Object volume reconstruction and pose estimation from monocular video

by

Alireza Nasiri Avanaki

B.Sc. in Electrical Engineering - Electronics,
Sharif University of Technology, 1995.
M.Sc. in Electrical Engineering - Communication Systems,
Sharif University of Technology, 1997.

A THESIS SUBMITTED IN PARTIAL FULFILMENT OF THE REQUIREMENTS FOR THE DEGREE OF

Doctor of Philosophy

in

The Faculty of Graduate Studies

Department of Electrical and Computer Engineering

We accept this thesis as conforming to the required standard

The University of British Columbia

December 2003

Abstract

Acquiring a three-dimensional perception of an object or a scene, from regular (single-camera and 2-D) video, is a trivial task for humans. The automatic implementation of such a task has been, and still is, one of the major problems of computer vision. The new approach introduced in this thesis focuses on volume reconstruction of an object from image sequences taken by a single camera. One of the numerous applications of this approach is 3-D object tracking in video. This can be used in very low bit-rate customized video transmission schemes.

A multi-objective pose estimation method is introduced that computes object relative pose between two input frames. One advantages of this method is that it does not use any feature point, thus it does not suffer from problems with feature point detection and tracking. Also, the method does not assume any model for the object at the outset, hence it can be applied to an arbitrary object. The method, however, requires a depth-map, which is not readily available from an image sequence. To overcome this requirement, an iterative scheme is employed. The first round of pose estimation between consequent frames is performed, assuming flat depth-maps. Pose estimates are then adjusted to reduce the error by maximizing a novel quality factor for shape-from-silhouette volume reconstruction. Shape-from-silhouette is applied to construct a 3-D model (volume), which provides depth-maps for the next round of pose estimation. The feedback loop is terminated when pose estimates do not change much, as compared to those produced by the previous iteration. Based on our theoretical study of the proposed system, a test of convergence to a given set of poses is devised. To handle input sequences with unknown frame order, the input sequence undergoes a preprocessing stage, in which the frames of the sequence are re-ordered to obtain the most accurate pose estimation. A theoretical validity criterion for volume reconstruction by shape-from-silhouette is established. This criterion is used to produce a volume reconstruction quality factor, which plays an important role in pose estimation adjustment. The reliable performance of our system is proved via several simulations carried on both synthetic and real image sequences. Effects of pose sampling rate, distribution of pose samples, and error in input pose on volume reconstruction quality by shape-from-silhouette are studied. It is shown that high levels of pose error cannot be compensated by increase in pose sampling rate, and that volume reconstruction at high pose sampling rates is more sensitive to pose error.

Table of Contents

List of Tables

List of Figures

List of Abbreviations

2-D	Two-dimensional
3-D	Three-dimensional
CBIR	Content-based image retrieval
CBVR	Content-based video retrieval
DMSA	Deterministic minimum seeking algorithm
FDMPE	Flat depth-map pose estimation
GA	Genetic algorithm
HCI	Human-computer interface
MD	Mask difference
PTC	Preliminary translation compensation
RPE	Relative pose estimation
SFS	Shape-from-silhouette
TD	Texture difference
TRPE	Translation-compensated RPE

Acknowledgments

I wish to thank my Ph.D. advisor, professor Babak Hamidzadeh, for all types of support and intellectual guidance he provided me during my doctoral program. I wish to thank professor Rabab Ward for her invaluable advice and support. I would like to thank my co-supervisor, Dr. Faouzi Kossentini, for his kind support.

My appreciation and many thanks go to my parents, my dear wife, Zahra, and my sweet daughter, Niayesh. Completion of this work was not possible without their care and emotional support.

I am also thankful to my teachers, my colleagues at UBC, and people or institutions that helped me with my studies and their names are not mentioned here.

1: Introduction

Through the process of imaging, the three-dimensional (3-D) world is projected onto a two-dimensional (2-D) image plane. That is, a whole dimension is lost by imaging. Animals use vision, inarguably their most powerful sense, to recover the lost depth information, hence perceive and then interact with the world. The task of extraction or recovery of 3-D information for a subject from its 2-D images is called 3-D modeling, 3-D reconstruction, or, volume reconstruction.

For known subjects, the problem of volume reconstruction can be reduced to object recognition and model fitting. For unfamiliar subjects, however, we need more information to understand their 3-D structure. In order to do this, we use the so-called *visual clues*[1] (Figures 1.1 and 1.2), available in 2-D images of subject, that point out its original 3-D structure. All visual clues illustrated in Figures 1.1 and 1.2 are, however, *static*. That is, they do not utilize the fact that the input we receive from the real world is almost always a sequence of images, not a single image. The dynamic visual clue to the 3-D structure is the relative motion between camera and subject. Motion is the most general visual clue we use: it usually works when other visual clues are not present (e.g., unknown texture) or fail to function (e.g., stereo vision for far-camera field, or single-camera sequence). For example, when we encounter an unknown new object, we usually look around it to perceive its shape. Motion is also the most important visual clue available

[1] Also known as visual cues.

in a monocular (regular single-camera) video as this also helps us to understand the shape of an unfamiliar object shown in a video sequence.

Motivated by the ability of the human visual system to understand the 3-D volume of an object, shown in an image sequence, the problem of automatic retrieval of 3-D information from monocular video is addressed in this thesis. The significance of this attractive problem is highlighted through a number of its crucial applications as discussed in Section 1.2.

1.1 Problem statement

The central problem of the thesis is stated as follows.

> Volume reconstruction of a general arbitrary object shown in a monocular image sequence is desired.

1.1.1 Input

The monocular image sequences are chosen as input because of the following reasons.

- **Abundance.** Almost all of the existing image and video archives and databases are monocular. Selection of monocular image sequences as input allows for application of the proposed solution to the image sequences in such archives, hence, it allows the creation of enriched multimedia content from regular image and video.

- **Availability.** For objects that no longer exist (the head of a deceased person, for instance), pictures or video footages are the only data available. In

such cases, any other type of input (e.g., stereo or range images; Chapter 2) cannot be obtained.

- **Ease of capture.** The input required for capturing the volume of a new object is a video footage showing the object in various poses. Acquisition of such input is easy and inexpensive as compared to other types of input, such as multi-camera image sequences or range images, mentioned in Chapter 2.

Note that unlike video sequences, in which the frames are inherently ordered by the time of capture, the order of images in an image sequence may not be known. The resulted reconstructed volume, however, should be invariant with respect to the frame order in the input sequence.

1.1.2 Output

The reconstructed volume (3-D or CAD model) of the object is the desired output, which can be optionally textured, using the texture information available in the input images.

1.1.3 Subject

The subject is considered a general arbitrary object. It means we do not deal with scenes. This does not cause any harm to the generality of the problem, since a scene can be decomposed to several objects, which can be treated individually by the proposed method.

Our focus is on reconstruction of general arbitrary objects. That means we consider the subject unknown and cannot assume any model for it at the outset.

On the bright side, we do not have to worry about object recognition, which is a difficult problem by itself.

Furthermore, we also assume that the subject is opaque and rigid. Non-rigid subjects can either be decomposed to rigid parts, or approximated by a rigid body, plus small variations. The reflection from the object's surface is preferred to follow the Lambertian model [3] closely (i.e., not to be shiny). The Lambertian model requires uniform distribution of the reflected light in all directions. This preference is made to avoid generation of bright spots (reflected images of the light sources) that may not move according to the object motion. Note that this is not a strict requirement, since the system can handle bright spots as outliers.

Several classes of semantic models for video data are compared in [1] (Figure 1.3), based on the level of abstraction and the granularity of data processing. It is imperative to see where our methods fit in this general picture: our methods involve processing of frames (i.e., fine granularity of data processing). The information extracted by our methods from video is the relative pose of the object in each frame and a 3-D model of the object. Such information corresponds to an average level of abstraction. Therefore, our methods fall into the same region as "motion detection" (object tracking).

1.2 Applications

To demonstrate the significance of the central problem of this thesis (Section 1.1), we describe three crucial applications of the proposed solution in this section.

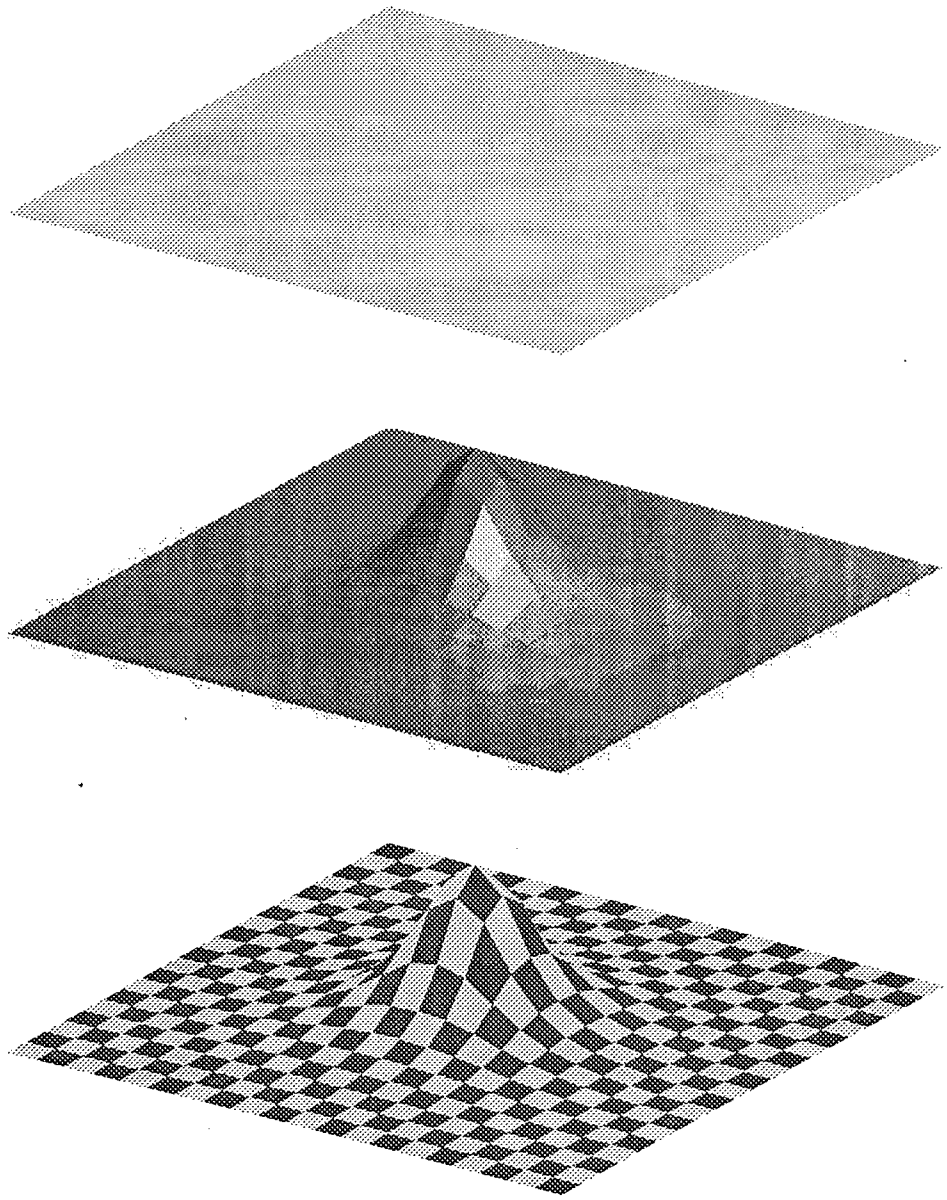


Figure 1.1: Some visual clues to depth information. There is no visual clue in the first picture. In the second picture, the same object is rendered under a light source: depth-clue is shading and reflection. A checkered (known) texture is mapped onto the object in the third picture.

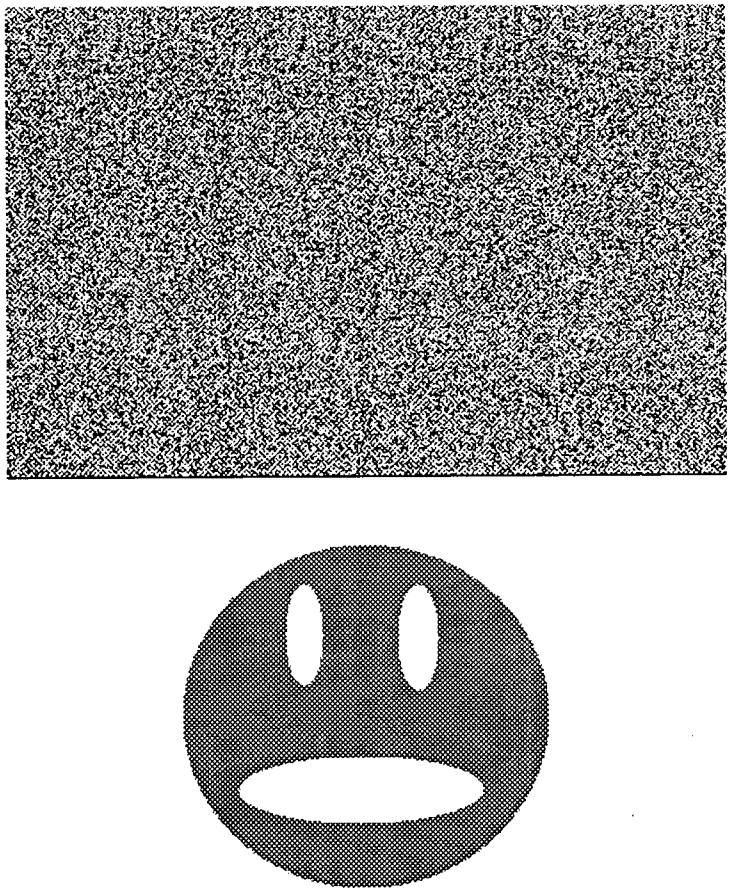


Figure 1.2: Random-dot stereogram (courtesy of coggeshall.org). The sole depth clue in stereograms is stereo vision. To see the depth structure, one has to cross his or her eyes so that the image seen by one eye partially overlaps with the one seen by the other eye. The successful viewer sees the figure of the second picture standing out of the background.

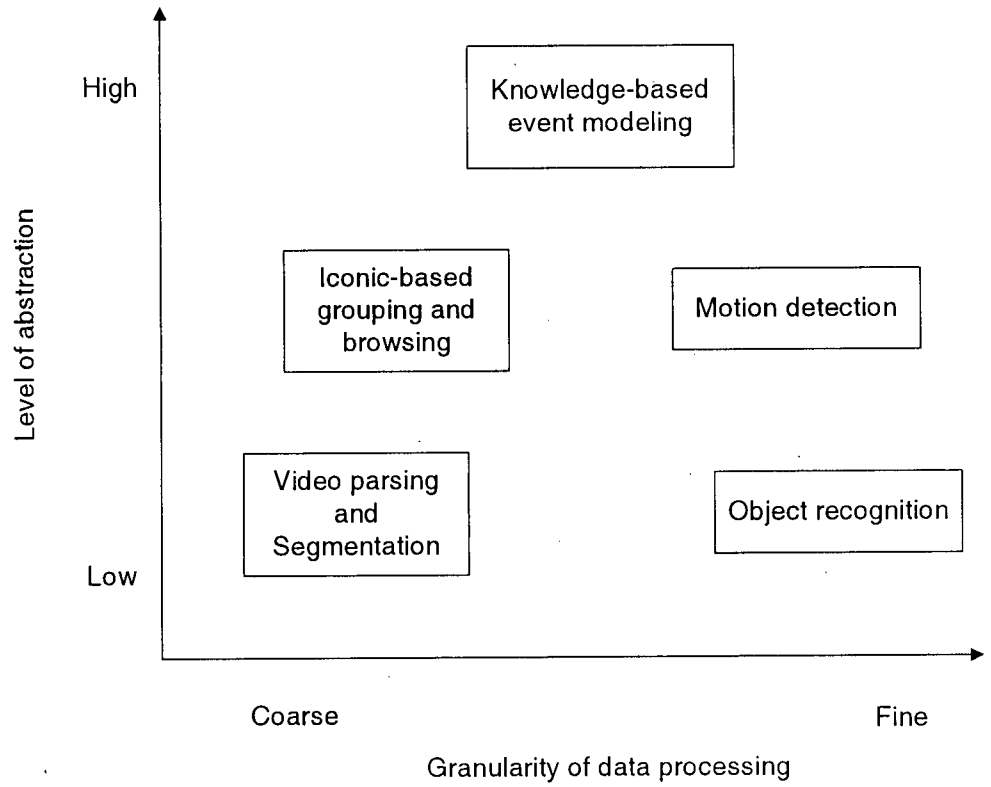


Figure 1.3: Comparison of video data semantic models [1]. Our methods fall into the "motion detection" category.

1.2.1 Content-based video retrieval (CBVR)

The wealth of information in images and videos, particularly in large archives, is almost useless if it cannot be searched and retrieved effectively. The traditional search and retrieval techniques, developed in database engineering, are suitable for raw data, that is the type of data with no distinction between the representation and the content (meaning). These techniques are not capable of handling images and videos whose contents are very different from the representation: unlike raw data, images and videos have rich high level meaning, while their low-level representation (e.g., at pixel level) does not carry the content. The research on content-based image and video retrieval began in response to this need.

One kind of information available in videos is the 3-D motion and structure of the subject. This information is required for high level scene/object description or representation, navigation through scenes, 3-D reconstruction of objects or scene, rendering images of scenes from arbitrary viewpoints and so on. However, to the best of the author's knowledge (Chapter 2), the state-of-the-art CBVR techniques are unable to fulfill queries about 3-D motion or geometry of subject.

The method proposed in this thesis can be used in the following fashion to equip an existing CBVR engine for handling 3-D-related queries. We first apply the method to every clip. This yields a 3-D model of the object, for each video clip, as well as an index of the amount of relative 3-D motion of the object for each frame in the clip. Using the newly generated index, the CBVR engine is now capable of answering new high-level queries, such as the following:

- Motion-related. Find (parts of) a sequence in which the object undergoes a specific type of motion, such as pure translation, a specific amount of

rotation at a given direction, or rotation at a given speed, etc.

- Structure-related. Find a sequence in which the 3-D model of the subject is "closely" approximated by a specified 3-D shape (e.g., sphere, or cube), or a given 3-D model.

Note that before being able to answer any structure-related query, a function should be defined for the CBVR engine to measure the similarity of two 3-D models. For an example, see the function that is used for measuring the similarity of the reconstructed volume to a sphere in Section 4.3.1.

The method can also be used, together with a content-based image retrieval (CBIR) engine, operating on an image database, for answering 3-D-related queries. In order to do this, the images in the database should first be classified, according to the subject shown in them, making a "virtual clip" for each subject. Thus, each virtual clip consist of every image that shows a certain subject in a variety of poses. The rest of the procedure is the same as the one just described for CBVR enhancement, operating on the virtual clips instead.

1.2.2 3-D scanner: building a virtual copy from (videos of) an object

The concept of a digital museum is that, instead of displaying an original object, which can be invaluable, its 3-D model is displayed, even on the Internet, and the original object is securely stored in a safe place. Also, in generation of virtual reality environments (e.g., for military training, and entertainment), we often need to insert a model of a real object in the virtual world.

In such circumstances, a laser range scanner is used to capture several range images (depth-maps) of the object from known viewpoints. The volume of the ob-

ject is made by stitching the range images together. The volume is then textured, using the regular images captured from the same viewpoints.

The drawback of this method is that it requires use of very expensive equipment (namely, a laser range scanner). The operation also has to be performed in a controlled environment. Finally, the method requires availability of the original object. Therefore, the method cannot be used at all, when some footage or a few pictures are the only things left from the object (Figure 1.4).

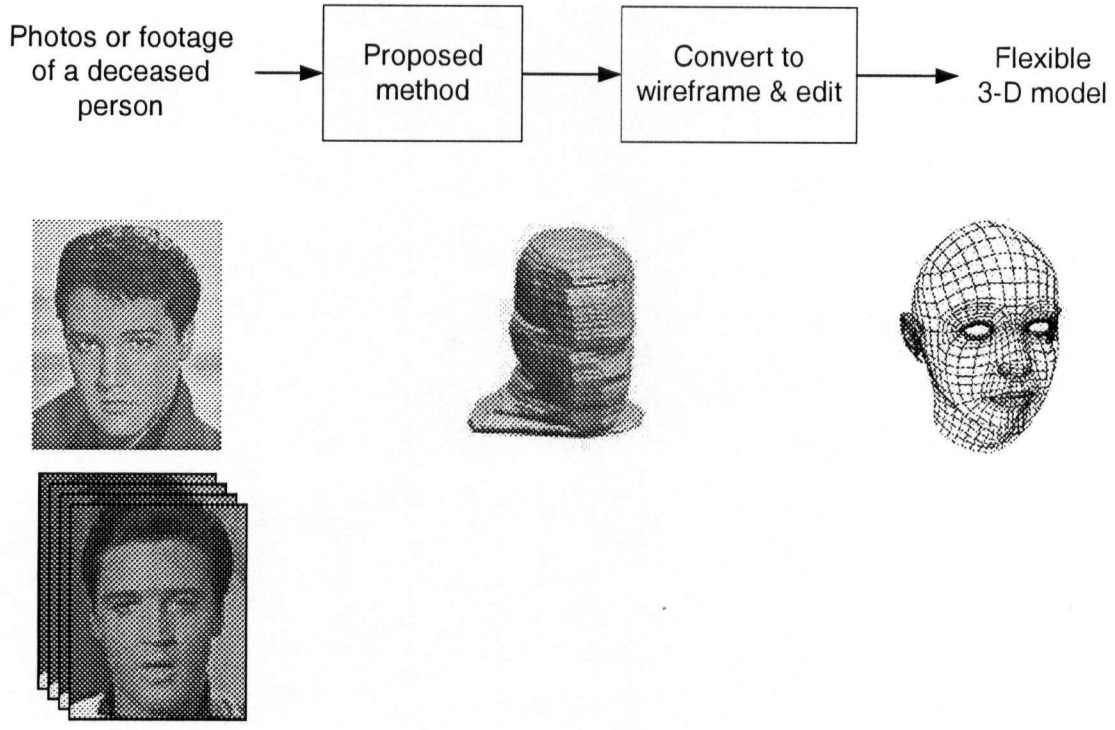


Figure 1.4: Head reconstruction and animation. In the example shown, the original object (head) is not available. The final result, a flexible model of the head, can be animated with the original or a modified texture (e.g., with added facial hair).

The method proposed in this thesis can be used at the centre of a 3-D object scanner. Such a 3-D scanner provides an inexpensive, versatile, and easy

alternative to the currently used procedure for object scanning.

1.2.3 3-D object tracking in video

In addition to a 3-D model of the subject, our proposed method generates an index of the 3-D motion of the subject for each frame of the input video sequence. Therefore, our method can be used as a 3-D object tracker.

Similar to the MPEG-4 video coding standard, our method can be used in a coding scheme (Figure 1.5) that provides a very low transmission bit rate for interactive video. By interactive video, we mean that the output video can be customized by the user (see receiver side in Figure 1.5). The difference with the MPEG-4 standard is that, here, the objects and their motion parameters are 3-D rather than 2-D.

Note that the transmission bit rate can be reduced even further by replacing the 3-D models inputs of the scene rendering block with generic 3-D sprites that are locally available. Thus, the one-time transmission of the 3-D object models are no longer necessary, leaving only the motion parameters to be sent, a task which requires only an ultra-low amount of channel capacity.

1.3 Thesis organization

The remainder of this thesis is organized as follows. Chapter 2 reviews the research work related to the subject. Chapter 3 introduces the method of multi-objective relative pose estimation. In Chapter 4, the method of shape-from-silhouette (SFS) is reviewed; the effects of pose error and sampling rate on reconstruction quality are discussed; and a validity criterion for volume reconstruction by SFS, along with its application in pose estimation, are introduced. Chapter 5 presents the volume

feedback loop structure that, among its other merits, eliminates the depth-map requirement for relative pose estimation. The thesis is concluded in Chapter 6 with an extended summary, limitations, and suggested lines of future work.

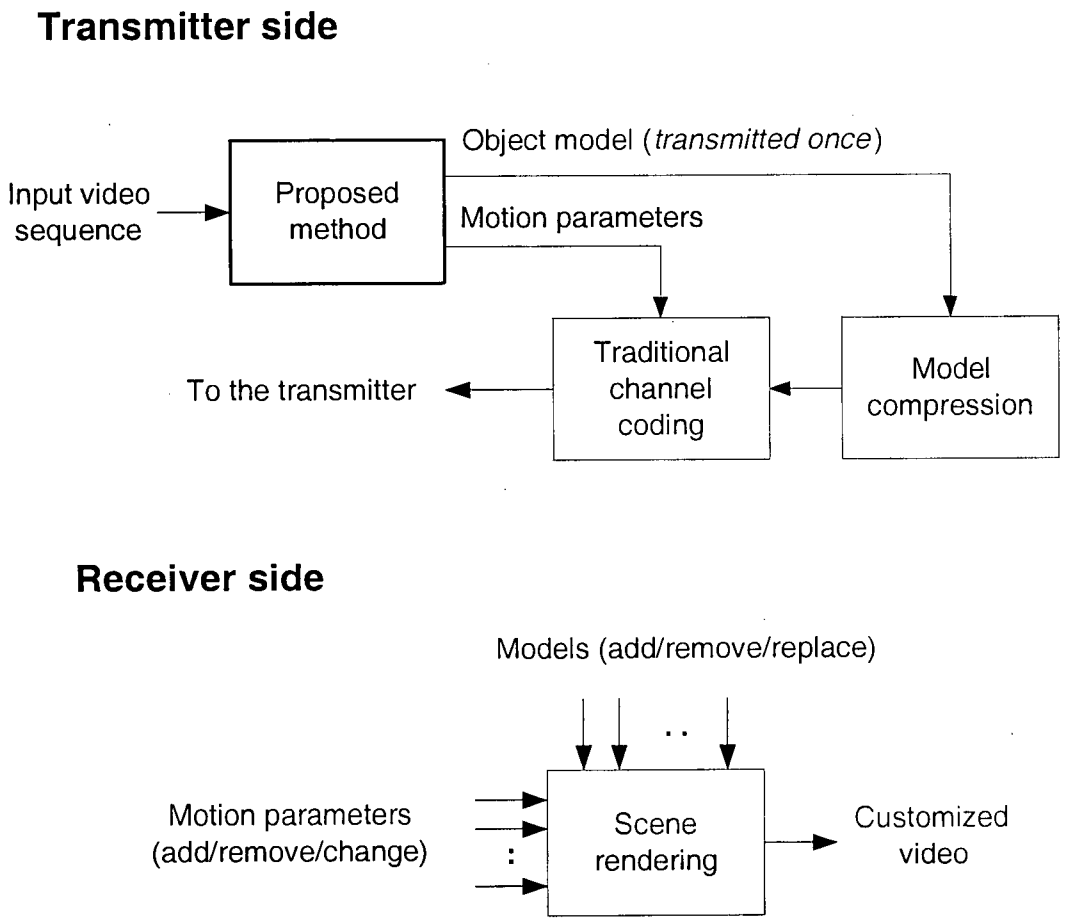


Figure 1.5: A very low bit-rate 3-D video coding scheme for interactive video transmission, an example of the proposed method application in 3-D object tracking in video. The 3-D model of the object is transmitted only once, and sending the motion parameters only requires a low bit-rate. Therefore, for longer sequences, the bit-rate will be lower. At the receiver, the user may have several streams coming from different sources for different objects. The user can control all inputs to the scene renderer, in order to customize the output video (hence the term "interactive" video coding is used).

2: Related research

The central problem of this thesis (defined in Section 1.1) lies on the borders of computer vision, computer graphics, and image processing with direct applications in content-based image and video retrieval. In this chapter, we briefly review the related literature in these fields to show the recent advances and the diversity of approaches to relevant problems. Thesis contributions, along with a comparison between the existing and the proposed methods, conclude this chapter.

2.1 3-D volume reconstruction (3-D modeling)

To begin with, we should clarify our meaning of the terms "3-D volume reconstruction or 3-D modeling". By these terms we mean the act of making a whole, solid 3-D model of an object. Figure 2.1 shows the relationships among different (classes of) methods relevant to the subject, their inputs and their outputs. In this section, we review and compare some representative methods, in the order of their output type. Methods that do not take monocular video as input (e.g., range imagery), or do not produce a solid volume as output (e.g., feature-based methods) are also discussed, for comparison.

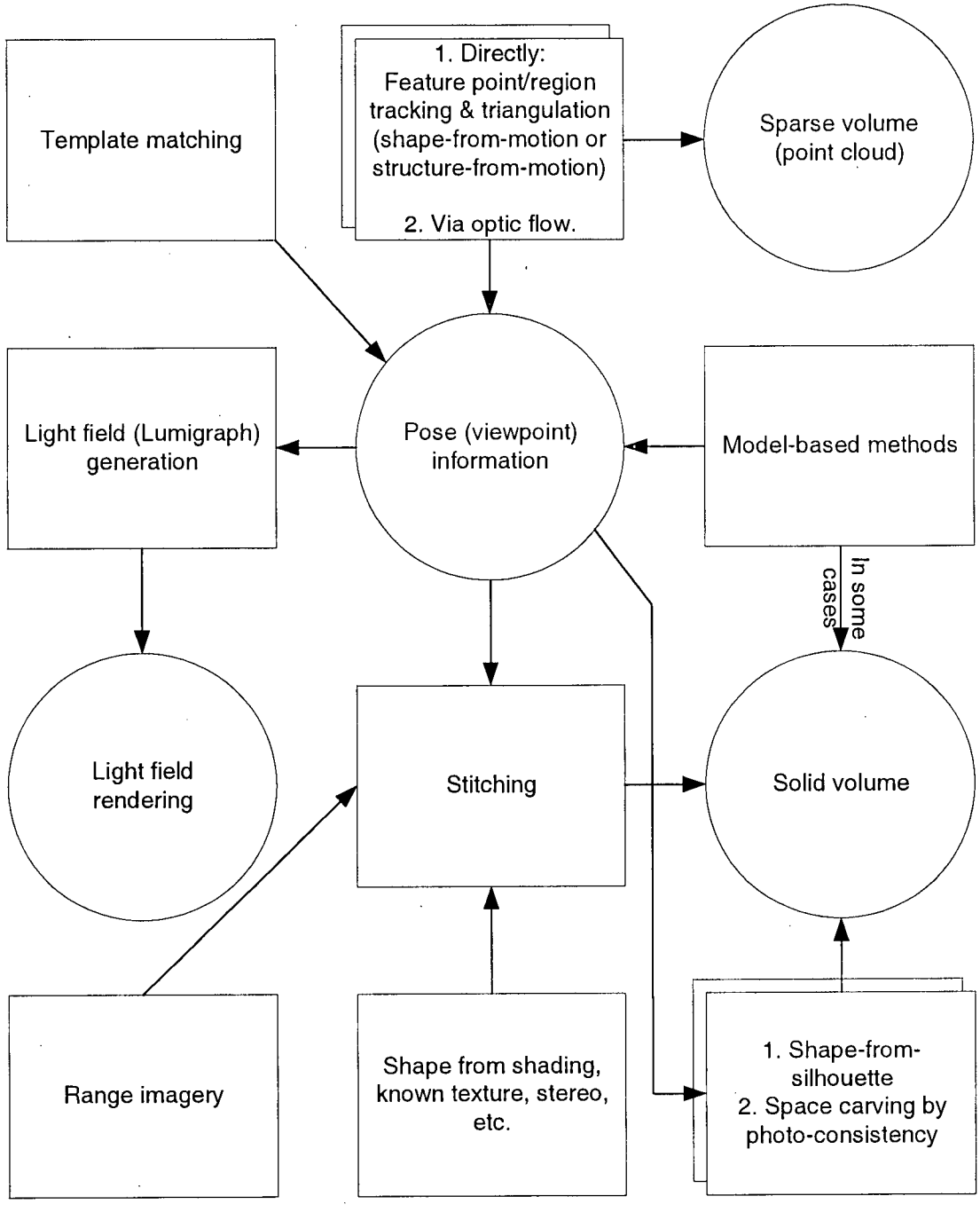


Figure 2.1: Topics in the literature that are related to 3-D volume reconstruction, and their relationships. Methods are shown by rectangles and data (input or output) are shown by circles. The common input to all methods, except for some cases of range imagery (bottom left), is the image sequence of the subject. Note that pose information (at the centre) is almost always required for volume reconstruction.

2.1.1 Solid volume output

Much work in computer vision has been devoted to extracting a *depth-map*[1] from various visual clues. These methods have a categorical name: "Shape from X", where X is the visual clue based on which the depth-map is computed. These methods extract the part of the object's external surface that is facing the camera. Needless to say, not only many depth-maps are required for reconstruction of the object, but also a tool is necessary to integrate (stitch) all surface pieces into a seamless volume.

Some visual clues used to make depth-maps are stereo, known texture, contour, shading. See [4, 5] on how these clues and their combinations are used to derive depth-maps. Stereo methods need pictures taken from two (or more) cameras having known relative geometry. Use of shading requires the reflectance map and the lighting of the scene to be known. Another class of algorithms requires additional information about texture of, or contours on, the surface of the object, which are not usually available. Given the input described in the problem statement (Section 1.1), we realize that our solution cannot be based on any of these clues. Regular video footage is not taken by stereo cameras; the object's texture and the lighting of the scene are not known a priori. Therefore, we do not discuss the literature on such methods any further.

The problem of integrating surface pieces has a 2-D counterpart called "image registration or alignment", which has been solved through several approaches [6, 7, 8]. Such methods are able to put small images, each showing part of a large

[1] For a pixel that represents an actual point on the object, depth value is defined as the distance between the camera and the corresponding point on the object. The depth-map of an image gives the depth value for each pixel in the image.

image, together to make the larger one. Unlike a jigsaw puzzle, whose pieces do not have any part of the large image in common, the input images to registration methods need to have some overlap. In fact, the common part should be the major area of the images or the methods fail to function. We did not find any registration method performing the same task for surface pieces. What we did find [9] helps to integrate a set of points in 3-D space into a surface. In other words, this method requires the poses of depth-maps (their places in the 3-D space) to be known. Poses are not usually known and should be estimated.

To date, the most accurate approach to 3-D modeling is "Range Imagery" [10, 11]. In this approach, the original object is scanned by laser range scanners from several viewpoints. The captured range images[2] are then integrated to a seamless volume of the object. The volume can be textured arbitrarily (e.g., with the images taken from the object).

The silhouette of an object from a viewpoint is the same as the extremal boundary of its image from the same viewpoint. Seales [12] builds a 3-D model of an object using the extremal boundaries of its images from various viewpoints. This task is performed by extracting all edges and classifying them into two categories, namely fixed edges (physical edges, e.g., the edge between two adjacent faces of a cube), and occluding contours (extremal boundaries that do not necessarily correspond to any physical edge, e.g., silhouette of a sphere). The object pose is then estimated by tracking the fixed edges. Knowing the pose of each occluding contour, the corresponding piece of the object's surface can be recovered. Finally, all surface portions are put together into the whole object surface, which

[2] A range image is not necessarily accompanied by a regular image. A depth-map, however, usually comes with a regular image of the object, since it is derived from a regular image using visual clues.

can be manipulated with traditional graphics tools. Local surface reconstruction, using the extremal boundary of the object, was addressed for the first time in [13]. The problem with [12] is that it requires a trinocular image sequence of the subject (i.e., all views of the subject should be captured by a camera rig consisting of three cameras). The method of [14] has the same problem. Such methods, therefore, cannot be applied to common single-camera video or image sequences.

There are also SFS methods that reconstruct the object volume, instead of its surface. The basic idea is to build the smallest bounding volume for the object. The task is accomplished by realizing the intersection of cylindric (or conic) volumes, made by stretching the silhouettes (extracted from the input images) in the direction of the viewpoints (from which the image is taken) to the origin and beyond, to obtain the part of the 3-D space that is shared by all of those volumes. The result is intuitively the tightest possible bounding volume for the object, considering the available viewpoints. The choice between conic or cylindric types of volume depends on whether perspective or orthographic projection is assumed. The latter is a special case of the former in far-camera cases. In [15], an algorithm for fast implementation of the method, using an octree representation of the object, is developed. The work is mainly based on [16], which, for the first time, introduced 3-D volume reconstruction out of images taken from arbitrary views. The object surface can be obtained from the reconstructed volume, as described in Section 4.2, where a graphical description of volume reconstruction by SFS is also given.

After comparing to other 3-D reconstruction methods, we found this class of methods well suited to our problem. That is because it can work with monocular image sequences; it provides a scalable solid volume reconstruction (i.e., it gives

the best possible model with any amount of available data); and it functions for arbitrary rigid opaque objects (does not assume any model for the subject). None of the other methods discussed so far, and in the remainder of this chapter, can fulfill all of these conditions. Hereafter, when we use the terms "SFS" or "SFS-based (volume) reconstruction", we mean the method that reconstructs the object volume out of the input silhouettes and is elaborated in Section 4.2.

Another approach to 3-D modeling is space carving by photo-consistency (e.g., [17, 18]). The basic idea of this approach is an extension of SFS: elements of the prototype volume (originally opaque) are projected to pixels on the input images (viewpoints are known), and are coloured accordingly. The volume elements that do not project to any pixel, are made transparent (or carved out, hence the name space carving). An advantage of this approach over SFS is the capability it has to reconstruct concave objects. The method is, however, much more complex computationally.

2.1.2 Light field rendering

In 1996, Levoy and others [19] introduced "light field rendering," a new approach to 3-D object modeling for use in computer graphics. Gortler and others [20] also reported a similar approach at the same time. These methods attempt to comprehensively sample all possible appearances of the object: the input images are considered as the 2-D slices of a 4-D function called "light field". The light field, which is the 3-D model of the object, should be constructed by sampling the space of possible appearances densely. To retrieve a specific object appearance, one should compute a slice of the 4-D light field at the desired position and orientation. The problem with this approach is that, in order to construct a

usable light field, a large number of samples (viewpoints) are required. Therefore, these methods cannot be applied to cases where a whole lot of views are not available in the capture phase.

2.1.3 Sparse volume (point cloud) output

See Section 2.2.1.

2.2 Pose estimation methods

All 3-D reconstruction approaches, discussed in Section 2.1, share a common requirement: they all need the poses (camera viewpoints) of the input images to be known. Such information is readily available, when the input images are taken in a controlled environment (e.g., camera position is controlled by a robot, or its location is estimated by detection of special markers, which are put in the scene for this purpose, in the image). We do not, however, assume a controlled input sequence. Therefore, the pose information should be estimated using one of the following approaches: model-based, template matching, and feature-based methods. Each class of methods is described below.

2.2.1 Feature-based methods

These methods are based on the premise that some points on the objects are special: side tips of the mouth and the pupils of the eyes are examples of such points in a human face. These points, which can be easily recognized (detected) and tracked through multiple images, are called feature points.

When an object moves (relative to camera), the projections of the feature points move in the images of the object. Assuming the object is rigid (i.e., feature

points are moving together coherently), one is able to derive the trajectory of the points and their relative geometry (a "point cloud" or sparse volume). In some simple situations (e.g., reconstruction of a planar object), this information is enough to estimate the structure of the underlying object. Other names for this problem are "shape-from-motion" and "structure-from-motion". The camera should be calibrated at the outset, or calibration should be performed in parallel.

As an advantage of this class of methods (e.g., [21, 22]), we can mention simplicity of concept and implementation. However, the number of feature points are not usually enough to describe the structure of the object to be modeled to the desired resolution. So, one need to use less easily-recognized points to achieve a higher resolution. In that case, the task of feature points detection and tracking in multiple images becomes the a substantial problem. This problem (also addressed in stereo imaging) is known as (feature) point matching or point correspondence, and in spite of the large volume of research on it, the methods are still computationally intensive and prone to error.

Some examples of the feature-based methods are briefly discussed in the following. Paraperspective (a simplified perspective) projection of feature points is assumed in [23], rather than orthographic projection, so that in addition to far-camera motions, near-camera and in-depth motions of the feature points can be tracked as well. An improved point correspondence method with a reduced number of mismatches is addressed in [24]. Extended non-linear Kalman filters are used in [25, 26] for motion and structure estimation from feature points. Camera calibration and motion estimation from feature points are performed simultaneously in [21].

Another class of methods for motion and structure estimation operate on

the optic (or optical) flow as their input. The concept of optic flow is loosely defined [27] as "apparent motion of brightness patterns, due to relative motion of camera and the object." The computation of optic flow is not a well-defined problem. Therefore, an additional constraint is added to derive the desired result. A usual constraint is smoothness of the resulting optic flow. One problem is that in multi-object scenes (even an object and the background), this constraint does not hold true on the boundaries. Moreover, the computation of optical flow involves numerical differentiation, which is a noisy process. Most accurate methods for optic flow computation rely on corresponding "feature-regions" or points in consequent images, which, as mentioned before, is not a straight-forward task. As a result of this, we put the optic flow class of methods in the feature-based category. Below are examples of the related literature in this area. Adiv [28] takes a single optic flow field as input and derives motion parameters and depth-map of the subject, which is assumed to be a single rigid object. In a similar work [29], computation of the optic flow from consequent frames is also included in the method. In [30, 31], attempts are made to analyze and alleviate the effects of noisy input (optic flow) on accuracies of the estimated the structure and motion. The input optic flow might be induced by several independent moving rigid objects. These multi-object motions are a source of severe errors in structure and motion estimates, that are also studied.

Feature based methods cannot be used effectively for 3-D reconstruction of sophisticated objects. Since there are a limited number of real feature-points in objects, such methods either resort to derive only a sparse structure of the object, or sacrifice the reconstruction accuracy for a higher resolution.

2.2.2 Model-based methods

Methods are reported that estimate the parameters of a 3-D model for a planar surface. The method in [32] requires poses of the input images to be known. The work in [33] tackles the same problem with unknown pose input images, but it is not stable. Both of the methods have a main concept similar to that of optic flow-based methods (Section 2.2.1); they get around optic flow computation, using simplifying assumptions of rigidity and being planar for the subject. The former assumption is very restrictive, making such methods inappropriate for our problem.

Some feature-based methods are designed for specific domains. Such methods put their knowledge of the domain into a model, making the task of structure estimation merely parametric. We use human head (face) pose estimation as an example. As an application to face pose estimation, the user's face pose (and gaze) can be used to make a convenient human-computer interface (HCI).

The methods developed for human head (face) pose estimation rely on detection of a number of feature points on the face (e.g., center and tips of eyes, tips of nose and mouth) and on tracking those feature points in image sequences [34, 35]. For pose estimation, the geometry of these points is compared with the geometry of such points stored in the database, thus deriving an estimate for the face pose. In [36], the relationship between visible regions of skin and hair is used to determine the pose. The method fails for bald people, who do not have the usual hair region. An advantage of this method is that it does not depend on (facial) feature point detection or tracking. In [37], the authors use a similar technique for gesture recognition. In [38], the authors use the eye contours to estimate poses of the head. A relatively different method [39] uses neural network processing

on regions of the face that are automatically extracted, based on colours. After the learning phase, the system is able to estimate the head and face pose. The severe problem with the method is that it should undergo learning, each time the pose of a new person's head is to be estimated. Bozdagi and others [40] fit a 3-D wireframe model to the head, shown in the input image sequence. The pose of the person's head is estimated by the pose of the fitted model's head.

Model-based methods cannot be used for pose estimation of a general object. They are, however, optimized to perform efficiently for their destined objects.

Some pose estimation methods assume depth-map information is available for one frame [41], or all frames [42] in the input sequence. Although these can be used for the pose estimation of a general object, these methods fall into the model-based category since the depth-map is actually information about the 3-D structure of the object.

2.2.3 Template matching

In methods based on template matching (e.g., [43]), the appearances of the object in various poses are recorded in a learning phase. In the operation phase, the appearance of the object, with an unknown pose, is compared to the recorded (known pose) appearances. The pose of the object is then estimated, based on the recorded appearances that are most similar to it. The problem is that it is not always easy to capture the appearance of an object from all poses. Moreover, if the desired pose estimation precision is doubled, the number of appearance captured from different poses in the learning phase should be increased 4 to 8 times, depending on use of 2- or 3-variable poses. The bottom line is that template matching is more a technique for pose recognition than pose estimation.

2.3 Content-based image retrieval (CBIR)

Before development of content-based techniques, retrieval systems relied on annotations (metadata) of each item of the database to answer a query. The task of annotating was done by an expert. For large image databases, such a task is very cumbersome and labor-intensive. Moreover, images contain much more information than one or two paragraphs worth of annotation. In other words, images can be interpreted differently, based on viewer's purpose or perspective. Therefore, static annotations or keywords, written once in the past, cannot necessarily play their role in the future. Thus, it would be useful if annotation could be performed automatically or not performed at all. That is the motivation for CBIR. A good perspective of this field can be found in [44].

The main concept behind a typical CBIR system is to retrieve a number of images from the database in response to user's query about the contents of images. The use of image processing methods allows utility of information residing in images, rather than annotations, for indexing the image database. There are two types of queries: syntactic and semantic. A semantic query is natural to humans, while a syntactic query may not be. For example, the query "Find an image with 30% or more pure yellow coloured pixels" is syntactic, and the query "Find an image of a snow-capped mountain" is semantic. There are several primitive features (such as area and circumference of the object mask, that area divided by that circumference squared, chain code or Fourier descriptor for that circumference, colour histogram for the object, etc.) that can be computed for an image, using image processing techniques. While syntactic queries can be answered, using only these pre-computed features, semantic queries should be

first decomposed to several syntactic ones and then put forward to the system's search engine. Not all semantic queries can be broken down to syntactic ones. That is why generic CBIR systems still use text annotations or other cues from images' content, if any is available.

Attempts to close the loop of searching for an image, using a CBIR system, by putting the user as the feedback box, have been made [45, 46]. This technique is called "relevance feedback". A CBIR system, equipped with relevance feedback, retrieves a specific number of images having the highest relevancy to the user's query. As the meaning of "relevance" is different for the user and the machine, the user should select some of the output images which are relevant, to help the machine to understand his or her intention. The system retrieves relevant images again, but this time with a better idea of relevance. This process is repeated until the desired images are retrieved or the user stops.

The query types that current CBIR systems support are given in the following. Colour distribution in the form of colour histogram can be handled by all systems. For gray-scale and binary images, this role is played by texture, distribution, or density of pixel whiteness. Sketch or outline is another type of query: the user depicts the outline of the desired object and the system will retrieve some images having the most similar sketch to the input. See [47] for an online showcase. Some methods [48, 49] allow for spatial relationships (relative location) among the objects (shown in the image) to be queried. A category of queries, which is built on the above mentioned concepts, is the "same X", where X can be colour, texture, or outline. The input to such a system is an image and the desired output is image(s) with the same property specified in the input image. A very high level type of query, made possible by the relevance feedback tech-

nique, is "I can recognize what I want, when I see it". To fulfill this query, the system provides the user with some randomly chosen images from the database; the user selects those which are (most) relevant to what he or she has in mind. This process is repeated until the favorite matches are found.

There are many image search and retrieval systems available. Many of them have CBIR engines. QBIC (Query By Image Content [50]) is a good example: it was one of the first practical CBIR engines that is now a commercial product from IBM. It accepts most of the query types discussed above. An online demo is available at [51].

Before concluding our account on CBIR, we would like to mention some works on how CBIR systems can be evaluated and compared to each other. A couple of quantitative measures of performance are defined in [52]: "recall" is ratio of the number of relevant images retrieved to the total number of relevant images in the database; "precision" is the number of relevant images retrieved divided by the total number of retrieved images. A system has a better performance if both of these ratios are higher. In [53] some qualitative measures are defined.

2.4 *Content-based video retrieval (CBVR)*

The simplest approach to CBVR is to consider each video clip in the database as a set of frames, and treat the collection of frames as an image database and run a CBIR engine on it. The problem with this approach is that due to the very large number of frames (5 minute footage = 9000 frames), the resulting image database becomes very large. Therefore, the CBIR engine cannot work efficiently.

A solution to this problem is to reduce the number of frames put into the image database. This can be done by keeping only one (or a few) representative

frames for each shot (a group of frames showing actions, continuous in time and space), in the image database, and removing the rest. This approach is called "shot and key-frame(s)". A number of shot detection methods are compared in [54]. A clue for detection of shot boundaries is the sudden changes in frames that consist of considerable variations in some or all primitive features. Some of the shot parameters (e.g., duration) are also useful and are stored in the database as metadata. Then a CBIR engine is run on the resulting image database. This is the way videos are treated by QBIC [50], a CBIR engine which is extended to a CBVR system.

The main disadvantage of "shot and key-frame" method of video representation for CBVR is that it cannot be used to answer spatio-temporal queries. A query type in this category is the "animated sketch" introduced in [55]. In addition to texture, colour, and sketch of the object being sought, the user can specify the motion and zoom factor change of the object. The system rejects possible sketch matches if their motion, considering previous and next frames, do not conform with the motion specified in the query. Answering queries of this type, which is the actual strength of CBVRs over CBIRs, is an advantage for the object-based approaches discussed in the following.

Shot detection can also be performed, based on objects: objects in the videos are detected and tracked and when they appear or disappear, we are on a shot boundary. This kind of shot detection involves segmentation of a video into object(s) (and background), and tracking each of the objects during their life-span through the video. The currently used tracking methods do not actually track the (3-D) objects: they just track the 2-D (deformable) mask of the objects. That is why these methods cannot provide 3-D information about the tracked objects.

Much work (e.g., [56, 57]) is done and is being done in this area because of its immediate application in object-based video representation and manipulation within the MPEG-4 [58] standard. For example, in [57] object tracking is practiced by finding the texture content of the initial object mask in the next frames, and collecting the regions found to make a new object mask, ensuring the spatial integrity of the object mask. The method is designed with the aim of extracting MPEG-4's Video Object Planes (VOP).

Note that to our best knowledge, none of the existing approaches to CBVR the retrieval of 3-D structure or motion of objects from video.

2.5 Thesis contributions and comparison with existing solutions

A summary of contributions made in the thesis follows. For more details see Section 6.1, or the corresponding chapters.

Chapter 3: contributions

A method for multi-objective relative pose estimation (with depth-map) is introduced and experimentally validated. Two pose similarity measures are also introduced and compared. This work is published in [41].

Chapter 3: comparison with existing methods

While providing pose estimates with a good (superb, in some cases) accuracy, the proposed method has the following advantages over the existing pose estimation approaches.

- It is not model-based. Therefore, it can be used for an arbitrary object.

- Unlike template matching methods, it does not require a learning phase prior to operation.

- It is not vulnerable to the usual problems of feature-based methods, such as feature-point detection (or selection), tracking, and low density of feature points.

Chapter 4: contributions

The effects of pose sampling rate, viewpoint distribution, and error in input pose on volume reconstruction quality (experimental work), along with a validity criterion for SFS-based volume reconstruction (theoretical work) are studied. The latter study yields a quality factor for SFS-based volume reconstruction that is used in stand-alone pose estimation and volume reconstruction method of Chapter 5. Part of this work is published in [59], and the rest in [60].

Chapter 4: comparison with existing methods

To our best knowledge, there are not any comparable work in this area.

Chapter 5: contributions

A method for stand-alone simultaneous pose estimation and reconstruction by volume feedback is introduced. The method is successfully tested with both real and synthetic input sequences. Parts of this work are published in [61, 62, 63], and as a whole in [60]. The details follow.

- A theoretical convergence study of the system provides a test for convergence to the given poses.

- Computational complexity of the system is studied (theoretical work).
- The effects of sampling rate and spatial (voxel) resolution on pose estimation error are studied.
- As compared to the multi-objective relative pose estimation [Chapter 3], the method has the following advantages.
  - No depth-map is required as input.
  - A new pose similarity measure is introduced that is functional for all types of objects.
  - A two-step translation compensation scheme, based on single vector colour optic flow computation, is employed. This allows for translation of the object in the input sequence as well as rotation.
  - The input sequence undergoes a pre-processing stage, in which the frames of the sequence are re-ordered to obtain the most accurate pose estimation, hence volume reconstruction quality. This allows for handling input sequences with unknown frame order.

Chapter 5: comparison with existing methods

The method introduced in this chapter can be used for pose estimation, volume reconstruction, or both. When used as a pose estimator, the proposed method has all advantages of the method of Chapter 3 over existing methods, plus the merits mentioned above.

All existing approaches to volume reconstruction require pose information for input images (or silhouettes). The proposed method, however, only needs an

image sequence of the object as its input. As compared to light field rendering method, which requires a comprehensive sampling of viewpoints, the proposed method has the advantage of being scalable. That is, volume reconstruction can be performed with any number of input images, and the reconstruction quality can be improved by providing images from other viewpoints.

If one accepts point-clouds as volumes, the feature-based methods can be considered as functionally compatible with the proposed method, in the sense of performing pose estimation and volume reconstruction at the same time. Our method, however, does not suffer from inherent problems of feature-based methods (mentioned above), and its output is a solid volume regardless of feature point density.

3: Multi-objective relative pose estimation

3.1 Introduction

In this chapter, we introduce a method for estimating the relative pose of a rigid object shown in an image (typically taken from a video sequence), with respect to a reference pose of the object. The method can be used to index a video sequence, based on the object pose. This, in turn, can aid 3-D information retrieval from video. We evaluate different objectives for measuring pose similarity and different optimization algorithms for finding a match between a rotated 3-D reference patch and an unknown pose image.

A few terms that are used throughout the chapter are defined in the following. The relative pose of the object shown in frame B with respect to frame A, is the amount of rotation needed to take the object from its pose shown in frame A *(reference)*, to the pose shown in frame B *(target)*. We abbreviate this parameter as the *relative pose* between frames B and A. For each frame, an *absolute pose* is defined as the pose of the object with respect to some global (fixed) reference for the pose. Thus, the relative pose between frames B and A can also be defined as the difference between their absolute poses.

3.2 The method

The input of the method is a sequence of images (or frames) of an object. the poses of the object in the input frames is to be determined. The object undergoes 3-D rotation in the sequence. The poses are to be computed relative to a reference pose in the input sequence for which we have also a depth-map. We use the depth information to rotate part of the object visible in the reference pose and compare it with the unknown-pose image. When the correct pose of the object in the input sequence is encountered, the rotated part of the reference pose becomes very similar (ideally coincides) with the unknown-pose image of the object.

The idea of this method is inspired by [64]. Koch describes a method of 3-D scene analysis by synthesis. We use this idea in our pose estimator, where the transformation is performed on the reference image (projected on the depth-map), and the rendered result is compared to the real image with the unknown pose, to estimate the pose. The structural difference of this work with [64] is that Koch assumed a parametric model for the subject (human face and shoulder); we, on the other hand, do not assume any model for the object to keep the method functional for arbitrary objects.

The block diagram of our pose estimator is depicted in Figure 3.1. I_1 and I_2 indicate the image of reference pose and an image of unknown pose, respectively. The depth-map of I_1 is assumed to be known. The output, P, is the pose estimated for I_2. To compute the pose, the system rotates I_1, a 3-D surface patch, to match I_2. The rotated I_1 is projected into the image plane and segmented. Then, it can be compared to I_2. Comparison can be based on the object masks of the two images or the combination of that and the texture difference of the images. A

threshold is imposed to decide whether or not the two masks coincide (i.e., the amount of rotation is enough). The loop is repeated until the pose is estimated (P).

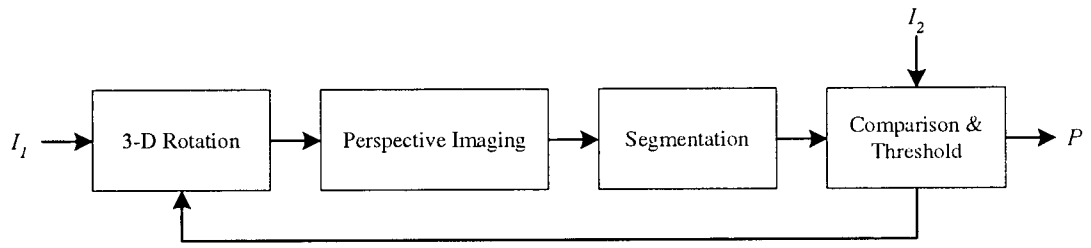


Figure 3.1: The block diagram of the proposed pose estimator. I_1 is the reference frame with known depth-map. P is the estimated pose of I_2 w.r.t. I_1.

In the following, we outline our assumptions. One assumption is the availability of the depth-map for the reference frame. A depth-map can be obtained by a stereo technique (e.g., triangulation) anytime the object undergoes a known amount of pure translational motion in the input image sequence. The frame ending such a motion can be considered as the reference frame. We also assume that the motion of the object from its reference pose (that in I_1) to the pose in I_2 is a pure 3-D rotation. Finally, we assume that the weak perspective projection closely approximates the imaging process, and the object is opaque and rigid.

3.2.1 3-D rotation

The first module of the system is in charge of 3-D rotation of the 3-D patch of I_1. Let a point specified by a vector $r = (x, y, z)^T$ rotate around the origin and reach $r' = (x', y', z')^T$, where ^{T} denotes transposition. r' can be computed from r by $r' = Rr$. R can be determined by the amount of rotation around each axis. It

can be shown easily that $R = R_x R_y R_z$, where

$$R_z = \begin{pmatrix} \cos\phi & \sin\phi & 0 \\ -\sin\phi & \cos\phi & 0 \\ 0 & 0 & 1 \end{pmatrix},$$

$$R_x = \begin{pmatrix} 1 & 0 & 0 \\ 0 & \cos\theta & \sin\theta \\ 0 & -\sin\theta & \cos\theta \end{pmatrix},$$

$$R_y = \begin{pmatrix} \cos\psi & 0 & -\sin\psi \\ 0 & 1 & 0 \\ \sin\psi & 0 & \cos\psi \end{pmatrix},$$

and ψ, θ, and ϕ are the amounts of rotation around X-, Y- and Z-axes, respectively.

I_1 is generated from a 2-D image and the corresponding depth-map by first fitting a surface to the depth-map. Then, the surface is textured with the 2-D reference image.

3.2.2 Perspective imaging

The imaging module is a virtual pinhole camera. It makes a 2-D perspective image out of the 3-D object. According to the notation introduced above, the projection of a point r in 3-D space to the image plane is (X, Y), where

$$\begin{aligned} X &= \frac{z_0 x}{z}, \\ Y &= \frac{z_0 y}{z}, \end{aligned}$$

and z_0 is the vertical distance from the origin of the object space (I_1 location) to the image plane. With no loss of generality, we take $z_0 = 1$.

While computing a perspective image of a 3-D scene from a certain view, one should take care in identifying which parts of the object occlude the other parts from that view. A standard technique in computer graphics for performing such a task is z-buffering [65]. Since the graphics functions in Matlab, which we use for our experiments, provide mechanisms for modeling occlusion transparently, we do not elaborate on this matter any further.

3.2.3 Segmentation

Although we assume segmentation is already performed, the object masks are still required by our algorithm and should be computed. To fulfill this need, we paint sides of the object not visible in the reference frame as well as the background, by a single colour not often seen on the object. Then the object mask for the rotated I_1 is easily computed by separating those pixels having a colour different than the background. This method may set aside single pixels having exactly the same colour as the background from the mask. The remedy is to process the mask by a morphologic filter [66, 67] to cover small uncovered areas within the mask. The object mask for I_2 and the initial object mask for the image of I_1 (before integration with a textured surface) are made semi-automatically. These images can be segmented automatically, if the footage is made in a controlled environment (e.g., having a uniform colour and lighting in background).

3.2.4 Comparison and threshold

This module is the main part of the system. We investigate two methods to compare the rotated 3-D mask (rotated I_1) to the unknown pose image, I_2. The first method is to compare the object masks, which are binary images, in I_1 and I_2 by computing the area of the regions where they do not overlap. That is, the area of $(I_1 \cup I_2) - (I_1 \cap I_2)$, where $\cup$ and $\cap$ indicate set union and intersection operations respectively. This is reasonable since we know that the two masks coincide when I_1 undergoes the same amount of rotation as the pose of I_2 with respect to I_1 (Figure 3.2).

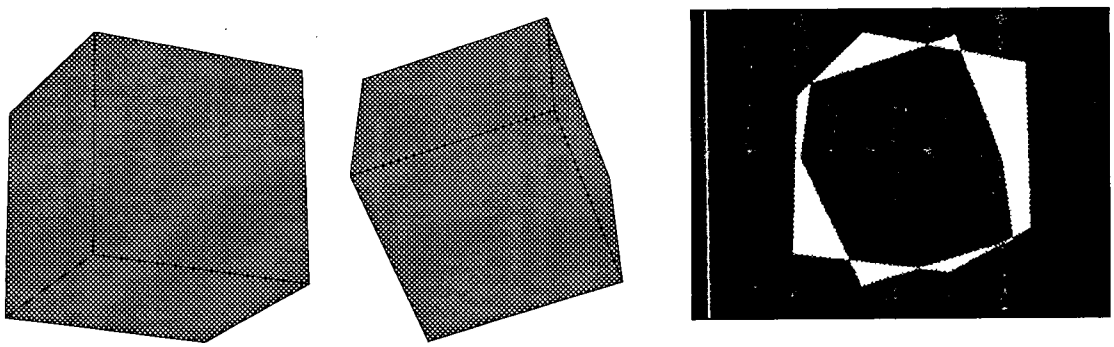


Figure 3.2: Object mask comparison. Left: reference frame of a cube, I_1. Middle: frame with unknown pose, I_2. Right: object mask of I_1 and I_2 compared. The white area is the difference between the two masks and it shrinks (here vanishes) as I_1 rotates up to the cube pose in I_2.

Mask comparison is a good measure of pose similarity as long as no new part of the object is revealed in I_2. When I_2 has some newly revealed parts, the mask of rotated I_1 can cover only a part of I_2 mask (i.e., the newly parts in I_2 cannot be covered by I_1). This is the case when the object is "smooth" (e.g., a sphere) so that even a small rotation reveals a new part not seen before, or when the object

rotation is large in "rough" objects (e.g., a cube). In such cases, the area of mask difference is not minimized at the correct pose; thus, the mask difference cannot be used as the measure of pose similarity between I_2 and rotated I_1.

Our solution for the case discussed above is to add a measure of texture difference to the mask difference and use the sum to measure the similarity of I_2 and rotated I_1. The intuition behind this measure is firstly, as we said above, the fact that the mask difference is not minimized at the correct pose. Secondly, the texture difference, which can be measured only in the common region of masks $(I_1 \cap I_2)$, is minimized at least in two poses: the correct pose and when I_1 turns its back to the camera (when all previously visible parts disappear). In the former case, the texture of both I_1 and I_2 is exactly the same all over $I_1 \cap I_2$. In the latter case, I_1 has shrunk to null, making $I_1 \cap I_2 = \emptyset$. Thus, any texture difference measure integrated over $I_1 \cap I_2$ becomes zero.

We demonstrate our intuition with an example. The experiment of this example is designed to show inability of either mask difference or texture difference alone as pose similarity measures and the functionality of their combination. In Figure 3.3(left) the view from the reference pose of a 3-D object is given. It is also regarded as the 3-D mask, I_1, built from the reference frame and the corresponding depth-map. The object is a textured symmetric prism. Note that only two faces of the object, which is "rough" in the sense introduced above, are visible from the reference pose. The object is then rotated to a known large amount. The captured view is given in Figure 3.3(right). Three faces are visible from this pose, one is seen before and the others are not.

Now we rotate I_1 for a set of angles and measure both the area of $(I_1 \cup I_2) - (I_1 \cap I_2)$ and the texture difference integrated over $I_1 \cap I_2$. The results are given

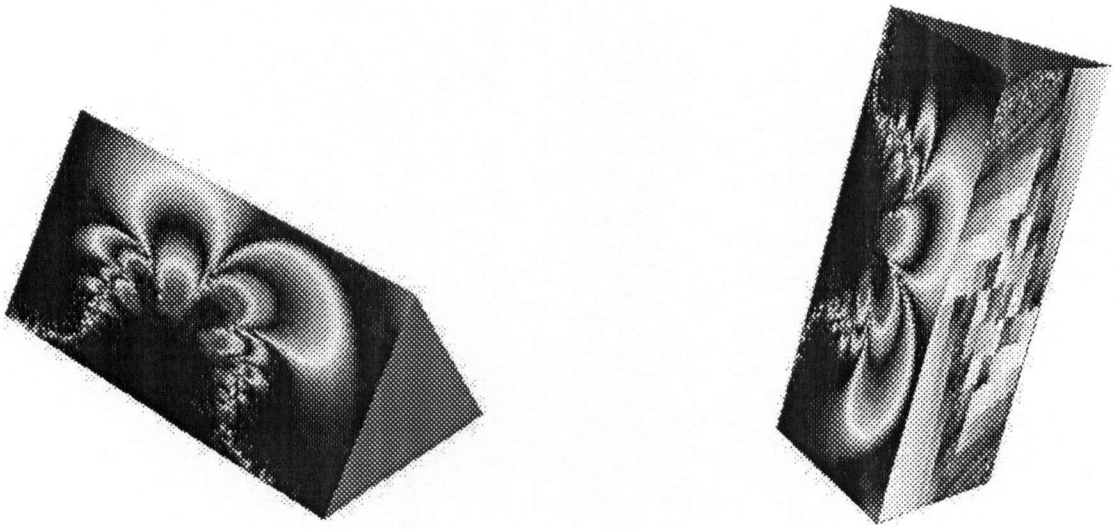


Figure 3.3: Setup for the experiment, comparing three pose similarity measures namely, texture difference, mask difference and combination of texture and mask. Left: Reference frame of a textured prism, I_1. Right: I_2. The pose is actually known, (60, 70) degrees, and is used as the ground truth.

in Figure 3.4. Before interpreting the results, we review a few points.

The set of 3-D rotations we use for the experiment are limited to combinations of horizontal (around Z-axis of the object coordinate system) and vertical (around the line of horizon) rotations only. In other words, we disregard possible rotations around the camera optical axis (the so called "roll" rotation). This simplifying assumption not only does not hurt the generality of the solution, but also helps us to visualize the results because in this case the desired pose similarity measure is a function of two rotation variables (i.e., a 3-D surface reaching its minimum in the correct 2-variable pose). That is, if we consider the roll rotation, the measure is a function of three rotation variables (i.e., a 4-D surface reaching its minimum at the correct 3-variable pose). The target of the camera is set on the origin of the object coordinate system, where the object is also centred. The mask difference is

measured as the number of pixels that lie in $(I_1 \cup I_2) - (I_1 \cap I_2)$. The last note is on how the texture difference is measured: we simply take the absolute difference of colour values for each pixel of I_2 and the rotated I_1. These values are then summed over $(I_1 \cap I_2)$ to form the texture difference measure.

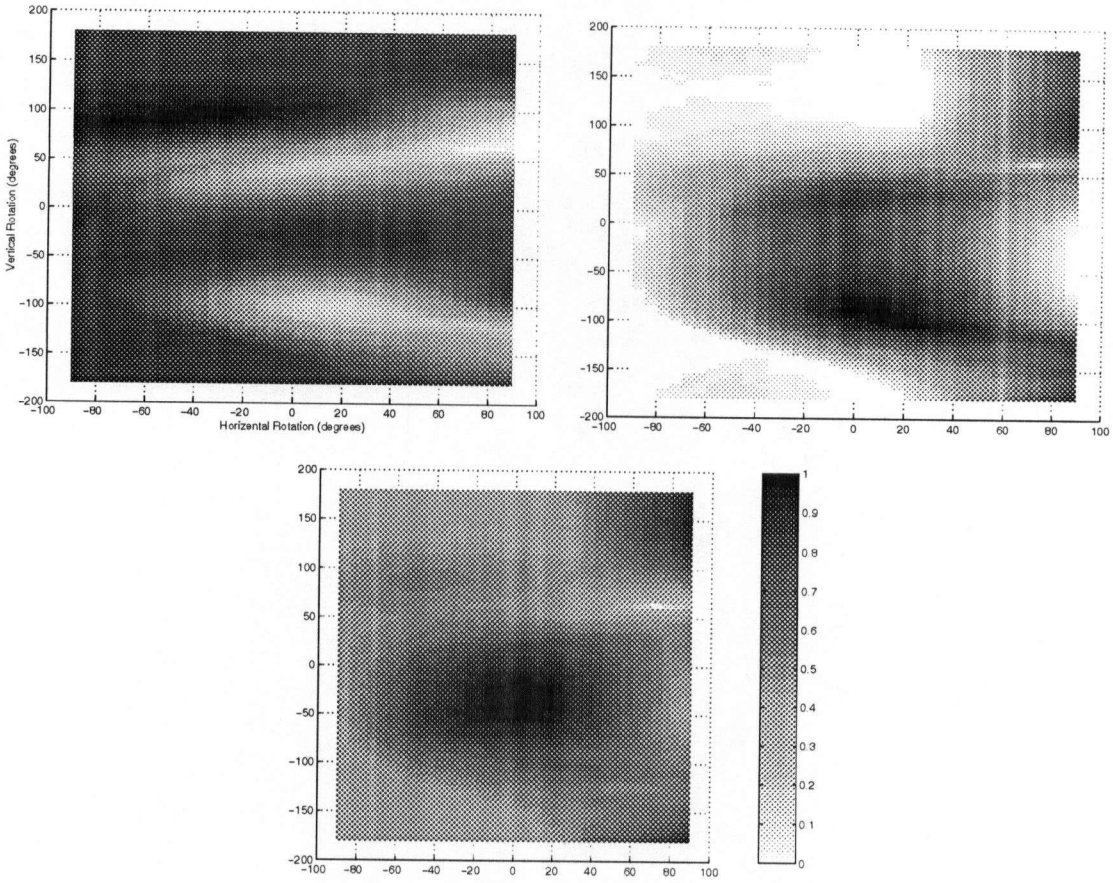


Figure 3.4: Comparing mask difference (left), texture difference (right) and their sum (bottom) as pose similarity measures. The sound measure should have its minimum at the correct pose (60, 70) degrees. All graphs are produced using values linearly normalized between 0 and 1 (see the colour scale on bottom right).

Interpretation of the results of the experiment (Figure 3.4) follows. The correct pose at which we expect a good similarity measure to have a global minimum, preferably the sole minimum, is (60, 70) degrees. The view from this pose is given

in Figure 3.3(right). We observe that both mask difference and texture difference have a local minimum at the correct pose. However, both of the measures have outlier regions that easily deflect any optimization algorithm seeking a minimum. For the mask difference (Figure 3.4(left)), there are two outlier regions like valleys horizontally stretched: one at the middle, and the other at the bottom of the graph. For the texture difference (Figure 3.4(right)), there are two outlier regions on the left (bottom and top), and one more at the right side of the graph. Confirming our intuition, the sum of texture difference and mask difference (Figure 3.4(bottom)) has a global minimum at (60, 70). The minimum is located at the correct position considering that the sampling period of the rotation angles is 5 degrees.

This experiment is conducted to demonstrate the open loop operation of the pose estimator system. It shows that the sum of mask and texture difference is an appropriate pose similarity measure for large rotations of rough objects. In Section 3.3, we give experimental results to prove good performance of the system in a closed loop (normal) operating mode.

3.2.5 Feedback algorithms

Here, we discuss a couple of optimization algorithms we use to close the feedback loop of the system: a deterministic minimum search algorithm [68] and a genetic algorithm [69].

The deterministic minimum seeking algorithm (DMSA) we use is multidimensional, unconstrained and nonlinear [68]. This method requires an initial guess (which is naturally relative pose of (0, 0) for small rotations), from which it senses the gradient of the surface of the similarity measure and follows that gradient to

get into a local minimum. The method is not helpful when one looks for the global minimum probably located far from the initial guess.

The genetic algorithm (GA) [69] is appropriate in cases that a global minimum is sought. The genetic algorithm is not deterministic; its random nature helps not to be trapped in the local minima. It requires a range of permitted values for arguments (in our case, the range of permitted rotations), as well as the resolution with which the arguments should be coded. For example, if the resolution is set to 5 bits and the range of the argument is 2 to 10, the argument can take $2^5 = 32$ equidistant values between 2 and 10, yielding a resolution of $(10 - 2)/32 = 0.25$.

3.3 Experiments

The first experiment is designed to show that the mask difference pose similarity measure is good enough to estimate the pose of a rough object undergoing a small rotational motion (i.e., no new faces of the object shows up). The object is that of Figure 3.3. The reference pose, I_1, is the same as that in Figure 3.3(left) and the pose to be determined, which is also the ground truth, is set to several values between (-85, -85) and (60, 70) degrees.

We use a DMSA to close the feedback loop of our system with an initial guess of (0, 0), which is a reasonable guess for small rotations. The search is performed on a rectangular grid from (-180, -90) to (180, 90). If the method converges, the grid size (the distance between adjacent nodes of the grid) is the maximum value for pose estimation error. There is a tradeoff between the grid size and the convergence speed of the method. Considering the value of grid size, our system performs well for small rotations using the mask difference as the pose similarity measure (Table 3.1). As we expected, the mask difference does not

perform effectively for large rotations.

Ground Truth	Grid Size	Estimated Pose	CPU Time
(5, 0)	2	(4, 0)	29.76
(10, -10)	3	(9, -9)	51.25
(0, 20)	3	(0, 18)	52.01
(-40, 30)	5	(-40, 25)	98.88
(60, 70)	10	(40, 10)	72.25
(-85, -85)	5	(-5, 16.56)	36.19

Table 3.1: Experiment 1. Angle pairs in Ground Truth and in Estimated Pose columns designate the amounts of horizontal and vertical rotations. The angle amounts are in degrees. The CPU time is in seconds.

The second experiment demonstrates that the sum of mask difference and texture difference performs effectively in estimating the pose of a rough object undergoing large rotations from the reference frame. The object shown in Figure 3.3(left) (at the reference pose) is used for the experiment. Two algorithms are used to close the feedback loop and their results are compared. The first one is a GA coding the first (from -180 to 180 degrees) and the second pose (from -90 to 90 degrees) variables with 6 and 5 bits, respectively. Hence the grid size becomes 5.7 (5.8) degrees for the first (second) pose variable. Using overlapping generations in our GA (i.e., copying the best four individuals of each generation to the next), our GA locks on a set of deepest minima of each generation. Results are shown in Table 3.2. Note that GAs, inherently, have moderate convergence properties, although they usually find a point near the global minimum. For a more accurate estimate, one may run a DMSA after a GA and use the result of the GA as the initial guess for the DMSA.

To the aim of devising a method faster than GA, we apply the DMSA, that

Ground Truth	Grid Size	Estimated Pose	CPU Time
(5, 0)	6	(2.86, -8.71)	2409
(10, -10)	6	(14.29, -8.71)	990
(80, 70)	6	(82.86, 66.77)	2640
(-85, -85)	6	(-2.86, 66.77)	3300
(5, 60)	6	(8.57, 55.16)	1815
(80, -10)	6	(77.14, -8.71)	1320

Table 3.2: Experiment 2.a (GA feedback). Angle pairs in Ground Truth and in Estimated Pose columns designate amounts of horizontal and vertical rotations. Angle amounts are in degrees. The CPU time is in seconds. Grid Size value is approximate.

was used in the first experiment, five times with initial guesses at (0, 0), (-60, -45), (60, -45), (-60, 45), and (60, 45) in the second part of Experiment 2. These points are selected so that each quadrant of the search space (i.e., the rectangular region between (-180, -90) and (180, 90)) has one initial point near its centre. The selected points performed better than the exact centres of quadrants (i.e., (-90, -45), (90, -45), (-90, 45), and (90, 45)). Initial guess of (0, 0) is added to compensate for mediocre performance of DMSAs started from the centre of quadrants that do not perform well for target poses near (0, 0). Among the resulted five minima, we select the deepest one as the final estimated pose. To be comparable with GA's grid size, we select a grid size of 5 degrees for both pose variables. The results of this experiment are given in Table 3.3. According to the results, faster and better convergence is achieved by the DMSA method with multiple initial points, as compared to the GA method.

We run the pose estimator on a smooth object, a textured sphere (Figure 3.5), in our third experiment. The reference pose is taken right above the equator (in Figure 3.5 the north pole is visible on the top). We try estimation for both large

Ground Truth	Grid Size	Estimated Pose	CPU Time
(5, 0)	5	(5, 0)	591.48
(10, -10)	5	(10, -10)	571.82
(80, 70)	5	(80, 70)	545.6
(-85, -85)	5	(0, 75)	654.73
(5, 60)	5	(5, 60)	822.53
(80, -10)	5	(80, -10)	632.14

Table 3.3: Experiment 2.b (Multiple DMSA feedback). Angle pairs in Ground Truth and in Estimated Pose columns designate amounts of horizontal and vertical rotations. Angle amounts are in degrees. The CPU time is in seconds.

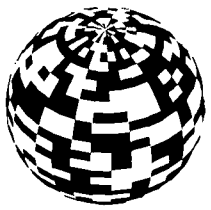


Figure 3.5: Textured sphere used in the third experiment.

and small rotations with both of the feedback mechanisms used in Experiment 2. The sum of mask difference and texture difference is used as the pose similarity measure. The results are given in Tables 3.4 and 3.5.

Incorrect estimation for pose (5, 0) in Experiment 3.a can be justified by the random nature of the GA. No feedback can do better for the pose (-85, -85) because the mask area of the rotated I_1 at that pose is almost zero, that is, parts of the object visible in I_2 in this pose were not visible in I_1 at all.

While multiple DMSA is faster than GA, it fails to converge to the correct pose in a couple of cases more than GA (the last two rows of Table 3.5).

Based on the experiments reported in this section, we list the pose similarity measures and feedback methods suitable for our pose estimator in different

Ground Truth	Grid Size	Estimated Pose	CPU Time
(5, 0)	6	(180, -8.71)	1515
(10, -10)	6	(8.57, -8.71)	949
(80, 70)	6	(82.86, 72.58)	2555
(-85, -85)	6	(-157.14, 2.9)	620
(5, 60)	6	(2.86, 60.97)	1598
(80, -10)	6	(77.14, -8.71)	4002

Table 3.4: Experiment 3.a (GA feedback). Angle pairs in Ground Truth and in Estimated Pose columns designate amount of horizontal and vertical rotations. Angle amounts are in degrees. The CPU time is in seconds. Grid Size value is approximate.

Ground Truth	Grid Size	Estimated Pose	CPU Time
(5, 0)	6	(6, 0)	223.89
(10, -10)	6	(12, -12)	193.3
(80, 70)	6	(-178.42, -4.36)	210.25
(-85, -85)	6	(72, 42)	117.72
(5, 60)	6	(174, 0)	483.1
(80, -10)	6	(114, -48)	199.29

Table 3.5: Experiment 3.b (Multiple DMSA feedback). Angle pairs in Ground Truth and in Estimated Pose columns designate amount of horizontal and vertical rotations. Angle amounts are in degrees. The CPU time is in seconds.

situations in Table 3.6.

All experiments reported in this section are coded in Matlab and conducted on an Ultra-Sparc 10 Sun machine.

3.4 Concluding remarks

Results of our experiments show that cases where smooth objects have undergone small or large rotation, or cases where rough objects have undergone large rotation, cannot be handled if there is no discriminating information in texture (e.g., the

Object Type	Small Rotation	Large Rotation
Rough	MD / DMSA	MD+TD / Multiple DMSA
Smooth	MD+TD / Multiple DMSA	MD+TD / GA

Table 3.6: Suitable pose similarity measure / feedback for different situations. MD: Mask Difference. TD: Texture Difference. DMSA: Deterministic Minimum Seeking Algorithm. GA: Genetic Algorithm.

texture is uniform on the entire surface of the object).

The distribution of texture difference or mask difference values might be useful to set up a better feedback or to identify useful information. For example, if the texture difference does not change for different pixels, we can conclude that the underlying object is uniformly textured. Therefore, in this case, the texture information may not help in pose estimation at all.

A limitation of our method is that it loses accuracy in estimation of the pose along with the increase in amount of rotation. That is because the target pose has smaller previously seen part, so the texture difference is measured by integration over a smaller area, causing more noise in the result. The phenomenon is responsible for the wrong pose estimation for ground truth value of (-85, -85) degrees in all reported experiments, because the mask of reference frame has an area of almost zero when rotated to (-85, -85) degrees. Note that the mask difference term cannot alleviate the problem, since it just removes the outliers and has no contribution to the determination of the correct pose. This limitation encourages gradual pose estimation as described below.

Small rotations of rough objects are more tractable in two aspects: one, mask difference is enough for comparing different views and needs less computation over mask-and-texture difference. Two, the feedback loop can be closed by a DMSA,

which is much faster than a GA. These advantages encourage estimation of gradual change in pose that occurs frequently in frames of a video sequence.

Pose estimation, specially when a GA is used for minimization to handle large rotations, is very time consuming. Therefore, further work, perhaps code optimization, is needed to reduce the processing time to make the task more manageable.

Finally, more computationally efficient comparison measures that converge onto a single minimum at the correct pose should be devised.

4: Shape-from-silhouette

4.1 Introduction

The extraction of 3-D information from an image sequence involves two steps. The first step is finding the poses/camera viewpoints of/from which the images are taken. The next step is fusion of images, now with known poses, into a 3-D model (volume) of the object.

Our choice for the second step above is "shape-from-silhouette" (SFS) method because of its simplicity in concept and use. The rest of the chapter is organized as follows. The tool we implemented for SFS-based volume reconstruction is reviewed in Section 4.2. Section 4.3 investigates the effects of viewpoint (pose) distribution, pose sampling rate, and pose error in volume reconstruction quality. In Section 4.4, we introduce a validity criterion for volume reconstruction by SFS, along with the application of the criterion in pose estimation. Section 4.5 concludes the chapter.

4.2 A reconstruction tool based on SFS

Following [15], we perform volume reconstruction from known-pose silhouettes. We intersect cylindric volumes made by stretching silhouettes in the direction of pose (to the coordinate origin), one for each available silhouette, with a cube.

This cube is centered on the origin and contains discrete equidistant points, each called a "voxel" (volume element; by analogy with pixel). We call this cube the "material cube" from which the output object is made by "carving". The number of voxels in the material cube is determined by the resolution desired for the output volume.

Each voxel in material cube is projected orthographically onto planes whose normals are pose directions. For each pose, the projected voxels are then tested to see if they lie on or off the silhouette of that pose. Details on how to project the voxels onto the planes are given in Appendix A. A voxel remains on material cube if "all" of its projections into silhouette planes lie within silhouettes. Otherwise, the voxel is cut out from the cube.

After the cube is carved by all available silhouettes, we have the tightest possible bounding volume of the object with desired resolution. While such representation for a volume has its benefits (see comparison below), one may want to find the boundary surface to the object because most 3-D graphics tools work with surface models rather than volume models. To this aim (Figure 4.1), we first eroded the outermost layer of the carved cube by a 3-D morphologic filter. Then we XORed the carved cube with its eroded copy. The result is the voxels on the surface of the carved cube which are part of the original volume but do not exist in the eroded copy. A surface is then fitted to these voxels using 3-D graphics tools. The 3-D morphologic erosion filter used is simply a generalization of its 2-D type described in [66, 67]: a voxel will be present in the output (filtered) volume if all of 26 voxels in its cubic vicinity are present. Otherwise, the voxel will be omitted.

As compared to [15], our method is simpler but slower. That is because we

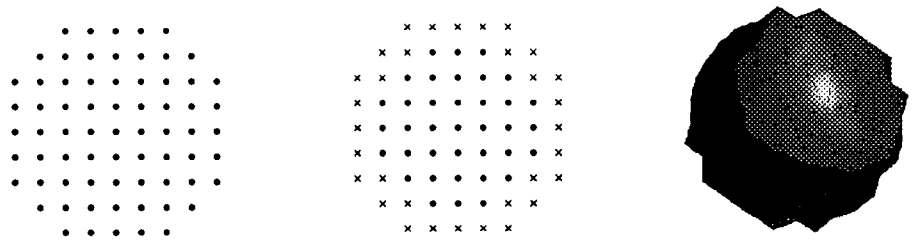


Figure 4.1: Making the cover surface of a volume. Left: cross section of a discrete sphere. Middle: cross section of the sphere eroded. "×"s are put in place of eroded voxels. Right: computed cover surface.

compute, for each voxel, whether it lies within a cylindric volume or not. However, this criterion is investigated for each block, comprised of several voxels, in [15]. The advantage of our method, however, is that it is easily broken into modules that can run in parallel: the silhouette from each pose can be used to carve the material cube independently. Next, all of the carved cubes are ANDed, since a voxel in the output volume should be present in "all" of the cylindric volumes. This is not the case for [15], since two volumes in "octree" representation cannot be unified in a computationally simple way.

Reliable performance of our reconstruction method is demonstrated in a few examples (Figure 4.2).

4.3 Factors influencing the reconstruction quality

The quality of volume reconstruction in the two-step reconstruction paradigm discussed in the beginning of Chapter 4 depends on several factors, namely:

- pose error,
- volume quantization,

- image quantization,
- pose sampling rate (number of views),
- viewpoint distribution (location of viewpoints), and
- object concavity.

The system we proposed for 3-D information retrieval from monocular video, uses a pose estimator (Chapters 3) and a silhouette-based volume reconstruction method (Section 4.2). For quality assessment of the reconstructed 3-D models, in this section we address sensitivity of this volume reconstruction method to choice of viewpoints, pose sampling rate and pose error.

Some of the factors given above are discussed in the literature: recent studies on sampling requirements of the light field rendering [19], a method for fusion of images into a 3-D profile of the object, are given in [70, 71]. To the best of authors' knowledge our work on sampling requirement of silhouette-based volume reconstruction method is the first in the in the subject. Although a class of studies (e.g., [72]) addressed the problem of camera viewpoint control to maximize a quality measure.

Object concavities are a source of reconstruction error, since silhouette-based reconstruction methods are inherently able to make a model of the object up to the object's "visual hull" [73], which does not include most of the concavities.

Our literature search for other issues of reconstruction quality assessment did not produce any result. From the above-mentioned issues of silhouette-based volume reconstruction quality analysis, here we explore the effects of pose error, pose

sampling rate (Section 4.3.2) and viewpoint selection/distribution (Section 4.3.1) on the reconstruction quality.

4.3.1 Viewpoint distribution

The capture phase of 3-D volume reconstruction consists of taking images of the subject from different viewpoints. In this section we introduce a new uniform distribution of viewpoints for 3-D volume reconstruction and compare it to the traditional viewpoints distribution through an experiment.

Consider the object is located at the center of a large sphere. And each viewpoint is corresponding to a point on that sphere. We call this sphere the "focal sphere". Note that for each viewpoint on focal sphere, there is another viewpoint so that the segment connecting these two viewpoints is a diameter of the focal sphere. Assuming orthographic projection, the silhouette of the object is the same from these two viewpoints (i.e., one of these viewpoints is redundant as long as silhouette-based volume reconstruction is concerned). Therefore, all of the information of object's silhouette can be captured from viewpoints located on only half of the focal sphere, which we call "focal hemisphere".

In [74, 15], the object is placed on a turn-table. While the table is turning, images are taken by a stationary camera. Changes in camera elevation result in viewpoint distribution depicted in Figure 4.3 (top). Note that this sampling scheme is denser near the pole of the focal hemisphere, thus it is not uniform.

To distribute the viewpoints evenly, one should have the same density of viewpoints on different spots of the focal hemisphere. To this aim, we first flattened a differential surface element of the focal hemisphere. Viewpoints are taken as nodes of a grid projected onto the flattened hemisphere. In other words, the

angular distance of neighboring viewpoints should be scaled by $1/\cos x$, where x denotes the latitude on which the viewpoints are placed. Thus, less number of viewpoints can be inserted near the pole, giving almost uniform viewpoint distribution all over the focal hemisphere. An example of such viewpoint distribution is depicted in Figure 4.3 (bottom).

To show better performance of the new sampling scheme, we use both traditional and the new sampling scheme to take samples for silhouette-based reconstruction of a sphere. The outcome is given in Figure 4.4 (top). As shown in the graph, for a certain reconstruction quality, considerably fewer samples are needed using our proposed sampling scheme. Here, the reconstruction quality is defined as the volume difference between the reconstructed and the ideal sphere normalized by the volume of the reconstructed sphere. The metric is estimated by a function of standard deviation and mean value of the radius of the reconstructed sphere.

We run the experiment again using a prism as the object. In this experiment, the reconstructed volumes are XORed with the representation of the object in the same voxelized space (a view of that is given in Figure 4.5), where the experiment is performed. The result (Figure 4.4 (bottom)) is the number of voxels remained after XOR operation, normalized by the number of voxels comprising the object.

Please note that the reconstruction of a sphere and a prism are different (Chapter 3) in the sense that a sphere is a smooth object (i.e., infinite number of samples is required for its exact reconstruction), while a prism is not. The new sampling scheme outperforms the traditional scheme in both situations.

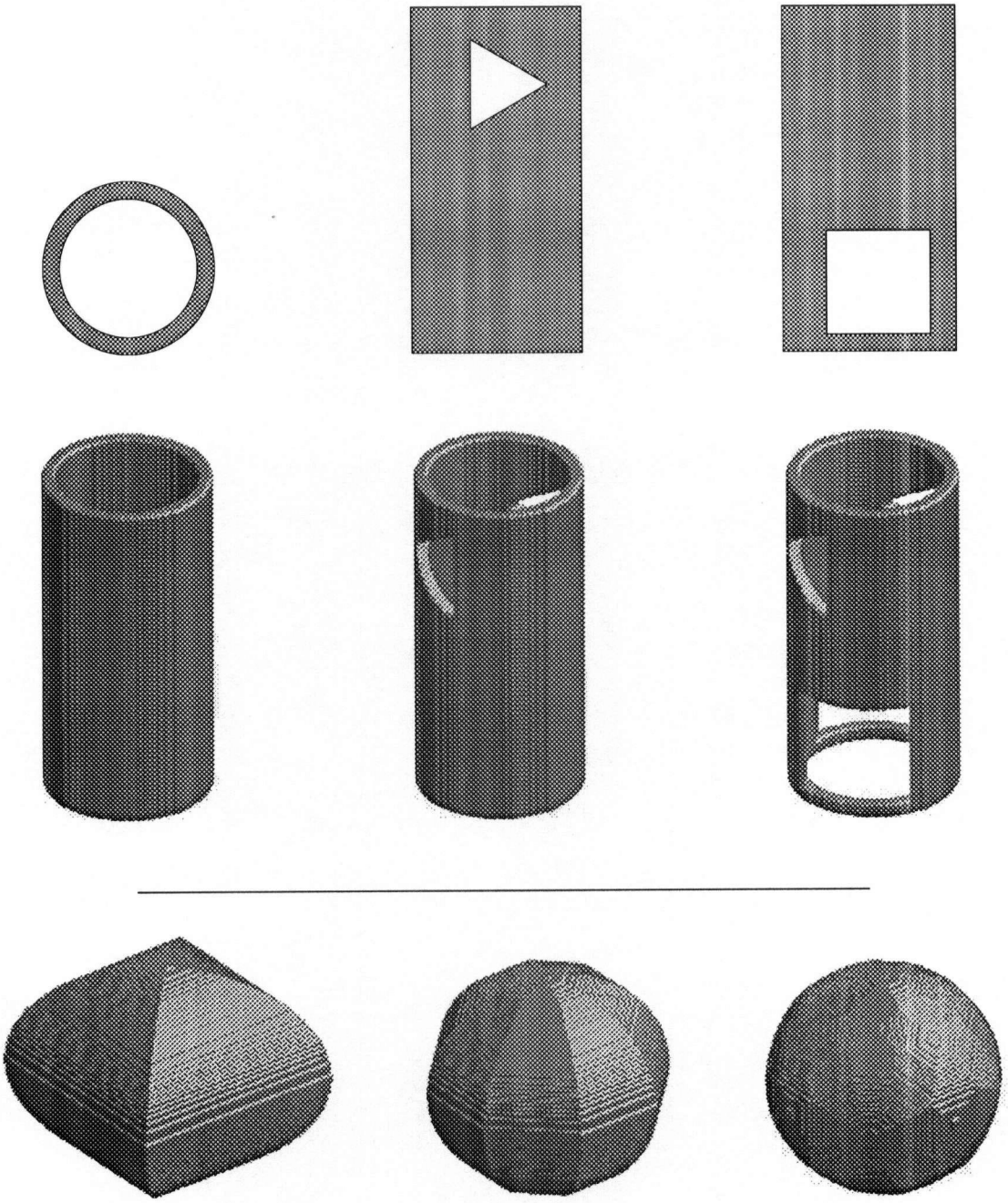


Figure 4.2: 3-D reconstruction examples. Top row: Input silhouettes of poses (left to right) $(0,0,1)$, $(1,0,0)$ and $(1,1,0)$. Middle row: 3-D models built from silhouettes in top row. Bottom row: Models all carved by a circular disk silhouette from different directions: $(1,0,0)$ and $(0,1,0)$ made the model at the left, $(1,1,0)$ and $(-1,1,0)$ made that at the center, $(0,0,1)$, $(1,1,1)$, $(-1,1,1)$, $(1,-1,1)$, and $(-1,-1,1)$ made one at the right. To build the models silhouettes were used cumulatively from left to right.

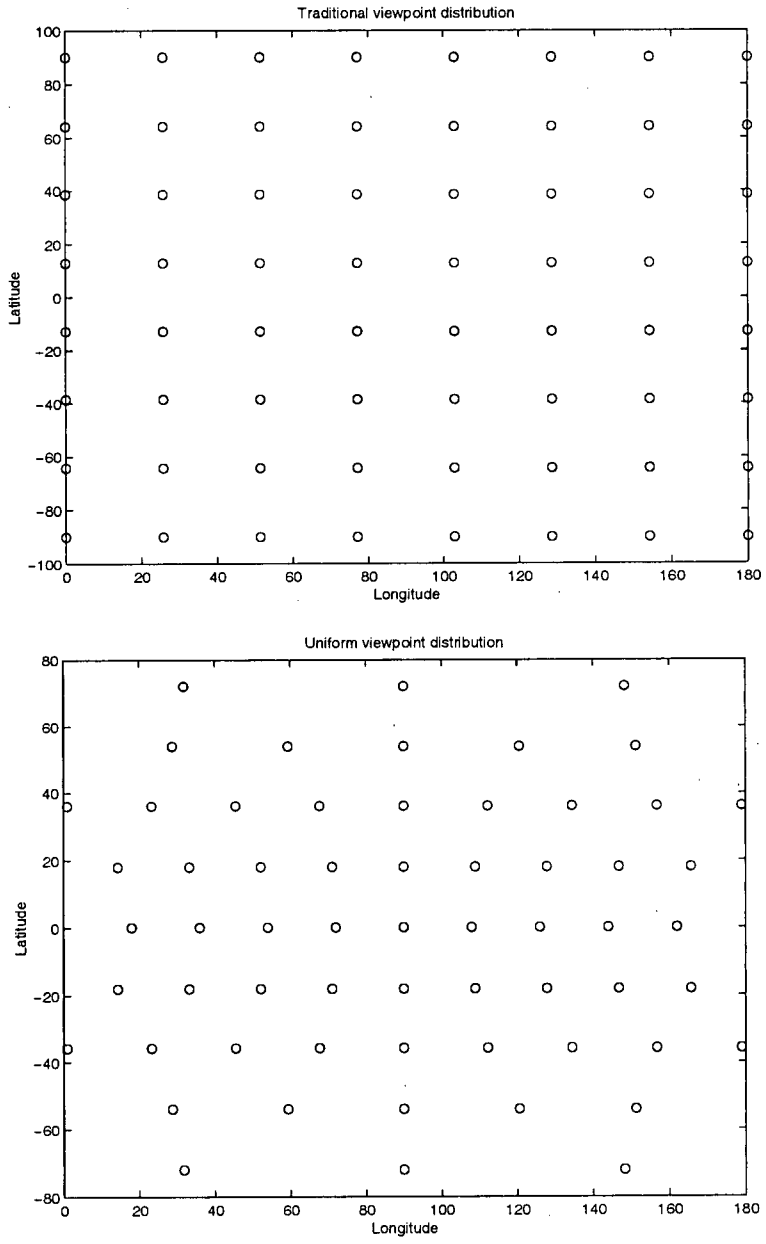


Figure 4.3: Viewpoint distributions: the traditional (top) and the uniform (bottom).

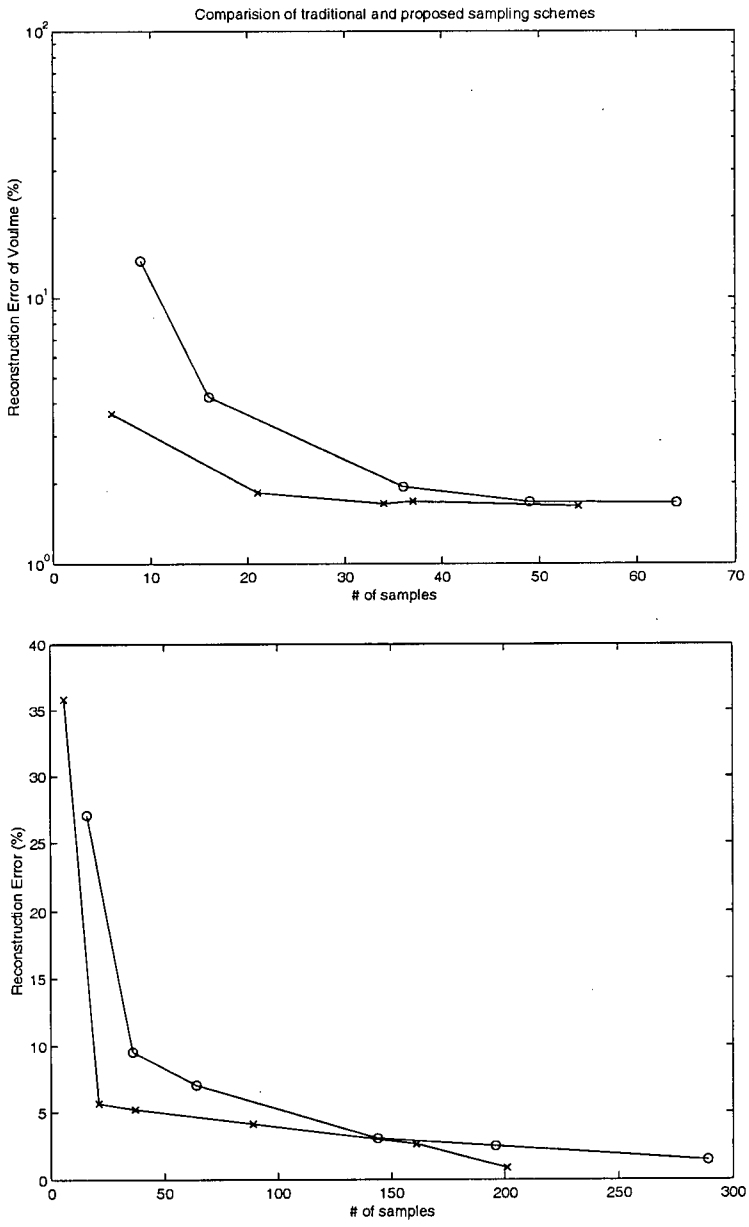


Figure 4.4: Comparison of the reconstruction quality using the traditional ('o') and the (new) uniform ('x') distribution of viewpoints in reconstruction of a sphere (top) and a prism (bottom).

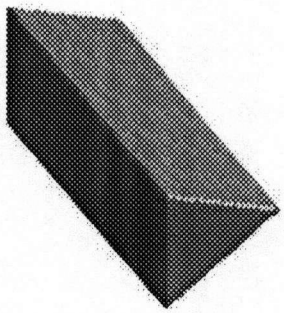


Figure 4.5: The prism used in the experiments in the voxelized space.

4.3.2 Effects of pose error and sampling rate

In this section, we discuss the effects of pose error and sampling rate variation on silhouette-based reconstruction quality. We model error of the pose estimator under study [Chapter 3] by a white random process with a uniform distribution ranging from ± 0.2 to ± 10 degrees, depending on working condition of pose estimator (e.g., grid size, image resolution, depth quality/resolution, etc). We also vary the sampling rate to see how many samples are sufficient for a specific level of pose error and reconstruction quality.

Because of its superior performance over the traditional sampling, the uniform sampling scheme (Section 4.3.1) is used. The object is a prism (Figure 4.5). We compare the reconstructed volume with the prism itself in voxel space and take the normalized difference as the measure for reconstruction error. The carving is performed over a cube of $201 \times 201 \times 201$ by silhouettes of size 512×512. The same pseudo-noise pattern is used in all of the experiments (i.e., only the noise level is changed) to limit the random effect of noise on the results.

The result is given in Figure 4.6 (left). For small levels of pose error (i.e., ± 0.2, ± 0.5 and ± 1 degrees; the lines that stick together on the graph), the effect of pose error is negligible (as compared to the true pose graph in Figure 4.4 (bottom)). That means errors of less than ± 1 degrees are tolerable by silhouette-based volume reconstruction, at least for simple "rough" (Chapter 3) objects like prisms. The reason is that the factors affecting the reconstruction process (Section 4.3) dominate the small errors in pose.

For higher values of error, we observe deviation from small error level curves. In high noise levels, interestingly, the increase of sampling rate cannot enhance the reconstruction quality (Figure 4.6 (left)). For example, with error of ± 10 degrees

in pose, the reconstruction quality becomes almost independent of the sampling rate, while at an error level of ± 5 degrees in pose, there is an inverse relationship between the number of samples and the reconstruction. Although some increase in the sampling rate causes some reconstruction degradation.

The result is also depicted in Figure 4.6 (bottom) from another point of view: it is observed that reconstruction with a large number of samples is more sensitive to pose error (i.e., reconstruction degradation with the increase of pose error level is faster for a large number of samples). That is, since the total reconstruction error is small, a small error in pose degrades the quality considerably. It is also observed that when pose error is large enough (± 10 degrees), the quality is degraded regardless of the sampling rate: all curves approach a single point with the increase of pose error. Small and local deviations of some of curves from the trend of quality degradation by error increase are because of the randomness in the model assumed for pose error.

4.4 *Validity criterion for volume reconstruction by SFS*

The input to a silhouette-based reconstruction method is a set of silhouettes, each having a pose value. In this section, we address the relationships between the input silhouettes and their poses that must hold to produce a volume by SFS. These relationships can be verified to see if the correct poses are assigned to the input silhouettes.

We denote the set of silhouettes (binary images: object 1, and background 0) by M_i, and their (absolute) poses by P_i, $i = 1, \cdots, N$. Volume reconstruction is performed by projecting each voxel of the initial volume, which should be large enough to contain the whole reconstructed volume, to each silhouette in the

direction of its pose. The voxel is carved out if it is projected to a background pixel in at least one silhouette. In other words, the voxel remains in the final reconstructed volume if it projects to foreground pixels in *all* silhouettes.

Note that we can always perform carving with one silhouette at a time (order does not matter) without a loss of generality. That is, for each silhouette, we carve out the voxels projecting to the background pixels. For the next silhouette, we continue with the remaining voxels.

Definition. For a pixel $f_i \in F_i$ (foreground pixels of M_i), $A(f_i)$ is defined as the set of voxels *marked by* f_i (a pixel in foreground of M_i). That is,

$$A(f_i) \equiv \{v \mid R_i(v) = f_i\},$$

where $R_i(v)$ is the projection on M_i in the direction of P_i. Note that each voxel is allowed to be marked by (at most) N pixels.

For better visualization, consider the line passing through all the members of $A(f_i)$: this line is perpendicular[1] to the M_i plane and pierces it at f_i. The following theorem holds if the correct poses are assigned to silhouettes.

Theorem 4.4.1 *For each foreground pixel f_i of M_i, $i = 1, \cdots, N$,*

$$\exists\, v \in A(f_i) \mid R_j(v) \in F_j \qquad \forall j = 1, \cdots, N.$$

That is, there is at least one voxel marked by f_i that is not carved out by any of the silhouettes.

[1] Orthographic projection is used for simplification. The same principles hold for weak perspective projection [75] with appropriate scaling of the input silhouettes.

Proof. By contradiction. Suppose all of the voxels marked by f_i are carved out. Then, there is no voxel v left in the reconstructed volume for which $R_i(v) = f_i$. However, the reconstructed volume is a superset of the visual hull [73] of the object, which is a superset of the object volume. Assuming an opaque object, there is no voxel of the object volume that can make pixel f_i in the foreground of M_i. This contradicts $f_i \in F_i$. Q.E.D.

4.4.1 Application to pose estimation

The relationship between silhouettes and poses of an opaque object stated in Theorem 4.4.1, can help us verify the input to the SFS method. For instance, to a specific set of silhouettes, one may only assign certain poses. Otherwise, the silhouettes and their poses cannot belong to a real object. This is desirable for one interested in the pose estimation of the object: it means shrinking the solution space for pose.

We are particularly interested in this application since we already have pose estimates (using methods of Chapters 3 or 5) we wish to verify. Thus, we adopt Theorem 4.4.1 to generate a measure of "validity" for a set of pose estimates, or equivalently, a measure of "wellness" for the reconstructed volume.

We count a *violation* when Theorem 4.4.1 is violated for a pixel $f_i \in F_i, i = 1, \cdots, N$. In other words, a violation happens when all voxels that are marked by f_i get carved out by other silhouette(s).

For a set of silhouettes and their poses, one can use the number of violations as a measure of validity of the set of input poses. That is, a better set of input poses makes fewer violations. Counting the number of violations by keeping track of voxels marked by each foreground pixel is cumbersome; fortunately, there is an

easier way to do this.

We run SFS on the set of input silhouettes and their poses to get the reconstructed volume (V). Next, we project V onto each silhouette's plane. Each violation corresponds to one foreground pixel that no voxel is projected on it. Therefore, the number of violations for foreground pixels of M_i is the number of "on" pixels in the image $M_i \oplus R_i(V)$, where $\oplus$ denotes the XOR operation. Therefore, the total number of violations is given by summation over $i = 1, \cdots, N$.

To use as a reconstruction quality measure, this method is integrated with our silhouette-based reconstruction tool (Section 4.2). The resulted routine receives a set of silhouettes and their poses as the input and gives the reconstructed volume and the reconstruction quality factor as the output. The reconstruction quality factor is defined as the number of violations negated. Thus, a better volume reconstruction process is associated with a higher quality factor. The quality factor thus defined is always non-positive and the best possible reconstruction is associated with a zero quality factor.

4.5 *Concluding remarks*

In this chapter, we introduced a new sampling scheme for silhouette-based volume reconstruction. The new scheme outperforms the traditional scheme for the same number of samples. We investigated the effects of pose sampling rate (the number of silhouettes used for reconstruction) and pose error (the amount of deviation between estimated pose for a silhouette and its true value). We demonstrated that (Section 4.3.2): (1) low levels of pose error (up to ± 1 degree) are tolerable by silhouette-based reconstruction for simple rough objects. (2) High levels of pose error cannot be compensated by increase in the sampling rate. (3) Reconstruction

at high sample rates is more sensitive to pose error.

We also studied the relationships between silhouettes and their poses in the input to a silhouette-based reconstruction method. This study furnishes a SFS-based volume reconstruction quality factor.

Some ways to extend this work are studies on: one, effects of a random sampling scheme. Two, effects of silhouette quantization. Three, effects of volume quantization. Four, other metrics for reconstruction quality such as one based on comparison of silhouettes of the reconstructed object and the original object from several (e.g., a random set of) viewpoints.

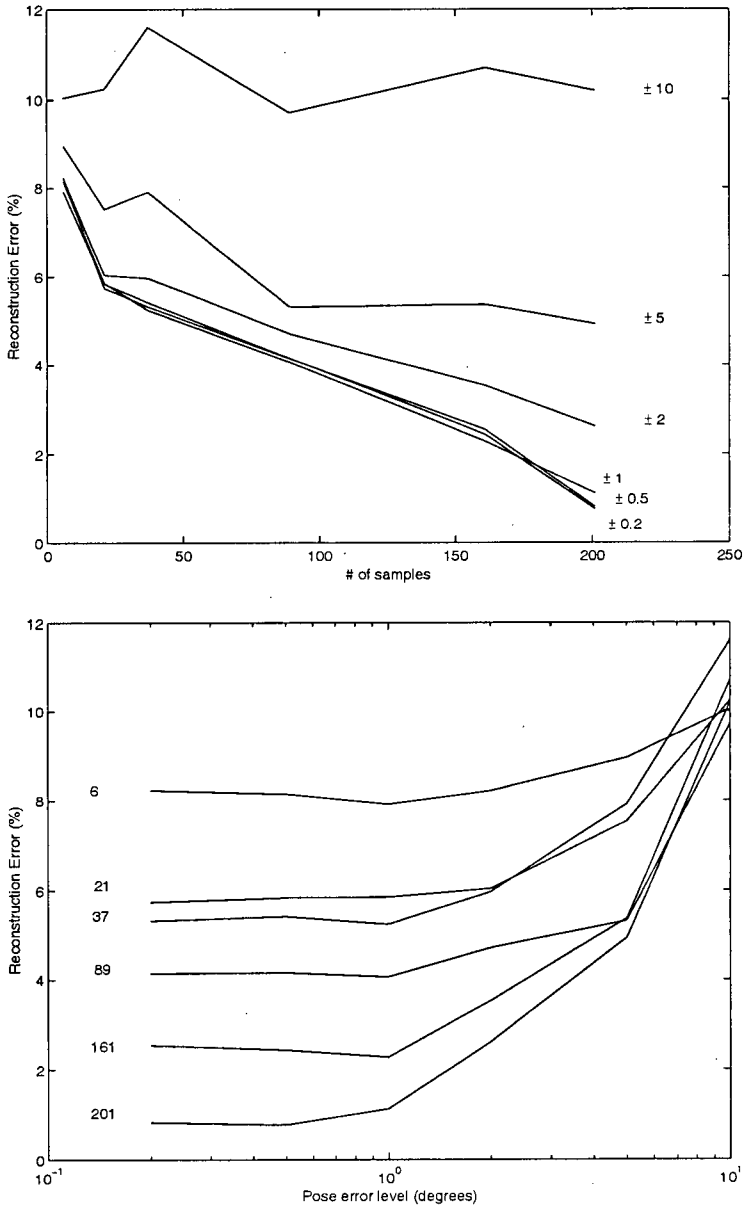


Figure 4.6: Comparison of the reconstruction quality at various levels of pose error and sampling rates. Top: pose error (degrees) used as parameter. Bottom: the number of samples used as parameter.

5: Pose estimation and object reconstruction by volume feedback

5.1 Introduction

In this chapter, we form a volume feedback loop from the 3-D reconstruction tool of Chapter 4, and a variation of the pose estimation method of Chapter 3. That is, first, initial pose estimates for images in the input sequence are derived in a pre-processing step. Next, an approximate volume of the object is reconstructed from input silhouettes and initial pose estimates. This volume is then used by a pose estimation method (variation of the method in Chapter 3), for a new round of pose estimations. The new set of pose estimates are used for volume reconstruction, hence closing the volume feedback loop.

This system proves to be functional (Section 5.5) by adding a block to improve pose estimates using the theoretically developed criterion of Section 4.4 for reconstruction quality by shape-from-silhouette.

Similar to the method of Chapter 3, the method of this chapter neither assumes any model for the object at the outset, nor uses any feature points.

Advantages of the method of this chapter over the method of Chapter 3 are: one, pose estimation and volume reconstruction are performed simultaneously for all images in the input sequence. Two, a single pose similarity measure is used

for all types of objects and all amounts of rotations. Three, this chapter's method does not require any depth-map at the outset since it builds up the depth-map for every input frame starting with the assumption of flat depth-maps for all frames. Four, thanks to the initial multi-reference pose estimation, the method can be also applied in situations where the frames are not necessarily ordered. These merits are achieved at the expense of longer processing times as compared to the method of Chapter 3.

The ideal conditions for operation of the proposed method follow.

- The object is convex and rigid.
- The camera model is effectively approximated by either orthographic or weak perspective projections.

Nevertheless, the method is successfully tested (Section 5.5.2) for inputs not satisfying some of these conditions. For more details and reasons of the above-mentioned limitations, see Section 5.6.

The remainder of this chapter is organized as follows. Section 5.2 provides an overview of the system and describes its building blocks. In Section 5.3, the integration of system blocks into the volume feedback loop structure is addressed. A basic account on complexity analysis of the system run time is given in Section 5.3.1. Section 5.4 is devoted to a theoretical study of convergence for the proposed system. To demonstrate the favorable performance of our system, results of several experiments on both synthetic and real input sequences are reported in Section 5.5. Some conclusions are drawn in Section 5.6.

5.2 Building blocks

The basic modules of our proposed system (Figure 5.1) are described in this section. Before going through a detailed description of the modules, an functional overview of the system is given.

First, an initial estimate of the object pose in each frame is made (Section 5.2.2). Then a volume of the object is reconstructed by shape-from-silhouette using the initial pose estimates. This volume provides the depth information required for enhanced relative pose estimation (RPE; Section 5.2.1). The new estimates are used to reconstruct a new volume of the object. This volume goes through a quality control mechanism (Section 5.3) and is fed to the RPE to close the feedback loop. Vital for the convergence to the true object volume, this block adjusts relative pose estimates to make a better volume in accordance with the quality factor introduced in Section 4.4. Convergence is checked by comparing the resulting set of relative poses to that of previous iterations. The loop is terminated if these two sets are similar.

5.2.1 Relative pose estimator

The relative pose estimator (RPE), the main part of the processing in the system (Figure 5.1), is used to estimate the relative pose between consequent frames. That is, each frame is used as the reference for the next one. According to this way, the very first frame has no reference. We consider the first frame as our global reference for the pose, having an absolute pose of 0.

The method we use for RPE is very similar to the method of Chapter 3 (published in [41]) for the pose estimation of a rigid body in several frames. The

method of Chapter 3 neither works with feature points, nor assumes a model for the object at the outset. The main restriction of this method is the requirement of a depth-map for at least one of the input frames. Inspired by the idea of [41], here we remove the depth-map requirement to build an stand-alone volume reconstruction method from regular image sequences.

The idea of the RPE method we use here is to rotate the reference 3-D patch until its image becomes "similar" to the target frame. To generate the reference 3-D patch, the texture of the reference frame is mapped onto its depth-map. The relative pose between the reference and target frames is then estimated by the amount of rotation required for the above mentioned matching. The selection of a similarity measure is quite important. As compared to the ones used in [41], the single similarity measure used here results in accurate pose estimates for both object types ("smooth" and "rough" as defined in [41]). In this way, the object type does not need to be identified at the outset for effective application of the method introduced in this chapter.

The depth-map information fed to the RPE is always in the form of the volume; for the first time the object volume is reconstructed based on initial pose estimates. Other times the volume is reconstructed based on estimates produced in the last RPE call. To generate a textured 3-D patch from the reference frame and the volume, we either obtain a depth-map from the volume and map the reference frame texture on it, or map the texture on the volume itself. To texture the input volume, we either use all of the available frames to texture the whole volume (e.g. by voxel coloring), or just rotate the volume to the absolute pose of the reference frame (computed using the last estimate for relative poses) and texture map the reference frame onto the volume. In both cases above, the second

choices are used since they are computationally cheaper; in the former case, one processing step is eliminated, and in the latter case, the required texture mapping is a standard procedure which can be performed swiftly by the existing graphics hardware on most computers.

Shape-from-silhouette builds the volume in the voxel space by carving [15]. Before mapping the texture of the reference frame onto the volume, we have to transform the volume from the voxel space to a mesh (vertices and faces) format. To this aim, we dilate the volume by a $3 \times 3 \times 3$ structuring element. Then, a surface (mesh) is fitted through all the voxels belonging to the dilated volume, but not present in the original volume.

After mapping the reference frame texture onto the reconstructed volume, the volume is rotated around its centre, and its images are compared to the target frame. To measure the similarity between the target frame and rotated reference patch, we use the sum of the filtered texture difference, normalized by the number of pixels in the common area of the silhouettes. More precisely,

$$\hat{P}_{ij} = \arg\min_{r} \{\tau(r)\} \tag{5.1}$$

where, $\hat{P}_{ij}$ is the relative pose between frames i (reference) and j (target). r denotes the amount of rotation with respect to the initial pose of the volume, which is set to $\sum_{k=1}^{i} P_{k-1,k}$, where $P_{k-1,k}$ is the pose between frames $k-1$ and k from the last round of estimation. The texture difference function, $\tau(.)$, is given by the following.

$$\tau(r) = \frac{\sum_p |I_{Ref}(p) - I_{Target}(p)|}{\|M_{Ref} \cap M_{Target}\|} \qquad p \in M_{Ref} \cap M_{Target} \tag{5.2}$$

That is, for all pixels in the common area of the silhouettes of the target (M_{Target}), and of the image of the rotated reference patch (M_{Ref}), sum up the absolute difference of intensity values of target and reference, normalized by the number of pixels in $M_{Ref} \cap M_{Target}$. In other words, compute the average intensity difference between target and reference per pixel.

Generally pose variables ($\hat{P}, P$ and r) are 3×1 vectors. The proposed system is fully capable of working with 3-D pose. However, in all experiments reported in this chapter scalar (1-D) poses are used. That is because much more time is required to run the system with 2-D or 3-D pose (the run time computed in Section 5.3.1 squared or cubed).

The above formulation (Equations (5.1) and (5.2)) is for gray scale images. To extend the method to color images, there are two options: (1) before going through the method, sum up all (2/3 in the case of YIQ/RGB coloring scheme) color components to get the gray scale reference and target images; and (2) for each color component use the method to determine $\hat{P}_{ij}$. An estimate of the relative pose can then be computed as a function of these color poses.

We observe that even a function as simple as the average of the color poses gives a better estimate than the estimate given by option (1), mentioned above.

To reduce the effect of noise introduced in the imaging process (due to mis-calibration, for instance), we employed a simple filter on the difference images ($\mid I_{Ref}(p) - I_{Target}(p) \mid$) before summation over all pixels. The filter eliminates low intensity pixels from the sum.

Throughout the experiments reported in this chapter, we use the average of color poses, derived from Equations (5.1) and (5.2), with filtered difference images.

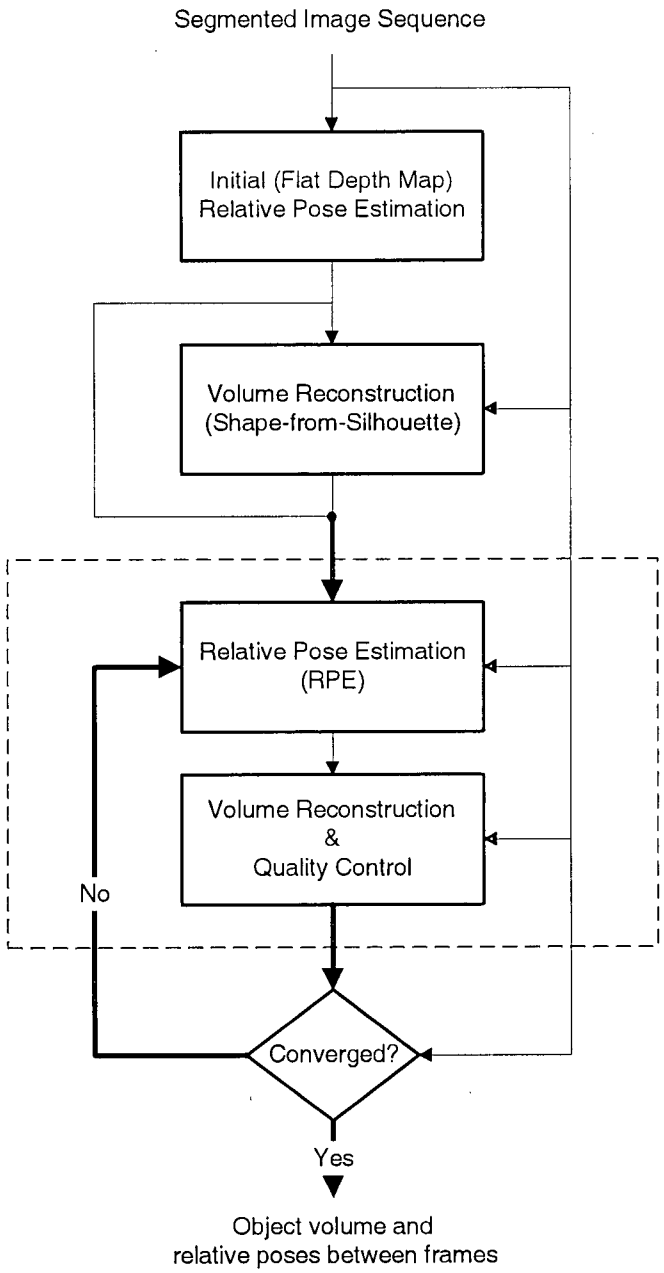


Figure 5.1: Block diagram of the proposed system. Thick lines indicate the data flow of volume and the set of relative poses. The part in the dashed box is studied in Section 5.4.

5.2.2 Flat depth-map pose estimation (FDMPE)

As the first step of our proposed method for volume reconstruction, we need to achieve an initial estimate of the object pose for each frame.

To this aim, we use our relative pose estimation (Section 5.2.1) method for every pair of consecutive frames, assuming flat depth-maps for all reference frames. Other assumptions for depth can be made, however a flat depth-map seems to be the most impartial.

Note that the assumption of depth (flat or other) does not restrict our method in any way. In other words, our method still works for arbitrarily shaped objects. However, the more the initial depth-map assumption is similar to the real object depth-map, the faster the method converges to the volume of the object.

After making this initial estimate, a volume of the object is made by shape-from-silhouette to be used for the RPE. To make this volume, we need the absolute poses assigned to all frames. Thus, we assign pose 0 to the first frame, and use the cumulative sum of the relative poses as the absolute poses of the other frames.

5.2.2.1 Unknown frame order case: multi-reference FDMPE

The method we just described can be used in cases where we know the order of input frames in advance, such as frames extracted from a video clip. When we do not have any information about the order of input frames, we use multiple reference frames for each target, and estimate the relative pose (with the assumption of a flat depth-map) between them. A table is made out of the results, pose estimates and difference measures for each possible pair of input frames (e.g., Table 5.1).

For each target, we select one frame as the reference according to the follow-

ing two rules, based on which Algorithm 1 is devised. Example 5.2.1 illustrates operation of Algorithm 1.

- **Maximum similarity.** The selected reference frame should give the minimum difference (maximum similarity measure) at the estimated pose among all other reference candidates for this target.

- **Connectivity.** One should be able to connect any arbitrary pair of frames through a chain of reference-target frames. That is because, for each frame, we should be able to compute the absolute pose required for volume reconstruction. To this aim, we may have to select the second most similar frame as the reference (i.e., relaxing the maximum similarity rule above).

Algorithm 1 Selection of the best reference for each target.

1: Sort pairs of frames in ascending order of their difference measures and put the result on the list L. Therefore, the first frame pair on L (i.e., $L(1,1)$ and $L(1,2)$) has the minimum difference measure.
2: Construct the frame list F of length N (number of frames); Each element has a list of possible connected frames (CF; initialized empty) and a binary flag for connectivity check (CCF; initialized 1 for one arbitrary element, and 0 for others).
3: **for** $K = 1$ to $length(L)$ **do**
4: Append $L(K,2)$ to $F(L(K,1)).CF$. {Connecting $L(K,2)$ and $L(K,1)$ ...}
5: Append $L(K,1)$ to $F(L(K,2)).CF$.
6: **if** $F(L(K,1)).CCF$ OR $F(L(K,2)).CCF$ **then**
7: Set CCFs of $F(L(K,1))$, $F(L(K,2))$ and every frame on their CF lists to 1.
8: **end if** {Connecting $L(K,2)$ and $L(K,1)$: Done.}
9: **if** all frames are connected (AND CCFs of all F elements) **then**
10: Break.
11: **end if**
12: **end for**

		Pose Estimates (Degrees)					Difference Measures				
Target:		A	B	C	D	E	A	B	C	D	E
Reference	A	n/a	36	56	68	72	n/a	87	129	181	225
	B	0	n/a	44	64	88	139	n/a	98	154	191
	C	0	0	n/a	52	88	191	151	n/a	119	156
	D	4	4	4	n/a	40	229	189	140	n/a	106
	E	0	0	0	68	n/a	294	251	187	91	n/a

Table 5.1: Multi-reference flat depth-map pose estimates and difference measures for Example 5.2.1.

Example 5.2.1 We use five consecutive frames, 10 degrees apart, from a prism (Figure 5.3). For simplicity, the relative pose estimation is performed over gray scale targets, and references are the results of summation over all color components. The results are given in Table 5.1.

The L list is as follows.

Difference measure	87	91	98	106	119	129	...
Frame pair	A B	E D	B C	D E	C D	A C	...

The connectivity is reached after the fifth iteration of the "for" loop (Algorithm 1). At this stage, the frames on the F list are connected as depicted in Figure 5.2. Therefore, A is the reference for B, B for C, C for D and D for E, as expected for an ordered sequence. **End of example.**

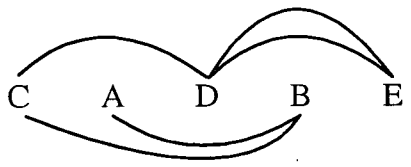


Figure 5.2: Connectivity diagram of the frames in Example 5.2.1.

The amount of pose estimates is approximately four times the ground truth values (10 degrees). We may use the method discussed in Section 5.3 to get a scaled set of relative poses giving the best volume reconstruction quality. In contrast, we may always scale down the resulting pose estimates. Since volume reconstruction quality control is performed later during the process anyway, this choice is not critical as long as the order of frames is correct. Thus, the second choice is used for simplicity and to lessen computational burden. Therefore, the relative pose estimates between consecutive frames in Example 5.2.1 are 9, 11, 13, and 10 degrees, using a scale factor of four.

Although performing the relative pose estimation among all pairs of frames is cumbersome, this has to be performed only once at the start of processing and does not need to be done in real time.

5.3 *Volume feedback loop*

The main contribution discussed in this chapter is to enhance pose estimations (equivalently, object volume estimations) through the feedback of volume (pose) information. That is, we reconstruct the object volume (based on the last round of pose estimations) and use depth information for a new round of pose estimations between consecutive frames.

During the experimental verification of this method, we noticed that, although relative pose estimates approach the same pattern for the ground truth relative poses between frames, the sum of the relative pose estimates remains almost constant through iterations of the volume feedback loop. Thus, even after several iterations, the pose difference between the last and the first frames is almost the same as that value computed with the initial pose estimates, assuming

a flat depth-map.

Example 5.3.1 In an experiment, 10 frames are taken from a textured prism every 10 degrees. Hence, the ground truth relative pose set is $\{10, \cdots, 10\}_{1\times 9}$, and the correct pose difference between the last and the first frames is 90 degrees. Except for the lack of corrective measures for the relative pose, the experiment in this example is the same as the first experiment reported in Section 5.5 with a high voxel resolution. The initial (flat depth-map) estimation gives a relative pose set of $\{5, 7, 8, 9, 11, 12, 15, 17, 13\}$. After five iterations, we reach a relative pose set of $\{11.8, 12.2, 11.5, 11.1, 10.8, 10.1, 9.5, 9.8, 11.1\}$ which does not change much during the next iterations (convergence). One may observe that the pattern of relative pose estimates appears to approach the ground truth: the standard deviation of the relative pose set changes from 3.9 degrees (initial estimation) to 0.9 degrees (fifth iteration). Considering the precision of the method (less than 2 degrees), the pattern of the relative pose at the fifth iteration is uniform, the same as that of the ground truth relative pose. However, the pose difference between the last and the first frames (i.e., the sum of the relative poses in each set) does not change much (from 97 to 97.9 degrees), and is several degrees off the true value (90 degrees). **End of example.**

Deviation of the FDMPE results from the true values are because of the assumption flat depth-map for all input images. However, we still do not know why this deviation is not corrected through iterations of the volume feedback loop by itself. The theoretical convergence study of Section 5.4 does not shed any light on this matter either, since it assumes deviations from the true pose values are small. While the reason of this phenomenon is unknown, its dramatic result is

known: the reconstructed volume does *not* converge to the object volume, unless some control mechanism is employed.

As mentioned above, the pattern of relative pose estimates converges to the ground truth, except for a scale factor. This observation, along with our work on validity criterion for shape-from-silhouette (Section 4.4), inspired a solution to remedy this problem. The method we developed in Section 4.4.1 takes a set of silhouettes and their relative poses as the input, and yields a quality factor for volume reconstruction as the output. Here, we use this quality factor in our algorithm to ensure convergence to the ground truth.

To achieve correct estimates for a relative pose, we run the method for volume reconstruction quality control on the original set of relative pose estimates and two scaled sets of it: one set with a scale factor of below 1, and the other with a scale factor of above 1. Among these three sets of relative poses, we use the set that gives the best reconstruction quality.

We use scale factors 0.9 and 1.1 based on the following observations. Scale factors very far from one (e.g. 0.2 and 5) do not make good volume candidates (see comments on initial pose estimation in Section 5.6). Scale factors very close to one (e.g. 0.9999 and 1.0001) make volumes same as the volume made by the original set of pose (unit scale) because of limited reconstruction resolution. Even if the accuracy were not limited, scale factors very close to one make the convergence time very long. Finding the optimal parameter values for the fastest convergence can be an extension to the method. For instance, scale factors farther from one in early iterations that approach one through the course of iterations may accelerate the method.

In Example 5.3.1, if we have a relative pose set[1] of $S_{1.0} = \{11.8, 12.2, 11.5, 11.1, 10.8, 10.1, 9.5, 9.8, 11.1\}$ by the end of the fifth iteration, we can make two scaled sets out of it: $S_{0.9} = 0.9\ S_{1.0} = \{10.6, 11, 10.4, 10, 9.7, 9.1, 8.6, 8.8, 10\}$ and $S_{1.1} = 1.1\ S_{1.0} = \{13, 13.4, 12.7, 12.2, 11.9, 11.1, 10.5, 10.8, 12.2\}$. Using our volume reconstruction quality factor, we see that $S_{0.9}$ gives the best reconstruction quality compared to $S_{1.0}$ and $S_{1.1}$. Hence, we choose $S_{0.9}$ as the relative pose set required in the next iteration.

Our experiments (Section 5.5) show that the method for volume reconstruction quality control ensures the convergence of estimated relative poses to the ground truth values.

After each iteration, the estimated relative poses are compared to values from the previous round of estimation. The feedback loop is terminated if the difference between the two sets is smaller than a certain threshold.

5.3.1 Complexity analysis

The proposed system (Figure 5.1) has one feedback loop (volume feedback) preceded by one processing block (FDMPE). The processing time also depends on how many times the loop is iterated before convergence.

FDMPE takes $O(N^2)$ to complete since it performs computations for all possible pairs of N input frames. In the case of ordered frames in the input, however, the run time of this block reduces to $O(N)$ since computations are performed for all pairs of consecutive frames only.

RPE takes the main part of processing time of the loop. Its processing time

[1] This set would change if the pose correction method were used in the previous iterations. It might be a good idea *not* to use the pose correction method within the first few iterations (Section 5.6).

is $O(N)$ since the frames are already ordered and computations are needed for all pairs of consecutive frames.

Assuming the number of iterations needed for convergence is almost constant (justified by observing the results reported in Section 5.5), the total processing time for the system is approximated by $O(N)$, and $O(N) + O(N^2)$ for the cases of known, and unknown frame order respectively.

5.4 Convergence issues

Although the estimated poses (and consequently the reconstructed volume) converge to their true values in a wide variety of experiments reported in Section 5.5, we want to know whether convergence occurs for *any* given input (i.e., segmented image sequence). In this section, we study convergence of the proposed method in the light of Poincaré-Lyapunov theorem [76] adopted from perturbation theory for multi-variable non-linear systems. The result of this study is a convergence test for the proposed system at a given vector of relative poses.

Consider the part of system inside the feedback loop (inside the dashed box in Figure 5.1). The input, s, and the output of this subsystem are $n \times 1$ vectors of relative poses[2], $n = N-1$ (N: number of frames in the segmented image sequence). We denote the non-linear function of this subsystem by $F(\cdot)$. Assuming the sum of all relative poses does not need adjustment (i.e., always unit-scaled vector is chosen; see Section 5.3), $F(\cdot)$ is a time-constant function.

[2] Actually we also have reconstructed volumes both at the input and the output of the subsystem. However, these volumes only facilitate the processing and have no new information because they are reconstructed from input/output vectors of relative poses and the segmented image sequence at the system input. Also, the segmented image sequence is not considered as an input in the convergence analysis, since it does not change through the course of iterations.

Since $F(\cdot)$ is put in a feedback loop, the iteration equation is given by

$$\frac{\Delta s}{\Delta t} = F(s) - s,$$

which is also the measure of convergence: if $\| F(s) - s \|$ is less than a threshold, the feedback loop is stopped. We are interested in studying behavior of $F(\cdot)$ in the neighborhood of q, a point (or vector; we use these two words interchangeably hereafter) for which $F(q) = q$. Substituting $s = y + q$ into the iteration equation above, we obtain

$$\frac{\Delta y}{\Delta t} = F(y + q) - y - q$$

The Poincaré-Lyapunov theorem [76] is quoted below.

Theorem 5.4.1 *If (a) all solutions of the linear system* $\frac{dx}{dt} = Ax$ *approach the origin as* $t \to \infty$, *(b) the initial value, c, is sufficiently close to the origin, (c) the non-linear term,* $h(x)$, *consists of a vector all of whose components are power series lacking constants and linear terms, then the solution of*

$$\begin{aligned} \frac{dy}{dt} &= Ay + h(y), \\ y(0) &= c \end{aligned}$$

approaches the origin as $t \to \infty$.

Restating our problem in terms of Theorem 5.4.1, we want to know whether or not the solution of

$$\frac{\Delta y}{\Delta t} = G(y),$$

$$y(0) \quad = \quad s_{init} - q$$

approaches the origin as $t \rightarrow \infty$, where $\mathcal{G}(y) \equiv F(y+q) - y - q$, and s_{init} is the initial relative pose estimate vector given by FDMPE (Section 5.2.2).

Before using Theorem 5.4.1, we have to expand $\mathcal{G}(y)$ in form of $Ay + h(y)$ according to condition (c) of the theorem. A is *total derivative*[77] of $\mathcal{G}(\cdot)$ at the origin. Functions $F(y)$ and consequently $\mathcal{G}(y)$ do not have closed forms. Therefore, we have to estimate the total derivative numerically.

Assuming small test vectors (y_i), $h(y_i)$s are small as well. Following the least squares (LS) approach, we want to find an A to minimize $\sum \parallel h(y_i) \parallel^2$. That is,

$$A = \arg\min_{B} \sum_{\forall m} \|\mathcal{G}(y_m) - By_m\|^2. \tag{5.3}$$

The solution (see Section 5.4.1 for derivation) is

$$A = GY^T(YY^T)^{-1}, \tag{5.4}$$

in which $Y_{n\times p} = \{y_1, y_2, \cdots, y_p\}$, $G_{n\times\ p} = \{\mathcal{G}(y_1), \mathcal{G}(y_2), \cdots, \mathcal{G}(y_p)\}$, and ^{T} stands for transposition.

We can improve LS estimation (A) of $\mathcal{G}(\cdot)$ total derivative by increasing the number of test vectors (p) and distributing the test vectors uniformly. In our experiments, we used $Y = \epsilon\{I, -I\}$ meaning $p = 2n$ vectors on both positive and negative directions on all n coordinate axes. Ideally, ϵ is desired to be very small, however since computation of $F(\cdot)$ involves quantized reconstruction and imaging, it cannot be less than half a degree in practice.

Condition (b) of Theorem 5.4.1 is met, if s_{init} is close to q. A given by

Equation (5.4) is a good estimate of the total derivative of $\mathcal{G}(\cdot)$ at the origin, therefore, condition (c) of the theorem is also satisfied. Condition (a) is satisfied if, and only if, all eigenvalues of A have negative real parts [78]. Hence, a test for convergence of the proposed method to $q_{n \times 1}$, a given vector of relative poses, is stated in Algorithm 2.

Algorithm 2 A test for convergence of the method to relative pose vector q.

1: $\mathcal{G}(\cdot)$ must be zero at the origin. That means $F(q)$ must be equal to q.
2: Select a number (p) of small test vectors, $Y_{n \times p} = \{y_1, y_2, \cdots, y_p\}$, uniformly distributed around the origin.
3: Compute $G_{n \times p} = \{\mathcal{G}(y_1), \mathcal{G}(y_2), \cdots, \mathcal{G}(y_p)\}$.
4: Use Equation (5.4) to estimate the total derivative of $\mathcal{G}(\cdot)$, A, at the origin.
5: Compute eigenvalues of A. The proposed method converges to q, if all eigenvalues of A have negative real parts.

All cases presented in Section 5.5.1 are examined by this test (Algorithm 2) for convergence to the ground truth vectors of relative poses. Results confirm convergence universally.

5.4.1 Least squares estimation of total derivative

The least squares estimation of $\mathcal{G}(\cdot)$'s total derivative at the origin requires minimization of the following error function, $E(B)$, with respect to $B = \{b_{ij}\}$. The notation is the same as that of Equation (5.3).

$$\begin{aligned} E(B) &= \sum_{m=1}^{p} \|\mathcal{G}(y_m) - By_m\|^2 \\ &= \sum_{m=1}^{p} (\mathcal{G}(y_m) - By_m)^T (\mathcal{G}(y_m) - By_m) \\ &= \sum_{m=1}^{p} [\mathcal{G}(y_m)^T \mathcal{G}(y_m) - 2y_m^T B^T \mathcal{G}(y_m) + y_m^T B^T B y_m]. \end{aligned} \tag{5.5}$$

The last expression is reached using $y_m^T B^T \mathcal{G}(y_m) = \mathcal{G}(y_m)^T B y_m$.

To this aim, we solve the following set of equations simultaneously for the elements of B.

$$\frac{\partial E(B)}{\partial b_{ij}} = 0, \qquad i = 1, \cdots, n, \qquad j = 1, \cdots, n.$$

We substitute $E(B)$ from Equation (5.5) and change the order of the two linear operators, $\frac{\partial}{\partial b_{ij}}$ and $\sum_{m=1}^{p}$. We reach the following equation.

$$\begin{aligned} 0 &= \frac{\partial E(B)}{\partial b_{ij}} \\ &= \sum_{m=1}^{p} \frac{\partial}{\partial b_{ij}} [\mathcal{G}(y_m)^T \mathcal{G}(y_m) - 2 y_m^T B^T \mathcal{G}(y_m) + y_m^T B^T B y_m]. \end{aligned}$$

In order to derive the solution in matrix form, we switch to a new notation: $Y_{n\times p} = \{y_{ij}\} = \{y_1, y_2, \cdots, y_p\}$, $G_{n\times p} = \{g_{ij}\} = \{\mathcal{G}(y_1), \mathcal{G}(y_2), \cdots, \mathcal{G}(y_p)\}$, and $\bar{x}_{*m}/\bar{x}_{m*}$ is the mth column/row of matrix X. In the following, the partial derivatives are computed after application of the new notation.

$$\begin{aligned} 0 &= \frac{\partial E(B)}{\partial b_{ij}} \\ &= \sum_{m=1}^{p} \frac{\partial}{\partial b_{ij}} (\bar{g}_{*m}^T \bar{g}_{*m} - 2 \bar{y}_{*m}^T B^T \bar{g}_{*m} + \bar{y}_{*m}^T B^T B \bar{y}_{*m}) \\ &= 2 \sum_{m=1}^{p} y_{jm} (\bar{b}_{i*} \bar{y}_{*m} - g_{im}). \end{aligned}$$

The last sum must be zero and can be written as the following vector product.

$$(\bar{b}_{i*} Y - \bar{g}_{i*}) \bar{y}_{j*}^T = 0. \tag{5.6}$$

The scalar Equation (5.6) is for a fixed i and a fixed j. For a fixed j and $\forall i$, we obtain an array of scalar Equations (5.6) that can be written as the following vector equation.

$$(BY - G)\bar{y}_{j*}^T = \bar{0}_{n\times 1}. \tag{5.7}$$

$\forall j$, we have an array of Equations (5.7) that can be summarized in one matrix equation:

$$(BY - G)Y^T = 0_{n\times n} \qquad \Rightarrow \qquad BYY^T = GY^T.$$

Under the circumstances discussed in Section 5.4, the inverse of YY^T exists. Therefore,

$$B = GY^T(YY^T)^{-1}.$$

5.5 *Experiments*

Results of several experiments are reported in this section to show the reliable performance of our proposed method in various conditions. The experiments are performed on three different textured objects: a sphere, a prism (synthetic sequences) and a human head (real sequence). For the synthetic sequences, we select pose samples and use our method to estimate the relative poses between frames and to reconstruct the volume of the object. The experiments are conducted for different pose sampling rates and space (voxel) resolutions. The good result of the head sequence shows viability and robustness of the proposed method.

5.5.1 Results for synthetic sequences (textured prism and sphere)

Three views of each object are shown in Figure 5.3. Color 3-D versions of the objects used in the experiments are available from [2]. A chroma-keying scheme is used to segment the synthetic sequences.

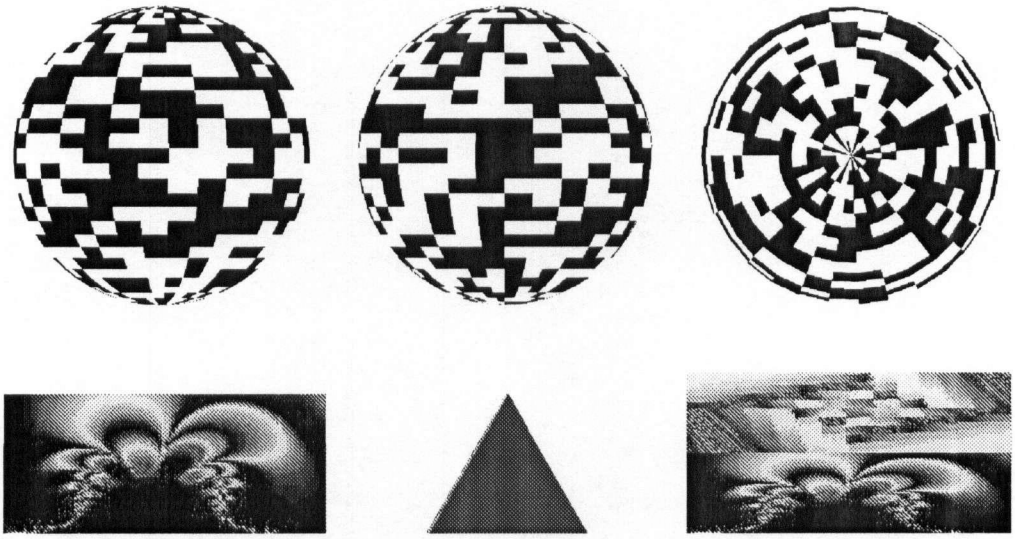


Figure 5.3: Front, left and top orthographic views of the two objects used in the experiments. For color 3-D models of the objects, see [2].

In our first experiment, 10 pose samples are taken every 10 degrees (ground truth) from the prism. These pose samples are fed to the system with voxel resolutions of 111 (Low), 161 (Moderate), and 211 (High). The voxel resolution is defined as the number of voxels making the length of the prism. The output of the system after each iteration is compared to the ground truth. That is, the resulting set of estimated relative poses after each iteration is compared to the set of ground truth relative poses. The results are given in Figure 5.4.

For all three voxel resolutions, the initial pose estimation is the same, hence the same error value. At first glance, it seems there is no relationship between the

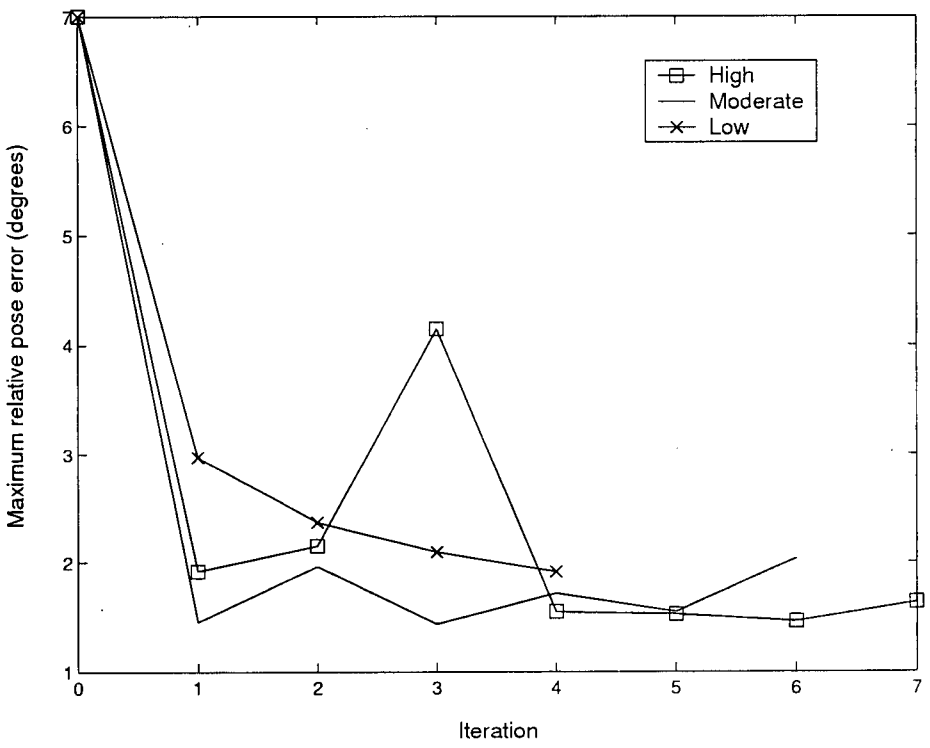


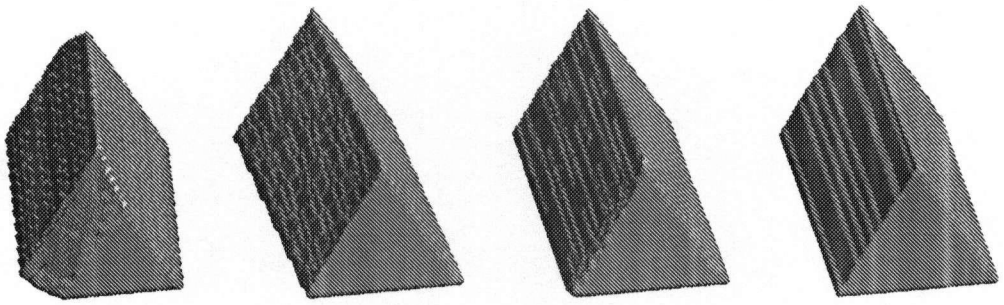
Figure 5.4: Maximum pose error for low, moderate and high voxel resolutions. Iteration # 0 is the flat depth-map (initial) pose estimation.

voxel resolution and quality of estimation. However, one can see that for a higher voxel resolution, the convergence criterion is reached later, and the maximum amount of pose errors of the final iteration is less.

In Figure 5.4, there is a substantial overshoot at the third iteration for high voxel resolution. The same overshoot also occurs in other graphs where the results of this experiment are used for comparison. We reviewed the experiment for possible mistakes; the results are correct and some fluctuations are normal through iterations before convergence.

It is imperative to see how the reconstructed volume evolves during iterations (Figure 5.5). The voxel resolution is set to High (211). It is observed that

the reconstructed volume gradually converges to the ground truth volume of the object. Note that although the maximum pose error value for the last iteration is slightly higher than that of its previous iteration (see Figure 5.4, "High" voxel resolution curve), the volume improves in the last iteration by having the same orientation as the ground truth volume.



Initial Iteration 6 Iteration 7 (Last) Ground truth

Figure 5.5: Evolution of reconstructed volume through iterations. From left to right: initial (flat depth-map results), one iteration before the last, the last iteration, and ground truth volume, un-textured, all shown from the same view.

The second experiment investigates the performance of the system for different sampling rates. At the constant voxel resolution of 211, we sampled the prism with periods of 6.9 (High sampling rate), 10 (Moderate), and 15 (Low) degrees. The sampling period is defined as the amount of rotational distance between two consecutive poses. The resulting set of estimated relative poses after each iteration is compared to the set of ground truth relative poses (Figure 5.6).

One may see that, although there seems to be no relationship between the sampling rate and the maximum pose error through iterations, the final value of the maximum pose error is lower at higher sampling rates. However, if we normalize the results by sampling periods (Figure 5.7), we see that during iterations,

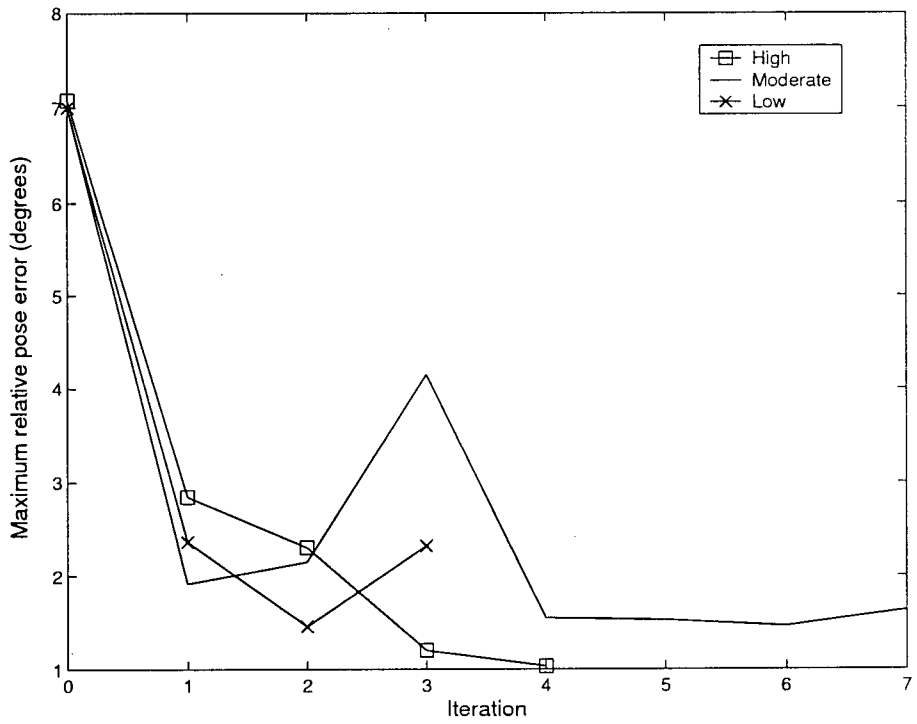


Figure 5.6: Maximum pose error for low, moderate and high sampling rate. Iteration # 0 is the flat depth-map (initial) pose estimation.

lower sampling rates result in lower normalized error levels. It is also observed that the final value of the normalized error is independent of sampling rate. The search for maximum similarity in the RPE routine is discrete. The final value of normalized error depends on the gap between two consecutive points on that search grid, if the convergence criterion is tight enough.

A sphere, an example of a smooth[3] object, with a black and white texture, is selected as the subject of the next experiment. 16 poses of the sphere, taken

[3] Unlike a rough object (e.g., a prism), a smooth object does not have any (sharp) edges, hence its silhouette does not change abruptly by small rotations; in the case of a sphere, the silhouette does not change at all. Also unlike a rough object, even a small rotation reveals part of the object not seen in the previous pose. See also [41].

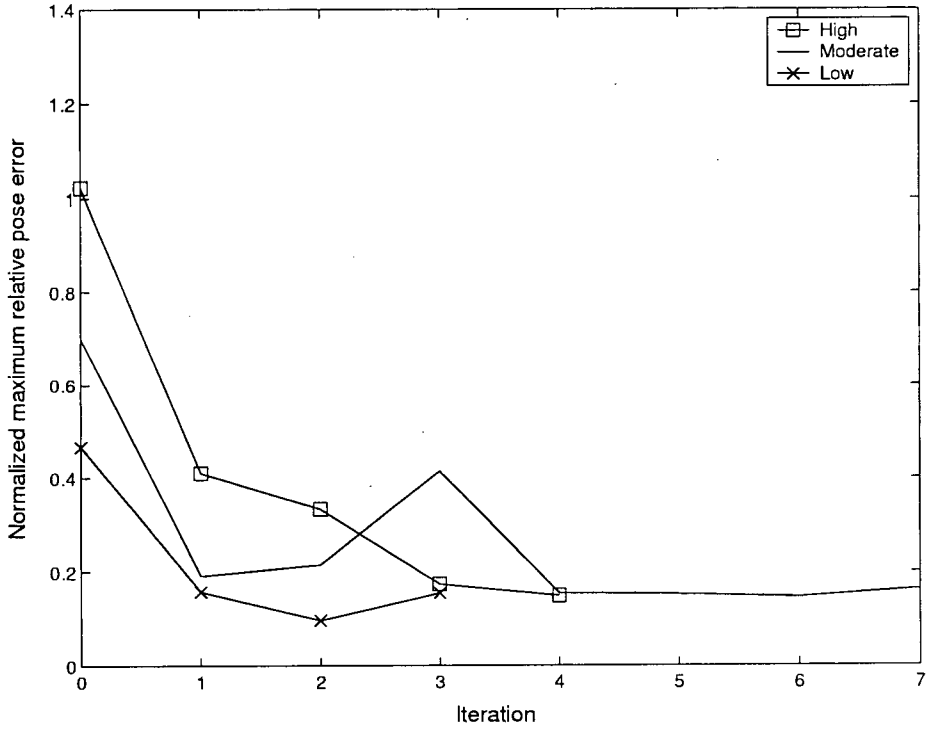


Figure 5.7: Maximum pose error (normalized by sampling period) for low, moderate and high sampling rate. Iteration # 0 is the flat depth-map (initial) pose estimation.

every 12 degrees, are fed to the system with a 211 voxel resolution. The results are satisfactory (Figure 5.8).

5.5.2 Human head sequence

In this experiment, eleven frames of a sequence showing a human head are fed to the system as input. Frames 1, 6, and 11 are shown in Figure 5.9. All frames and their silhouettes (semi-automatically computed based on color) can be obtained from [2]. The following merits of the system will be shown by good performance of the system, that is, successful reconstruction of the head volume in this exper-

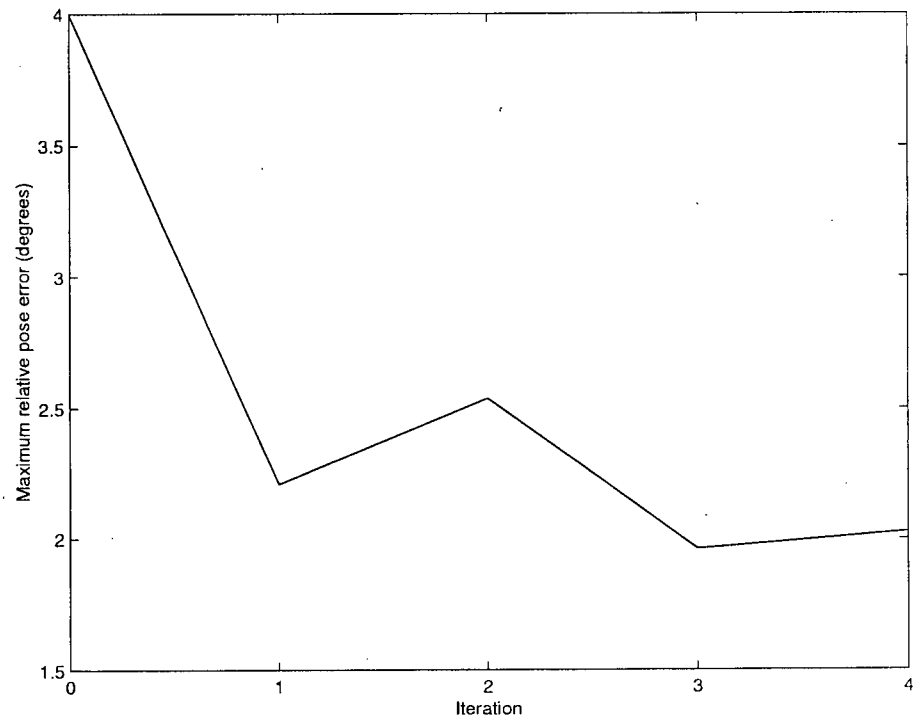


Figure 5.8: Maximum pose error for the sphere. Iteration # 0 is the flat depth-map (initial) pose estimation.

iment. The system is shown to be robust to the following effects.

- **Shadow.** Unlike frames of the synthetic sequences used in the experiments of Section 5.5.1, shadow is present in parts of the frames in the head sequence.

- **Imperfect segmentation.** The method used for segmentation classifies some background pixels as the silhouette of the object.

- **Translation.** The object (head) in this sequence undergoes translation, as well as rotation [2].

Figure 5.9: Frames 1, 6, and 11 of the head sequence, the input to the system in the experiment reported in Section 5.5.2.

5.5.2.1 Object translation compensation

The original proposed system (Figure 5.1) only permits rotation of the object in the input sequence. To handle translations, the following adjustments are made. One, all frames of the input sequence are translated so that the centres of mass of their silhouettes are placed at the same point. Two, in the RPE block (Section 5.2.1), we compensate for the translation between the reference patch and the target before searching for their relative pose.

The first task above is conducted by computing the centre of mass for each silhouette. Next, each frame is copied into a certain template so that the centre of mass of its silhouette falls at the same point as the centre of the template. Performing this task prevents the opposing silhouettes from carving out details generated by each other (Figure 5.10).

Compensation for the translation between the reference patch and the target in RPE block (the second task above) is performed by computing a single optic flow vector for all pixels on the foreground using the first method described in [79].

Figure 5.10: The silhouettes of frames 1 (M_1) and 11 (M_{11}) of the head sequence. M_{11} subtracted from flipped M_1. Details (black/white parts) carved by M_{11}/M_1 are destroyed by the other silhouette, unless the silhouettes are moved to have the same centre of mass (Section 5.5.2).

The 3-D reference patch is then moved according to the computed translation vector. The usual problems with optic flow computation methods are avoided since only a single vector is computed based on the motion information of a large number of pixels. For more details, see Appendix B.

5.5.2.2 Results

The results are given in Figures 5.11 and 5.12. Note that, except for the sum of relative poses (i.e., the amount of absolute pose for the last frame), we do not have any other information about the true values of poses, either relative or absolute. That is because the input sequence is shot in an uncontrolled environment with a regular consumer camera. Therefore, unlike the previous experiments, the performance cannot be checked by comparison of the pose estimates to the true values.

To see the improvement of the pose estimates through iterations, the sum of

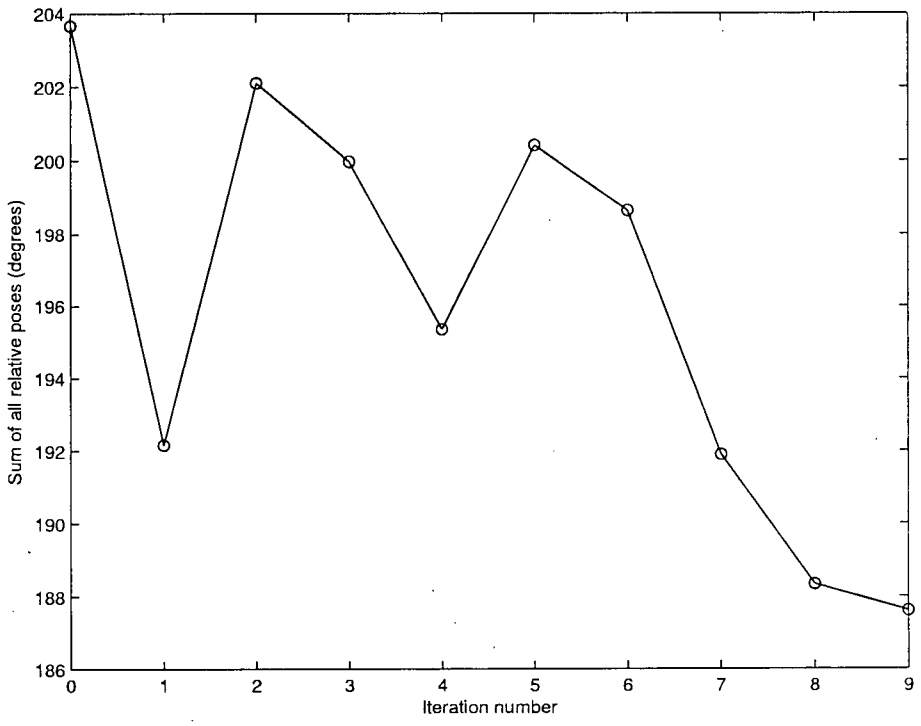


Figure 5.11: The sum of relative poses through iterations for the head sequence. The ground truth value of the sum is 180 degrees.

all relative poses is monitored (Figure 5.11). The sum approaches the true value of 180 degrees and, at the final iteration, the sum differs only 7 degrees from the true value, which means an average error of only 0.7 degree on each relative pose.

The reconstruction quality factor is also increasing through iterations (Figure 5.12); the best reconstruction quality occurs at the last iteration. Note that because of imperfect segmentation of the input sequence, the final value of the quality factor for this sequence is approximately 20% lower than the corresponding values for the synthetic sequences of Section 5.5.1.

One also observes the convergence to the true volume qualitatively (Figure 5.13). The reconstructed volumes at four iterations are used to render the

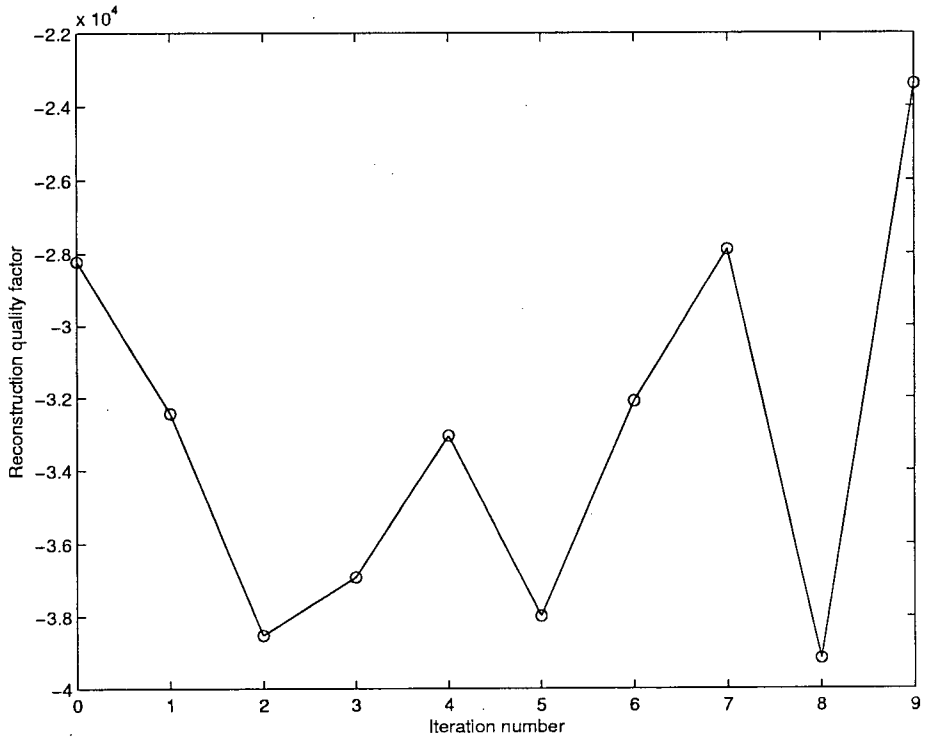


Figure 5.12: The reconstruction quality factor through iterations for the head sequence.

first frame of the sequence. In other words, the reconstructed volume is textured and observed at the absolute pose estimate of the first frame in several iterations. The details of the object (especially notice the nose and lips area) absent from the reconstructed volume in early iterations, appear gradually and are fully present in the final volume made at the final iteration.

Compared to the experiments with synthetic input sequences, the head sequence seems to require more time to converge, perhaps because it is a more complicated object with more details. Also note that the number of frames required for perfect reconstruction of a prism is 2 (one on the prism axis, the other perpendicular to the axis). This number is infinite for a human head. In the case

that only a certain level of reconstruction accuracy is required, we still do not know how many input frames are needed. This question is listed as a future work in Section 5.6.

5.6 Concluding remarks

In this chapter, the question of "How to reconstruct object volume using a certain set of input images?" is addressed. In some situations, however, we have to select a subset of the input images. For example, in the case of availability of a large number of input images, a subset of images are selected to maintain a desirable level of accuracy in volume reconstruction, as well as a reasonable processing time. Considering that the processing time of our method is $O(N^2) + O(N)$ (N is the input size; Section 5.3.1), the question of "How much input is needed for a certain reconstruction precision?" is of importance, and should also be addressed in future work.

The initial pose estimation with a flat depth-map assumption (Section 5.2.2) is of critical importance. That is, in the case that initial set of pose estimates are way off the ground truth, the system may not converge to a good solution at all. One might use a random search tool, such as a genetic algorithm, to search the solution space for the relative pose set that gives the best reconstruction quality (Section 4.4). In this way, although a good initial solution can speed up the method towards convergence, a bad one cannot lead to a total disruption.

In Section 5.5.2, successful volume reconstruction and pose estimation from an un-calibrated input image sequence is reported. That is, while volume reconstruction and feedback use orthographic projection (or weak perspective[4] [75]),

[4] The assumption of weak perspective projection, even in small distance/size ratios, introduces

the method is operational for regular video footage as long as perspective effects are small (i.e., the camera is not too close to the object and the object remains close to the optical axis of camera).

For segmentation of the object from background, we used a semi-automatic method for the real sequence and chroma-keying (a production technique) for the synthetic sequences. Although our experiments showed the robustness of the method to un-calibrated input images and errors in segmentation, comprehensive study of the effects of errors in segmentation and camera parameters has to be conducted in a continuation of this work.

Shape-from-silhouette is only able to reconstruct the object volume up to its visual hull [73]. Due to this inherent limitation, our method cannot reconstruct the volume of concave objects better than their visual hull accurately. In the case of a concave object, however, the resulting volume can be refined, using photo-consistency conditions [81], to a better approximation of the true object volume.

One possible extension to this work is using other visual cues (e.g., shading), when available, along with silhouette information to enhance the reconstruction accuracy, especially for concave objects. Another line of future work is to handle paraperspective [75], and full perspective input images.

Currently, the search for the estimate of relative pose (Section 5.2.1) is exhaustive. Use of a heuristic search instead, can expedite the whole system, since RPE is the major part of the processing. A possible disadvantage is that a heuristic search is prone to error. Thus, we have to study how much error we can afford in estimation of relative poses, and ensure the error level of the selected heuristic

negligible errors as long as the object lies close to the optical axis of camera [80] (i.e., the camera is focused on the object). Therefore, assumption of orthographic or weak perspective is not a serious limitation for our method.

search tool is less than the permitted error level.

In addition, to speed up the process, we are investigating the elimination of the relative pose correction step (Sections 5.3 and 4.4.1) in the few first iterations. The intuition behind this idea is to let the system learn the pattern of relative poses before trying to correct possible scaling errors. Questions, such as "How many iterations should be exempted from pose correction to leave the quality of the final reconstructed volume unchanged?" should be addressed in the feasibility study for this alteration.

Figure 5.13: The first frame of the sequence, rendered from the volumes reconstructed at iterations (from left to right, and top to bottom) 0 (initial estimation), 1, 7, and 9 (final). Note gradual improvement of details, especially in the lips and nose area.

6: Conclusion

In this chapter, an extended summary of the thesis is given. The summary includes contributions made to the literature (a distinctive list of contributions is given in Section 2.5). A number of ways to improve, and generalize the methods introduced in this thesis, along with their limitations, are also discussed.

6.1 Extended summary

Background

The three-dimensional perception of an object (a scene) is closely related to the ability to distinguish between various poses of the object (pictures taken from different camera viewpoints). Several methods [Chapter 2], such as shape-from-silhouette, light field rendering, and space carving using photo consistency, have addressed reconstruction of an object (a scene) from its images, taken from known poses (camera viewpoints). That is, these methods are not readily usable for extraction of 3-D information from monocular video sequences. Among the existing methods that can provide the required pose information, only feature-based methods function for general objects and are suited to images taken in uncontrolled environments. Feature-based methods, however, suffer from their need to detect and track feature points across images in the sequence.

Applications

Applications for the central method developed in this thesis include the following [Chapter 1]: creation of digital museums, 3-D extensions to MPEG video coding standards (by 3-D object tracking in video), and generation of virtual environment (e.g., photo-realistic, computer generated animation), all from regular video footage that are shot in uncontrolled environments. The system also generates an index of object poses for frames of the input sequence. Such an index can be used as an aid to existing image or video content-based retrieval engines, to answer higher-level, more complex queries.

Contributions

A detailed description of contributions is given below. A list of contributions, without details, is given in Section 2.5.

Multi-objective relative pose estimation

A novel approach for rigid object pose estimation selects one frame as the reference for object pose and projects it to its depth-map, to create a 3-D reference patch [Chapter 3]. The reference patch is then rotated until it reaches a view that seems "similar" to the unknown-pose view. A number of pose similarity measures were tested for different types of objects, undergoing various amounts of rotation from the reference pose. It is shown that the sum of texture and mask differences, in most cases, can be used as an effective pose similarity measure, capable of unique determination of the correct pose. A number of optimization methods (e.g., a genetic algorithm) are used as the feedback that relates pose

comparison to the reference frame rotation. The study recommends using different similarity measures and feedback methods for different types of objects. The proposed method has the following advantages over the existing pose estimation approaches: one, it is general, not model-based. Two, unlike template matching methods, it does not require a learning phase prior to operation. Three, it is not vulnerable to problems with feature-point detection, tracking, and low density of feature points.

3-D modeling by shape-from-silhouette: effects of sampling rate and pose error

As mentioned above, pose information is used by a reconstruction method (shape-from-silhouette, selected for simplicity) to generate a 3-D model of the object shown in the video. The quality of the reconstructed 3-D model depends on the silhouette samples (viewpoint distribution), among other things. An improved viewpoint distribution is introduced and its performance is compared to the existing method empirically. Also, the effects of pose sampling rate variation and different levels of pose error, on silhouette-based reconstruction quality, are investigated. It is shown that, in the presence of considerable pose error, increasing the sampling rate cannot enhance the reconstruction quality, and that the reconstruction process is more sensitive to pose error at higher sampling rates.

3-D modeling by shape-from-silhouette: validity criterion

Specific relationships between the input silhouettes and their poses must hold, in order to produce a valid volume by shape-from-silhouette. These relationships, which are introduced for the first time by the author, can be verified to check if correct poses are assigned to the input silhouettes. Based on verification of

these relationships, a novel quality factor for volume reconstruction by shape-from-silhouette, is developed and used to select the best set of pose estimates from several candidates.

Elimination of the depth-map requirement by volume feedback: simultaneous pose estimation and object reconstruction

Multi-objective pose estimation [Chapter 3] requires the depth-map for at least one frame of the input sequence. Thus, the method has to rely on another technique (e.g., stereo vision). To eliminate this requirement and to make the method the first stand-alone correspondence-less method for volume reconstruction and pose estimation from video, a volume feedback scheme is adopted [Chapter 5]. Initially, the object pose is estimated in each image, assuming flat depth-maps. Shape-from-silhouette is then applied to make a 3-D model (volume), which is used for a new round of pose estimations, this time by a model-based method that gives better estimates. Before repeating this process, by building a new volume, pose estimates are adjusted to reduce error by maximizing a novel quality factor for shape-from-silhouette volume reconstruction. The feedback loop is terminated when pose estimates do not change much, as compared to those produced by the previous iteration.

Based on a theoretical study of the proposed system, a test of convergence to a given set of poses is devised. Reliable performance of the system is also proved, by several experiments on both synthetic and real image sequences. Computational complexity of the proposed system, and effects of sampling rate and voxel resolution on pose error are studied. In addition to the merits of the multi-objective pose estimation technique, elaborated in Chapter 3, the new system has

the following advantages: one, translation of the object in the input sequence is allowed and is compensated for in two stages [Section 5.5.2, Appendix B]. Two, the frames in the input sequence are re-ordered to achieve the best possible pose estimation [Section 5.2.2]. Three, a single procedure works for different types of objects and different amounts of relative pose.

6.2 Limitations

Specific limitations of each method are discussed at the end of the corresponding chapters. In the following, general limitations of the approach are reviewed.

The proposed pose estimation methods fail to function for un-textured objects. Note that even the human visual system cannot effectively distinguish pose or volume of a un-textured object under uniform illumination. That is, when no visual clue (e.g., shading or texture) is available except for silhouette changes (if any).

As a result of using shape-from-silhouette for volume reconstruction, the reconstruction accuracy is limited to that of the *visual hull* of the object [73]. That is because SFS-based reconstruction relies on the object silhouettes, which are not affected by object concavities (e.g., the nostrils in human face) at all. Note that if the method is used for generation of a virtual object, the model usually looks fine after texturing (e.g., the nostrils are coloured black, which is the case, unless in a special lighting situation).

6.3 Future work

In addition to the following, a few improvements and extensions, for each specific method, are discussed at the end of the corresponding chapters.

Short term goals

- In what manner and how much errors in segmentation of the input images, and camera calibration affect volume reconstruction and pose estimation accuracy?

- Use of other reconstruction methods, instead of shape-from-silhouette, in the volume feedback structure. At a higher computational cost, they may result in a better reconstruction quality, especially for concave objects.

Long term goals

- Fusion of depth information provided by other visual clues (e.g., shading).

- Fusion of volume (depth) information acquired from other video sequences or pictures of the subject. This may require use of a new volume representation that accommodate minor changes (non-rigidity) in the subject.

- How much input is needed for a certain reconstruction accuracy?

- A nearly real-time 3-D information retrieval system from video, based on the proposed approach may be achieved by converting codes from Matlab to C++, implementation on dedicated hardware, and optimizing the methods for speed.

- Application of the accelerated method, as described above, to very low bit-rate 3-D video coding, as outlined in Section 1.2.3.

- Application of the enhanced method, as described above, to non-rigid vision-based tracking of human body (or parts, e.g. hands) for the purpose of human-computer interaction (HCI).

References

[1] S.-C. Chen, R. L. Kashyap, and A. Ghafoor. *Semantic models for multimedia database searching and browsing.* Kluwer Academic Publishers, 2000.

[2] http://www.ece.ubc.ca/~alin/objects.htm, 2003. The objects and the head sequence used in the thesis.

[3] J. H. Lambert. *Photometria sive de mensura de gratibus luminis, colorum et umbrae.* Eberhard Klett, Augsburg, 1760.

[4] B.K.P. Horn. *Robot Vision.* McGraw-Hill, New York, 1986.

[5] J. Aloimonos and D. Shulman. *Integration of Visual Modules.* Academic Press, San Diego, CA, 1989.

[6] P. Viola and W.M. Wells. Alignment by maximization of mutual information. *International Journal of Computer Vision*, 24(2):137–154, 1997.

[7] Z. Yang and F. S. Cohen. Cross-weighted moments and affine invariants for image registration and matching. *IEEE Trans. on Pattern Recognition and Machine Intelligence*, 21(8):804–814, 1999.

[8] V. Govindu and C. Shekhar. Alignment using distribution of local geometric properties. *IEEE Trans. on Pattern Recognition and Machine Intelligence*, 21(10):1031–1043, 1999.

[9] A.E. Johnson and S.B. Kang. Registration and integration of textured 3d data. *Image and Vision Computing*, 17:135–147, 1999.

[10] L.S. Chen, T.S. Huang, and J. Ostermann. Animated Talking Head with Personalized 3D Head Model. In *First workshop on multimedia signal processing*, pages 274–279, 1997.

[11] M.K. Reed and P.K. Allen. 3d modeling from range imagery: an incremental method with a planning component. *Image and Vision Computing*, 17:99–111, 1999.

[12] W.B. Seales and O.D. Faugeras. Building three-dimensional object models from image sequences. *Computer Vision and Image Understanding*, 61(3):308–324, 1995.

[13] R. Vaillant and O.D. Faugeras. Using extremal boundaries for 3-d object modeling. *IEEE Trans. on Pattern Recognition and Machine Intelligence*, 14(2):157–173, 1992.

[14] Jane Mulligan et al. Trinocular stereo: a real-time algorithm and its evaluation. In *Proceedings of the IEEE Workshop on Stereo and Multi-Baseline Vision*, pages 10–17, December 2001.

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

[16] M. Potmesil. Generating octree models of 3d objects from their silhouettes

in a sequence of images. *Computer Vision, Graphics, and Image Processing*, 40:1–29, 1987.

[17] K. N. Kutulakos and S. M. Seitz. A theory of shape by space carving. *International Journal of Computer Vision*, 38(3):199–218, 2000.

[18] A. Broadhurst et al. A probalistic framework for space carving. In *Proc. of IEEE International Conf. on Computer Vision*, pages 388–393, 2001.

[19] M. Levoy and P. Hanrahan. Light field rendering. In *Computer Graphics Proceedings*, pages 31–42, 1996.

[20] S.J. Gortler, R. Grzeszczuk, et al. The lumigraph. In *Computer Graphics Proceedings*, pages 43–54, 1996.

[21] M. Pollefeys et al. Self-calibrartion and metric reconstruction inspite of varying and unknown camera parameters. *International Journal of Computer Vision*, 32(1):7–25, 1999.

[22] C. Lu et al. Fast and globally convergent pose estimation from video images. *IEEE Trans. on Pattern Recognition and Machine Intelligence*, 22(6):610–622, 2000.

[23] C.J. Poleman and T. Kanade. A paraperspective factorization method for shape and motion recovery. *IEEE Trans. on Pattern Recognition and Machine Intelligence*, 19(3):206–218, 1997.

[24] J. I. Thomas and J. Oliensis. Dealing with noise in multi-frame structure

from motion. *Computer Vision and Image Understanding*, 76(2):109–123, 1999.

[25] S.D. Blostein and R.M. Chann. The use of image plane velocity measurements in recursive 3-D motion estimation from a monocular image sequence. In *Proceedings of Canadian Conference on Electrical and Computer Engineering*, pages 797–801, 1994.

[26] G. Calvagno et al. Three-dimensional motion estimation of objects for video coding. *IEEE Journal on Selected Areas in Communications*, 16(1):86–97, 1998.

[27] B.K.P. Horn. *Robot Vision*, chapter 12. McGraw-Hill, New York, 1986.

[28] G. Adiv. Determining 3D Motion and structure from optical flow generated by several moving objects. *IEEE Trans. on Pattern Recognition and Machine Intelligence*, 7(4):384–401, 1985.

[29] A.N. Netravali and J. Salz. Algorithms for estimation of three-dimensional motion. *AT&T Technical Journal*, 64(2):335–346, 1985.

[30] G. Adiv. Inherent ambiguities in recovering 3d motion and structure from a noisy flow field. *IEEE Trans. on Pattern Recognition and Machine Intelligence*, 11(5):477–489, 1989.

[31] D. J. Heeger and A. D. Jepson. Subspace methods for recovering rigid motion I: Algorithm and implementation. *International Journal of Computer Vision*, 7(2):95–117, 1992.

[32] J. Aloimonos and D. Shulman. *Integration of Visual Modules*, chapter 4. Academic Press, San Diego, CA, 1989.

[33] S. Negahdaripour and B.K.P. Horn. Determining 3d motion of planar objects from image brightness patterns. In *Proc. of the international joint conf. on AI*, 1985.

[34] E. Elagin et al. Automatic pose estimation system for human faces based on bunch graph matching technology. In *Third IEEE International conf. on automatic face and gesture recognition*, pages 136–141, 1998.

[35] J. Heinzmann and A. Zelinsky. 3-D facial pose and gaze point estimation using a robust real-time tracking paradigm. In *Third IEEE International conf. on automatic face and gesture recognition*, pages 142–147, 1998.

[36] Q. Chen, H. Wu, et al. 3d head pose estimation without feature tracking. In *Third IEEE International conf. on automatic face and gesture recognition*, pages 88–93, 1998.

[37] H. Wu et al. Spotting recognition of head gestures from color image series. In *Proceedings of Fourteenth International Conference on Pattern Recognition*, pages 83–85, 1998.

[38] C. Colombo and A.D. Bimbo. Real-time haed tracking from the deformation of the eye contours using a piecewise affine camera. *Pattern Recognition Letters*, 20:721–730, 1999.

[39] R. Rae and H.J. Ritter. Recognition of human head orientation based on artificial neural networks. *IEEE Trans. on Neural Networks*, 9(2):257–265, 1998.

[40] G. Bozdagi et al. 3-D motion estimation and wireframe adaptation including photometric effects for model-based coding of facial image sequences. *IEEE Trans. on Circuits and Systems for Video Technology*, 4(3):246–256, 1994.

[41] A. N. Avanaki, B. Hamidzadeh, and F. Kossentini. Multi-objective retrieval of object pose from video. In *Proceedings of IEEE Conference on Tools with Artificial Intelligence*, pages 242–249, November 2000.

[42] M. Harville et al. 3d pose tracking with linear depth and brightness constraints. In *Proc. of IEEE International Conf. on Computer Vision*, pages 206–213, 1999.

[43] T. Hogg, H. Talhami, et al. Practical approach to view-based synergetic pose estimation. In *Proceedings of IEEE TENCON Conference*, volume 2, pages 663–666, 1997.

[44] V. Gudivada and V. Raghavan. Content-based image retrieval systems. *Computer*, 28(9), 1995.

[45] I. Cox, M. Miller, et al. Bayesian relevance feedback for image retrieval. In *Int. Conf. on Pattern Recognition*, pages 361–369, 1996.

[46] Y. Rui, T. Huang, et al. Content-based image retrieval with relevance feed-

back in mars. In *Proc. of IEEE International Conf. on Image Processing*, pages 815–818, 1997.

[47] http://www.ee.surrey.ac.uk/Research/VSSP/imagedb/demo.html, 2003. SQUID description.

[48] M. Nabil, J. Shepherd, et al. An image retrieval system for distributed environments. In *Workshop on Research Issues in Database Engineering*, pages 67–74, 1998.

[49] J.R. Smith and S.F. Chang. Integrated spatial and feature image query. *ACM Multimedia Systems*, 7(2):129–139, 1999.

[50] M. Flickner, H. Sawhney, W. Niblack, et al. Query by image and video content: the QBIC system. *Computer*, 28(9):23–32, 1995.

[51] http://wwwqbic.almaden.ibm.com, 2003. QBIC Home Page.

[52] J. Smith. Image retrieval evaluation. In *IEEE Workshop on Content-Based Access of Image and Video Libraries (CBAIVL '98)*, pages 112–113, 1998.

[53] A. Gupta and R. Jain. Visual information retrieval. *Comm. of the ACM*, 40(5), 1997.

[54] R. Ford, C. Robson, et al. Metrics for scene change detection in digital video sequence. In *Int. Conf. on Multimedia Computing and Systems (ICMCS '97)*, pages 610–611, 1997.

[55] S.F. Chang, W. Chen, et al. A fully automated content-based video search engine supporting spatiotemporal queries. *IEEE Trans. on Circuits and Systems for Video Technology*, 8(5):602–615, 1998.

[56] F. Bremond and M. Thonnat. Tracking multiple non-rigid object in video sequences. *IEEE Trans. on Circuits and Systems for Video Technology*, 8(5):585–591, 1998.

[57] F. Marques and J. Llach. Tracking of generic objects for video object generation. In *Proc. of IEEE International Conf. on Image Processing*, 1998.

[58] Mpeg home page. http://www.cselt.it/mpeg, 2000.

[59] A. N. Avanaki, B. Hamidzadeh, and F. Kossentini. Effects of sampling rate and pose error on volume reconstruction by space carving. In *Proceedings of IEEE Pacific-Rim Conference on Multimedia*, volume 2195 of *Lecture Notes in Computer Science*, pages 1048–1053. Springer, 2001.

[60] A. N. Avanaki, B. Hamidzadeh, and F. Kossentini. Multi-reference object pose retrieval with volume feedback. *ACM Multimedia Systems Journal*, 2004. Accepted for publication.

[61] A. N. Avanaki, R. Ward, B. Hamidzadeh, and F. Kossentini. Multi-reference object reconstruction and pose indexing using volume feedback. In *Proc. of IEEE International Symp. on Circuits and Systems*, 2004. Accepted for publication.

[62] A. N. Avanaki, B. Hamidzadeh, and F. Kossentini. Object reconstruction and pose indexing by volume feedback. In *Proc. of IEEE International Conf. on Image Processing*, volume 2, pages 13–16, 2003.

[63] A. N. Avanaki, B. Hamidzadeh, and F. Kossentini. Pose estimation and 3-d modeling from video by volume feedback. In *Proceedings of SPIE: Visual Communications and Image Processing*, pages 1131–1142, 2003.

[64] R. Koch. Dynamic 3-D Scene Analysis through synthesis feedback control. *IEEE Trans. on Pattern Recognition and Machine Intelligence*, 15(6):556–568, 1993.

[65] J. Foley et al. *Computer Graphics: Principles and Practice.* Addison-Wesley, New York, 1990.

[66] The MathWorks Inc. *Image processing toolbox user's guide*, pages 8–4–8–10. The MathWorks Inc., 1999. Distributed with Matlab release 11.

[67] R.C. Gonzalez and R.C. Woods. *Digital image processing*, chapter 8.4. Addison-Wesley, Reading, Mass., 1992.

[68] J.C. Lagarias, J.A. Reeds, et al. Convergence properties of the nelder–mead simplex method in low dimensions. *SIAM Journal on Optimization*, 9(1):112–147, 1998.

[69] D.E. Goldberg. *Genetic algorithms in search, optimization, and machine learning.* Addison-Wesley, Reading, Mass., 1989.

[70] S. C. Chan and H.-Y. Shum. A spectral analysis for light field rendering. In *Proc. of IEEE International Conf. on Image Processing*, volume 2, pages 25–28, September 2000.

[71] Z. Lin and H.-Y. Shum. On the number of samples needed in light field rendering with constant-d epth assumption. In *Proc. of Computer Vision and Pattern Recognition*, pages 588–595, June 2000.

[72] M. Steinmetz W. Niem. Camera viewpoint control for the automatic reconstruction of 3d objects. In *Proc. of IEEE International Conf. on Image Processing*, pages 655–658, 1996.

[73] Aldo Laurentini. The visual hull concept for silhouette-based image understanding. *IEEE Trans. on Pattern Recognition and Machine Intelligence*, 16(2):150–162, 1994.

[74] E. E. Hemayed and A. A. Farag. Object modeling using space carving. In *Proc. of IEEE International Conf. on Image Processing*, volume 2, pages 760–763, September 2000.

[75] Y. Aloimonos. Perpective approximations. *Image and Vision Computing*, 8(3):177–192, 1990.

[76] R. Bellman. *Perturbation techniques in mathematics, physics, and engineering*, chapter 1.22, pages 41–42. Holt, Rinehart and Winston, Inc., 1964.

[77] T. M. Apostol. *Calculus*, volume 2, chapter 2.8.18. Waltham, Mass., Blaisdell Pub. Co., 1967.

[78] R. Bellman. *Perturbation techniques in mathematics, physics, and engineering*, page 40. Holt, Rinehart and Winston, Inc., 1964.

[79] J. Barron and R. Klette. Quantitative color optical flow. In *Proc. of 16th international conf. on pattern recognition*, volume 4, pages 251–255, 2002.

[80] R. Horaud et al. Object pose: the link between weak perspective, paraperspective, and full perspective. *International Journal of Computer Vision*, 22(2):173–189, 1997.

[81] G. Eckert et al. Shape refinement for reconstructing 3d objects using an analysis-systhesis approach. In *Proc. of IEEE International Conf. on Image Processing*, pages 903–906, 2001.

A: Orthographic projection

Here, we describe how to project a point of material cube (Section 4.2) $a = (a_1, a_2, a_3)^T$ to a plane with normal $n = (n_1, n_2, n_3)^T$ to find out if the projection lies inside or outside of the silhouette from that pose. T denotes transposition.

The material cube is assumed to be centered on the origin of the coordinates $O = (0, 0, 0)^T$. For the point a and the normal n, consider the plane P passing through a with the normal n. We want to derive 2-D coordinates of a in this plane. Then, we can tell if a is projected within silhouette or not (Figure A.1).

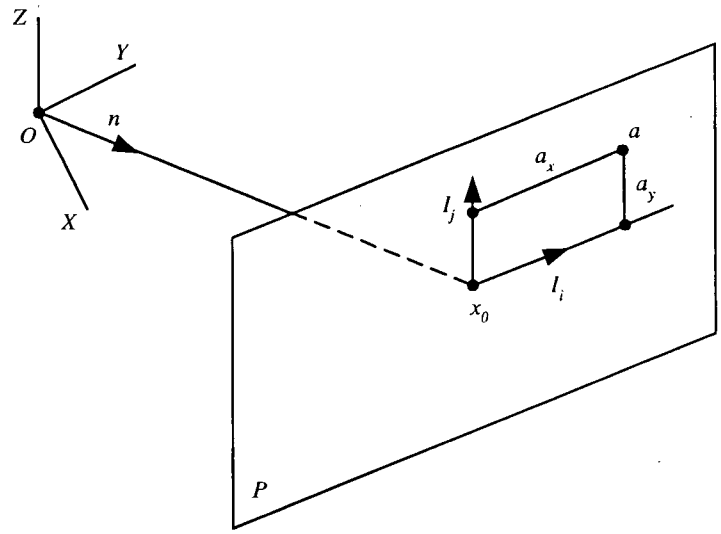


Figure A.1: Orthographic projection onto the silhouette plane

We define x_0, the center of view, as the point where a line emanating from O and perpendicular to the plane P, pierces that plane. x_0 will be regarded as

the origin of the 2-D coordinates in which silhouette information is given. We do so because in our method (described in Section 4.2), the cylindrical volumes are supposed to aim at O, the origin, as their middle point. x_0 must satisfy the following equations:

$$\begin{cases} n \cdot (x_0 - a) = 0 \\ x_0 \times n = O \end{cases}$$

where $\cdot$ and $\times$ denote dot and cross vector products respectively. The first equation states the fact that x_0 is on the plane P, while the second tells us that the direction of segment $\overline{Ox_0}$ and n are the same. Given that the length of n is 1, the solution to the system above is,

$$x_0 = (n \cdot a)\, n, \tag{A.1}$$

after simplification. The last step to find a 2-D coordinate system is to define unit vectors in the plane P. Assuming the images, from which silhouettes are extracted, are taken by a camera with 0^o of roll (i.e., the horizon line remains horizontal in the image), we define the horizontal unit vector, I_i, as the unit vector perpendicular to vectors n and $(0,0,1)^T$. And the vertical unit vector, I_j, is defined as the unit vector normal to I_i and n. Symbolically,

$$I_i = \frac{(0,0,1)^T \times n}{\|(0,0,1)^T \times n\|},$$

$$I_j = \frac{n \times I_i}{\|n \times I_i\|},$$

where $\|\cdot\|$ is vector length operator. Doing the math and simplification gives

(note that $\|n\| = 1$) the following.

$$I_i = \frac{1}{\sqrt{(1-n_3^2)}} \begin{pmatrix} -n_2 \\ n_1 \\ 0 \end{pmatrix},$$

$$I_j = \frac{1}{\sqrt{(1-n_3^2)}} \begin{pmatrix} -n_3 n_1 \\ -n_3 n_2 \\ 1-n_3^2 \end{pmatrix}. \tag{A.2}$$

When $|n_3| = 1$ (i.e., the normal vector and Z-axis are in the same direction), I_i and I_j become undefined. This situation occurs when silhouettes are from top and bottom views of the object. We define I_i and I_j for these situations as follows:

$$I_i = \begin{cases} (1,0,0)^T & \text{for} \quad n = (0,0,1)^T \\ (-1,0,0)^T & \text{for} \quad n = (0,0,-1)^T \end{cases} \tag{A.3}$$
$$I_j = (0,1,0)^T$$

The above definition ensures that $(I_i \times I_j) \parallel n$ (i.e., (I_i, I_j, n) make a basis for a right-handed coordinate system). Our choice makes definition of I_i and I_j, in special cases, compatible to the regular derivation discussed above.

Now its fairly simple to find projection of a onto the plane P, in which silhouette data are given:

$$a_x = (a - x_0) \cdot I_i,$$

$$a_y = (a - x_0) \cdot I_j,$$

where a_x and a_y are X and Y coordinates of a point a of the material cube projected orthographically onto the plane P (with normal n), in which silhouette data is given (Figure A.1). I_i and I_j are unit vectors in the plane P, which are given by Equations (A.3) and (A.2). x_0 is the middle point of the silhouette frame and is calculated by Equation (A.1).

B: Translation compensation in the RPE block by optic flow

The following lines are added to the RPE block to compute a single value optic flow (function `svOF(.)`), and to move the 3-D reference patch accordingly. All codes are in Matlab.

```
transComp=svOF(double(iRef),iTarget)
camdolly(-transComp(2), transComp(1), 0, 'movetarget', 'pixels')
```

The first method described in [79] is implemented in the following function. `Rx`, `Ry`, and `Rt` denote $\frac{\partial R}{\partial x}$, $\frac{\partial R}{\partial y}$, and $\frac{\partial R}{\partial t}$. `R`, `G`, and `B` are the colour components of the RGB colouring scheme. Spatial partial derivatives (such as `Rx`) are computed for `i0`, the "average" image of `iRef` and `iTarget`. All partial derivatives are computed for the centres of all four-pixel neighborhoods (i.e., four adjacent pixels).

```
function u=svOF_(iRef,iTarget,i0)
% Single Value Optic Flow for foreground pixels, where
% Rt+Bt+Gt are non-zero.
% This routine gives a correct answer of flows up to 1 pixel.
% Use svOF() for larger flows (translations).

if nargin==2, i0=(iRef+iTarget)/2; end
```

```
Dt=iTarget-iRef;
Rt=Dt(:,:,1);
Gt=Dt(:,:,2);
Bt=Dt(:,:,3);

Rt=filter2([.25 .25; .25 .25],Rt,'valid');
Gt=filter2([.25 .25; .25 .25],Gt,'valid');
Bt=filter2([.25 .25; .25 .25],Bt,'valid');

Rx=filter2([-.5 .5;-.5 .5],i0(:,:,1),'valid');
Gx=filter2([-.5 .5;-.5 .5],i0(:,:,2),'valid');
Bx=filter2([-.5 .5;-.5 .5],i0(:,:,3),'valid');
Ry=filter2([-.5 -.5;.5 .5],i0(:,:,1),'valid');
Gy=filter2([-.5 -.5;.5 .5],i0(:,:,2),'valid');
By=filter2([-.5 -.5;.5 .5],i0(:,:,3),'valid');

msk=(Rt~=0)|(Gt~=0)|(Bt~=0);
if ~any(msk(:)), u=[0;0]; return, end
Rx=Rx(msk);Ry=Ry(msk);Rt=Rt(msk);
Gx=Gx(msk);Gy=Gy(msk);Gt=Gt(msk);
Bx=Bx(msk);By=By(msk);Bt=Bt(msk);

A=[Rx Ry;Gx Gy;Bx By];
b=-[Rt;Gt;Bt];
u=inv(A'*A)*A'*b;
```

The function above, `svOF_(.)`, fails to compute an optic flow (translation) larger than one pixel. The following function, `svOF(.)`, calls `svOF_(.)` to recover the translation recursively, one pixel at a time.

```
function uv=svOF(iRef,iTarget)
% single value OF for foreground pixels, where
% Rt+Bt+Gt are non-zero.
% Each +1 in uv(1)/uv(2) means a down/right iRef shift
% is required to match iTarget.

for K=-1:1,for L=-1:1,
  nu=svOF_(circshift(iRef,[K,L]),iTarget);
  u(K+2,L+2)=nu(1);
  v(K+2,L+2)=nu(2);
end, end
n=u.^2+v.^2;
[K,L]=find(n==min(n(:)));
uv=[K(1),L(1)]-2;
if all(uv==[0 0]), return, end
uv=uv+svOF(circshift(iRef,uv),iTarget);
```