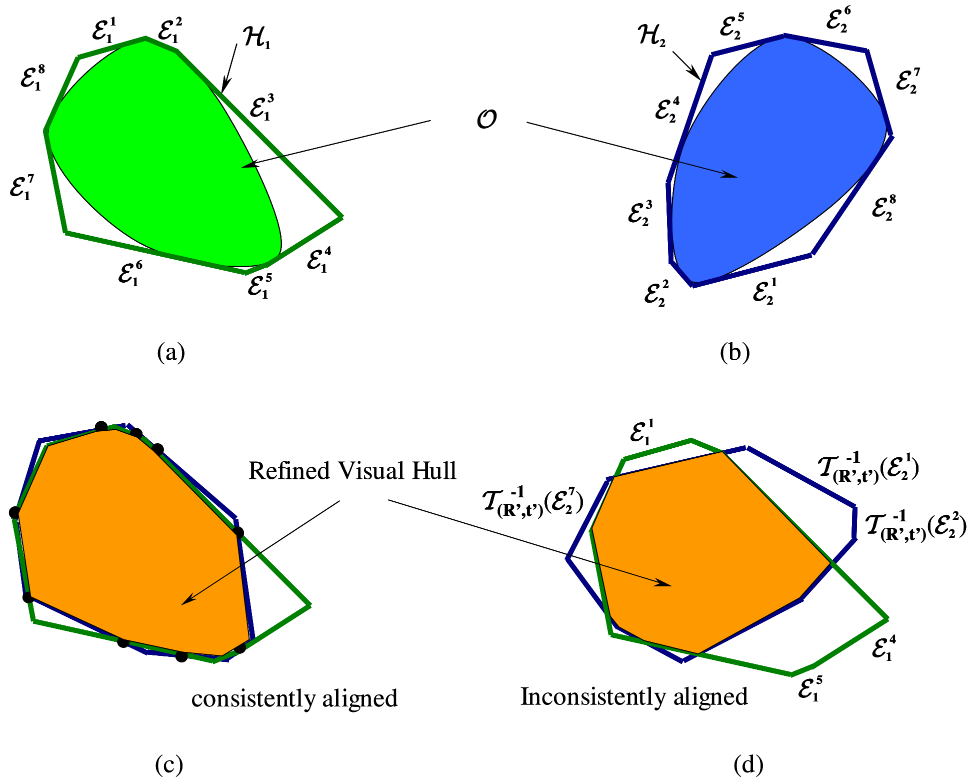


Figure 19. (a)(b) Two visual hulls of the same object at different pose. (c) All edges satisfy Lemma 3 when the alignment $(\boldsymbol{R}, \boldsymbol{t})$ is consistent, (d) edges $\mathcal{E}_1^1$, $\mathcal{E}_1^4$, $\mathcal{E}_1^5$, $\mathcal{T}^{-1}_{(\boldsymbol{R}',\boldsymbol{t}')}(\mathcal{E}_2^1)$, $\mathcal{T}^{-1}_{(\boldsymbol{R}',\boldsymbol{t}')}(\mathcal{E}_2^2)$, $\mathcal{T}^{-1}_{(\boldsymbol{R}',\boldsymbol{t}')}(\mathcal{E}_2^7)$ violate Lemma 3 when the visual hulls are not aligned consistently.

Lemma 3.1 is more readily usable than Lemma 3. Lemma 3.1 expresses the geometrical constraints in terms of the silhouette images, rather than the visual hulls as in Lemma 3. The difference in 2D is insignificant as 2D visual hulls can be represented easily by polygon with finite number of edges. For 3D objects, however, it is difficult to have a complete and exact representation for 3D visual hull (see the discussion in Section 6). The VH representation problem can be avoided by expressing the constraints in terms of silhouette images instead of VH.

To express Lemma 3.1 mathematically, we introduce a function $\mathcal{D}_{\mathcal{S}_j^k}(\boldsymbol{B})$ which returns the part of a segment $\boldsymbol{B}$ whose projection lies inside and on the silhouette image $\mathcal{S}_j^k$. In other words, for a point $\boldsymbol{P}$ on the segment $\boldsymbol{B}$, $\boldsymbol{P} \in \mathcal{D}_{\mathcal{S}_j^k}(\boldsymbol{B})$ i.f.f. $\boldsymbol{P} \in \boldsymbol{B}$ and $\boldsymbol{\Pi}^k(\boldsymbol{P}) \in \mathcal{S}_j^k$. We restate Lemma 3.1 in an equation form as follows.

Lemma 3.1 $(\boldsymbol{R}, \boldsymbol{t})$ is a consistent alignment of two visual hulls $\mathcal{H}_1$ and $\mathcal{H}_2$, constructed from silhouette sets $\{\mathcal{S}_1^k\}$ and $\{\mathcal{S}_2^k\}$ if and only if the following $(L_1 + L_2)$ conditions are satisfied

$$\left\{ \begin{array}{lcll} \bigcap_{k=1}^{K} \mathcal{D}_{\mathcal{S}_1^k}(\mathcal{T}^{-1}_{(\boldsymbol{R},\boldsymbol{t})}(\mathcal{E}_2^i)) & \neq & \emptyset; & i = 1 \cdots\cdots, L_2 \\ \bigcap_{k=1}^{K} \mathcal{D}_{\mathcal{S}_2^k}(\mathcal{T}_{(\boldsymbol{R},\boldsymbol{t})}(\mathcal{E}_1^i)) & \neq & \emptyset; & i = 1 \cdots\cdots, L_1 \end{array} \right. , \tag{12}$$

As an example, two synthetic 2D visual hulls (polygons) each has four edges are generated. Lemma 3 is used to search for the space of all consistent alignment. There are three degrees of freedom (2 in translation and 1 in rotation). The results are shown in Figure 20. From the solution space, it can seen that there are two disjoint groups of solutions, clustered around different rotation angles. Another set of synthetic 2D visual hulls with 6 edges are also shown in Figure 21. In this example, there is only one group of solutions.

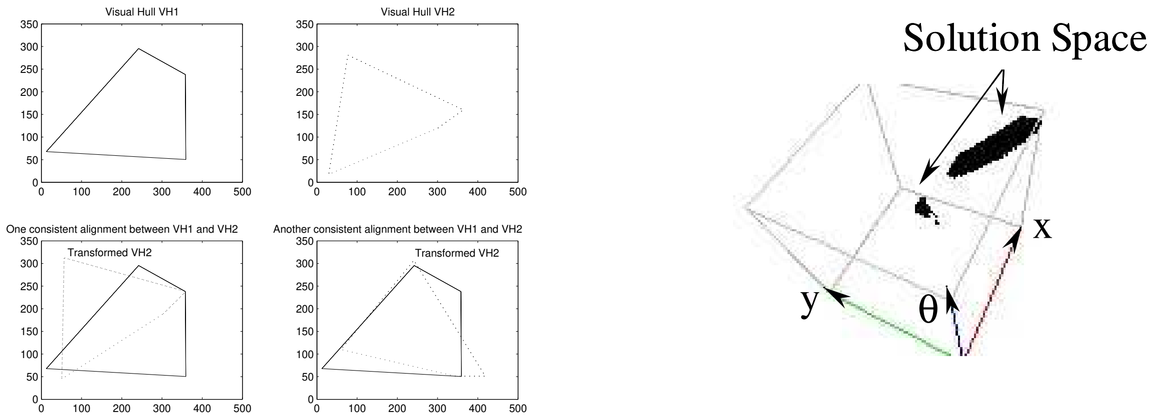


Figure 20. Example of two synthetic 2D visual hulls (each has four edges) and their solution space of consistent alignment.

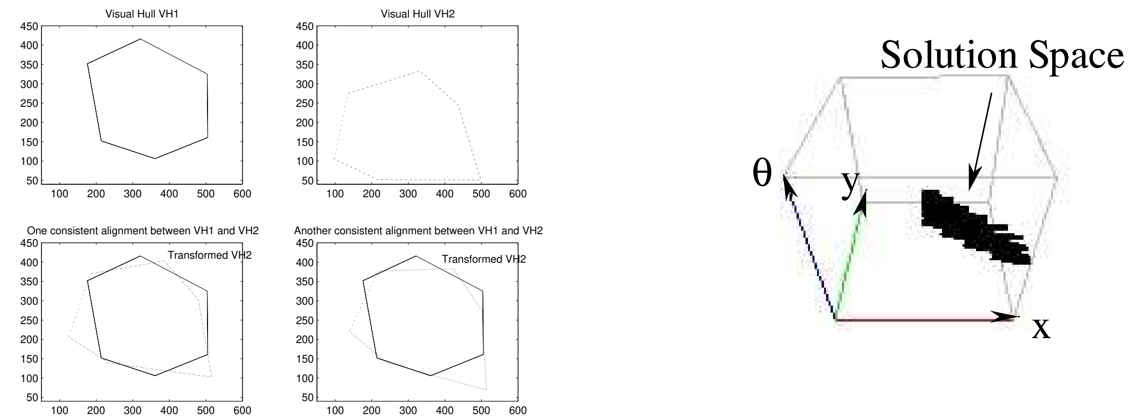


Figure 21. Example of two synthetic 2D visual hulls (each has six edges) and their solution space of consistent alignment.

5.2. Proposed Further Work

I propose to use Lemma 3.1 to derive a systematic algorithm for finding the space of all consistent alignment of two 2D visual hulls. The algorithm will also derive the space of all possible shapes of the object. I also propose to extend the theoretical analysis to 3D objects by deriving the 3D equivalent of Lemma 3. The aim is to find simple testing conditions for consistently aligning two 3D VHs.

6. Representation of 3D Visual Hull

6.1. Bounding Edge as Visual Hull Representation

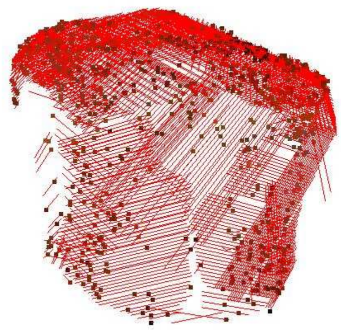


Figure 22. Example bounding edges and their corresponding colored surface points of the synthetic torso model used in Section 4.3.2 at time t_{11}.

In Section 4, we define bounding edge for the purpose of expressing FTVH and aligning visual hulls. Example bounding edges and their corresponding colored surface points of the synthetic torso data set used in Section 4.3.2 is shown in Figure 22. It is interesting to observe that the bounding edges outline the shape of the actual visual hull. Intuitively, I propose to represent VH by a list of bounding edges derived from all the points on the K silhouette boundaries of $\mathcal{S}_j^k$ for $k = 1, \cdots\cdots, K$. In fact, the construction of bounding edges is consistent with the bounding cones intersection method for visual hull construction described in Section 2.3.1. The difference lies in the basic geometrical entities used. Bounding edge uses segments (1D) while planes and/or surface patches (2D) are used in the bounding cones intersection.

There are several advantages in using bounding edges to represent visual hull. They lie exactly on the surface of visual hull and therefore are not approximation like the discrete voxels. In other words, bounding edge is exact and incomplete representation of 3D visual hull while the voxel representation is complete and inexact. Here a VH representation is said to be exact if the shape information embedded in the representation about the object is accurate information of the object. A VH representation is said to be complete if it has full geometrical information (though the information may not be accurate) of the object's shape. In applications like VH alignment, exactness is preferred over completeness because small amount of accurate information is more valuable than large amount of approximate information. As a second advantage, FTVH is naturally embedded in the definition of bounding edges. The essence of the SFS principle is used automatically when we represent VH by bounding edges. It is also easier to compute and parameterize the one-dimensional bounding edges than the 2D planes/surface patches used in bounding cones intersection. Bounding edge is a good compromise between the overly complex bounding cones intersection representation and the coarsely approximate voxel representation. Table

6.1 lists the differences and comparisons between the three visual hull representations : bounding cones intersection, discrete voxel and bounding edge.

Properties	Bounding Cones Intersection	Voxel	Bounding Edge
Basic Geometric Entities	2D Surface	3D Volume	1D Line Segment
Completeness	complete	complete	incomplete
Exactness	exact	inexact	exact
Computational Complexity	very high	low	moderate
natural FTVH embedding	yes	no	yes
Applications	None	real-time system, fast, approximate color and shape estimation, obstacles avoidance	visual hull alignment, precise color and shape estimation for high quality re-rendering

Table 1. Table comparing bounding cones intersection, voxel-based and bounding edge representations for visual hull.

6.2. Proposed Further Work

There are a lot of advantages of using bounding edge to represent visual hulls. One good example is the alignment algorithm I developed. I propose to further investigate this representation, discover more useful properties and potential applications in other problems. Possible directions include studying the geometrical meaning of two intersecting bounding edges, refining two visual hulls represented by bounding edges, investigating its use and relationship with the classic stereo paradigm and comparing its properties and differences with the rim mesh and visual hull mesh representation newly suggested by Lazebnik et. al. in [27].

7. Conclusion

7.1. Summary of Proposal

In this proposal, I studied the shortcomings of the classic shape reconstruction method called Shape from Silhouette and provided practical solutions to overcome some of these limitations. I used SFS in a real-time human motion reconstruction system by introducing a fast algorithm called **SPOT** which traded accuracy for speed. An error analysis of **SPOT** on noisy silhouette images was also presented. I pointed out the shape inexactness problem of SFS, especially when it was done by using voxel-based representation. A novel entity called bounding edge was proposed to represent 3D VH. By using bounding edge and its inherently embedded Fundamental Theorem of Visual Hull, I incorporated color information in the framework of SFS. Points on the object surface were extracted by using color consistency. These colored surface points were used to align visual hulls across time by using a technique similar to image alignment. From the alignment results, I used a straightforward algorithm to refine the shape of visual hull. Finally I presented a theoretical study on the inherent ambiguity in 2D visual hull alignment as a basis for similar study in 3D.

7.2. Related Work

Laurentini has published a series of papers studying various properties of visual hulls [23] [24] [25] [26]. In [24], he defines the concept of hard points which are the same as the touching points mentioned in FTVH. The application-based papers [7] [5] [6] by Laurentini et. al. have similar goal and idea as my real time human motion reconstruction system. They concentrate on the motion capture/fitting aspects from the reconstructed voxels and there are neither real-time consideration nor silhouette noise analysis in these papers. Niem performs error analysis of voxel-based reconstruction method in [32]. He focuses on local shape error caused by discrete resolution of the silhouettes and volume, finite number of views and to some extent camera calibration parameters. His analysis does not include silhouette image noises but provide guidelines for choosing number of viewpoints and voxel size from a given shape accuracy.

There are two major vision groups working on problem of visual motion of curves and surfaces [12], which is closely related to the problem of visual hull alignment. Cipolla, Wong and Mendonca [29] [45] [44] study the problem of estimating structure and motion of smooth object undergoing *circular motion* from silhouette profiles. They assume a single camera which is weakly calibrated (i.e. with known intrinsic but unknown extrinsic parameters). Either the camera (on a robotic arm) or the object (on a turntable) performs unknown circular motion while the silhouette images are taken. In [30] symmetric properties of the surface of revolution swept by the rotating object are used to recover the revolution axis, leading to the estimation of homographies and full epipolar geometries between images using one-dimensional search. In [45], they identity and estimate the *frontier points* (see [18] for detailed definition) on the silhouette boundary and use them to estimate the circular motion between images. Once the motion is estimated, the object shape can be reconstructed by using the classic SFS method.

The second group of researchers, lead by Ponce [18] [19] [43] also study the problem of recovering motion and shape of *smooth curved* object from silhouette images. They define a local parabolic structure on the surface of the object and use that, together with epipolar geometry, to locate corresponding frontier points on three silhouette images. Motion between images is then estimated by a two-step nonlinear minimization.

In a very recent, yet to be published paper [27], Lazebnik et.al. suggest entities called rim mesh and visual hull mesh to construct and represent VH. It turns out the edges of the visual hull mesh are indeed the bounding edges defined in this document. In [27], the bounding edges are used to build a mesh with frontier and triple points as vertices. The issue of voxel reconstruction from a probability point of view is studied in [4] and [9]. In the former paper responsibility values are assigned to each voxel to account for semi-transparent object. In the latter paper, Bayesian formulation is adopted to improve the performance of space carving. Both papers serve as inspirations for the robust visual hull alignment algorithm to be developed in the final thesis.

7.3. Work to be finished

1. Characterize the effect of alignment (or equivalently camera calibration) errors in SFS. From the analysis, devise an unified algorithm to be used in VH refinement across time and robust single time instant SFS against poorly calibrated cameras.

2. Apply the alignment and refinement algorithms to detailed human body shape acquisition and motion tracking in calibrated video sequences.

3. Derive an algorithm to find the space of all consistent alignments between two 2D visual hulls. Extend the ambiguity analysis to 3D and derive similar geometric constraints for aligning two 3D VHs.

4. Further investigate the properties and applications of bounding edge, especially its similarities and differences between the several literatures listed in the related work.

7.4. Expected Contributions

1. A real-time Shape from Silhouette Algorithm called **SPOT** with analysis of noisy silhouette images. The application of **SPOT** to real-time voxel-based dynamic human motion reconstruction.

2. Emphasis on the importance of Fundamental Theorem of Visual Hull as a key source of information in SFS methods and the use of bounding edges as new visual hull representation.

3. Algorithm for aligning 3D visual hulls across time by incorporating color information into the SFS framework.

4. Robust algorithm for refining 3D visual hulls across time.

5. Theoretical analysis of visual hull alignment ambiguity and geometrical constraints for consistent alignment of 2D and 3D visual hulls.

6. Applications of visual hull alignment/refinement to human body shape acquisition and motion tracking.

7.5. Schedule

1. Alignment error analysis and robust refinement algorithm : finished by June 2002.
2. Application to human shape acquisition and motion tracking : finished by September 2002.
3. Algorithm for finding the space of all consistent alignment for 2D and extend results to 3D : finished by November 2002.
4. Further investigation of properties and applications of bounding edges : finished by mid-January 2003.
5. Writing thesis and defend : finished by March 2003.

References

[1] H. H. Baker. *Depth from Edge- and Intensity-Based Stereo*. PhD thesis, University of Illinois, 1981.

[2] S. Baker, T. Sim, and T. Kanade. A characterization of inherent stereo ambiguities. In *Proceedings of ICCV'01*, Vancouver, Canada, June 2001.

[3] B.G. Baumgart. *Geometric Modelling for Computer Vision*. PhD thesis, Stanford University, 1974.

[4] J. S. De Bonet and P. Viola. Roxels: Responsibility weighted 3d volume reconstruction. In *Proceedings of ICCV'99*, Corfu, Greece, September 1999.

[5] A. Bottino and A. Laurentini. Non-intrusive silhouette based motion capture. In *Proceedings of 4th World Multiconference on Systemics, Cybernetics and Informatics SCI 2001*, pages 23–26, Orlando, FL, July 2000.

[6] A. Bottino and A. Laurentini. Interactive reconstruction of 3d objects from silhouette. In *Proceedings of 9th International Conference on Computer Graphics, Visualization and Computer Vision WSCG'2001*, pages 5–9, Orlando, FL, February 2001.

[7] A. Bottino, A. Laurentini, and P. Zuccone. Toward non-intrusive motion capture. In *Proceedings of ACCV'98*, pages 416–23, Hong Kong, Jan. 1998.

[8] E. Boyer and M. Berger. 3d surface reconstruction using occluding contours. *International Journal on Computer Visiosn*, 22(3):219–233, 1997.

[9] A. Broadhurst, T.W. Drummond, and R. Cipolla. A probabilistic framework for space carving. In *Proceedings of ICCV'01*, Vancouver, Canada, June 2001.

[10] C. Buehler, W. Matusik, L. McMillan, and S. Gortler. Creating and rendering image-based visual hulls. Technical Report MIT-LCS-TR-780, MIT, 1999.

[11] German K. M. Cheung, T. Kanade, J.Y. Bouquet, and M. Holler. A real time system for robust 3d voxel reconstruction of human motions. In *Proceedings CVPR'00*, Hilton Head Island, SC, June 2000.

[12] R. Cipolla and P. Giblin. *Visual Motion of Curves and Surfaces*. Cambridge University Press, 2000.

[13] Q. Delamarre and O. Faugeras. 3d articulated models and multi-view tracking with silhouettes. In *Proceedings of ICCV'99*, Corfu, Greece, September 1999.

[14] A. Elgammal, D. Harwood, and L. S. Davis. Non-parametric model for background subtraction. In *Proceedings of ICCV Frame-rate Workshop*, Corfu, Greece, September 1999.

[15] O. Faugeras. *Three-Dimensional Computer Vision : A Geometric Viewpoint*. MIT Press, 1993.

[16] P. Giblin and R. Weiss. Reconstruction of surfac from profiles. In *Proceedings of Image Understanding Workshop*, pages 136–144, 1987.

[17] T. Horprasert, D. Harwood, and L. S. Davis. A statistical approach for real-time robust background subtraction and shadow detection. In *Proceedings of ICCV Frame-rate Workshop*, Corfu, Greece, September 1999.

[18] T. Joshi, N. Ahuja, and J. Ponce. Towards structure and motion estimation from dynamic silhouettes. In *Proceedings of IEEE Workshop on Motion of Non-rigid and Articulated Objects*, pages 166–171, Austin, TX, November 1994.

[19] T. Joshi, N. Ahuja, and J. Ponce. Structure and motion estimation from dynamic silhouettes under perspective projection. Technical Report UIUC-BI-AI-RCV-95-02, UIUC, 1995.

[20] I. A. Kakadiaris and D. N. Metaxas. 3d human body model acquisition from multiple views. *International Journal on Computer Visiosn*, 30(3):191–218, 1998.

[21] Y. C. Kim and J. K. Aggarwal. Rectangular parallelepiped coding: A volumetric representation of three dimensional objects. *IEEE Journal of Robotics and Automation*, RA-2:127–134, 1986.

[22] K.N. Kutulakos and S. Seitz. A theroy of shape by space carving. *International Journal of Computer Vision*, 38(3):199–218, 200.

[23] A. Laurentini. The visual hull concept for silhouette-based image understanding. *IEEE Trans. PAMI*, 16(2):150–162, February 1994.

[24] A. Laurentini. How far 3d shapes can be understood from 2d silhouettes. *IEEE Trans. PAMI*, 17(2):188–195, February 1995.

[25] A. Laurentini. How many 2d silhouettes does it take to reconstruct a 3d object? *Computer Vision and Image Understanding*, 67(1):81–87, 1997.

[26] A. Laurentini. The visual hull of curved objects. In *Proceedings of ICCV'99*, Corfu, Greece, September 1999.

[27] S. Lazebnik, E. Boyer, and J. Ponce. On computing exact visual hulls of solids bounded by smooth surfaces. In *Proceedings CVPR'01*, Kauai, HI, December 2001.

[28] W. N. Martin and J. K. Aggarwal. Volumetric descriptions of objects from multiple views. *IEEE Trans. PAMI*, 5(2):150–174, March 1983.

[29] P. Mendonca, K. Wong, and R. Cipolla. Camera pose estimation and reconstruction from image profiles under circular motion. In *Proceedings ECCV'00*, pages 864–877, Dublin, Ireland, June 2000.

[30] P. Mendonca, K. Wong, and R. Cipolla. Epipolar geometry from profiles under circular motion. *IEEE Trans. PAMI*, 23(6):604–616, June 2001.

[31] S. Moezzi, L.C. Tai, and P. Gerard. Virtual view generation for 3d digital video. *IEEE Computer Society Multimedia*, 4(1), January-March 1997.

[32] W. Niem. Error analysis for silhouette-based 3d shape estimation. In *Proceedings of International Workshop on Synthetic-Natural Hybrid Coding and Three Dimensional Imaging (IWSNHC3DI'97)*, pages 5–9, Rhodes, Greece, September 1997.

[33] M. Okutomi and T. Kanade. A multiple-baseline stereo. *IEEE Trans. PAMI*, 15(4):353–363, 1993.

[34] M. Potmesil. Generating octree models of 3d objects from their silhouettes in a sequence of images. *Computer Vision, Graphics and Image Processing*, 40:1–20, 1987.

[35] P. Rander, P.J. Narayanan, and T. Kanade. Virtualized reality : Constructing time-varying virtual worlds from real world events. In *Proceedings of IEEE Visualization '97*, pages 277–283, Phoenix, AZ, October 1997.

[36] L. G. Roberts. Machine perception of three-dimensional solids. In J. K. Aggarwal, R. O. Duda, and A. Rosenfeld, editors, *Computer Methods in Image Analysis*, pages 285–323. New York : IEEE Press, 1977.

[37] S. Seitz and C.R. Dyer. Photorealistic scene reconstruction by voxel coloring. In *Proceedings of CVPR'97*, pages 1067–1073, San Juan, Puerto Rico, June 1997.

[38] K. Shanmukh and A. K. Pujari. Volume intersection with optimal set of directions. *Pattern Recognition Letter*, 12:165–170, 1991.

[39] R. Szeliski. Rapid octree construction from image sequences. *CVGIP : Image Understanding*, 58(1):23–32, July 1993.

[40] R. Szeliski. Image mosaicing for tele-reality applications. Technical Report CRL 94/2, Compaq Cambridge Research Laboratory, 1994.

[41] C. Tomasi and T. Kanade. Shape and motion from image streams under orthography. Technical Report CMU-CS-92-104, CMU, 1992.

[42] W. T. Vetterling and et. al. *Numerical Receipes in C.* Cambridge University Press, 1993.

[43] B. Vijayakumar, D. Kriegman, and J. Ponce. Structure and motion of curved 3d objects from monocular silhouettes. In *Proceedings of CVPR'96*, pages 327–334, San Franscisco, CA, June 1996.

[44] K. Y. K. Wong and R. Cipolla. Head model acquisition and silhouettes. In *Proceedings of International Workshop on Visual Form IWVF-4*, Capri, Italy, May 2001.

[45] K. Y. K. Wong and R. Cipolla. Structure and motion from silhouettes. In *Proceedings of ICCV'01*, Vancouver, Canada, June 2001.

[46] Z. Zhang. Flexible camera calibration by viewing a plane from unknown orientations. In *Proceedings of ICCV'99*, pages 666–673, Corfu, Greece, September 1999.