Univerza v Ljubljani
Fakulteta za računalništvo in informatiko

Jaka Krivic

Segmentacija in 3D sledenje objektom na osnovi superkvadričnih modelov

Doktorska disertacija

Ljubljana, 2006

Mentor: prof. dr. Franc Solina

University of Ljubljana
Faculty of Computer and Information Science

Jaka Krivic

Segmentation and 3d tracking of superquadric modeled objects

A Dissertation in Computer and Information Science

Ljubljana, 2006

Supervisor: prof. dr. Franc Solina

To my three pearls.

Acknowledgements

First, I owe a great debt of thanks to prof. dr. Franc Solina. Besides the invaluable scientific advice and ingeniously resourceful problem solving, he provided me with motivation and moral support throughout my career, contributing greatly to my studies and this thesis.

I would like to thank my friends and colleagues at the CV and VICOS laboratories, that have helped in so many ways.

I also thank Doc. Aleš Jaklič, and Prof. Robert Sablatnig for serving as dissertation committee members.

I will remain in debt to my parents, and the whole familia, for all their support, coming in all kinds of forms, from a nice word to a superb cake.

Abstract

Segmentation and 3D tracking of superquadric modeled objects

In this thesis methods needed to model 3D objects, segment them from 3D data, and track them throughout image sequences are studied. First, the detection of articulated 3D objects is investigated. The envisioned system accepts range images at the input. Image segmentation is then performed to acquire superquadric descriptions of the scene. Also the objects are modeled with part level models described by superquadrics. Parts of an object form a structure, that distinguishes it from other objects. In order to exploit the structural difference between objects, we propose a method based on interpretation trees, which compares scene and model part by part giving object hypotheses. Various types of part matching constraints are introduced that compare scene parts to model parts in order to reduce the search for object instances. Also, a verification procedure is proposed that verifies that hypothesized scene parts really represent the object in question. This procedure also determines object position and part configuration, at least to some extend.

Next, the object detection is introduced to the problem of 3D object tracking initialization. It provides the system with the initial object position and configuration, which are then further improved by fitting the part models directly to 3D data. The tracking phase takes advantage of the information about the object's position from previous frames to acquire the object's position efficiently.

Keywords: *superquadrics; part-level object modeling; range images; object recognition; 3D object tracking*

Povzetek

Segmentacija in 3D sledenje objektom na osnovi superkvadričnih modelov

Doktorska disertacija obravnava modeliranje 3D objektov, njihovo segmentiranje iz 3D podatkov, kakor tudi njihovo sledenje na sekvencah slik. Najprej preučimo zaznavanje strukturiranih 3D objektov, kjer imamo na vhodu sistema globinske slike. Le-te najprej segmentiramo, da dobimo superkvadrične opise scene s slike. Hkrati tudi modeliramo objekte, ki jih želimo zaznavati na slikah. Modeli so zgrajeni iz delov, ki so modelirani s superkvadriki in predstavljajo strukturo objekta. Objekti se med sabo ločijo predvsem po različni strukturi. Da bi v čim večji meri uporabili strukturno razliko med objekti, predlagamo metodo za razpoznavanje, ki temelji na interpretacijskih drevesih, in katera s pomočjo primerjave delov scene z deli objekta generira hipoteze. Predlagamo različne vrste primerjav med superkvadričnimi deli, ki pomagajo zožiti iskanje dobrih hipotez. Predlagamo tudi metodo za preverjanje hipotez o instancah objekta, ki hkrati vsaj do neke mere poda tudi morebiten položaj in konfiguracijo objekta.

Nadalje uporabimo zaznavanje objektov pri problemu inicializacije 3D sledenja. Ker predstavljena metoda poda grob položaj in konfiguracijo objekta, predlagamo dodaten korak, ki ju izboljša. Korak izboljšave temelji na prilagajanju superkvadrikov, ki opisujejo objekt direktno na regije 3D točk v določeni bližini do dela. Predstavimo še fazo sledenja, kjer se informacija o stanju objekta na zadnji sliki sekvence uporabi za učinkovito določanje položaja in konfiguracije na novi sliki.

Ključne besede: *superkvadriki; modeliranje po delih; globinske slike; razpoznavanje objektov; 3D sledenje objektom*

Contents

List of Tables

List of Figures

Part I

Thesis

Chapter 1

Introduction

In computer vision many different models have been used for describing various aspects of objects and scenes. Part-level models are one way of representing 3D objects, when particular entities that they describe, correspond to perceptual equivalents of parts. Therefore, several part-level shape models are required to represent an articulated object. Such descriptions are suitable for path planning or manipulation, but they are sometimes not exhaustive enough to represent all the necessary details needed in object recognition.

To obtain part-level description of a scene the image has to be partitioned into segments corresponding to individual parts, and a part model for each of these segments has to be recovered. If the two tasks are separated, segmentation does not take into account the shapes that part models can adopt. To avoid this problem, segmentation and recovery can be combined, so that images can only be segmented into parts which are instances of selected part models. To achieve concurrent segmentation and shape recovery, the *recover-and-select* paradigm can be used (Leonardis et al. (1995)).

One of the more popular types of volumetric models are superquadrics (Jaklič et al. (2000)). These are volumetric models that represent standard geometrical solids as well as shapes in between and are defined by only 11 parameters.

1.1 Problem Statement

In this thesis methods needed to model 3D objects, segment them from 3D data, and track them throughout image sequences are studied. First, recognition of structured 3D objects is investigated. Parts of an object form a structure, that distinguishes it from other objects. For the task of recognition of such objects, the relations between parts, the object's structure, are therefore even more important than the shape of the parts itself. Second, tracking objects in 3D space is studied. The process of tracking an object in 3D space can be divided into two phases. In the initialization phase the object's presence is determined and (if present) its 3D position and part configuration initialized. The tracking phase takes advantage of the information about the object's position from previous frames to acquire the object's position easily.

1.2 Related Work

This thesis relates to research in 3D data modeling and segmentation as well as 3D object tracking. Related work is briefly discussed in the following subsections.

1.2.1 Modeling and Segmentation of 3D Data

Pentland (1986) was the first who used superquadrics in the context of computer vision. The method of Solina and Bajcsy (1990) for recovery of superquadrics from pre-segmented range images, however, became more widespread (Jaklič et al. (2000)).

Several methods for segmentation with superquadrics have been developed. A tight integration of segmentation and model recovery was achieved by Leonardis et al. (1997) by combining the *recover-and-select* paradigm (Leonardis et al. (1995); Leonardis (1996)) with the superquadric recovery method of Solina and Bajcsy (1990). The paradigm works by independently recovering superquadric part models everywhere on the image, and selecting a subset which gives a compact description of the underlying data. SEGMENTOR is an object-based implementation of the *recover-and-select* segmentation paradigm using superquadrics and other parametric models (Jaklič (1997)).

The applicability of the SEGMENTOR has been explored in several contexts, for example for reverse engineering (Solina et al. (1998)). Segmentation and shape modeling of smooth and regular man-made objects with SEGMENTOR is fairly stable, if the objects can be easily represented with superquadric shapes. Segmentation of rough, natural shapes which are not very close to ideal superquadric shapes is less reliable. The superquadric models can not expand as easily on rough surfaces and complex shapes as on smooth regular objects, which results generally in over-segmentation. Automatic adaptation of the granularity of models to the scale/roughness of the scene is in the context of superquadrics still unresolved (Solina and Leonardis (1998)). Despite of those deficits we decided to test the applicability of the SEGMENTOR for object recognition of articulated objects in complex scenes.

The main goal of this thesis was how the results of SEGMENTOR can be used for object recognition and subsequent object tracking. The aim of this thesis was to investigate the possible use of part-level descriptions obtained by the SEGMENTOR for recognition of articulated objects. We hypothesized that the configuration of parts and their rough shape should provide enough constraints for successful matching with the models of known objects. The recognition system would search for matches between scene and model parts, a procedure known as *model based matching*. The object hypotheses can be subsequently verified by fitting the object model directly to the range data. Fig. 1.1 depicts the object detection scheme proposed in this thesis (see also Krivic and Solina (2004)). Taking the segmented range image (i.e. superquadric models reconstructed by SEGMENTOR and their respective 3D point regions) and the object model at the input (top row), the system would output the objects found, and their positions and part configurations (bottom row). Such recognized objects could be further used for higher level reasoning, such as developed by Chella et al. (2000). As means for scene understanding they used the notion of conceptual space, to link between subconceptual information (in the form of superquadrics) and symbolically organized knowledge.

Superquadrics have been used in several computer vision systems. Raja and Jain (1992) tried to relate superquadrics and geons, part primitives introduced by Biederman (1985).

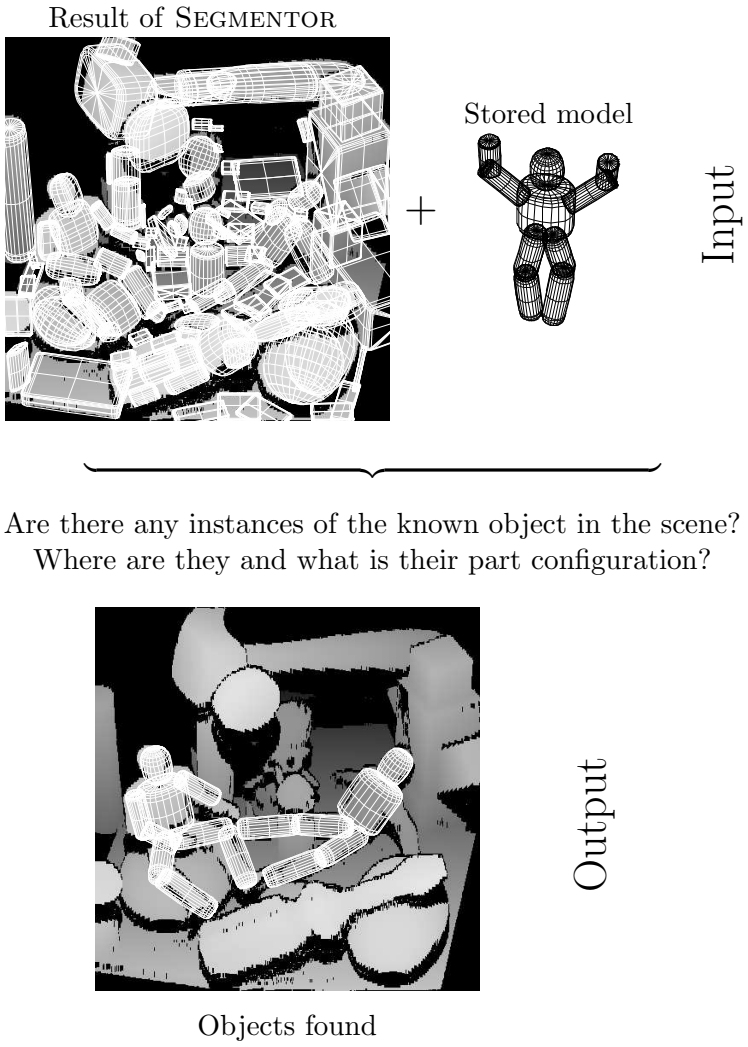


Fig. 1.1: Overview of the object recognition system proposed in this thesis (see also Krivic and Solina (2004)). The system takes segmented range image produced by SEGMENTOR along with the object model to detect objects presence and position.

They investigated recognition of geons from superquadrics fitted to range data, but did not deal with object made of those parts. Dickinson et al. (1992) used superquadrics as modeling primitives to construct objects. The recognition is based on aspects, which are used to model the superquadric parts. Aspects are recovered from an image, and aspect hierarchy is used to infer a set of volumetric primitives and their connectivity relations. The verification of object hypothesis is then basically the topological verification of the recovered graph.

Since superquadrics are part-level descriptions, an object recognition system that searches for matches between parts in the scene and parts of the modeled object can be used (Dickinson et al. (1992)). One of the first such methods by Nevatia and Binford (1977) uses a relational graph structure to represent an object. The recognition then be-

comes a matter of matching two graphs. The $3DPO$ vision system developed by Bolles and Horaud (1986) uses a *local feature focus* method for constraining the size of the solution search space. Kim and Kak (1991) used bipartite matching for fast rejection of inapplicable models, and a combination of bipartite matching and discrete relaxation to prune the possible object hypotheses. Grimson (1990) developed the *interpretation tree* method. He arranged all possible matches of a scene part with a model part in a tree structure—an interpretation tree. The problem of recognition is to find consistent interpretations without exploring all possible ways of matching the scene and model parts, which was done using geometric constraints.

1.2.2 Initialization of 3D Object Tracking

Pose reconstruction of articulated objects as proposed by Taylor (2000) (or some variation of it) can be found in most approaches to object tracking that use semi-automatic initialization. Based on the locations of joints in a single uncalibrated image, the pose of an articulated object can be determined. Because a stick model is used (only joints matter, parts themselves are not modeled), the pose can only be expressed as a function of a scaling factor. With additional ad-hoc constraints also expressed as functions of the scaling factor, and minimization, the scaling factor can be determined. For the approach to become an automatic initialization procedure, it first and foremost lacks the estimation of object's joint locations in the image. The second deficiency is the ad-hoc approach to scaling factor computation. A similar approach is the one from Barron and Kakadiaris (2000), where learned statistics of anthropometric properties of human body are used for human pose reconstruction.

In the approach of Rosales et al. (2001b) image preprocessing with the SMA system (*Specialized Mapping Architecture*) is used. The SMA system extracts possible hypotheses for joint positions in images from multiple views. SMA is a supervised machine learning architecture (Rosales et al. (2001a)), that is used to transform input image features to (hypothetical) joint locations in 2D planes of the so called virtual cameras. From the resulting set of hypotheses, the object pose is reconstructed by a derivation of the Expectation Maximization algorithm. The results indicate that 3 or more views should be used for a simple human model. Another drawback of the system is also the hypothesis generation, which is based on the background removal.

1.2.3 3D Object Tracking

Most of the work done in the field of articulated object tracking deals with tracking human figures or present the results on those. Gavrila and Davis (1996) use a superquadric model of human body with 22 degrees of freedom. Their approach uses a generate-and-test strategy: based on previous states/poses the valid model parameter intervals are first determined, followed by a minimization of distances between model and image edges in a discretized hierarchical state space. The input to the system are four synchronized sequences from different views. Initialization begins by background removal and determination of a region of interest by a threshold operator. The main 3D axis of the body is computed by PCA (Principal Component Analysis) from the main axes of the regions of interest of at least two views. It is worth noting that the initialization procedure is

limited to a human body model, which has to be in an upright, stretched pose, with no overlapping parts and a steady background.

Bregler and Malik (1998) parameterized a kinematic chain of articulated objects by exponential maps and twist motions, as used in robot control. They orthographically project their 3D articulated model, each rigid part's projection is an ellipse with a support map that holds the probability that the points in the ellipse belong to the tracked object. The approach poses the motion problem as linear estimation problem using optical flow information. The initialization is semi-automatic in the sense that an operator has to label the joint locations in the sequences' first images. From this input the model parameters are computed using various minimization techniques. The approach is a multiview-one and does not deal with model to image ambiguities and multimodality.

In the approach of Delamarre and Faugeras (1999), objects are modeled by spheres, truncated cones and parallelepipeds. Input to the system are synchronized sequences from two or more cameras. First, object contours are detected by a powerful method of active geodesic contours. Then in an iterative process of solving dynamical equations of 3D model movement, forces are applied between each model part and the detected contours. The process is stopped upon convergence, i.e. when the 3D pose of the object is (almost) stable in relation to the detected contours. An initialization procedure is not presented. The results of tracking a human figure from three cameras in a controlled environment are quite good, but the method relies on the quality of the object contour detection, which is not always possible.

Wachter and Nagel (1999) use a kinematic model with truncated cones as building blocks. Object localization is based on prediction by extended Kalman filtering with a simple constant velocity model for every degree of freedom, and state estimation using optical flow and image edge information. Results are presented on the case of a human figure with 10 to 15 degrees of freedom. In an uncontrolled real environment using monocular sequences the results are good, nevertheless the movement is parallel to the image plane and therefore the depth ambiguities are reduced to the minimum.

Deutscher et al. (2000) introduced simulated annealing and a crossover operator to particle filtering methods (or CONDENSATION by Isard and Blake (1998)), thus improving robustness and speed. The weighting function of annealed particle filter consists of two types of image features, gradient image edges and the object silhouette. Truncated cones are used for object modeling. Although the presented 29 degrees of freedom human figure tracking results are satisfactory, the test sequences are captured from three different views in a dark background, thus eliminating clutter and depth ambiguities. The initialization process is not presented. In a similar approach by Sidenbladh et al. (2000), optical flow is used instead of edges and silhouettes. In addition, learned temporal models of movement are applied for dealing with depth ambiguities and overlapping.

Drummond and Cipolla (2001) model objects with quadrics connected by a kinematic chain. The approach is based on an algorithm which efficiently propagates statistics of probability distributions through the kinematic chain to obtain maximum a posteriori estimates of the object motion. Statistics defines the probability, that near the sampled points on the model contour there is an edge in the image, that is supposedly the object contour. The authors do not deal with the initialization process nor with the recovery from mistracking. For more complex objects such as a human upper-body, the system is fed with sequences from 3 different views to achieve decent results.

The approach of Plaenkers and Fua (2002) uses special articulated models, which are based on implicit surface formulation (Plankers and Fua (2001)). The system is fed two sequences of images from a stereo camera pair. By using stereo disparity maps to compute clouds of 3D points, and using objects pose, the object silhouette is extracted more accurately. State estimation is achieved by gradient minimization of distances between 3D point clouds and model surface, and between extracted silhouette and model contour. The strength of the approach lies in the ability to analytically and precisely compute all the gradients used, what originates from the type of models used. On the other hand the 3D information from a stereo pair is used, and the approach does not deal with occlusion, which occurs in real footage of articulated objects. Again, an initialization process is not proposed.

Zhang and Kambhamettu (2002) use a single extended superquadric for modeling and tracking a human head in 3D from a monocular sequence. Rigid 3D head movement is extracted by using optical flow. For eliminating errors originating in noisy optical flow estimation and the lack of 3D information, a further post regularization stage using edge flow is applied. The proposed system robustly tracks a human head in the presence of clutter and occlusion. The authors do not indicate how the approach could be used in tracking articulated objects.

Sminchisescu and Triggs (2003) model the object surface with superquadrics. These are not used directly, but are discretized as parameterized meshes, which are transformed into 3D points through the model kinematic chain. Initialization is semi-automatic in the sense that the operator labels joint locations in the first frame. The system fits the model to the input by using nontrivial optimization. Tracking itself is based on a hybrid search algorithm by combining gradient optimization and sampling in the search space around local minima scaled by covariance, or *covariance scaled sampling*.

In the approach of Sigal et al. (2004) the object model is a collection of loosely connected parts. The system is based on the assertion that it is easier to extract positions and movements of each of the model parts as it is to extract the pose of the whole object. For individual parts appearance based detectors using PCA also permit automatic initialization and recovery from mistracking. Tracking is based on a variant of particle filtering, which exploits the general cyclic graph, that describes temporal and spatial part connections. Key strength of the approach is in the fact that it does not distinguish between initialization and tracking, because on every image candidate parts are detected. On the other hand, the approach is based on appearance based part detectors, and therefore the tracked object's appearance has to be known in advance. The results are presented on the case of human figure tracking with footage from four different views, where the system works quite well.

Most of the methods do not deal with initialization at all, or aim just at simplifying user interaction (e.g. Taylor (2000)). The few that do, either use an ad-hoc approach (e.g. Gavrila and Davis (1996)), or use some offline steps (such as learning appearance in Sigal et al. (2004)). The main contribution of this thesis is therefore a tracking initialization step, that needs no user interaction, and only uses 3D object model, and can manage many objects of the same classes without any additional processing. Also, no methods that would be using superquadrics directly in the process exist, this thesis investigates a direct use of superquadric models in object modeling as well as the scene processing and tracking.

1.3 Thesis Overview

The rest of this thesis is composed as follows.

Chapter 2 first introduces superquadrics as mathematical solids, listing their geometrical properties and relationships important for this thesis. Homogeneous transformations and Euler rotation angle notation are also presented in order to describe a superquadric in general position and orientation in space. Next, range image segmentation is presented, using an implementation of the *recover-and-select* paradigm in the SEGMENTOR system. The contents of this chapter is based on the work by Solina and Bajcsy (1990), Leonardis et al. (1995) and Jaklič (1997), and is presented here for introductional purposes only.

A novel object recognition scheme using articulated superquadric built object models is proposed in Chapter 3. The proposed scheme is a model based matching technique based on interpretation trees. The chapter proposes various part match constraints for reducing the interpretation search, as well as an interpretation verification procedure. At the end of the chapter the experiments verifying the proposed scheme's effectiveness are presented and discussed.

Following in Chapter 4 is the proposed application of object recognition scheme to the problem of object tracking initialization. The proposed initialization uses interpretation tree search to detect object instances in the scene along with another step of improving the quality of object position by fitting the superquadric part models directly to the underlying 3D data. The chapter ends with experimental results of the initialization procedure.

Chapter 5 explores the possibility to use the part fitting step similar to the initialization position improvement for frame to frame object pose estimation and tracking. Experimental results demonstrate the performance of the proposed method.

Finally in Chapter 6, the contributions of the thesis are summarized and future work is discussed.

Chapter 2

Superquadrics and 3D Data Segmentation

Superquadrics are a family of volumetric models, which were first introduced in computer graphics by Barr (1981) and later gained popularity in computer vision (Pentland (1986); Solina and Bajcsy (1990); Raja and Jain (1992); Dickinson et al. (1992); Leonardis et al. (1997); Jaklič et al. (2000)). This chapter summarizes the properties of superquadric important for this thesis, as well as segmentation of superquadrics from 3D point sets (e.g. range images).

2.1 Superquadrics

Basic superquadric shapes are compact representation of 3D shapes as they are described by only 11 parameters $\Lambda = \langle a_1, a_2, a_3$ [size], ϵ_1, ϵ_2 [shape], ϕ, θ, ψ [rotation], p_x, p_y, p_z [translation] $\rangle$.

2.1.1 Superquadric Surface

The surface of a superquadric in local coordinate frame is defined by

$$\mathbf{s}(\eta,\omega) = \begin{bmatrix} s_x \\ s_y \\ s_z \end{bmatrix} = \begin{bmatrix} a_1 \cos^{\epsilon_1}\eta \cos^{\epsilon_2}\omega \\ a_2 \cos^{\epsilon_1}\eta \sin^{\epsilon_2}\omega \\ a_3 \sin^{\epsilon_1}\eta \end{bmatrix}, \qquad \begin{array}{ccccc} -\frac{\pi}{2} & \leq & \eta & \leq & \frac{\pi}{2} \\ -\pi & \leq & \omega & < & \pi \end{array} \tag{2.1}$$

The *surface vector* $\mathbf{s}$ originates in the center of the coordinate system and defines the surface of a superquadric. The parameter ω is the angle between the x axis and the projection of $\mathbf{s}$ to the x-y plane (latitude angle), whereas the parameter η is the angle between $\mathbf{s}$ and x-y plane (longitude angle). The parameters a_1, a_2, a_3 determine the size of superquadric in the direction of x, y in z axes, respectively. The parameters ϵ_1 and ϵ_2 determine the shape of the superquadric in the plane containing z axis and a plane parallel to x-y plane, respectively. Fig. 2.1 shows some superquadric shapes. By varying the ϵ_1 and ϵ_2 parameters and with the size parameters a_1, a_2, a_3 being equal and fixed, shapes from a cube to a cylinder, a sphere and a prism can be achieved.

The above notation of the surface vector is somewhat simplified (basic notation of Barr (1981)). Since a result of rising negative real number to an arbitrary positive real number

is generally a complex number, a new exponent function is defined as

$$x^y = \text{sign}(x)\,|x|^{y*} \tag{2.2}$$

where x^{y*} is an ordinary exponent function. This form of exponent function is implicitly assumed where necessary throughout the thesis.

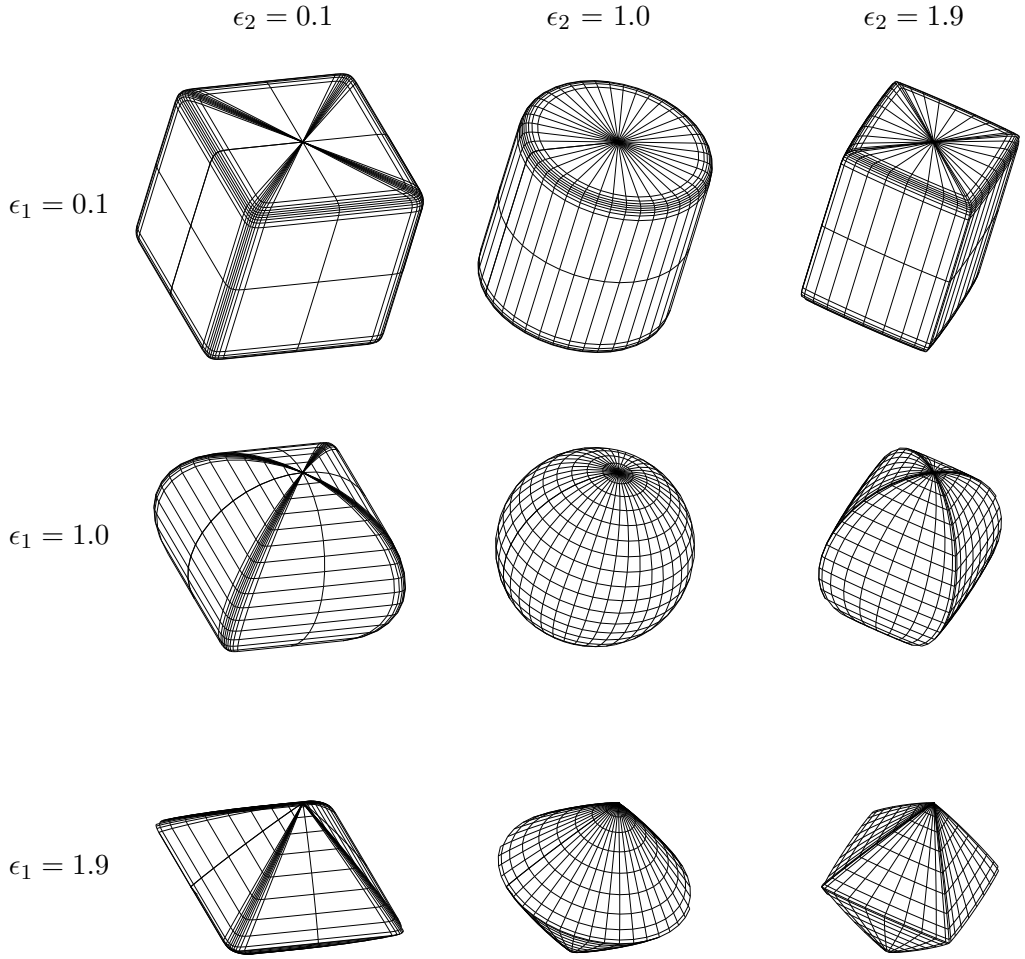


Fig. 2.1: Superquadric shape as a function of parameters ϵ_1 and ϵ_2 with size parameters being constant and equal.

2.1.2 Implicit Superquadric Equation

The surface of a superquadric can also be defined as a set of solutions of an implicit superquadric equation. The equation can be derived from Eq. 2.1 by eliminating parameters ω and η

$$F(x,y,z) = \left(\left(\frac{x}{a_1}\right)^{\frac{2}{\epsilon_2}} + \left(\frac{y}{a_2}\right)^{\frac{2}{\epsilon_2}} \right)^{\frac{\epsilon_2}{\epsilon_1}} + \left(\frac{z}{a_3}\right)^{\frac{2}{\epsilon_1}}, \tag{2.3}$$

with superquadric surface points satisfying the equation $F(x, y, z) = 1$.

2.1.3 Distance Between a Point and a Superquadric

The function $F(x, y, z)$ can also be used for computing the radial Euclidean distance between a point and a superquadric by finding a scalar β that scales the point's vector $\mathbf{p_0} = (x_0, y_0, z_0)$ so that the scaled vector $\mathbf{p_s} = \beta\mathbf{p_0}$ lies on the surface of the superquadric. Thus

$$F(\beta x_0, \beta y_0, \beta z_0) = 1 \tag{2.4}$$

$$F(x_0, y_0, z_0) = \beta^{-\frac{2}{\epsilon_1}} \tag{2.5}$$

leading to the radial Euclidean distance

$$d = |\mathbf{p_0} - \mathbf{p_s}| = |\mathbf{p_0} - \beta\mathbf{p_0}| = |\mathbf{p_0}||1 - F^{-\frac{\epsilon_1}{2}}(x_0, y_0, z_0)| = |\mathbf{p_s}||F^{\frac{\epsilon_1}{2}}(x_0, y_0, z_0) - 1|. \tag{2.6}$$

Note that the function $F(x, y, z)$ divides the space in a local coordinate system of a superquadric into three parts

$$F(x,y,z)\begin{cases} < 1, & \text{point } (x,y,z) \text{ lies inside} \\ = 1, & \text{point } (x,y,z) \text{ lies on the surface} \\ > 1, & \text{point } (x,y,z) \text{ lies outside} \end{cases} \tag{2.7}$$

Function $F(x, y, z)$ is also called the *inside-outside function*.

2.1.4 Normals

Normal vectors play an important role with 3D geometrical solids (e.g. determining visible part of the solid). In superquadrics, normal vector can be calculated as any other normal vector – by calculating vector product of two vectors tangential to the surface $\mathbf{n} = \frac{\partial \mathbf{s}}{\partial \eta} \times \frac{\partial \mathbf{s}}{\partial \omega}$. The *normal vector* of a superquadric therefore is

$$\mathbf{n}(\omega, \eta) = \begin{bmatrix} n_x \\ n_y \\ n_z \end{bmatrix} = \begin{bmatrix} \frac{1}{a_1} \cos^{2-\varepsilon_1}\eta \cos^{2-\varepsilon_2}\omega \\ \frac{1}{a_2} \cos^{2-\varepsilon_1}\eta \sin^{2-\varepsilon_2}\omega \\ \frac{1}{a_3} \sin^{2-\varepsilon_1}\eta \end{bmatrix}. \tag{2.8}$$

2.1.5 Superquadric in an Arbitrary Position

So far only the properties of superquadrics in their canonical coordinate system were presented. To represent superquadric in a certain global coordinate system, a homogeneous rigid transformation can be used. This can be decomposed into rotation and translation. The global coordinate system is transformed into a local coordinate system by first rotating and then translating it. This can be expressed as a relationship between the homogeneous vectors to the same point in space as

$$\mathbf{p_{global}} = \mathbf{T_{rotation}}\mathbf{p_{canonical}} + \mathbf{p_{translation}} \tag{2.9}$$

Rotation can be parameterized by only three independent parameters. Using ZYZ-Euler rotation, which is used in this thesis, angles ϕ, θ and ψ signify the rotation around the z

axis, rotation around the new y axis, and the rotation around the new z axis, respectively. The rotation matrix therefore is

$$\mathbf{T_{rotation}} = \begin{bmatrix} \cos\phi\cos\theta\cos\psi - \sin\phi\sin\psi & -\sin\phi\cos\theta\cos\psi - \cos\phi\sin\psi & \sin\theta\cos\psi \\ \cos\phi\cos\theta\sin\psi + \sin\phi\cos\psi & -\sin\phi\cos\theta\sin\psi + \cos\phi\cos\psi & \sin\theta\sin\psi \\ -\cos\phi\sin\theta & \sin\phi\sin\theta & \cos\theta \end{bmatrix}. \tag{2.10}$$

To summarize, a superquadric in arbitrary position is defined by shape, rotation and translation

$$\Lambda = \langle a_1, a_2, a_3, \epsilon_1, \epsilon_2, \phi, \theta, \psi, p_x, p_y, p_z \rangle. \tag{2.11}$$

2.1.6 Volume and Moments of Inertia

Other important properties of a superquadric are its volume and moments of inertia. The volume can be derived by integrating the area of intersection of a plane parallel to the $x-y$ plane over the z axis (Jaklič et al. (2000)). The equation for volume is

$$V = 2a_1a_2a_3\epsilon_1\epsilon_2\, B\left(\frac{\epsilon_1}{2}, \epsilon_1 + 1\right) B\left(\frac{\epsilon_2}{2}, \frac{\epsilon_2 + 2}{2}\right), \tag{2.12}$$

or alternatively by some algebraic manipulation

$$V = 2a_1a_2a_3\epsilon_1\epsilon_2\, B\left(\frac{\epsilon_1}{2} + 1, \epsilon_1\right) B\left(\frac{\epsilon_2}{2}, \frac{\epsilon_2}{2}\right), \tag{2.13}$$

where the term $B(x, y)$ is a beta function.

A similar procedure leads to derivation of the moments of inertia (Jaklič et al. (2000)). Following are general equations for the moments of inertia.

$$I_{xx} = \frac{1}{2}a_1a_2a_3\epsilon_1\epsilon_2(a_2^2 B(\frac{3}{2}\epsilon_2, \frac{1}{2}\epsilon_2)B(\frac{1}{2}\epsilon_1, 2\epsilon_1 + 1) + 4a_3^2 B(\frac{1}{2}\epsilon_2, \frac{1}{2}\epsilon_2 + 1)B(\frac{3}{2}\epsilon_1, \epsilon_1 + 1)), \tag{2.14}$$

$$I_{yy} = \frac{1}{2}a_1a_2a_3\epsilon_1\epsilon_2(a_1^2 B(\frac{3}{2}\epsilon_2, \frac{1}{2}\epsilon_2)B(\frac{1}{2}\epsilon_1, 2\epsilon_1 + 1) + 4a_3^2 B(\frac{1}{2}\epsilon_2, \frac{1}{2}\epsilon_2 + 1)B(\frac{3}{2}\epsilon_1, \epsilon_1 + 1)), \tag{2.15}$$

$$I_{zz} = \frac{1}{2}a_1a_2a_3\epsilon_1\epsilon_2(a_1^2 + a_2^2)B(\frac{3}{2}\epsilon_2, \frac{1}{2}\epsilon_2)B(\frac{1}{2}\epsilon_1, 2\epsilon_1 + 1). \tag{2.16}$$

Inertial moments for some special cases of superquadrics like a sphere, an ellipsoid and a cube are listed in Tab. 2.1

2.2 Superquadric Recovery with Segmentor

This section describes Segmentor (Jaklič (1997); Jaklič et al. (2000)) for range image segmentation and superquadric recovery. Segmentor uses the *recover-and-select* paradigm (Leonardis (1996)) in the segmentation process. As the name states, the main components of the paradigm are model recovery, and selection of models that describe the 3D data best. Parametric model recovery is difficult because two problems have to be

Tab. 2.1: Inertial moments for special cases of superquadrics (from Jaklič et al. (2000)).

	Sphere	*Ellipsoid*	*Cube*
	$\epsilon_1 = 1, \epsilon_2 = 1$	$\epsilon_1 = 1, \epsilon_2 = 1$	$\epsilon_1 = 0, \epsilon_2 = 0$
I_{xx}	$\frac{8\pi}{15}r^5$	$\frac{4\pi}{15}abc(b^2+c^2)$	$\frac{1}{12}abc(b^2+c^2)$
I_{yy}	$\frac{8\pi}{15}r^5$	$\frac{4\pi}{15}abc(a^2+c^2)$	$\frac{1}{12}abc(a^2+c^2)$
I_{zz}	$\frac{8\pi}{15}r^5$	$\frac{4\pi}{15}abc(a^2+b^2)$	$\frac{1}{12}abc(a^2+b^2)$

solved. First, a set of points belonging to a model has to be established, thus segmenting the data, and second, model parameters have to be determined.

Superquadric model parameters

$$\Lambda = (a_1, a_2, a_3, \epsilon_1, \epsilon_2, \phi, \theta, \psi, p_x, p_y, p_z) \tag{2.17}$$

can be determined from a set of 3D points by minimization of the error function

$$G(\Lambda) = a_1 a_2 a_3 \sum_{i=1}^{N} (F^{\epsilon_1}(x_i, y_i, z_i) - 1)^2, \tag{2.18}$$

where $(x_i, y_i, z_i), i = 1...N$ are 3D points in a canonical coordinate frame of the superquadric. The method was developed by Solina and Bajcsy (1990) and is practically the standard procedure for recovering single superquadrics from 3D point sets. Conversely, if model parameters are known, corresponding 3D points can be determined by pattern classification techniques. The *recover-and-select* paradigm solves the two problems simultaneously by an iterative method of growing the models to acquire suitable point sets and selecting models that describe the whole data set best, by using the *Minimum Description Length*criterion.

2.2.1 Model recovery

One of the main problems, that has a major effect on the success of the whole procedure, is where to find initial estimates (seeds) for a given data set. SEGMENTOR searches for the points that could belong to a single parametric model in a grid-like pattern of windows laid over the range image. Thus the problem of classifying all data points of a certain model is relaxed to only a small subset of points belonging to a single model. There is however no guaranty that every seed will grow to a good model, since some can be recovered over areas belonging to different models. SEGMENTOR independently builds models from all statistically consistent seeds and uses them as hypotheses that could compose the final solution.

Initially the whole data set is partitioned in many subsets (regions) using small windows that are laid over the range image in a grid fashion. Every subset is then fitted a superquadric. A decision if the seed is good is based on comparing the average error-of-fit

$$\overline{\xi_i} = \frac{1}{|R_i|} \sum_{\mathbf{x} \in R_i} d(\mathbf{x}, M_i) = \frac{1}{|R_i|} \xi_i \tag{2.19}$$

with a threshold (Jaklič (1997)). Because of the redundant nature of the method this decision is not critical, it only removes models lying on the image part boundaries and reduces the number of initial seeds.

After seed descriptions are placed, the growing stage can take place. It consists of a search for new compatible points in the vicinity of the description's region. Possible new points have to satisfy two constraints: first, they have to be neighboring the region R_i that belongs to the model M_i, and second, they have to be sufficiently close to the model M_i. New points satisfying both criteria are added to region R_i and model parameters for new model M_i' are recomputed for this extended region R_i'. After that, the average error-of-fit between the region R_i' and M_i' is computed and compared to a threshold, leading to (possible) further growth or termination of it.

In order to prevent sudden changes in orientation of a superquadric due to the ambiguous superquadric parameters, parameters of model M_i are used as initial estimates when computing parameters of model M_i'. But, since this initialization would possibly stuck the model in a local minimum, a second set of superquadric parameters M_i'' is computed the usual way (Solina and Bajcsy (1990)). As the new model the one with a smaller error-of-fit to the region R_i is used.

Individual models are recovered independently (parallel implementation is also possible). The result of the model recovery procedure for a model M_i consists of three terms:

1. region R_i (set of 3D points), corresponding to model M_i,
2. model parameters P_i for model M_i and
3. error-of-fit ξ_i between model M_i and region R_i.

These three terms are subsequently used in the description selection procedure.

2.2.2 Description Selection

Since the search for parametric models is initiated everywhere in the image, the resulting grown descriptions completely or partially overlap. The following procedure takes that redundant representation and combines a subset of descriptions that best describe the whole range image. The task of combining different models is reduced to a selection procedure which considers many competitive solutions and takes the subset that makes up the simplest solution, the one that includes as many points as possible while keeping the models' error-of-fit low. The *Minimum Description Length*principle can be used to derive such a solution.

The objective function to be maximized in order to produce best description in terms of models has the following form

$$F(\mathbf{m}) = \mathbf{m}^T \mathbf{Q} \mathbf{m} = \begin{bmatrix} c_{11} & \dots & c_{1N} \\ \vdots & & \vdots \\ c_{N1} & \dots & c_{NN} \end{bmatrix}, \tag{2.20}$$

where $\mathbf{m}^T = [m_1, m_2, ..., m_N]$ is a presence vector, having $m_i = 1$ where model M_i is included in the description, and $m_i = 0$ where model M_i is absent from the solution. The diagonal terms of matrix $\mathbf{Q}$ express the cost-benefit value of a particular model M_i

$$c_{ii} = K_1|R_i| - K_2\xi_i - K_3|P_i|, \tag{2.21}$$

while the off-diagonal terms express the interaction between the overlapping models

$$c_{ij} = \frac{-K_1|R_i \cap R_j| + K_2\xi_{ij}}{2}. \tag{2.22}$$

K_1, K_2, K_3 are weights (Jaklič (1997), the term $|R_i \cap R_j|$ is the number of points included in both descriptions i and j, and ξ_{ij} is a correction for the diagonal error term in case that both models are selected

$$\xi_{ij} = \max(\sum_{\mathbf{x}\in R_i\cap R_j} d^2(vecx, M_i), \sum_{\mathbf{x}\in R_i\cap R_j} d^2(\mathbf{x}, M_j)). \tag{2.23}$$

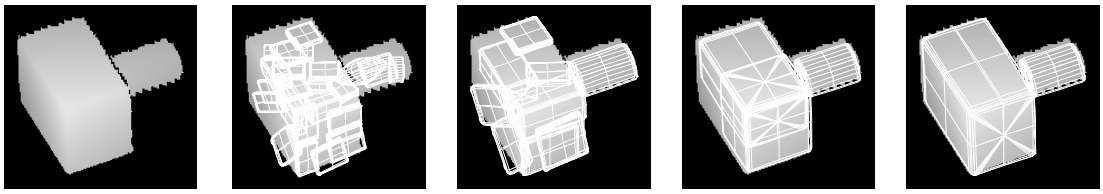
Maximization of the objective function $F(\mathbf{m})$ belongs to the class of combinatorial optimization problems, and since the number of possible solutions increases exponentially with the size of a problem, it is usually not feasible to explore every single solution. In this case, due to the specific nature of the problem, a suboptimal solution can be obtained by a direct application of the *greedy algorithm*, which sequentially selects the option which is locally optimal. Models are selected in the order of their contributions to the objective function. The model that at some stage maximizes the objective function is selected, if the objective function gains with its inclusion. The algorithm stops when the best model does not contribute to the objective function or when all the models are selected. SEGMENTOR uses a fast implementation of the *greedy algorithm* with worst-case time complexity $O(N^2)$ and space complexitiy $O(N)$ (Jaklič (1997)).

2.2.3 Interleaving Model Recovery and Selection

In order to accomplish a computationally efficient procedure, model recovery and description selection can be interleaved. Every few steps of the model recovery of current descriptions are interrupted with the selection of descriptions that describe the data best. This way the computation of growing and recovery of models that would very likely not be included in the final descriptions is avoided. The two phases are interleaved until all of the models are recovered. Positive and negative sides of interleaving model recovery and description selection are discussed in Leonardis (1996).

2.2.4 Segmentation Example

The input to SEGMENTOR is a range image, or a set of 3D points with defined point neighborhoods. Fig. 2.2 shows a sample segmentation of a range image with SEGMENTOR. Fig. 2.2(a) shows the input range image of an object composed of a box and a cylinder. First, seeds are laid over the range image in a grid like fashion in Fig. 2.2(b), followed by first iterations of growth and a selection in Fig. 2.2(c). In the growing stage, new points that are close to the model are added to each description, and a new model is reconstructed on this extended set of points. Compare Fig. 2.2(b),(c),(d),(e) as models grow in size. After certain number of growing iterations several descriptions may completely or partially overlap, indicating a good time for a selection. Using *Minimum Description Length*, a subset of descriptions is selected, that produce the simplest description of the range image. Compare Fig. 2.2(b),(c),(d),(e) as the number of models decreases. To make the whole process more efficient, the growing and selection stages are interleaved with a rule of a thumb that selection is performed when descriptions grow to twice their size (so there possibly is significant overlap, Jaklič et al. (2000)).



(a) range image (b) seeds (c) $g = 2$ (d) $g = 4$ (e) $g = 8$

Fig. 2.2: Example of segmentation with SEGMENTOR. (a) range image, (b) placed seed descriptions, (c) - (e) data descriptions after (additional) g growth steps and a selection. After 14 growth steps altogether with intermediate selections the descriptions are not able to grow anymore and the final description (e) is obtained.

2.3 Chapter summary

In this chapter superquadrics were introduced along with their mathematical properties important for this thesis. Superquadrics have very compact representation as just 11 parameters can be used to represent the basic geometric solids in an arbitrary position in 3D space. Next, range image segmentation with superquadric models using SEGMENTOR system was briefly described, including the *recover-and-select* paradigm that SEGMENTOR is based upon. This chapter describes the work done by other authors, and presentation of this work is primarily aimed at introducing the SEGMENTOR, which results are used as the input to the object recognition system proposed in Chapter 3, and for object tracking proposed in Chapters 4 and 5.

Chapter 3

Modeling and Recognition of 3D Objects

Part-level object models have been used in various object recognition methods, such as the ones from Nevatia and Binford (1977), Bolles and Horaud (1986), and Kim and Kak (1991). In such methods, correspondences between the object model and the input data have to be established for a successful recognition. The problem of recognizing an object can be based on comparison of the parts in the scene to the parts of the modeled object. Scene parts that constitute the input data have to be extracted from the scene first. Parts have to be detected in the scene, or the scene has to be partitioned into segments that are described by those parts. SEGMENTOR is a system that does just that in the case of range image input, such as the one we used (see Section 3.6.1). SEGMENTOR outputs the scene descriptions, each description consisting of A) a range image region that is described by B) a superquadric.

The work in this thesis uses the output from SEGMENTOR and proposes a novel method for object recognition based on interpretation trees. The main contributions of the thesis in the field of object recognition are the superquadric part matching constraints needed for pruning the interpretation search, and the interpretation verification procedure for superquadric object models.

In this chapter a 3D object recognition scheme is presented. The basic principle of the scheme is a search for feasible interpretations arranged in tree structures, i.e. using interpretation trees Krivic and Solina (2004). First, the object model is briefly presented, a very simple one using joint structure and superquadric parts. Then, object recognition is presented as a search for feasible interpretations through interpretation trees. This consists of a part matching for pruning the tree search, and structural verification for determining the soundness of an interpretation. The latter also leads to 3D pose estimation, which is subsequently used in the proposed 3D tracking system as an initialization step. Finally, the experimental results for the scheme are presented.

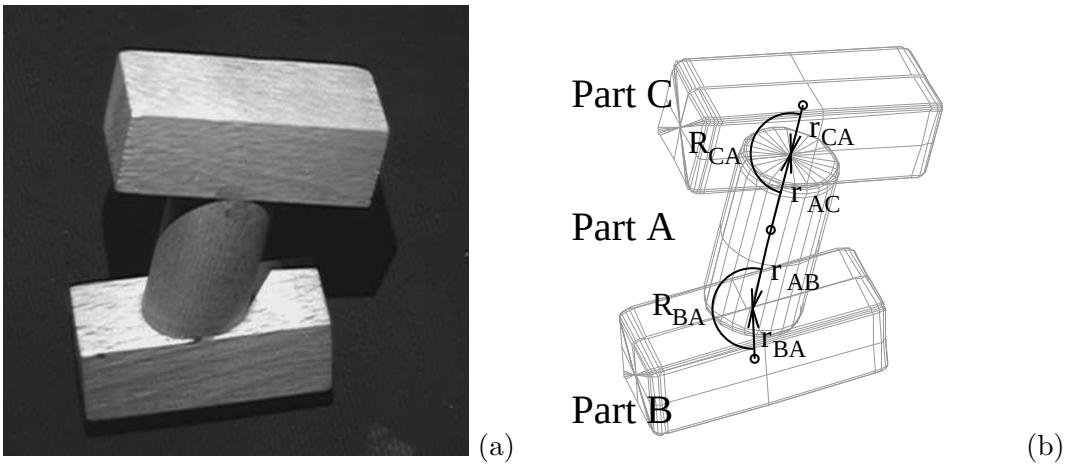
3.1 Object Model

Some kind of object model must exist if an artificial system is to be able to perform 3D object recognition. A fairly simple model was designed for the proposed system, in which

the object is modeled on two levels. On the first level, object's parts are modeled with superquadrics that define the part's size and shape, such as the superquadrics in Fig. 3.1b (see also Fig. 3.5b). On the second level the part structure is described by defining the connections between parts, such as connections in Fig. 3.1b (see also Fig. 3.5b). One part is given the central role in the object's model. To the central part the object position and general orientation can be assigned. Other parts are connected to their "parent parts" by a joint. Vector $\mathbf{r_{ij}}$ denotes the position of joint connecting parts i and j relative to the center of part i. Therefore, to define a joint two vectors $\mathbf{r_{ij}}$ and $\mathbf{r_{ji}}$ are needed.

There are two types of joints: rigid and flexible. Rigid joints contain besides positional parameters, predefined (constant) rotational parameters, denoted by rotational matrix $\mathbf{R_i}$, and therefore rigidly 'glue' the two parts together. The object in Fig. 3.1, for example, contains two rigid joints. Flexible joints, however, do not have fixed rotational parameters, but can be assigned any value from a given interval for rotating the connected parts into the right configuration. Such flexible joints connect parts of non rigid objects such as the figurine in Fig. 3.5. Of course, the values of rotational parameters could also be constrained, as, for example, would be the case of a human arm Filova et al. (1998), which can only move in certain ways, but this is beyond the scope of this thesis.

Throughout this thesis, $\mathbf{r_{ij}}$ stands for the joint position connecting parts i and j, as described above, $\mathbf{c_i}$ is the center of the superquadric that matches, or should match part i, $\mathbf{R}$ is a ZYZ rotation matrix, and ϕ_i, θ_i and ψ_i are rotation angles for part i.



(a) (b)

Fig. 3.1: Simple object (a) and its model (b).

The models are built manually by measuring the parts and approximating the superquadric and other parameters for each part and joint. Fig. 3.1 shows a three part object, consisting of two blocks connected by a cylinder, and its model.

3.2 Model Matching

The output of the SEGMENTOR is therefore a set of N superquadrics, which compose the scene, called *scene parts*. One can easily imagine the process of recognizing an object as matching scene parts with part models of the stored body model. All possible matches

arranged in a tree structure are called an *interpretation tree* (Grimson (1990)). Nodes in an interpretation tree represent a match between a part of the scene and a part of the model. The search for correct interpretation begins at the root of the interpretation tree. The root expands to all possible matches for the first model part. At the first level of interpretation tree search, every one of N scene parts is therefore matched to the first model part. From a given node, the search continues in depth only if the match represented by that node is consistent, i.e. if the two parts represented by that node are similar. On a given level of the interpretation tree search, the corresponding model part is matched to all scene parts from the set of N parts, except the ones that have already been matched on some higher levels of the tree.

In depth-first search, which was used in our system to examine the interpretation tree, the order of model parts as they pertain to the depth of the tree can be important for finding an interpretation quickly. The parts that are reconstructed more consistently and from more viewpoints should be closer to the tree root. The system thus searches more probable interpretations first. One can, of course, easily implement many other enhancements, such as tree pruning, where two parts are matched only if they are in the right distance to the parts already matched. Best-first search could also be implemented by sorting the matches based on part similarity.

In real scenes some parts of an object may be hidden to the viewer and some occluded by other objects or parts. Also, the part detector can miss some parts or introduce some spurious ones. To enable the system to deal with such cases, a fictitious scene part that matches every model part is introduced. A match between this fictitious scene part and a model part is called a *wildcard match* and is simply appended to the list of scene parts.

When the search through the interpretation tree reaches a leaf one gets a consistent interpretation. But because the constraints involved in match consistency test are local in nature, the interpretation does not have to make sense globally. In general, there is no guarantee that a found interpretation makes global sense. These interpretations must therefore be taken only as hypotheses. For most problems one can come up with a test for global consistency which discards wrong hypotheses, a procedure called *interpretation verification*.

Tab. 3.1 outlines the model matching procedure used in the proposed system. *Stack* holds the nodes that have yet to be expanded, while $Interp$ is a list of consistent part matches that form the current interpretation. At the start, the *root* label, signifying the tree root, is pushed on stack. After a node is popped from the *Stack*, the match it represents is checked for constraints by *consistent(*X*,*$\mathcal{T}$*)* function, which returns *true* when X is a *root*, a wildcard match (W, m_k), or is $\mathcal{T}$-consistent real part match (see Section 3.3). On consistency, the match is added to the current interpretation $Interp$ and the tree is explored depth-first, by pushing all the possible matches on stack. When a leaf node is reached, the current interpretation $Interp$ is verified for structural soundness (see Section 3.4).

3.3 Part Match Consistency

As stated in the previous section, if an object is present in some scene, it should consist of the same parts as the object's model. In reality, of course, the parts are not ex-

```
// Stack - stack of nodes to be expanded
// Interp - list of consistent part matches that form an interpretation
// MaxSize - currently maximum interpretation size
// size(Interp) - no. of real part matches in Interp
// root - label for tree root
// consistent(X,T) - returns TRUE when X is a root, or X is a wildcard match, or X
is T consistent real part match
// verify(Interp) - returns TRUE when Interp is sound
Stack = [root], Interp = [], MaxSize = 0;
while (Stack not empty)
   pop next match X = (f_j, m_k) from Stack
   determine the set of constraints T for part m_k
   if (consistent(X,T) AND max. possible interpret. size ≥ MaxVel)
      add X in Interp;
      if (leaf reached)
         if (verify(Interp) )
            save Interp;
            MaxSize = size(Interp)
         end if
      else /* not a leaf, but still consistent */
         push (W, m_{k+1}) on Stack
         for i = 1..N
            if (part f_i not in Interp)
               push (f_i, m_{k+1}) on Stack
            end if
      end if
   end if
end while
```

Tab. 3.1: Interpretation tree search algorithm used in the object recognition scheme proposed in this thesis. The two main components are match consistency procedure *consistent(X,T)* and interpretation verification procedure *verify(Interp)*, defined in Section 3.3 and Section 3.4, respectively.

actly the same, but should be similar enough. The comparison between two superquadric parts, should therefore be tolerant to slight (or great) changes in part shape and size. Superquadric parameters cannot be used directly for comparison of two superquadrics because several sets of parameters can lead to the same size and shape of a part (Jaklič et al. (2000)). Therefore when comparing scene part f_j with model part m_i, a set of constraints $\mathcal{T}_i$ is used to determine the part similarity (i.e. match consistency), which is dependent on the superquadric recovery on the model part m_i. In this thesis, (m_i, s_j) denotes a match between a model part i and a scene part j. A consistent match (m_i, s_j) (where matches for the parts m_0 to m_{i-1} are consistent) means, that the search can continue with the next match (m_{i+1}, s_j) at next level $i+1$ of the interpretation tree, whereas an inconsistent match (m_i, s_j) stops further search in depth and continues with the next match (m_i, s_k)

on the same level i of the interpretation tree.

A basic constraint, that can be included in every part's constraint set $\mathcal{T}_i$, is a *volume constraint* (Krivic and Solina (2001))

$$\mathcal{V}_i(V) : V \in [V_i^{min}, V_i^{max}]. \tag{3.1}$$

If a volume V of a scene part is within the model's part interval, $V_i^{min} \leq V \leq V_i^{max}$, the two parts represent a possible match.

Superquadric recovery on some shapes is not reliable (Jaklič (1997)) and produces many (two or more) overlapping superquadrics, that correspond to a single model part. In those cases, the volume constraint can be extended (Krivic and Solina (2001)) to

$$\mathcal{V}'_i(V) : (\exists sq \in S_i \wedge (\sum_{center(sq) \in S_i} V_{sq}) \in [V_i'^{min}, V_i'^{max}]) \vee (\bar{\exists} sq \in S_i \wedge V \in [V_i^{min}, V_i^{max}]), \tag{3.2}$$

as follows:

- if there are any superquadrics, whose centers are less than some distance S_i from the center of the considered superquadric, the sum of their volumes $\sum V_{sq}$ (including the volume of the considered superquadric) should be in the predefined interval $V_i'^{min} \leq \sum V_{sq} \leq V_i'^{max}$. Distance S_i can be assigned a value of the perimeter of the largest sphere, that can fit into the model part being matched.
- if there are no other superquadrics at such a distance, the part's volume V should be in the usual interval $V_i^{min} \leq V \leq V_i^{max}$.

The two cases are dealt with separately (with different threshold values), because the shared space of the superquadrics is not taken into account.

For parts with reliable superquadric reconstruction, such as the parts of the object in Fig. 3.1, size and shape along minimal inertia axis can be used. The *size constraint* (Krivic and Solina (2001)) is defined as

$$\mathcal{S}_{i,l}(a'_l) : a'_l \in [a_{i,l}^{min}, a_{i,l}^{max}], l = 1, 2, 3. \tag{3.3}$$

Similarly, the *shape constraint* is defined as

$$\mathcal{H}_{i,m}(\epsilon') : \epsilon'_m \in [\epsilon_{i,m}^{min}, \epsilon_{i,m}^{max}], m = 1, 2. \tag{3.4}$$

Constraints can be computed as follows: first, inertial moments along x, y, z axes are computed for part f_j, and sorted. Next, a'_l is assigned the a_k parameter that corresponds to the l-th lowest inertial moment value (e.g. when inertial moment along y axis is lowest, $a'_1 := a_2$, since a_2 is the size along y axis). ϵ'_1 is assigned the ϵ_1 parameter when inertial moment along x or y axis is smallest, and ϵ_2 when inertial moment along z is smallest. ϵ'_2 is assigned the remaining ϵ parameter. For parts with a very reliable reconstruction the *volume difference* (Chen et al. (1997)) constraint could be used in order to match parts to shape and size as accurately as needed. The volume difference constraint was not implemented in our system though, because of its high time complexity.

Note that the properties used above are unary. When including a scene part in an interpretation, there are possibly other parts already included. In order to reduce the search

space, binary (n-ary) constraints, such as distance between two parts, can be introduced into the part matching procedure. The *distance constraint*

$$\mathcal{D}_{i,j}(d) : d \in [d_{i,j}^{min}, d_{i,j}^{max}] \tag{3.5}$$

is based on the distance d between the centers of scene superquadrics that represent parts m_i and m_j.

The purpose of part match consistency is to prune the interpretation tree, leading to faster interpretation discovery. The constraints involved should be adjusted so that the part matching procedure rejects as many unsuitable parts, while accepting any possible part matches that may appear in scene reconstructions. In this way the system does not "overlook" any objects and finds them quickly.

3.4 Interpretation Verification

A search through an interpretation tree is guided by match consistency constraints, which are local and only compare parts (or relate few parts). Therefore, when the search reaches a leaf signifying a consistent interpretation, the interpretation is not necessarily sound, i.e. the parts from the interpretation do not represent the modeled object. Consistent interpretations have to be only taken as hypotheses and further verified. *Interpretation verification* is a procedure which should answer the question: "Does the given set of parts really represent the object X?" The proposed scheme answers this question in a few steps.

First, the system can reject interpretations that include too many wildcard matches, by setting a threshold P on the real interpretation size. For example, if the object is a part of a fence with twenty iron poles welded to a frame (24 parts altogether), the object could be recognized even if 16 poles are missing, thus a threshold $P = 8$ parts (one third of model parts) would be reasonable. On the other hand when recognizing the figurine from Fig. 3.5, three matched parts (approximately one third of model parts) does not necessarily indicate the object's presence. One would expect at least five matched parts to be sure of the object's presence. By rejecting interpretations that include too few real matches, the system may therefore reject some correct interpretations (false negatives), but it will reject many more wrong ones (false positives), since the probability that some parts will randomly form a structure similar to the structure of the object decreases as the number of matched parts increases. The threshold P can be set to some fraction of the number of model parts and depends on the modeled object as well as on the application. For most objects the threshold P can be set to around half of the number of model parts.

Second, for a given interpretation, the hypothetical object position and part configuration can be computed (Krivic and Solina (2002), Krivic and Solina (2004)). The work in this thesis is focused on articulated objects consisting of elongated parts with unreliable reconstructions, that are joined near the longer ends. Using this assumption, the configuration can be computed efficiently. Let the part's main axis be the axis of minimal inertia (Jaklič and Solina (2003); Jaklič et al. (2000)). Analysis of such objects showed that the main axes of scene parts are well aligned with true main axes of the object model parts. When a joint is configured so that it connects two parts, the following rotation of the subordinate part is the rotation that aligns its main axis with the main axis of the

matched scene part:

$$\begin{aligned}
\mathbf{R_{X\to s}}: &\quad \phi = \arctan\tfrac{-s_x}{s_y}, && \theta = \tfrac{\pi}{2}, && \psi = \arctan\tfrac{\sqrt{s_x^2+s_y^2}}{s_z} \\
\mathbf{R_{Y\to s}}: &\quad \phi = 0, && \theta = \arctan\tfrac{s_z}{-s_x}, && \psi = \arctan\tfrac{\sqrt{s_x^2+s_z^2}}{s_y} \\
\mathbf{R_{Z\to s}}: &\quad \phi = \arctan\tfrac{s_y}{s_x}, && \theta = \arctan\tfrac{\sqrt{s_x^2+s_y^2}}{s_z}, && \psi = 0,
\end{aligned} \tag{3.6}$$

where $\mathbf{R}_{...}$ is a ZYZ rotation matrix, ϕ, θ and ψ are rotation parameters and $\mathbf{s} = [s_x, s_y, s_z]^T$ is a scene part's unit main axis vector rotated in the local coordinate frame of the part superior to the part being configured. In the general case, the object's configuration is hard to determine due to inherent rotational ambiguity of superquadrics. For computing the configuration, custom procedures tailored to particular objects, or types of objects, would have to be developed.

After an object model is approximately configured to its interpretation, this configuration can serve as the basis for the third step in the interpretation verification. The individual superquadric part of the object model can then be fitted to those regions in the range image that correspond to their position given by the approximated configuration. To fit individual superquadric models to such part regions the standard fitting method was used (Solina and Bajcsy (1990)). The fitting function (Jaklič (1997); Jaklič et al. (2000))

$$G(\Lambda) = a_1 a_2 a_3 \sum_{i=1}^{N} (F^{\epsilon_1}(x_i, y_i, z_i) - 1)^2, \tag{3.7}$$

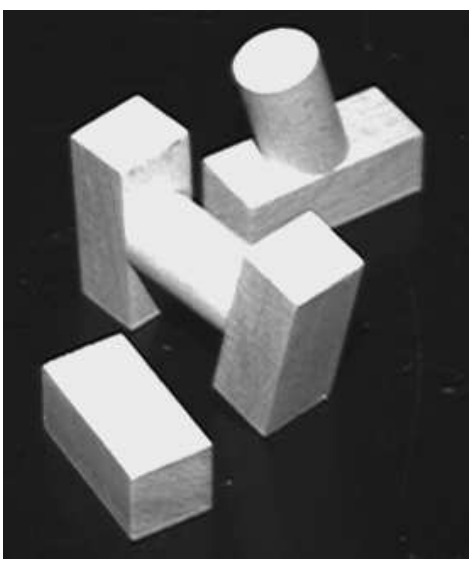
where F is the superquadric implicit function from Eq. 2.3 and $[x_i, y_i, z_i]^T$ the point i from the range image region, was minimized only for the position and orientation parameters, i.e. $\Lambda = (t_x, t_y, t_z, \phi, \theta, \psi)$, while the size (a_1, a_2, a_3) and shape (ϵ_1, ϵ_2) parameters were fixed to the values of the tested model part superquadric. The position and orientation parameters of the tested superquadric were used in the minimization as initial parameters. For model parts with reliable reconstructions, the interpretation is rejected, if the error of the fitting function (Jaklič (1997); Jaklič et al. (2000)) is greater than threshold $E = 2.5$ (the same as used in range image segmentation). For parts with unreliable reconstruction, the model superquadric is fitted in the same way, but the interpretation is rejected if the poles (points where the main axis pierces the superquadric surface) move more than a threshold D_i.

The final interpretation therefore consists of the object model whose configuration in 3D has been refined by fitting each superquadric of the object model to its corresponding region in the range image (Krivic and Solina (2004)).

Rigid joints are then further verified for consistency. Due to the rotational ambiguity of superquadrics, the proposed scheme does not deal thoroughly with joint rigidity, but rather just compares the angle between the two main axes.

There is another aspect of interpretation verification, namely the feasibility of the object's configuration. If an articulated object is set by the interpretation process into a non-feasible configuration, the verification process should reject it. Configuration feasibility is beyond the scope of this thesis, but could nevertheless be applied to the presented scheme by defining sets of valid intervals for joint rotation parameters. The joint rotation parameters extracted from the final interpretation would then be compared to those sets thus determining if self-penetration and other non-feasible poses have occurred.

3.5 Interpretation Search Example


(a)

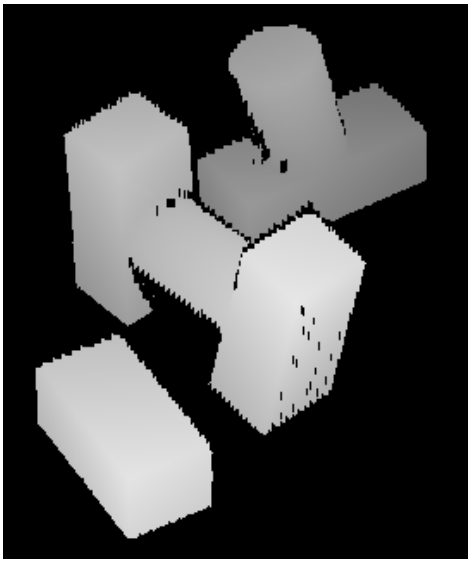
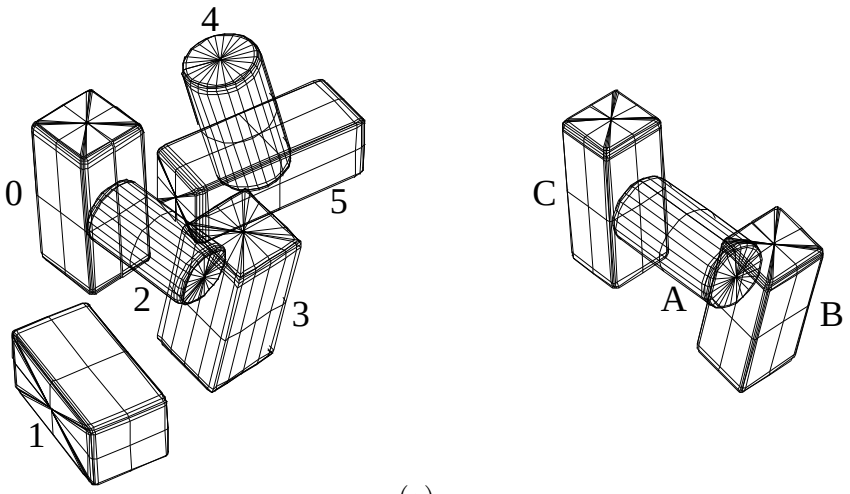

(b)

Fig. 3.2: Simple scene (a) containing the object from Fig. 3.1, and the corresponding range image (b).

Let us look at an example of interpretation search in greater detail. The input scene and its range image are depicted in Fig. 3.2, in which a simple test object seen in Fig. 3.1 and Fig. 3.3b is searched for. The parts of the test object are labeled A,B and C. The first step of the recognition process is superquadric recovery using the SEGMENTOR. The superquadric reconstruction on scenes such as the one in Fig. 3.2, where all parts can be perfectly modeled by superquadrics, is very reliable. For each scene part one superquadric is reconstructed, which describes the corresponding scene part very well, and there is almost no overlapping. The result can be seen in Fig. 3.3a. Parts in the reconstruction are labeled $0-5$. The threshold P on real interpretation size is set to 2 (the interpretation must include at least 2 parts).

Next, the search for possible interpretations begins using interpretation trees. The interpretation tree for finding the test object is shown in Fig. 3.4. The search begins at the root. The root expands into nodes representing matches between model part A and scene parts from 0 to 5. First, part A is compared to part 0. Since part A is a cylinder and part 0 is a block, the match is not consistent and the search does not continue in depth. It rather proceeds on the same level by visiting the right sister node and comparing parts A and 1. Again, this match is discarded due to part shape mismatch. Visiting right sister node on the same level, the search continues by comparing part A with part 2. Size and shape approximately match, so that the search can continue in depth, by searching for a match for model part B. The comparison to scene part 0 delivers a consistent match, since the size and shape match. Continuing one level deeper, matches for model part C are searched. The first node yields a consistent match between parts C and 1. Since this is a leaf node, a consistent interpretation $(A, B, C) = (2, 0, 1)$ is obtained.

Although the parts have pairwise the same size and shape, a glance at Fig. 3.3 can



(a) (b)

Fig. 3.3: (a) Superquadric reconstruction of the scene from Fig. 3.2 and (b) labeled model from interpretation verification.

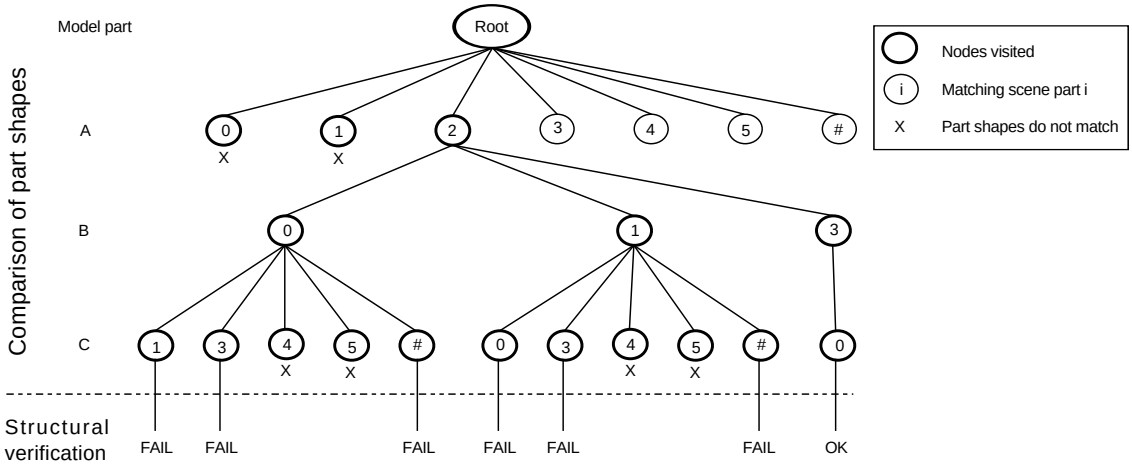


Fig. 3.4: Interpretation tree for scene in Fig. 3.2

tell that the recovered parts $(2, 0, 1)$ do not represent the object in question, since their configuration is wrong. This is why every consistent interpretation derived by the interpretation tree must be verified using the properties of the whole object. The parts in the interpretation should conform to the same structure as parts that compose the model. The system can verify if the configuration occupied by the scene parts in the interpretation corresponds to the model using structural information described in Section 3.1.

The first step in the process of verifying the interpretation $(A, B, C) = (2, 0, 1)$ is putting a threshold P on its size. Since the interpretation includes 3 real matches, it passes the first test. Next, the configuration of the hypothesized object is compared to the object's model. Scene part labeled 1 is too distant from part 2, so the interpretation is rejected,

and the search continues at the right sister node, with interpretation $(A, B, C) = (2, 0, 3)$. When comparing the hypothesized configuration with the model, the distances between part centers match. But since the joints in the object are rigid, the joint rotations of the hypothesized object do not match the modeled ones, because the model parts A and B are slightly tilted whereas scene parts 2 and 0 are perpendicular. If both joints were flexible, the configuration would match, the rotational parameters would be computed, and the superquadric part fitting would proceed. Since the hypothesized object's configuration would be accurate, the superquadric parts of the model wouldn't move or rotate much in the process of fitting, and the interpretation would succeed.

Let us skip forward in the interpretation tree search until the interpretation $(A, B, C) = (2, 0, \#)$ is found. After trying all possible matches for model part C, there is also a possibility, that the part in question is missing (is occluded, or the reconstruction isn't appropriate). The hash sign (#) in the interpretation stands for a wildcard, that is a fictitious part that matches every model part. A wildcard match is simply appended (at the end) of the list of scene parts. Interpretation $(A, B, C) = (2, 0, \#)$ is thus consistent, but fails again on structural verification. For the purpose of demonstration, let us imagine that part 0 is occluded from the scene. The interpretation tree search would then lead to interpretation $(A, B, C) = (2, 3, \#)$, which is structurally sound, and also represents the best interpretation for the scene.

The search continues with three more consistent interpretations, which fail on structural verification, until the correct interpretation $(A, B, C) = (2, 3, 0)$ is found.

3.6 Experimental Results

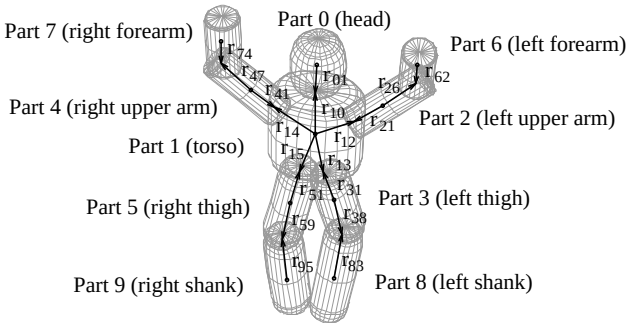
Human figures were used as generic articulated test objects for the recognition task. We were not interested in the specific problem of modeling the human form and do not want to compete with dedicated human form capture systems, although it should be mentioned that systems using superquadrics for modeling humans do exist (Jojic and Huang (2000)). Due to the rather small workspace of the range scanner used (see next subsection), toy figurines were used instead. Figurines representing "Commander Data" from the Star Trek series were selected. Fig. 3.5 shows the object and its model. Its arms and legs are flexible and the figurine can thus be configured into many different poses.

3.6.1 Setup

The experimental setup for the system was as follows: range images were obtained by the structured light range scanner. Its main components are an ABW LCD projector for projecting the structured light sequence onto the scene, a Sony XC-75CE camera for capturing the image sequence, and Linux based software (Skočaj and Leonardis (2000)) that controls the projector and camera, and generates the range image from the captured sequence. A range image is an array of 450×450 elements signifying the distance between the element and the camera. The work space of the scanner is about $25cm \times 25cm \times 20cm$, so that objects larger than that can not be scanned as a whole. It takes the scanner about ten seconds to capture a range image.

Captured range images were processed with the SEGMENTOR. On a 400Mhz PC, the processing took from 1:30 (simple scenes) to 3:00 hours (complex scenes).

(a)



(b)

Fig. 3.5: Toy figurine (a) is modeled in two levels (b): superquadric part models define the size and shape of individual parts (grey models) while the structural level (vectors $\mathbf{r_{ij}}$) defines how parts are connected to each other.

Input	Model parts	Scene parts	Objects	Computation time
H-object 1, Fig. 3.2,3.3	3	6	1	2.9s
H-object 2, Fig. 3.6	3	8	1	3.4s
Toy figurine 1, Fig. 3.8	10	183	2	48.7s
Toy figurine 2	10	121	2	42.5s

Tab. 3.2: Processing times for some input sets.

The resulting sets of superquadric descriptions were processed with the recognition system as described in the previous sections. The system was implemented in C++, and the processing times for some examples can be seen in Tab. 3.2. Tab. 3.4 and 3.5 show constraints and verification parameters' values that were used in the experiments, respectively.

Model of the figurine was built manually. The model consists of superquadrics (Fig. 3.5b). Each superquadric represents one of the major body parts: head, torso, a pair of upper arms and forearms, and a pair of thighs and shanks. Due to the limited scale of parts which can be recovered on the selected range image resolution by the SEGMENTOR, the model does not include distinct models of hands and feet. Each body part is described by a superquadric of a particular size and shape. The torso is given a central role in the model. The head and upper arms and legs are attached to it via joints (Fig. 3.5b). For each of those parts the joint position in relation to the center of the part itself ($\mathbf{r_{i1}}$) and to the center of the torso ($\mathbf{r_{1i}}$) is defined. Similar is true for lower extremities. The parameter values for all parts were obtained by measuring the figurine and are listed in Tab. 3.3.

The figurine is interesting in several ways. It is fairly realistic and naturally shaped and therefore cannot be perfectly modeled by superquadrics. Since the surfaces are not smooth, the reconstruction of superquadrics on their range images is less stable. There can be several superquadrics reconstructed on a single scene (object) part, or a single superquadric can span over several scene (object) parts. The flexibility of body joints makes the matching problem even more complex than if the object part configuration

No.	*Part*	a_1	a_2	a_3	ϵ_1	ϵ_2	Volume
0	head	8	8	10	0.7	1.0	3185
1	torso	14	10	15	0.3	0.9	13077
2, 4	upper arm_x	5	5	13	0.1	1.0	2027
3, 5	thigh_x	7	7	17	0.3	1.0	4995
6, 7	forearm_x	5	5	10	0.1	1.0	1559
8, 9	shank_x	6	6	17	0.3	1.0	3670

Joint positions

$\mathbf{r_{10}} = [0,0,15]^T, \mathbf{r_{01}} = [0,0,-10]^T, \mathbf{r_{12}} = [15,0,8]^T, \mathbf{r_{21}} = [0,0,7]^T,$
$\mathbf{r_{13}} = [5,0,-22]^T, \mathbf{r_{31}} = [0,0,7]^T, \mathbf{r_{14}} = [-15,0,8]^T, \mathbf{r_{41}} = [0,0,7]^T,$
$\mathbf{r_{15}} = [-5,0,-22]^T, \mathbf{r_{51}} = [0,0,7]^T, \mathbf{r_{26}} = \mathbf{r_{47}} = [0,0,-9]^T,$
$\mathbf{r_{62}} = \mathbf{r_{74}} = [0,0,10]^T, \mathbf{r_{38}} = \mathbf{r_{59}} = [0,0,-14]^T, \mathbf{r_{83}} = \mathbf{r_{95}} == [0,0,17]^T$
$x = \{1,2\}$

Tab. 3.3: Model parameters for toy figurine object from Fig. 3.5.

would be rigid.

3.6.2 Constraint Values and Verification Parameters

Reconstructions of superquadrics on range images of the object taken in different poses and from different viewpoints differ greatly. The exception is the head since the analysis of superquadric reconstructions of the human body showed that the head was the most consistently reconstructed body part. At the same time, the head is also the only part that does not change significantly its relative position in relation to the torso. It was therefore reasonable for the head part to use in part matching beside the volume constraint also the size $\mathcal{S}$ and the shape $\mathcal{H}$ constraints, to early on reject as many unsuitable parts as possible.

Superquadrics reconstructed on the torso region differ the most from the torso's model superquadric. On this region several possibly overlapping superquadrics can be recovered, which can partially extend even into regions belonging to extremities. Thus, the extended volume constraint $\mathcal{V}'$ was used for the torso part.

Tab. 3.4 lists the constraint values, while Tab. 3.5 lists verification parameters used for the figurine object. Values were defined on the basis of thirty superquadric reconstructions of the object's range images.

3.6.3 Results

Let us first present an example recognition of the simple object from Fig. 3.1. Fig. 3.6a to Fig. 3.6d show the scene, the superquadric reconstruction of the scene, the best hypothesized interpretation and the verified interpretation, respectively. The interpretation found is a valid one, consisting of matches for all object parts, which are configured correctly.

As previously mentioned, the recognition system was tested using the figurine object on two types of scenes:

- scenes containing only one figurine in different configurations, and

No.	*Part*	*Constraints*	*Constraint Values*
0	head	$\mathcal{T}_0 = \{\mathcal{V}_0, \mathcal{S}_{0,1}, \mathcal{S}_{0,2}, \mathcal{S}_{0,3}, \mathcal{H}_{0,1}, \mathcal{H}_{0,2}\}$	$V_0^{min} = 1000$, $V_0^{max} = 6000$, $a_{0,1}^{min} = 6.5$, $a_{0,1}^{max} = 15$, $a_{0,2}^{min} = 5$, $a_{0,2}^{max} = 11$, $a_{0,3}^{min} = 3$, $a_{0,3}^{max} = 9$, $\epsilon_{i,1}^{min} = 0.5$, $\epsilon_{i,1}^{max} = 1.4$, $\epsilon_{i,2}^{min} = 0.4$, $\epsilon_{i,2}^{max} = 1.5$
1	torso	$\mathcal{T}_1 = \{\mathcal{V}'_1, \mathcal{D}_{1,0}\}$	$V_1^{min} = 3500$, $V_1^{max} = 14000$, $V_1'^{min} = 5500$, $V_1'^{max} = 20000$, $S_1 = 12$, $d_{1,0}^{min} = 16, d_{1,0}^{max} = 27$
2,4	upper arm	$\mathcal{T}_{(2\|4)} = \{\mathcal{V}_{(2\|4)}, \mathcal{D}_{(2\|4),1}\}$	$V_{(2\|4)}^{min} = 300$, $V_{(2\|4)}^{max} = 6500$ $d_{(2\|4),1}^{min} = 14, d_{(2\|4),1}^{max} = 31$
3,5	thigh	$\mathcal{T}_{(3\|5)} = \{\mathcal{V}_{(3\|5)}, \mathcal{D}_{(3\|5),1}\}$	$V_{(3\|5)}^{min} = 500$, $V_{(3\|5)}^{max} = 8000$ $d_{(3\|5),1}^{min} = 20, d_{(3\|5),1}^{max} = 34$
6,7	forearm	$\mathcal{T}_{(6\|7)} = \{\mathcal{V}_{(6\|7)}, \mathcal{D}_{(6\|7),(2\|4)}\}$	$V_{(6\|7)}^{min} = 300$, $V_{(6\|7)}^{max} = 3500$ $d_{(6\|7),(2\|4)}^{min} = 7, d_{(6\|7),(2\|4)}^{max} = 23$
8,9	shank	$\mathcal{T}_{(8\|9)} = \{\mathcal{V}_{(8\|9)}, \mathcal{D}_{(8\|9),(3\|5)}\}$	$V_{(8\|9)}^{min} = 300$, $V_{(8\|9)}^{max} = 4000$ $d_{(8\|9),(3\|5)}^{min} = 9, d_{(8\|9),(3\|5)}^{max} = 38$

Tab. 3.4: Match consistency test values for the toy figurine object from Fig. 3.5.

Meaning	*Variable*	*Value*
Interpretation size threshold	P	5
Maximum pole rotation thresholds	D_1	3
	D_2, D_3, D_4, D_5	11
	D_6, D_7	6
	D_8, D_9	7

Tab. 3.5: Values of interpretation verification parameters used in the experiments.

- complex scenes containing one or two figurines along with a large number of other parts.

With the first set of test images we wanted to test systematically the system's performance for isolated figurines. The figurine was configured into seven different poses and for each pose, range images from eight different viewpoints were captured, which makes a total of 56 images.

Fig. 3.7 shows one of the results, while Tab. 3.6 summarizes the recognition results. The object was detected in 39 cases. In 24 of those cases, the model computed from the best interpretation fitted the object very well. An interpretation included on the average 7.2 real matches. The object was not detected in 17 cases. In 9 of those cases, the reason for the failure was a singular object configuration as seen from that particular viewpoint. Due to occlusion, some parts, mainly the torso or the head, were not recovered properly, thus leading to a part configuration, which was later rejected when superquadric refitting

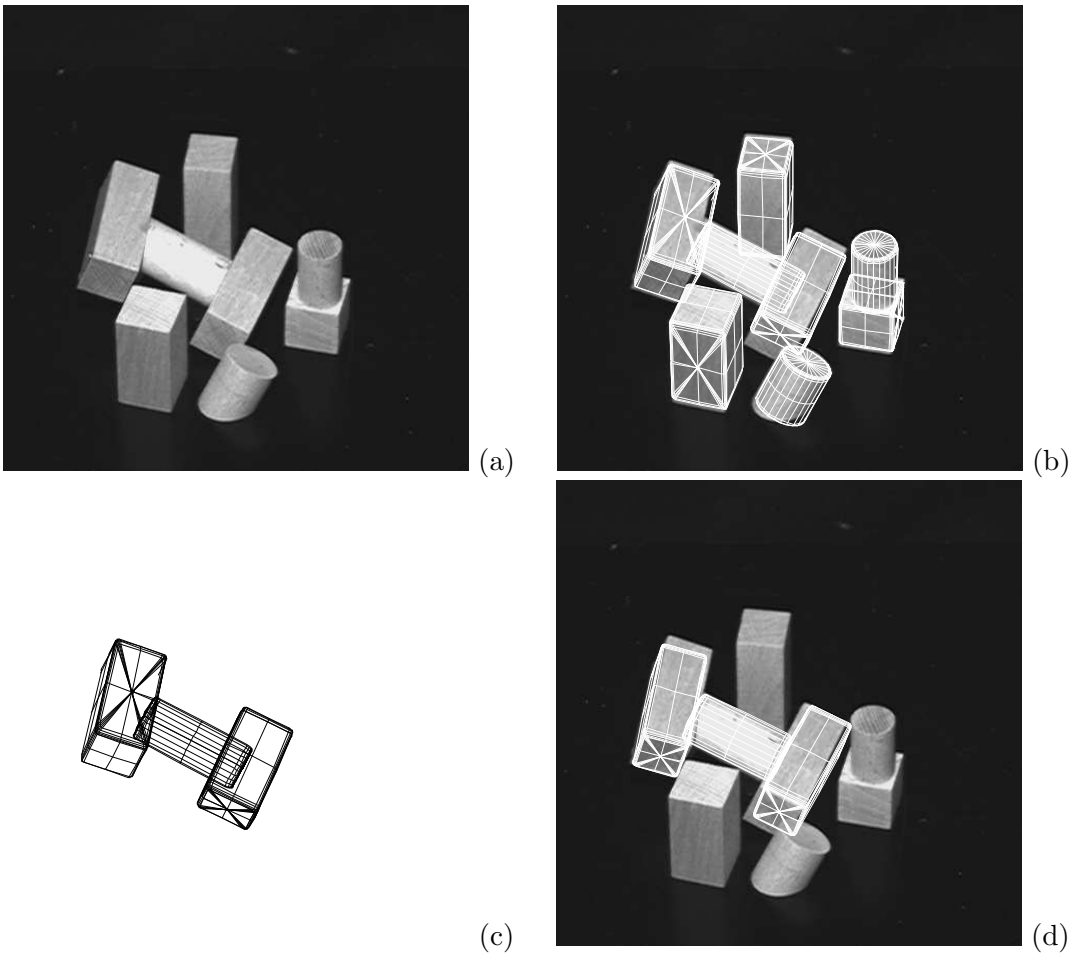
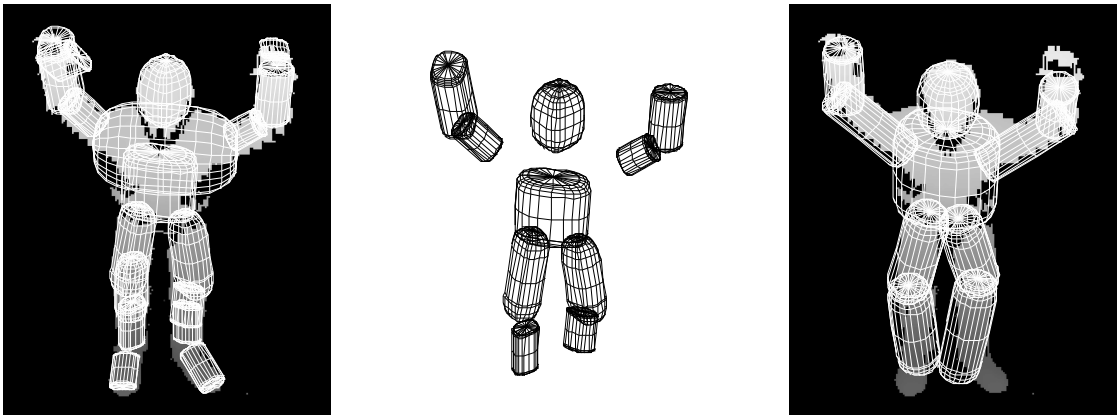


Fig. 3.6: Interpretation of a simple scene: (a) intensity image of a scene, (b) input range image with superimposed reconstructed superquadrics, (c) superquadrics selected for the hypothesis, (d) verification by refitting superquadrics of the model to corresponding segments in the range image.

was done. In the 8 other cases of failure the best interpretation found included less than five real matches, and was therefore rejected.

The system's performance was also tested on 20 different complex scenes. Complex scenes included several appearances of the figurine, as well as many unknown objects (see Fig. 3.8,3.9). Nevertheless, there were no false positive recognitions of the human form, although there were many at least partially misleading part configurations. It is much harder to test a complex scene in a systematic fashion because of so many possible variables. One can observe that the reconstructions of the supporting surfaces in complex scenes were not appropriate, because such surfaces cannot be modeled well by superquadrics.



(a) (b) (c)

Fig. 3.7: Single figurine scene: (a) the input range image with superimposed reconstructed superquadrics, (b) superquadrics selected for the hypothesis, (c) verification by refitting superquadrics of the model to their corresponding segments in the range image.

Total number of scenes: 56			
Object detected: 39		Object not detected: 17	
Model fit:		Cause:	
Good	Poor	Occluded head or torso:	Too few real matches:
24	15	9	8

Tab. 3.6: Results of recognition on 56 scenes consisting of only one object.

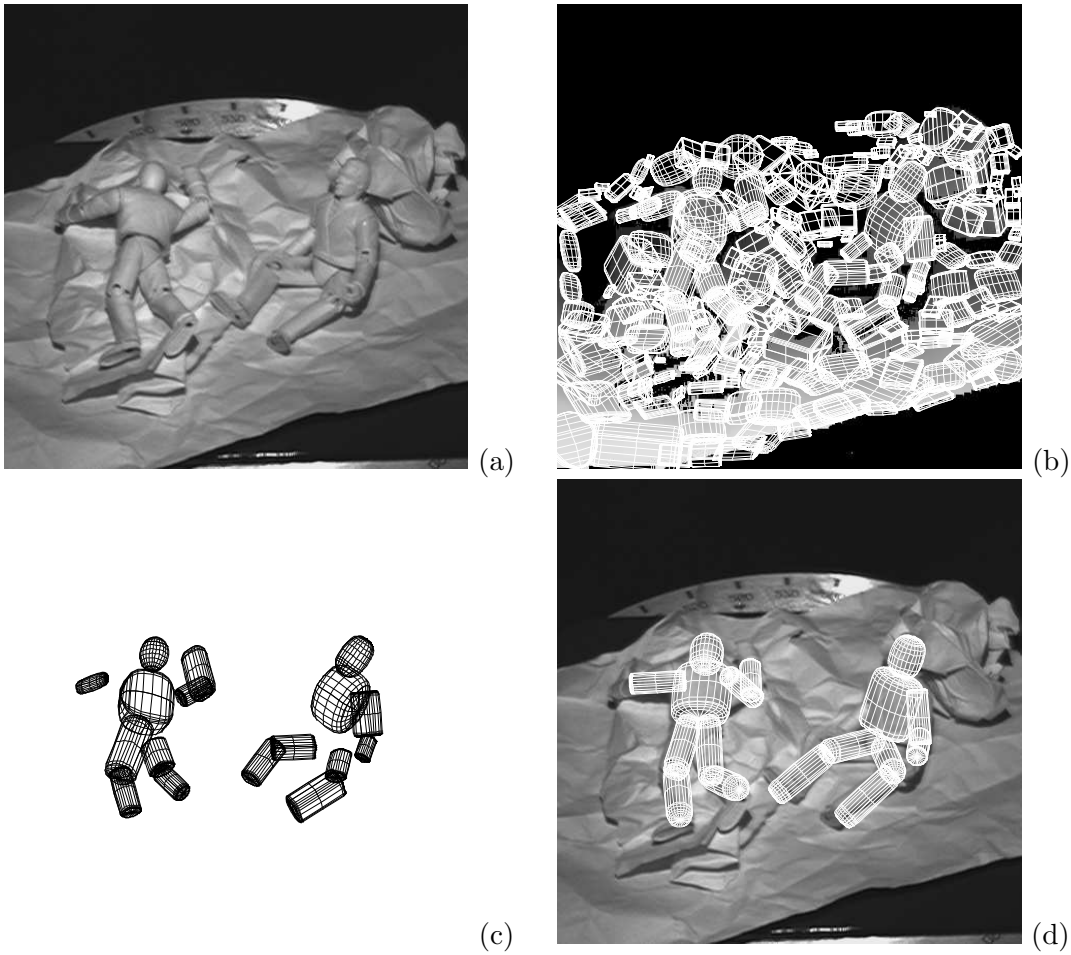
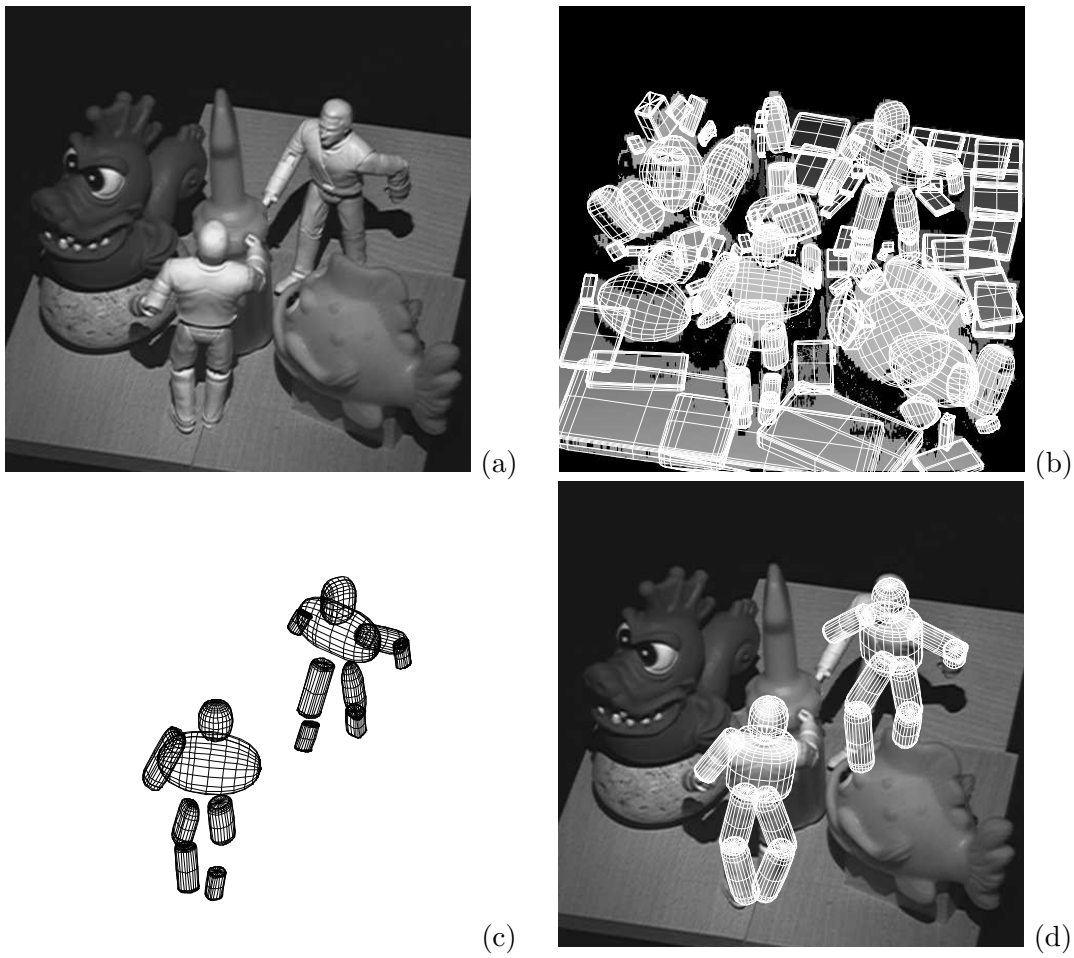


Fig. 3.8: Interpretation of a complex scene: (a) intensity image of a scene, (b) input range image with superimposed reconstructed superquadrics, (c) superquadrics selected for two hypotheses, (d) verification by refitting superquadrics of the model to corresponding segments in the range image.



(a) (b) (c) (d)

Fig. 3.9: Interpretation of a complex scene: (a) intensity image of a scene, (b) input range image with superimposed reconstructed superquadrics, (c) superquadrics selected for two hypotheses, (d) verification by refitting superquadrics of the model to corresponding segments in the range image.

3.7 Chapter Summary

This chapter proposes a novel object recognition scheme, which takes an an input a set of superquadric descriptions of a scene obtained by the SEGMENTOR system, and outputs the instances of known objects (if any) and their likely configurations. Objects are modeled with superquadric part descriptions assembled together by rigid or flexible joints. Recognition is presented as a search for a subset of scene parts that match the object model. In order to perform this search, part match constraints such as volume, volume difference, size, shape and distance constraint, were defined that prune interpretation tree and perform basic object matching. The procedure for determining the soundness of an interpretation was defined next, which is based on calculation of object configuration followed by fitting the superquadrics from object model to the range image. Presented results of the object recognition system were reasonably good, and mostly showed quality accordance of interpretation based object configuration with the scene reality. Preliminary results of this recognition scheme were published in the *Computer Vision and Image Understanding* journal (Krivic and Solina (2004)).

Chapter 4

Initialization of 3D Object Tracking

Object tracking consists of determining the object position in the input image or images. Most approaches to 3D object tracking conceptually divide object position assessment to object detection with initial pose estimation and subsequent object movement. Primarily the reason is in the relative ease of the latter part as compared to the former.

Most of the existing methods for 3D object tracking do not deal with system initialization. In such systems it is the user's responsibility to set up the various object parameters for the system to even begin tracking an object. Some of the 3D object tracking systems are able to facilitate user interaction in the sense that the user has to input some higher-level information about the object. This higher-level information usually consists of joint locations in the input image(s), and most of the systems use some variation of the pose reconstruction method of articulated objects as proposed by Taylor (2000). Using user labeled joint locations in a single uncalibrated image, the pose of an articulated object can be determined up to a scaling factor. While the ad-hoc approach to scaling factor computation is burdensome, the main hurdle for automating such procedures lies in determining the joint locations. An approach by Rosales et al. (2001b) targets joint positions in images for multiple views by preprocessing with the SMA system (*Specialized Mapping Architecture*). SMA transforms input image features to hypothetical joint positions for the view images, and the object pose is afterwards reconstructed by a derivation of the Expectation Maximization algorithm. Drawbacks include the need for supervised learning, background removal needed for hypothesis generation as well as the need for three or more views for a simple human model, as results suggest.

As mentioned above, the main advantage of separating object detection (with its initial pose estimation) and object movement computation is efficiency. A scheme consisting of object detection on every single frame of the sequence is a valid one, but has been avoided primarily because few methods even provide the detection part, and in those which do, there are more efficient ways of computing the movement from frame to frame. But the approach of Sigal et al. (2004) does just that by detecting candidate parts on every image, thus determining the object and its pose through a variant of particle filtering. Object detection on every single frame has two direct and desired consequences: automatic initialization and recovery from mistracking are inherent. Nevertheless, the approach uses

appearance based part detectors, requiring the object's appearance be known in advance.

In this chapter a 3D object detection scheme for initialization of object tracking is presented. The scheme consists of an object detection step presented in the previous chapter, yielding the object instances on the initial frames along with their approximate positions, and another fitting step aimed at improving the object pose. Finally, the experimental results for the scheme are presented.

4.1 Object Detection

In order to be able to provide range data for segmentation with SEGMENTOR, while still being able to produce images at video rate, we opted for a stereo camera input to the system. A stereo camera system can provide range measurements based on images obtained from slightly offset cameras (e.g. Baker and Binford (1981), Jones and Malik (1992)). This is accomplished in three steps. In the first step correspondences between image features in the two views of the scene are established. Usually this is done by some correlation measure between image regions. In the second step relative displacements, also called disparities, for feature coordinates are computed. And finally in the third step, the 3D locations for the matched features (and therefore also their respective image points) can be computed using the knowledge of the camera geometry. The 3D position of the matched feature is a function of the disparity, the focal length of the lenses, resolution of the image sensor and the displacement between cameras. We used a commercially available stereo camera system from PGR (2006), see Section 4.3.1 for details of the system.

The first step of the tracking initialization is object detection. Input range data is first segmented to obtain superquadric descriptions. Then, using the proposed object recognition scheme from the previous chapter, object pose is determined from hypothesized descriptions that best match the object model. If no objects are detected, the next frame can be processed. When an object is detected, the proposed object recognition scheme determines its approximate position as well as the part configuration. This can, however, be further improved to better match the given range data.

4.2 Improving Object Pose

Since object position and its part configuration is at least to some extent initialized after the object is detected, this configuration can be used as a starting point for fitting the object model parts directly to 3D data in the proximity of initialized object, further improving object position and configuration.

A better position of some part can be computed by fitting the part's model to an appropriate region of 3D points. The fitting is done by minimizing the error of fit function just for parameters of translation and rotation, or parameters of rotation around the attaching joint, for central or attached parts, respectively. The key question in this process is how can the appropriate region of 3D points be determined. We distinguish two different cases, one for a central part and one for an attached part. In the case of a central part i, all points in a certain proximity to the part's model (i.e. the points lying at most some distance d_{prox_i} from the central part's model) are included in the region. In the second

case of an attached part i, the region consists of 3D points in the range image that meet the following criteria:

1. the point lies at some maximal distance d_{prox_i} to the part's model as rotated around the attaching joint in last known configuration,

2. the point lies inside of a paraboloid centered in the joint, rotated in the direction of the part's model from the previous frame, and wide enough to encompass the majority of the part's model rotated by some maximum possible rotation, and

3. the point does not lie closer than some distance d_{prox_j} to any part j which is higher in the kinematic chain (i.e. in a region corresponding to the parts higher in the kinematic chain).

The first and second criteria should provide enough points in the proximity to compensate for some possible joint rotation, while not including possible points from other parts. The third criterion aims at stabilizing the fitting and somewhat reduces self-penetration and other infeasible configurations, because the parts' respective regions do not overlap.

There are two possible sources of instability with this approach. The first is the determination of 3D points region. On one hand, the region may not include all the points that belong to the corresponding part. This can be avoided by increasing d_{prox_i}. On the other hand, the region may include points from other parts, which can be reduced by decreasing d_{prox_i}. We tried to reduce the influence of the two antagonistic modes by performing the model fitting in several steps, reducing the distance d_{prox_i} with each step. This can be accomplished in a simple fashion by setting the starting value and ending value for the d_{prox_i} parameter, and linearly reducing it according to the number of steps.

The second possible source of instability comes from the quality of object model. The influence of model quality is twofold. First, it influences the 3D point region, as the distance from a point to the model does not correspond necessarily to the distance from the point to the part. The better the quality of the model, the better the two distances match and thus the better the points from a region correspond to the to the part. Second, the model quality influences the fitting process. The better the match between the size and shape of the model to the part, the more stable the fitting process is, which in turn leads to more accurate position estimation.

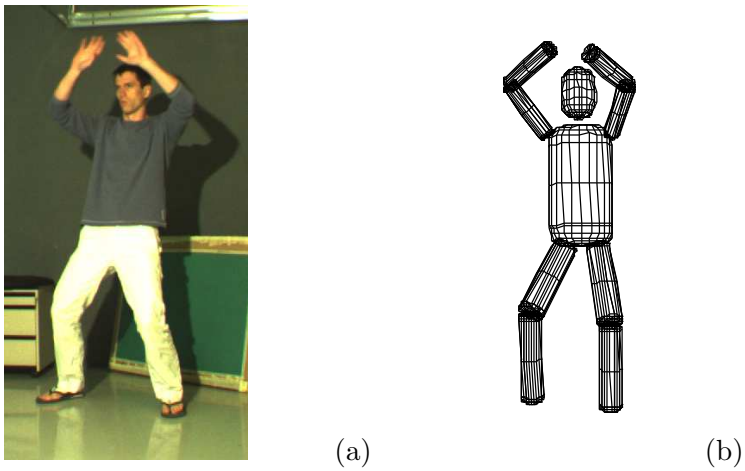
The procedure can also in some cases determine the position of the parts that are not initialized by the object detection procedure. Of course, if a part is completely hidden, it does not produce any points in the image, and the procedure fails. But if a part is not initialized by the object detection procedure because of an inconsistent match between scene and model part (poor fit to segmentation results), then there are some points in the image that belong to the part. Points that possibly belong to the part in question can generally be found in the proximity of the attaching joint, by using the part's most usual configuration (set with object model) and by using different, larger, distances for proximity and larger paraboloids. Parts with no initial configuration are processed after all the initialized parts, so they do not take the regions that probably correspond to those parts.

The pose improvement procedure works as follows. The parts of an object are first sorted according to the part's kinematic chain depth. This is a once-for-object step and

can be done when loading the model of an object. One begins with a central part i, computing the region, and fitting to it the part's superquadric model. When several steps of fitting are used, d_{prox_i} is reduced with each step. Then one descends the kinematic chain level by level, taking part i and producing its current superquadric model, taking into account the adjustments of the object (i.e. central part) position change and its parent parts joint adjustments. The region is then computed using the superquadric model and the three criteria above. The part superquadric model is fitted to the region by using just the rotation around the joint. Again, when several steps are used, d_{prox_i} is reduced with each step.

4.3 Experimental Results

The test object for the experiments in initialization of object tracking was a person, mainly for being articulated and not perfectly modelable with superquadric parts. Additional quality is ease of animation. For the articulated object tracking application moving objects with changing pose are welcome, and animation is much easier with a living person compared to puppets or similar objects. Fig. 4.1 shows a person that is acting in subsequent initialization and tracking samples.



(a) (b)

Fig. 4.1: (a) The person acting in subsequent tracking sequences, and (b) structured superquadric part models.

4.3.1 Setup

The experimental setup consisted of a Point Grey Research's Bumblebee stereo camera, which is factory calibrated with internally stored calibration parameters. The camera was connected to a 1,8GHz AMD Athlon PC. Images in a resolution of 1024×768 pixels were captured and processed using Point Grey Research's Triclops SDK in order to produce the depth information for the rectified frames in a resolution of 320×240 pixels. This processing was done online and took about 25ms for each frame.

The images and processed results were stored and processed offline. Initial frames for the sequences were processed with the SEGMENTOR, taking from 45s to 300s on the

designated PC, depending on number of extracted 3D points and size and number of largest recovered region(s). Segmentation parameters (maximum error of a description and maximum point distance for region growth) were defined by performing segmentation on 3 different scenes, using ten different equidistant values for each parameter from their respective speculated valid intervals. The resulting 300 segmentations were visually inspected and qualitatively evaluated. Since the estimated most quality segmentations appeared on all 3 scenes at almost the same parameter value pair, there was no need for segmentation of further scenes, and those values were used in all subsequent segmentations.

4.3.2 Object Model

Model of a person was structurally the same as the one for the toy figurine in the previous chapter (see Fig. 3.5). The size and joint position parameters were assigned approximate measured values from a person's respective body part. Tab. 4.1 shows model parameters used for the person from Fig. 4.1.

No.	*Part*	a_1	a_2	a_3	ϵ_1	ϵ_2	Volume
0	head	9	9	13	0.7	1.0	5241
1	torso	16	10	30	0.3	0.9	29610
2, 4	upper arm_x	4	4	16	0.1	1.0	1596
3, 5	thigh_x	6	6	20	0.3	1.0	4490
6, 7	forearm_x	4	4	16	0.1	1.0	1596
8, 9	shank_x	5	5	20	0.3	1.0	3118
Joint positions							
$\mathbf{r_{10}} = [0,0,35]^T, \mathbf{r_{01}} = [0,0,13]^T, \mathbf{r_{12}} = [16,0,26]^T, \mathbf{r_{21}} = [0,0,16]^T,$ $\mathbf{r_{13}} = [8,0,-33]^T, \mathbf{r_{31}} = [0,0,20]^T, \mathbf{r_{14}} = [-16,0,26]^T, \mathbf{r_{41}} = [0,0,16]^T,$ $\mathbf{r_{15}} = [-8,0,-33]^T, \mathbf{r_{51}} = [0,0,20]^T, \mathbf{r_{26}} = \mathbf{r_{47}} = [0,0,-16]^T,$ $\mathbf{r_{62}} = \mathbf{r_{74}} = [0,0,16]^T, \mathbf{r_{38}} = \mathbf{r_{59}} = [0,0,-16]^T, \mathbf{r_{83}} = \mathbf{r_{95}} == [0,0,16]^T$ $_{x = \{1,2\}}$							

Tab. 4.1: Model parameters for the person from Fig. 4.1.

Again, similarly to the figurine model from the previous chapter, the same constraints were used with the respective parts (listed in Tab. 4.2), as well as verification parameters (listed in Tab. 4.3). Parameter values were defined based on thirty reconstructions of superquadrics of different scenes containing the person.

A position improvement was only engaged for a single step. The parameters were set to encompass a relatively small proximity of the part models.

4.3.3 Results

Fig. 4.2 to Fig. 4.7 present results of the improved pose estimation for object tracking initialization. Each figure shows a reference scene image, disparity image, regions of respective segmented range image descriptions, superquadric models of respective segmented range image descriptions labeled (a) to (d), respectively. The latter two are produced by the SEGMENTOR. Further, object hypothesis and detected object are shown, labeled (e)

No.	*Part*	*Constraints*	*Constraint Values*
0	head	$\mathcal{T}_0 = \{\mathcal{V}_0, \mathcal{S}_{0,1}, \mathcal{S}_{0,2}, \mathcal{S}_{0,3}, \mathcal{H}_{0,1}, \mathcal{H}_{0,2}\ \}$	$V_0^{min} = 1000$, $V_0^{max} = 6000$, $a_{0,1}^{min} = 6.5$, $a_{0,1}^{max} = 15$, $a_{0,2}^{min} = 5$, $a_{0,2}^{max} = 11$, $a_{0,3}^{min} = 3$, $a_{0,3}^{max} = 9$, $\epsilon_{i,1}^{min} = 0.5$, $\epsilon_{i,1}^{max} = 1.4$, $\epsilon_{i,2}^{min} = 0.4$, $\epsilon_{i,2}^{max} = 1.5$
1	torso	$\mathcal{T}_1 = \{\mathcal{V}'_1, \mathcal{D}_{1,0}\}$	$V_1^{min} = 3500$, $V_1^{max} = 14000$, $V_1'^{min} = 5500$, $V_1'^{max} = 20000$, $S_1 = 12$, $d_{1,0}^{min} = 16, d_{1,0}^{max} = 27$
2,4	upper arm	$\mathcal{T}_{(2\|4)} = \{\mathcal{V}_{(2\|4)}, \mathcal{D}_{(2\|4),1}\}$	$V_{(2\|4)}^{min} = 300$, $V_{(2\|4)}^{max} = 6500$ $d_{(2\|4),1}^{min} = 14, d_{(2\|4),1}^{max} = 31$
3,5	thigh	$\mathcal{T}_{(3\|5)} = \{\mathcal{V}_{(3\|5)}, \mathcal{D}_{(3\|5),1}\}$	$V_{(3\|5)}^{min} = 500$, $V_{(3\|5)}^{max} = 8000$ $d_{(3\|5),1}^{min} = 20, d_{(3\|5),1}^{max} = 34$
6,7	forearm	$\mathcal{T}_{(6\|7)} = \{\mathcal{V}_{(6\|7)}, \mathcal{D}_{(6\|7),(2\|4)}\}$	$V_{(6\|7)}^{min} = 300$, $V_{(6\|7)}^{max} = 3500$ $d_{(6\|7),(2\|4)}^{min} = 7, d_{(6\|7),(2\|4)}^{max} = 23$
8,9	shank	$\mathcal{T}_{(8\|9)} = \{\mathcal{V}_{(8\|9)}, \mathcal{D}_{(8\|9),(3\|5)}\}$	$V_{(8\|9)}^{min} = 300$, $V_{(8\|9)}^{max} = 4000$ $d_{(8\|9),(3\|5)}^{min} = 9, d_{(8\|9),(3\|5)}^{max} = 38$

Tab. 4.2: Match consistency test values for the person from Fig. 4.1.

Meaning	*Variable*	*Value*
Interpretation size threshold	P	6
Maximum pole rotation thresholds	D_1	4
	D_2, D_3, D_4, D_5	13
	D_6, D_7	8
	D_8, D_9	8

Tab. 4.3: Values of interpretation verification parameters used in the experiments.

and (f), respectively, superimposed over the reference scene image. These two are produced by the first step of the initialization procedure, the object detection. The last couple of images in the figures, labeled (g) and (h) show the final result of the initialization procedure, the object superimposed over the reference scene image, and a 90° view of same, from the viewers right, respectively.

In the example initialization A in Fig. 4.2(f) one can observe slightly erroneous configuration of the torso, upper arms and right lower arm parts, after the object detection phase, while the image (g) shows improvement of the whole object configuration. Similar can be stated for examples B to D from Fig. 4.3 to Fig. 4.5, respectively. Example B in Fig. 4.3 depicts an interesting interpretation of the scene parts. Since the implemented system does not constrain joint rotations, lower legs are pointing forward in a way that is impossible for a human to take. Because the interpretation induced a part configuration which assumed a strong knee rotation, the lower legs were appointed to the region of

points close to the knee, and fitting rotated the parts as to fit the flatter upper part to the region. A point to notice is the fact that the whole superquadric is used in a part model, even with parts that are connected in a way that there is significant overlap of space that both parts occupy. We believe that refinement of the model that would not be using the part of the superquadric model that cannot represent object surface at all, combined with additional joint constraint checking would greatly prevent such configurations. Example C in Fig. 4.3 shows that while the model fits reasonably well, the configuration lacks precision in the person's left arm. Due to great initial error, the fitting process could not satisfy the whole shift necessary, but instead fitted the bottom part of the lower arm's model to the lower arm's top region. Example D in Fig. 4.3 displays a good improvement of slight part configuration errors arising from object detection method.

The two example initializations from Fig. 4.6 and Fig. 4.7 show erroneous part configurations. Example E in Fig. 4.6 produced a verified interpretation, but one based on rather poor part matches (see Fig. 4.6(e)). The stereo disparity data (Fig. 4.6(b)) is rather poor in the region of torso, producing a barely consistent match for that part. This in turn provided for non-quality position estimation in Fig. 4.6(f), which was not substantially improved (Fig. 4.6(g,h)). Example F in Fig. 4.7 shows a similar case, with the difference that the improvement step did recover from great initial pose estimation error. For additional improvement (left arm, right lower leg) similar notes as with example B would have to be taken into account. Although one would wish otherwise, it has to be noted, however, that improvement of pose of such magnitude were very rare, and were mostly exception to the rule.

Further, Fig. 4.8 shows two cases where object was not detected. In these cases there was no verified interpretation, the reason being mainly missing disparity data from the torso region, which SEGMENTOR was not able to overcome.

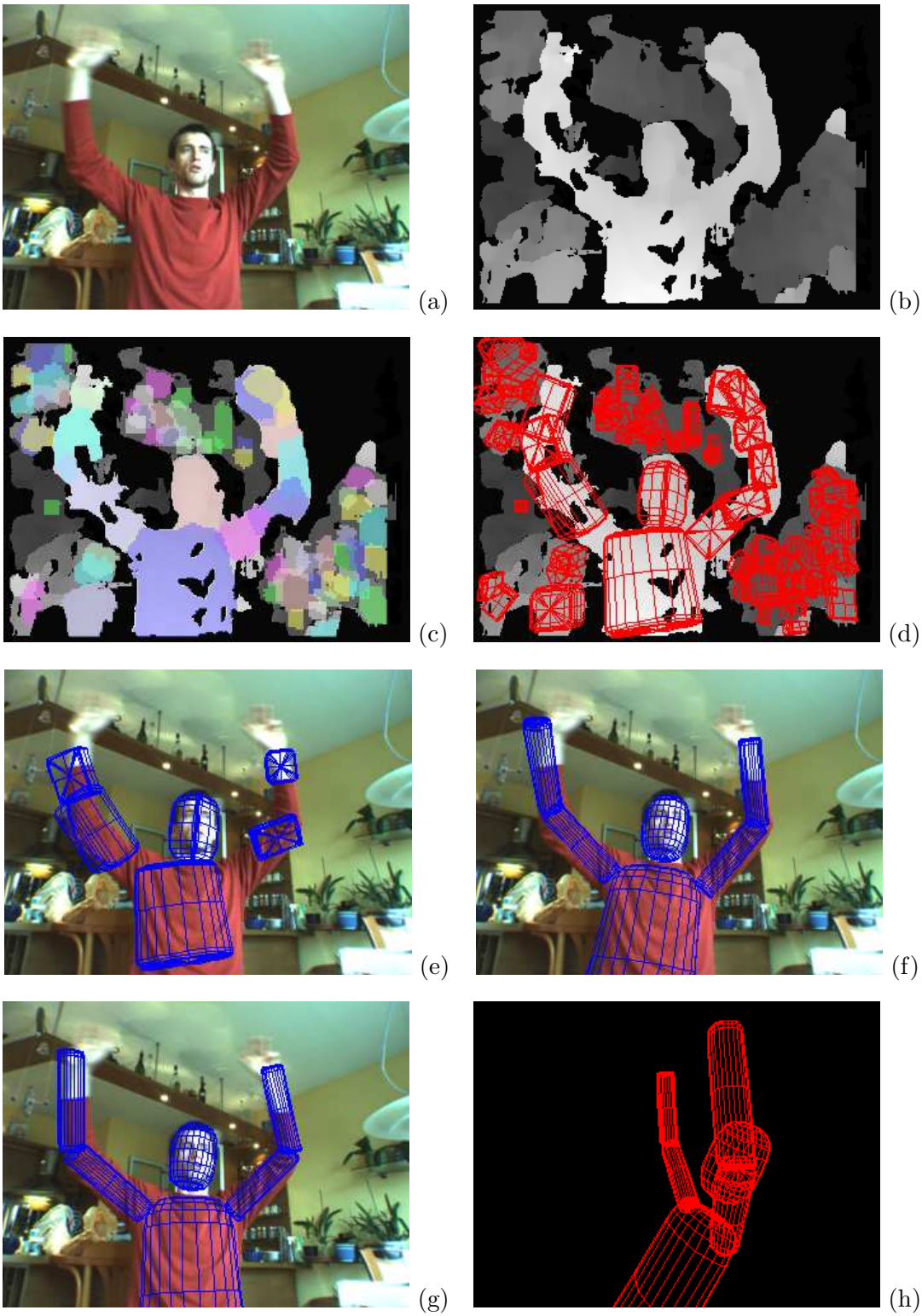


Fig. 4.2: Initialization of object position, example A. (a) scene, (b) range (disparity) image, (c) segmented range image regions, (d) segmented superquadrics, (e) object hypothesis, (f) detected object, (g) improved object position, and (h) 90° side view from the viewers right.

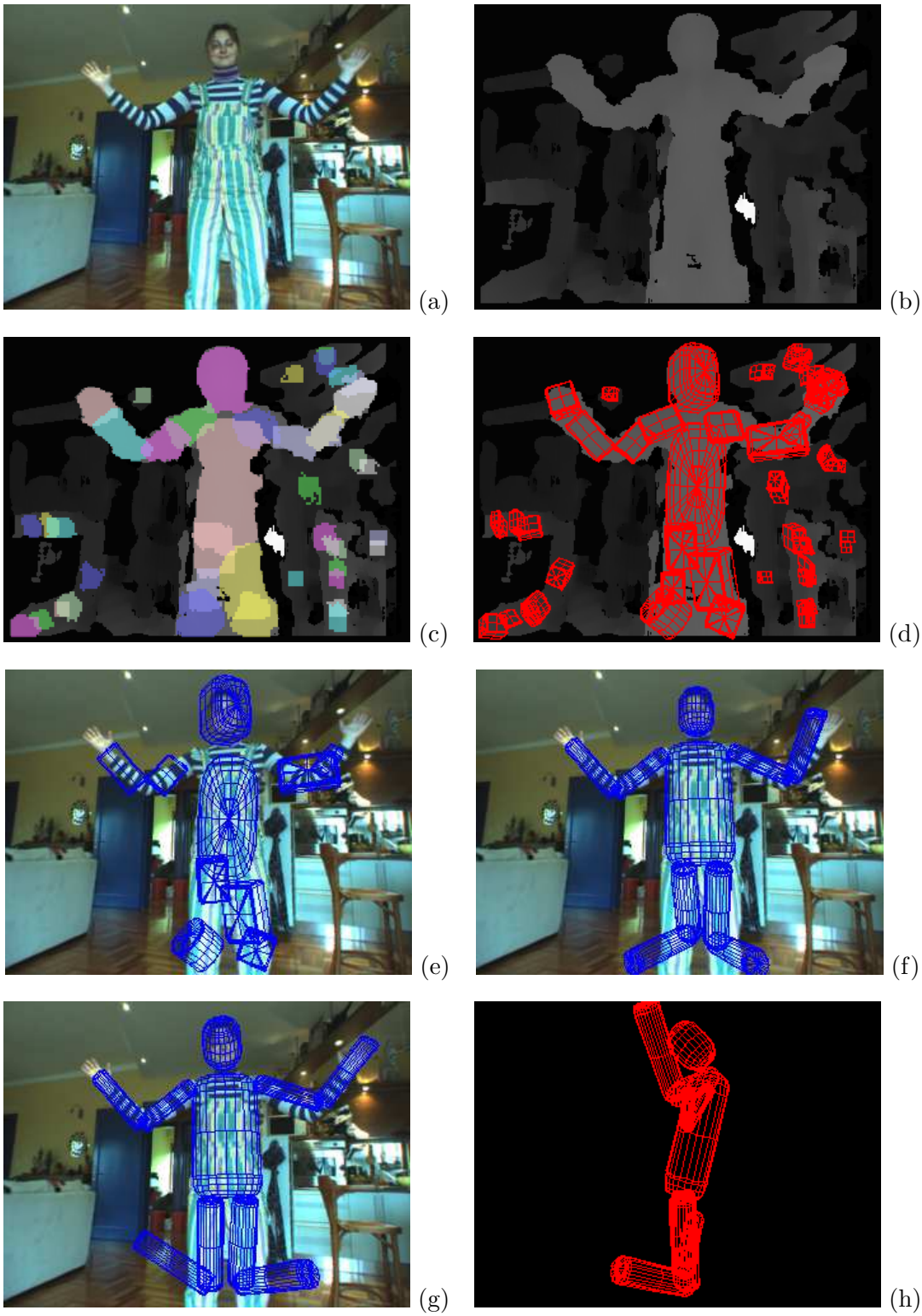


Fig. 4.3: Initialization of object position, example B. (a) scene, (b) range (disparity) image, (c) segmented range image regions, (d) segmented superquadrics, (e) object hypothesis, (f) detected object, (g) improved object position, and (h) 90° side view from viewers right.

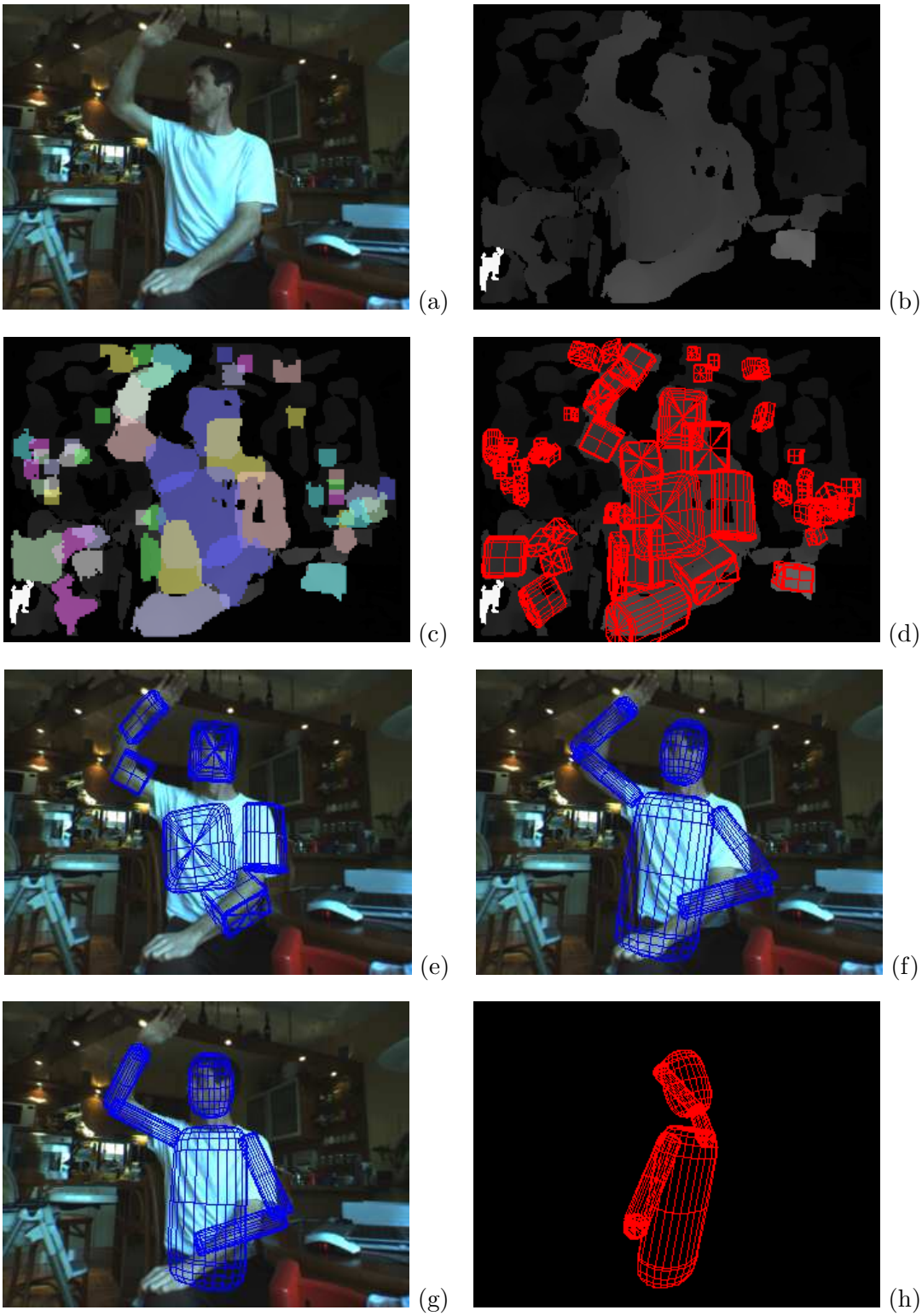


Fig. 4.4: Initialization of object position, example C. (a) scene, (b) range (disparity) image, (c) segmented range image regions, (d) segmented superquadrics, (e) object hypothesis, (f) detected object, (g) improved object position, and (h) 90° side view from viewers right.

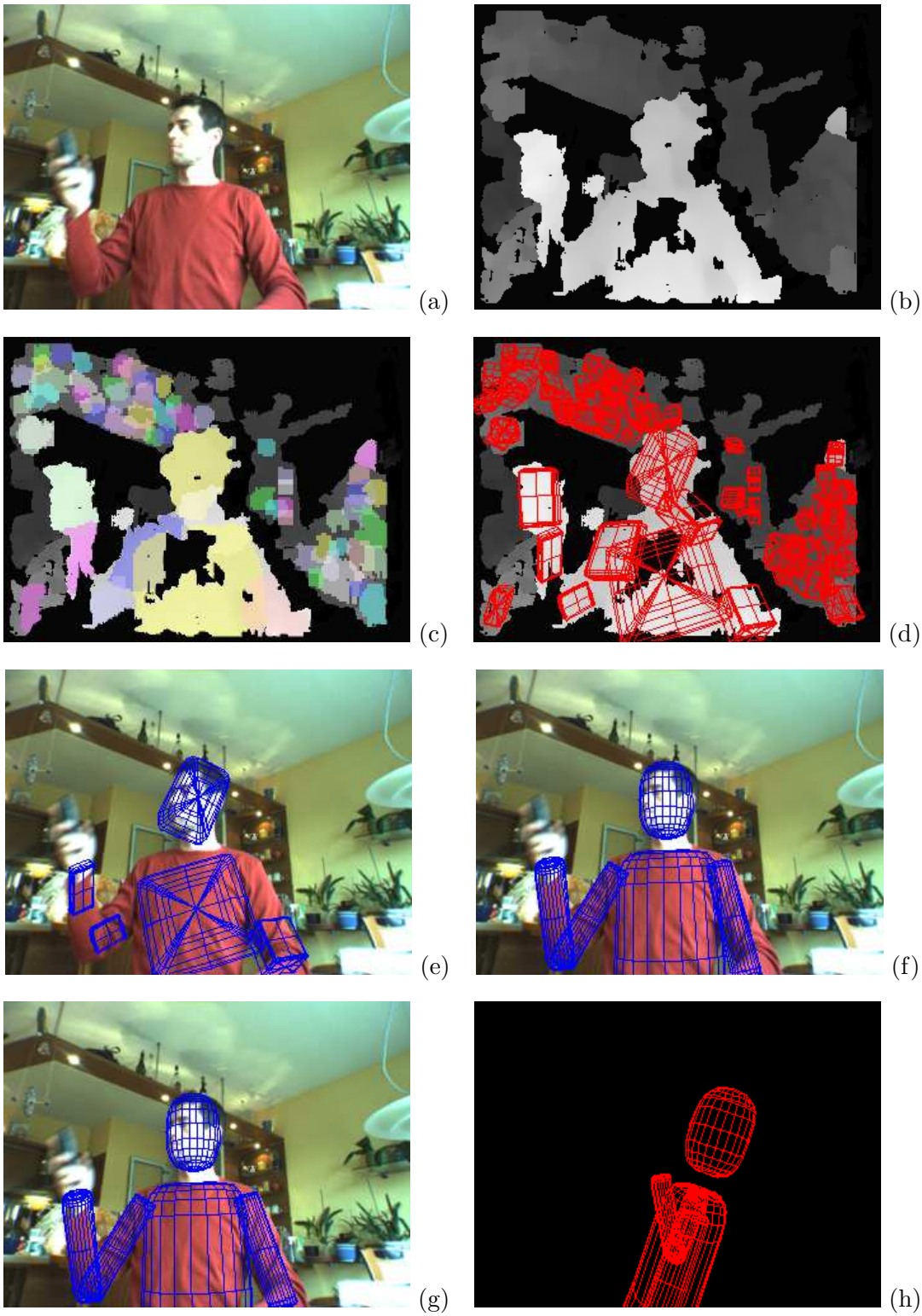


Fig. 4.5: Initialization of object position, example D. (a) scene, (b) range (disparity) image, (c) segmented range image regions, (d) segmented superquadrics, (e) object hypothesis, (f) detected object, (g) improved object position, and (h) 90° side view from viewers right.

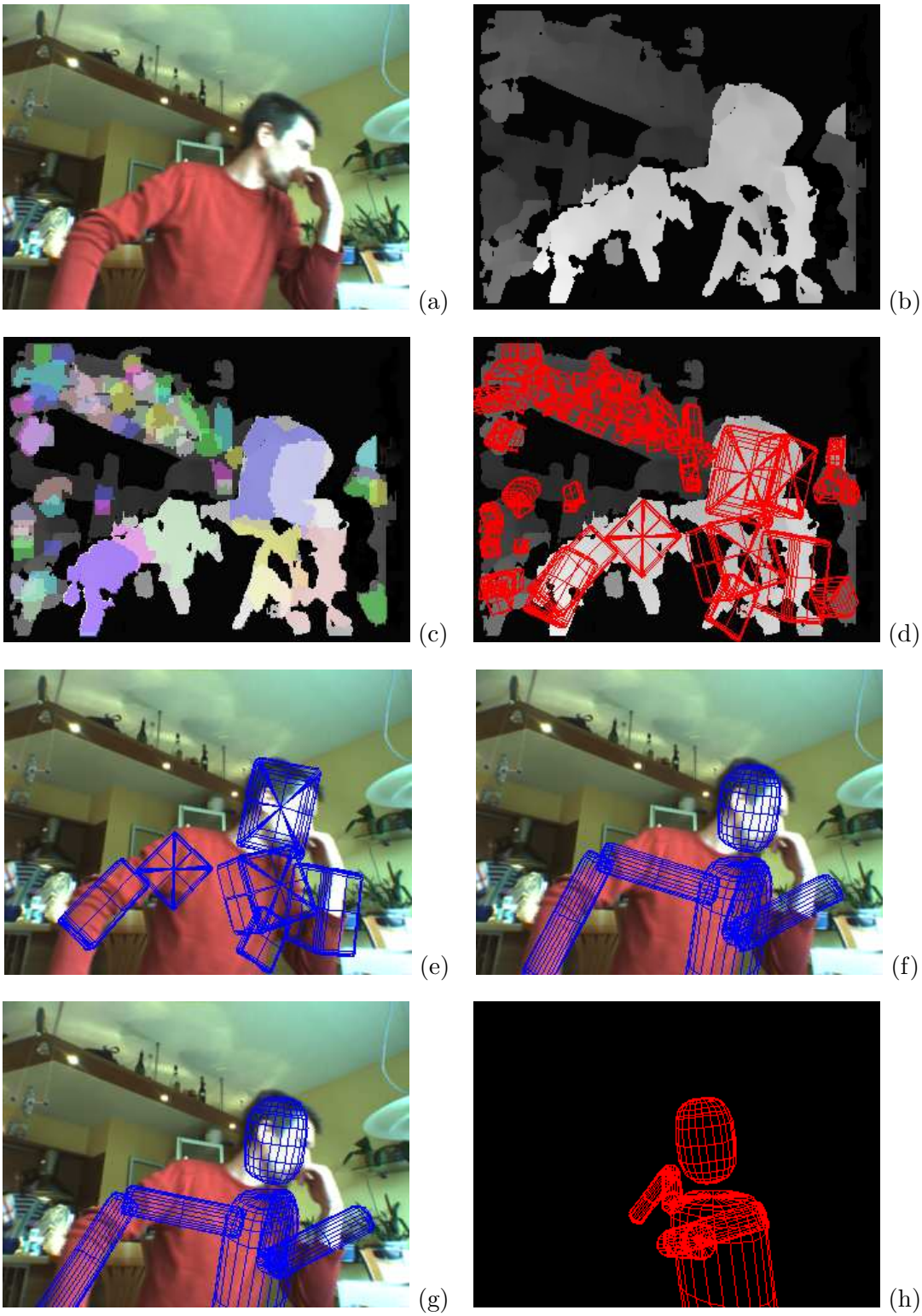


Fig. 4.6: Erroneous initialization of object position, example E. (a) scene, (b) range (disparity) image, (c) segmented range image regions, (d) segmented superquadrics, (e) object hypothesis, (f) detected object, (g) improved object position, and (h) 90° side view from viewers right.

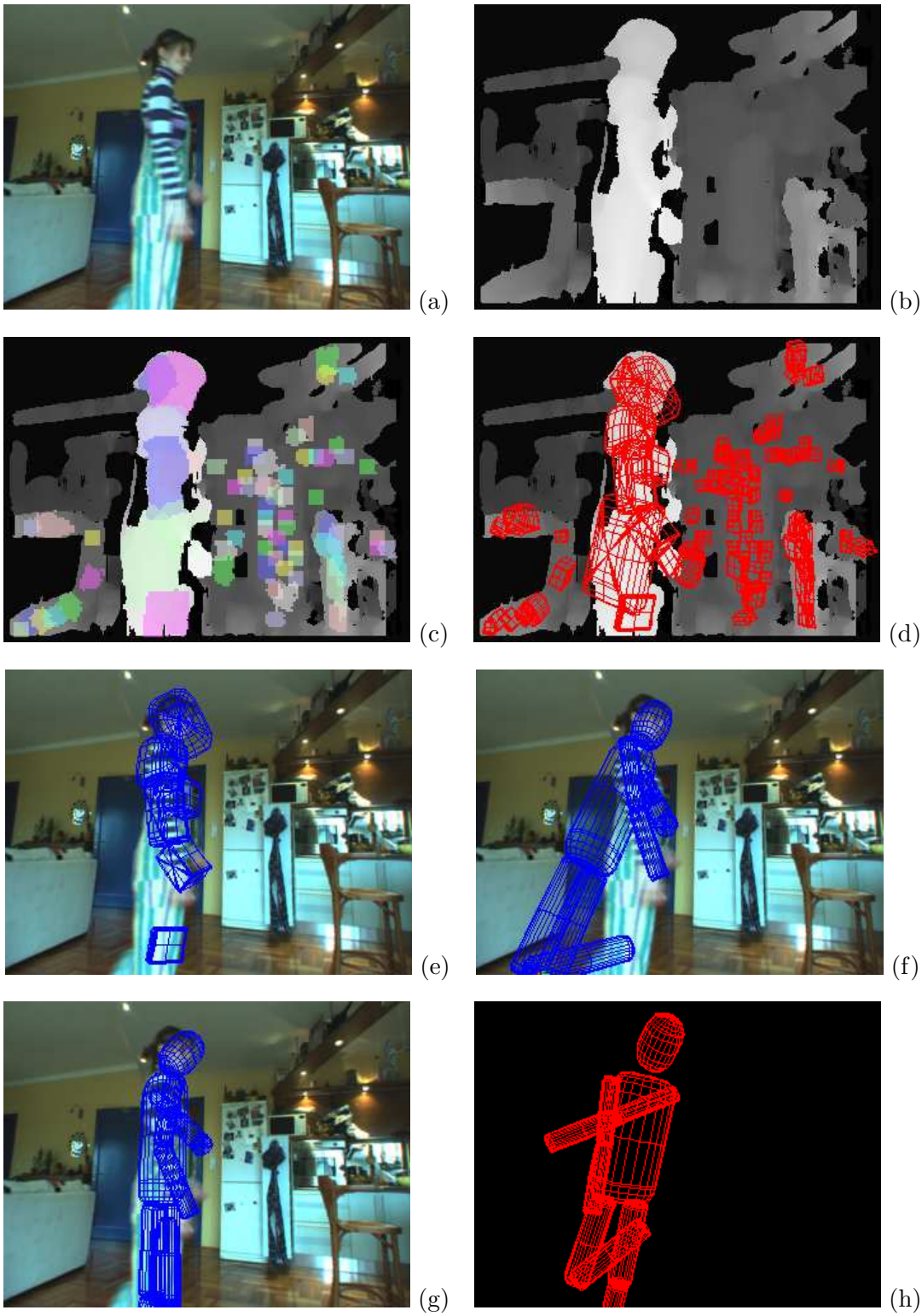


Fig. 4.7: Erroneous initialization of object position, example F. (a) scene, (b) range (disparity) image, (c) segmented range image regions, (d) segmented superquadrics, (e) object hypothesis, (f) detected object, (g) improved object position, and (h) 90° side view from viewers right.

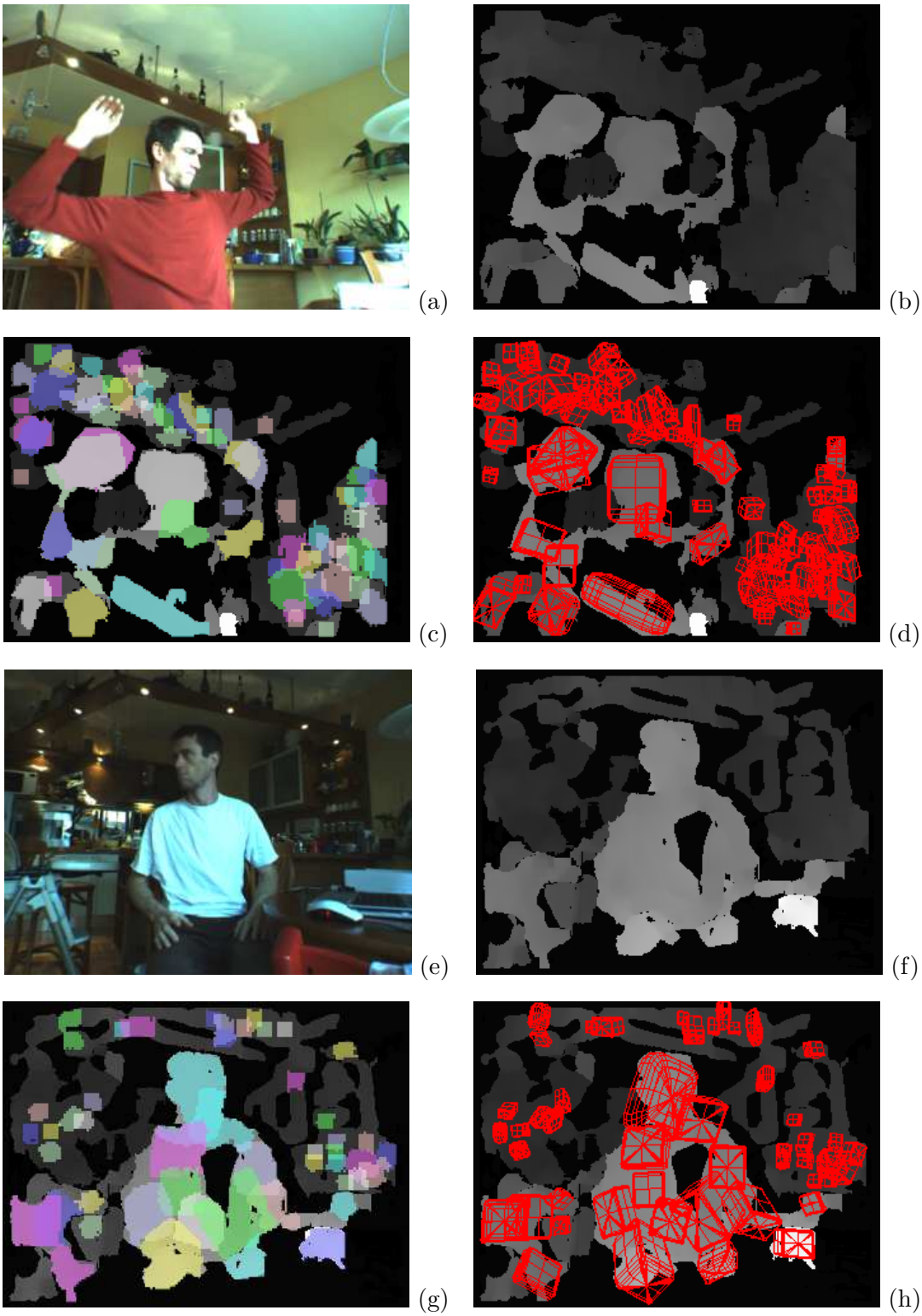


Fig. 4.8: Failed initialization of object position, examples G (a)-(d) and H (e) - (f). (a,e) scene, (b,f) range (disparity) image, (c,g) segmented range image regions, (d,h) segmented superquadrics. The object was not detected.

4.4 Chapter Summary

This chapter presented the application of the object recognition scheme proposed in Chapter 3 to the problem of 3D object tracking initialization. Using depth information extracted from a stereo camera pair, the initial images are segmented with superquadric descriptions and object's presence and location is determined by the means of interpretation search and verification. The object pose is further improved by fitting superquadric part models to the 3D data. Results were presented that show match between the detected and positioned object on one hand and the 3D stereo depth data on the other is of reasonable quality and surely good enough to be used as initial estimate for object tracking. The method somewhat suffers from instability of segmentation for the objects that can not be modeled well with superquadrics. The quality of the model as well as of stereo camera produced 3D data also contribute to reducing stability.

Chapter 5

3D Object Tracking Using Superquadric models

In this chapter the method for object tracking is discussed. The method is basically the same as the object position and improvement step from previous chapter, as it uses object position and configuration from previous frame to ease the search for part-compatible points in the range image, which are further used for fitting the object part model. Problems resulting from using only depth information are discussed next, and possible solutions are proposed.

5.1 Movement Estimation

When tracking an object in an image sequence, the tracking procedure should be able to compute the object position in the image from time t, based on the object position on time $t-1$. The hypothesis that object position and part configuration changes can be determined by fitting the object part models to the appropriate regions of range data was somewhat verified in Section 4.2. It describes the procedure for object position improvement, which can correct relatively small errors in object position and part configuration. Since part movements from frame to frame should be relatively small, it can also be used for the object movement estimation. The main problem is determination of appropriate 3D region for fitting. In the procedure, the parameters d_{prox_i} signify the largest movement of a part from frame to frame, that the procedure can compensate for. Similarly, the paraboloid parameters signify the maximum rotation around the attaching joint. Once the region is known, the part models are fitted by only allowing general translation and rotation for the central part model or rotation around attaching joint for the attached part models. The models are fitted in the order of depth in the kinematic chain.

Tracking begins with object initialization described in Chapter 4 in the first frame. The object model is positioned and configured according to the object detected in the first step and with improved position and configuration in the second step. If the object is not detected, the next frame is processed, until the object is detected. Once the object position is known and its part configuration refined, the movements from frame to frame are computed by determining the parts regions and fitting the part models to the regions.

Fig. 5.1 shows an example tracking of a person waving his hands. The sequence is

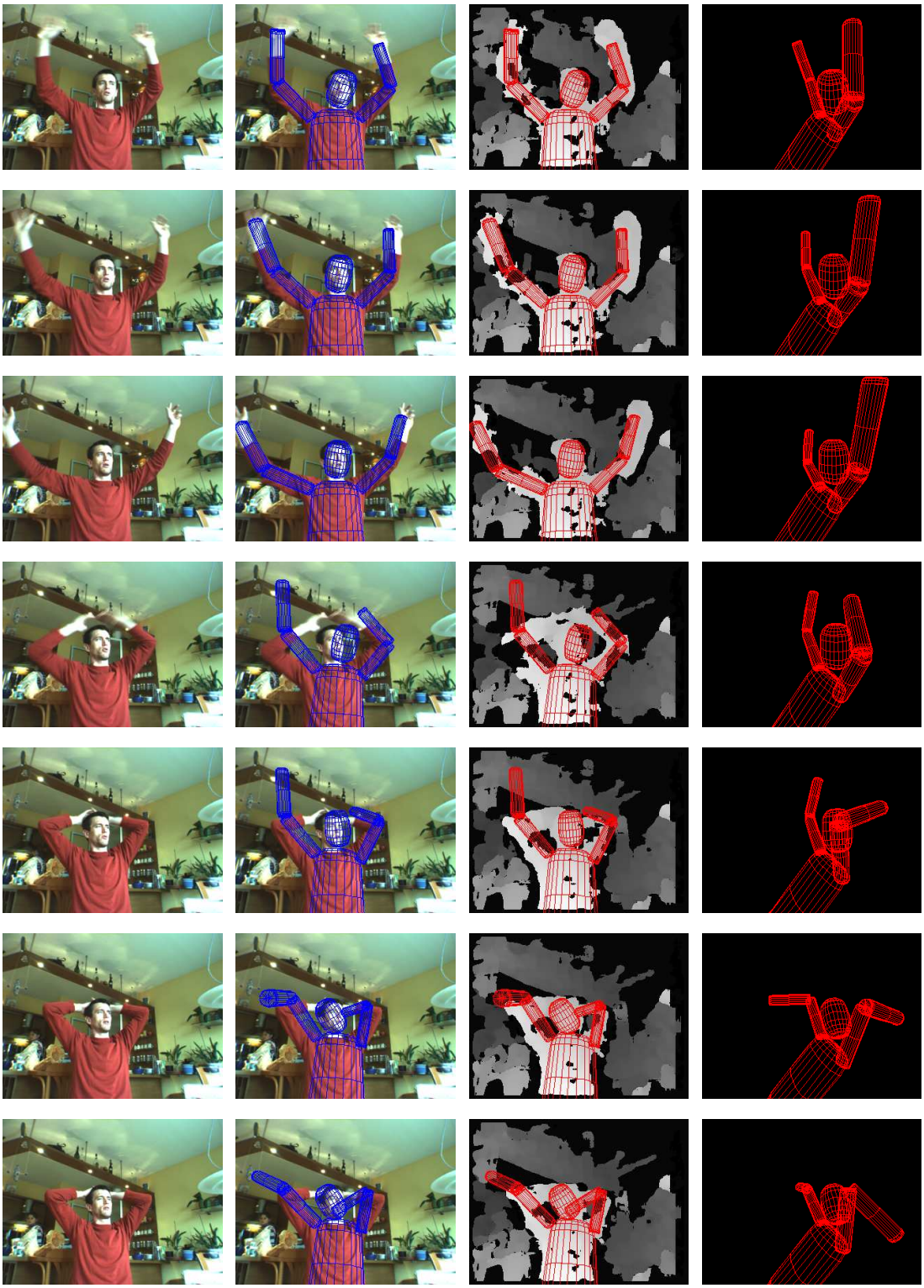


Fig. 5.1: Tracking articulated motion. Frames sequent from top to bottom, each frame in a row with columns depicting reference image, object model overlaid on reference image, object model overlaid on disparity image, and side view of object model.

shown with a frame in each row, with the first frame on top. Each frame is displayed as a reference image in the leftmost column, followed to the right by the object model superimposed over the reference image and over the disparity image, and by a side view from the viewers right. The object was initialized on the first frame. Each next frame was processed by taking object model position from previous frame and estimating movement of parts in three steps. The first three frames show satisfactory tracking in of the arms, and a bit lower quality of fit of the head. In the fourth frame, however, a considerable error in the configuration is made, especially for the right arm and the lower left arm. Also notice that the head is tilted even further to the right and slightly in front, which is the exact opposite of the movement made by the person. Although one could argue that there is significant movement of parts from third to fourth frame, and although in the next three at least the head configuration somewhat recovers, this error shows a serious fault that hinders the movement estimation method.

Let us analyze the causes for such behavior. Fig. 5.2 presents the inner workings for the case of estimating movement for the fourth frame of sequence from Fig. 5.1. Fig. 5.2(a) and (b) show the initial model over the reference and the disparity image, respectively. Fig. 5.2(c) depicts initial part models laid over the regions used in the first step of the fitting. Note that the individual parts are positioned according to their already fitted parent parts (that is why the object appears disconnected). Fig. 5.2(d) depicts the configuration resulting from the first step of fitting. Similarly, Fig. 5.2(e,f) and (g,h) show steps two and three, respectively. The main cause of the error is the initial movement of the torso model (compare Fig. 5.2(c) and (d)). The fitting moves the torso superquadric up and right, and rotates it a bit (note the head in Fig. 5.2(c) as compared to (b)). This causes the regions for fitting the parts attached to the torso to be out of place, since the joint locations for those parts are offset greatly. Note that in subsequent steps the torso stops moving. Some of the error can also be labeled pose-specific, such as the head taking the whole of the region of the right lower arm, causing the lower arm to lose its orientation.

The reason for this error lies in the fact, that the method computes the object translation and rotation by fitting a single, central, part to its hypothetic region. Even if the central part of and object was of such shape, that it would allow for unambiguous fitting (superquadrics are just the opposite, being very symmetrical), there would still be problems regarding this approach.

1. The region can in general only be determined to some degree. Several points not belonging to the part are likely to be included in the region, and some points that belong to the part can be missing from the region. The degree of erroneously included or excluded points affects the superquadric fitting process and influences the quality of its results.

2. For the fitting to produce fine results one would need the object model to be perfect. Since this is rarely the case (even more so with superquadric part models of naturally shaped objects), the effects of the non-reliable region selection from the point above are further amplified. The region 3D points are selected based on the part model, therefore the region itself is only adequate to some degree. And further, since the model only matches the true object part, stability of the superquadric fitting procedure is decreased.

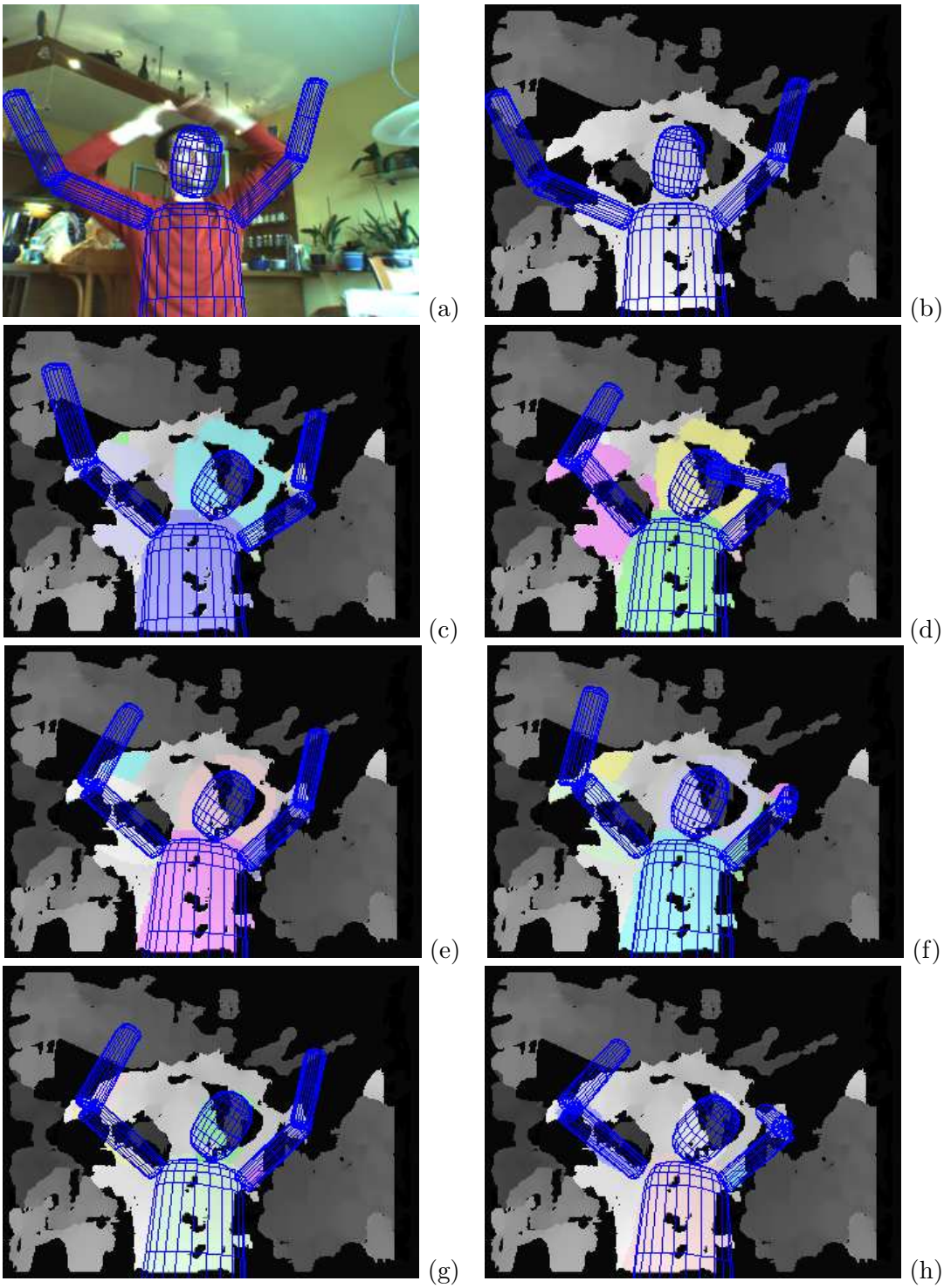


Fig. 5.2: Three steps for movement estimation for fourth frame from Fig. 5.1. (a) reference frame with initial model superimposed, (b) disparity image with initial model superimposed, (c,e,g) initial part models laid over regions (shown colored) in fitting steps 1,2 and 3, respectively, and (d,f,h) part models after fitting (step 1,2 and 3, respectively) laid over regions.

3. The two problems from above are even further magnified due to the fact that the real input 3D data contains noisy measurements.

It can be assumed that for a specific application of the method, the input 3D data noise is reduced to the minimum (e.g. by fine tuning the stereo processing parameters), and that the object is modeled as truly as possible (e.g. by some evaluation of the model with acquired 3D data). Since some discrepancy between the object and its model will still exist and noise will not be completely removed, at least when dealing with naturally shaped objects and real 3D data, the fitting regions will remain inaccurate. Improvement should be sought in using more parts to determine object position.

The idea that we implemented was that the parts at the first level of kinematic chain would be fitted in a way that would alow them not only to rotate around the joint but also to move the joint position. From the joint offset we speculated more precise object position (and primarily orientation) could be extracted. First, the central part is fitted in the same way as above. Next, the parts attached directly to the central part are fitted to the same region as above, with the only difference that minimization is done for position (arbitrary translation and rotation), not the rotation around the attaching joint. The new positions of the attached parts produce hypothesized joint locations, denoted $\mathbf{r'_i}$, relative to each part i. The new position of the central part produces a starting point for minimizing the distances between the hypothetical joint locations for attached parts relative to the central part (denoted by $\mathbf{c_i}$ in local frame of the central part). The error function is

$$E_{pos}(\mathbf{T}) = \sum_i |\mathbf{T}\mathbf{r'_i} - \mathbf{c_i}|, \tag{5.1}$$

where $\mathbf{T}$ is homogeneous matrix, that transforms a global frame into a central part's local frame. After the central part is fitted to hypothesized joints, all of the remaining parts are fitted by the same procedure as above. Note that this implies that the parts from the first level in the kinematic chain are fitted twice, but not (necessarily) to the same region.

Standard Levenberg-Marquardt minimization was used for the minimization of E_{pos}, as we had it readily implemented, although simpler methods could be probably used. The method produced fine results on synthetic data. Fig. 5.3 presents improvements made by the additional step for the case of fourth frame from Fig. 5.1. Fig. 5.3(a) shows initial model superimposed over the reference image. Fig. 5.3(b) depicts the fitted central and first level part models laid over the regions used, in the first step of the joint fitting. Fig. 5.3(c) depicts the configuration resulting from the first step of fitting the remaining parts. Similarly, Fig. 5.3(d,e) and (f,g) show steps two and three, respectively. Finally, Fig. 5.3(h) depicts final configuration achieved by this method. While some error has been reduced, mainly in the head and left arm configuration, the erroneous right arm configuration remains, and can be attributed mainly to missing data from the elbow region, which seems to mislead the model wrongly lift the arm.

This process could be taken one step further, by doing the same for the parts lower in the kinematic chain. Ideally, every single part of the object should influence its position. The object's kinematic chain as a whole could be fitted to the input data. Fitting could be done as a nonlinear minimization of an error function, which would evaluate the fit between the 3D data and object model in a way similar to fitting single superquadric parts. The function would be minimized for parameters of object position and joint rotations.

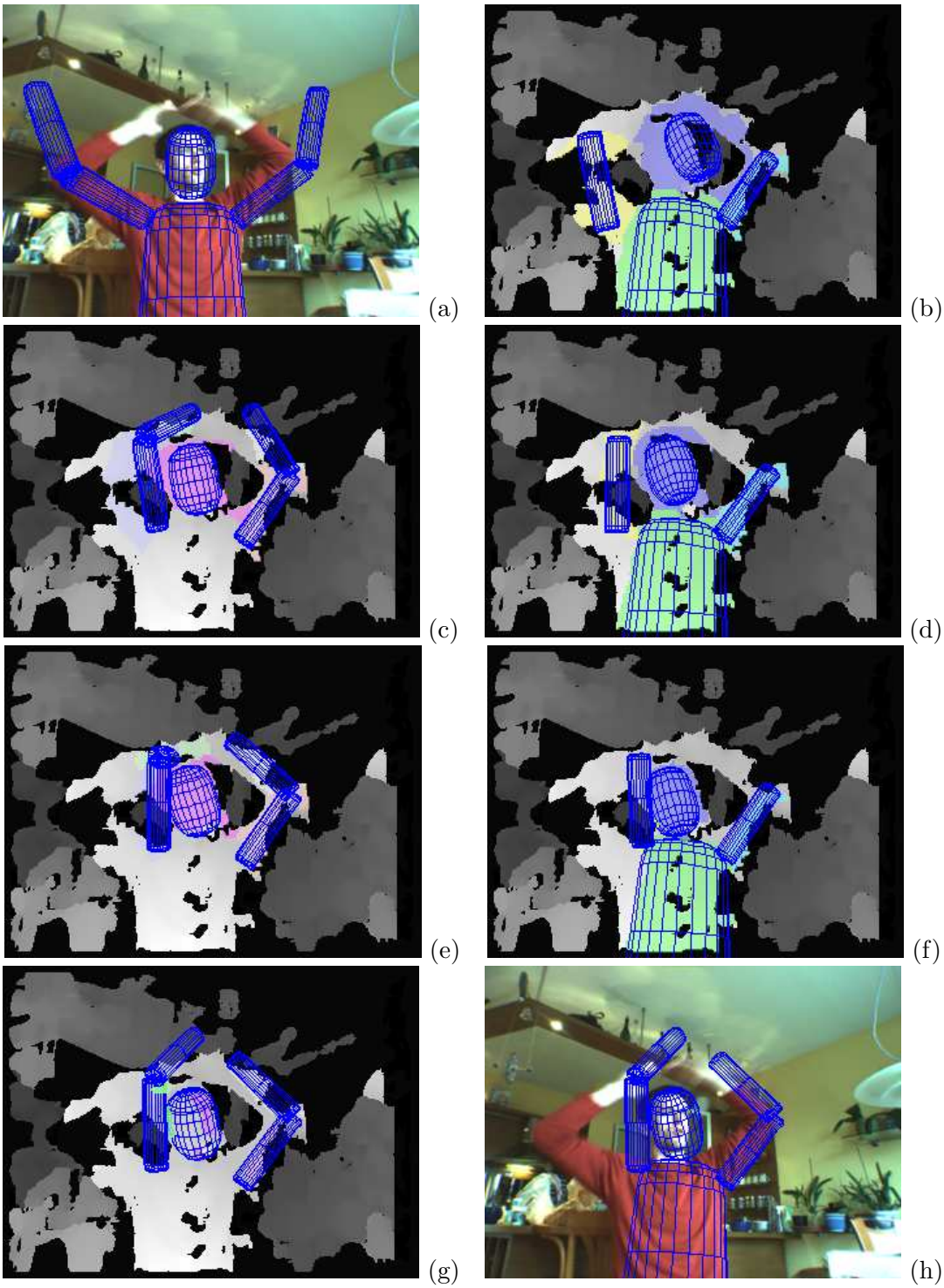


Fig. 5.3: Improved movement estimation for fourth frame from Fig. 5.1. (a) reference frame with initial model superimposed, (b,d,f) fitted central and first level part models, laid over regions (shown colored) in steps 1,2 and 3, respectively, (c,e,g) part models after fitting (step 1,2 and 3, respectively) laid over regions, and (h) final model configuration superimposed on reference image.

The first problem with such an approach would be in determining the point cloud, the 3D point region, to fit the object to. Some kind of a measure of outlying would have to be devised to get rid of the points not belonging to the object, but this would probably have to be made dynamic and integrated into the fitting procedure, in order not just to eliminate the outlying points, but also to include the good ones as the object is being fitted. Second problem would be in the error function itself, which is based on the distance of a point to the model and would have great many local minima. As can be seen in the results presented here, even a relatively straightforward fitting of a single superquadric part to the stereo 3D data of average quality is quite error prone. The lower the quality of object model and the input 3D data would be, the greater the number of local minima and less stable the method is.

The proposed method was not stable enough to produce a tracking of a person over a sequence of reasonable length, given the quality of the stereo data and object model.

5.2 Chapter Summary

This chapter presented the method for estimating object motion from frame to frame in a sequence of range images, by determining the regions that belong to object parts and then fitting the superquadric part models to the regions. The method suffers from poor object model quality and input data noise, which influence both region determination as well as part model fitting.

Chapter 6

Summary and Conclusions

In this work we first developed a method for articulated 3D object recognition from range images, based on superquadric part models as a variant of model based matching. The method makes use of superquadric scene descriptions provided by processing the input range images with the SEGMENTOR. Interpretation tree search is employed for feasible object to scene part matches, leading to object instance detection. To facilitate the search, we proposed several types of part match constraints for general use with superquadric models. We also proposed the interpretation verification procedure and showed the efficiency of the scheme on experimental results.

The object tracking initialization that we proposed is based on the above mentioned object recognition scheme. The image sequences for our proposed tracking system are stereo image sequences, that are processed to extract depth information. Tracking initialization works by segmenting the initial range image with SEGMENTOR and detecting possible object instances with the proposed recognition method. Object positions acquired this way are further refined by fitting the part models directly to range data. The main problem lies in determining the regions that correspond to particular parts. Experimental results show some improvement on pose estimation.

The proposed method for estimating frame to frame movement is based on fitting the part superquadric models to the point regions, that are hypothesized from object configuration from previous frame. The inherent instability of the method, that is due to the fact that in general, a single part can not determine the object position, is addressed by including the parts from the first level of kinematic chain to the object position estimation procedure. The idea is that the parts from first level can determine joint positions more accurately, and this information can be used to improve on object position. Hence, the procedure is extended by fitting (to the same hypothesized point regions) the central and its attached parts for arbitrary position, giving the hypothesized joint positions. Those are used in a step, where distances between joint positions in the central part and their respective hypothesized locations are minimized. While there is some improvement to the method, it still does not work stably enough to produce quality tracking information of reasonable length. Main causes are estimated to be low quality of model and low quality of input stereo acquired 3D data. Both causes interact strongly to further diminish the practicability of the method.

One of the greatest problems with the methods proposed in this thesis, lies in their relative sensitivity to acquired 3D data. The proposed methods for object detection and

tracking initialization rely on the output of SEGMENTOR . Since SEGMENTOR was developed for input from capturing devices that are accurate (such as the structured light range scanner we used in the experiments in Chapter 3), it is sensible to to noisy and sparse 3D data. Fitting of the superquadric models, region growing and description selection, which constitute the core of SEGMENTOR , are all based on distances to superquadric surfaces, making them sensible to input noise and sparseness. Also, the position improvement and movement estimation steps of the proposed methods use superquadric fitting. The methods would perform better on improved input quality.

6.1 Contributions of the Dissertation

The contributions of the dissertation in the order of importance to the thesis, are as follows:

- Object recognition scheme for detecting and segmenting articulated 3D object from range data using superquadric models was proposed and experimentally verified. The scheme is build upon interpretation trees, and proposes part match constraints and interpretation verification suitable for general superquadric-built objects.
- A method for initialization of 3D object tracking from stereo image sequences based on the object recognition scheme with additional pose improvement step was proposed and experimentally verified.
- Pose estimation step for the case of superquadric models was proposed and experimentally tested.
- Superquadric models were used directly without any intermediate representations in the problem of articulated object tracking.
- Segmentation of stereo depth data with superquadric models was experimentally verified to a certain degree.

6.2 Future work

While we tackled important subproblems in the problem of 3D object tracking, namely object detection as a means to position estimation and system initialization, several subproblems remain to be solved.

The initialization and tracking methods still need to be tested on greater variety of objects and image sequences. Such testing would require extensive work in the areas of object modeling and capturing image sequences, which requires much more manpower than was available. Therefore the rigorous testing was beyond the scope of this thesis.

The proposed tracking method only makes use of the 3D data for movement estimation. For stabilizing the movement estimation, intensity information could be used additionally. The part configuration could be further constrained by fitting the contours to object edges. Another way of improving movement estimation would be an introduction of some elasticity to the object model. A joint attaching two parts could be allowed an offset for its two locations as they pertain to the two respective parts, as well as some attraction measure

that would force the two positions to overlap. In the description of our tracking method we did not deal with recovery from mistracking. In order to recover from mistracking, a procedure for assessing the quality of detected object would have to be designed.

The proposed methods also do not deal with pose feasibility. In this regard, joint constraints and self-penetration constraint could be implemented.

Part II

Doktorska disertacija

Chapter 7

Razširjen povzetek disertacije

7.1 Uvod

V raziskavah s področja računalniškega vida so bili uporabljeni mnogovrstni modeli za opis najrazličnejših lastnosti objektov in scen. En izmed načinov za predstavitev 3D modelov je opisovanje po delih, kjer model opisuje delom ustrezne entitete. Tako je za predstavitev strukturiranega modela potreben skupek modelov za dele. Taki opisi dostikrat niso dovolj izčrpni za predstavitev vseh detajlov potrebnih pri razpoznavi objekta.

Če želimo pridobiti opise delov neke scene, moramo sliko scene razdeliti na segmente, ki ustrezajo posameznim delom, kakor tudi pridobiti modele za vsak tak segment. V primeru, da se ti dve opravili ločita, segmentacija ne upošteva oblik, ki jih modeli za dele lahko zavzamejo. Problemu se lahko izognemo s kombiniranjem segmentacije in rekonstrukcije delov, tako da lahko slike segmentiramo le na dele, ki so neke pojavitve izbranih modelov. Hkratno segmentacijo in rekonstrukcijo oblik lahko dosežemo z uporabo paradigme *gradi-in-izberi*.

Eni priljubljenejših volumetričnih modelov so superkvadriki. To so matematična telesa, ki lahko s samo enajstimi parametri opisujejo standardne geometrijske oblike kot tudi vmesne oblike.

7.1.1 Postavitev problema

V disertaciji obravnavamo metode potrebne za modeliranje strukturiranih 3D objektov, segmentacijo njihovih delov iz 3D podatkov, in njihovega sledenja skozi sekvence slik. Najprej raziščemo zaznavanje 3D objektov. Deli objekta določajo strukturo, ki objekt ločuje od ostalih objektov. Pri nalogi zaznavanja objektov tako damo večjo težo ugotavljanju strukture objekta kot podobnosti samih delov. Nadalje raziščemo sledenje objektov v 3D prostoru, ki ga razdelimo v fazo inicializacije, kjer ugotavljamo prisotnost objekta in njegov položaj, in fazo premikanja, kjer uporabimo informacijo o položaju na prejšnjih slikah za učinkovito ugotavljanje premikov.

7.2 Superkvadriki in segmentacija superkvadrikov

Superkvadriki so družina volumetričnih modelov, katerih opisi so zelo kompaktni, saj za osnovne oblike v splošni legi potrebujemo le enajst parametrov, $\Lambda = \langle a_1, a_2, a_3$ [velikost],

ϵ_1, ϵ_2 [oblika], ϕ, θ, ψ [rotacija], p_x, p_y, p_z [translacija] $\rangle$. Poleg tega obstajajo analitično izvedena formule za lasnosti kot so površina, razdalje do točk, normale, volumen, vztrajnostne momente, itd.

SEGMENTOR je sistem za segmentacijo globinskih slik in gradnjo superkvadričnih modelov (Jaklič (1997)), ki uporablja paradigmo *gradi-in-izberi* (Leonardis (1996)). Kot pove že ime sta glavna dela paradigme gradnja modelov in izbira modelov, ki najbolje opisujejo 3D podatke. Gradnja parametričnih modelov je težka predvsem, ker mora hkrati rešiti dva problema, in sicer določiti množico točk, ki pripadajo modelu (segmentacija), kakor tudi določiti parametre modela. Parametre superkvadričnega modela, ki najbolje ustrezajo dani množici točk, lahko določimo z minimizacijo funkcije napake, ki poda prileganje modela množici točk. Metoda (Solina and Bajcsy (1990)) je praktično standardna metoda za gradnjo enega superkvadrika iz množice točk. Nasprotno pa, če poznamo parametre modela, lahko z tehnikami klasifikacije vzorcev določimo modelu ustrezajoče točke. Paradigma *gradi-in-izberi* rešuje oba problema hkrati z iterativno rastjo modelov, da pridobi ustrezne množice točk, in izbiro modelov, ki najbolje opisujejo celotno množico podatkov z uporabo principa *Minimum Description Length*. Na začetku se vhodna množica točk razdeli na majhne podmnožice, tako da se razkosa globinsko sliko z rešetom. Vsaki podmnožici se nato prilagodi model, množico pa lahko nato razširimo tako, da vključimo njej sosednje točke, ki so dovolj blizu modela. Modeli tako rastejo, rast pa nadzorujemo z parametri za vključevanje novih točk in kontrolo napake prileganja modela. Ker gradimo mnogo redundantnih modelov po vsej sliki, se na koncu modeli oz. regije, ki jih opisujejo, deloma ali v celoti prekrivajo. Z uporabo principa *Minimum Description Length*izberemo tako podmnožico vseh zgrajenih modelov, ki je najpreprostejša vendar zavzema kar največ točk in ima čim nižjo napako prileganja modelov.

7.3 Modeliranje in zaznavanje 3D objektov

Za predlagano metodo zaznavanja 3D objektov smo napravili precej preprost model objekta, ki določa objekt na dveh nivojih. Na prvem nivoju modeliramo dele objekta tako da jim določimo velikost in obliko. Na drugem nivoju pa model opisuje strukturo objekta, tako da določa mesta (sklepe), kjer se deli med sabo povezujejo. Model objekta vsebuje še parametre za položaj (translacijo in rotacijo) v prostoru, konfiguracijo pa določamo s parametri rotacije okoli posameznih sklepov. V tem delu se nismo ukvarjali z omejevanjem teh vrednosti na realno možne, ampak smo ločili samo trde in fleksibilne sklepe.

7.3.1 Ujemanje modelov

SEGMENTOR zgradi množico superkvadrikov, ki opisujejo prizor in jih imenujemo *deli prizora*. Ko imamo še superkvadričen model objekta, si lahko postopek prepoznavanja objekta predstavljamo kot iskanje ujemanja med deli prizora in deli modela. Vsa taka ujemanja postavljena v drevesno strukturo imenujemo *interpretacijsko drevo*. Vozlišča v tem drevesu predstavljajo ujemanje med delom prizora in delom modela. Neki poti od korena do lista drevesa pravimo interpretacija, ker razlaga (morebitni) pomen delov prizora v smislu objekta. Iskanje prave interpretacije začnemo v korenskem vozlišču, ki ga razširimo za vsa možna ujemanja za prvi del modela. Iz vsakega vozlišča se drevo razširi v vsa ujemanja za naslednji del modela, iskanje pa nadaljujemo v globino le, če

je ujemanje konsistentno, se pravi če sta dela podobna. V naši izvedbi smo uporabili preiskovanje najprej v globino. V realnih prizorih so lahko nekateri deli objekta skriti oz. zakriti. Da sistem lahko obravnava take primere dodamo delom prizora nepravi del, za katerega določimo, da se dobro ujema z vsemi deli objekta. Ko iskanje interpretacije doseže list drevesa, dobimo konsistentno interpretacijo. Vendar, ker so omejitve pri preverjanju ujemanja lokalnega značaja, ni nujno da je celotna interpretacija res smiselna. V splošnem ni nekega jamstva, da je najdena interpretacija globalno pravilno. Interpretacije moramo torej jemati zgolj kot hipoteze, ki jih moramo nekako globalno preveriti z postopkom, ki mu pravimo *preverjanje interpretacije*.

Primerjava delov

Če je torej objekt prisoten na nekem prizoru, potem bi moral biti zgrajen iz enakih delov kot njegov model. V resničnem svetu pa deli seveda niso povsem enaki, temveč le do neke mere podobni. Primerjava superkvadričnih delov mora torej biti tolerantna do manjših (ali večjih) razlik pri obliki in velikosti. Za primerjavo dveh superkvadrikov ne moremo uporabiti direktno parametrov superkvadrikov, temveč primerjavo izvedemo kot množico omejitev, katere določimo posameznim delom. Predlagamo več vrst omejitev. Osnovna omejitev, ki jo lahko vključimo v množico omejitev vsakega dela je *omejitev prostornine*, ki enostavno določa kolikšna je lahko prostornina dela prizora, da se ta še ujema z delom modela. Omejitev prostornine lahko še razširimo s tem da upoštevamo prostornino morebitnih prekrivajočih delov prizora. Nadalje lahko za dele z zanesljivejšo rekonstrukcijo uporabimo *omejitev velikosti* in pa *omejitev oblike*, ki ju definiramo kot interval veljavnosti za parameter velikosti oz. oblike ki ustreza osi z najmanjšim vztrajnostnim momentom. Uporabimo lahko tudi *omejitev razdalje*, pri kateri določimo koliko je lahko del oddaljen od nekega dela, ki je že vključen v interpretacijo. Namen omejitev ujemanja je rezanje interpretacijskega drevesa, kar pomeni hitrejšo pot do interpretacije. Omejitve morajo biti nastavljene na način, ki proceduri primerjanja delov omogoča da zavrne čim več neprimernih delov, pri čemer pa mora sprejeti vse mogoče rekonstrukcije. Sistem tako ne spregleda nobenega objekta in objekte najde hitro.

Preverjanje interpretacije

Konsistentno interpretacijo moramo preveriti zato, ker so ujemanja med deli, ki so nas privedla do interpretacije, lokalnega značaja. Preverjanje interpretacije je torej postopek, ki mora odgovoriti na vprašanje, ali dana množica delov res predstavlja znani objekt. Predlagani postopek odgovori na to vprašanje v nekaj korakih. Najprej lahko sistem zavrne interpretacije, ki vsebujejo preveč nepravih ujemanj, tako da postavimo prag za velikost interpretacije. Prag postavimo na nek del celotnega števila delov in je odvisen tako od objekta samega kot tudi od aplikacije. Nadalje lahko za dano interpretacijo ugotovimo hipotetičen položaj in konfiguracijo delov, tako da hipotetični objekt zavzema približno isti prostor kot deli prizora v interpretaciji. V tretjem koraku preverjanja prilegamo superkvadrik modela na regije globinske slike, ki ustrezajo ujemajočim delom prizora. Za prileganje uporabljamo standardno metodo, pri čemer funkcijo prileganja minimiziramo samo za parametre translacije in rotacije. Interpretacijo lahko zavrnemo, če je napaka ujemanja tako prilagojenega superkvadrika prevelika oz. pri delih z manj zanesljivo rekonstrukcijo če se del tekom prileganja preveč premakne .

Preverjana interpretacija torej sestoji iz modela objekta, katerega hipotetična konfiguracija je bila popravljena z prileganjem superkvadrikov modela regijam globinske slike, ki ustrezajo ujemajočim delom prizora. Nadalje lahko preverimo rotacijo trdih sklepov. V tem delu se ne ukvarjamo detajlno s preverjanjem trdih sklepov zaradi rotacijske dvoumnosti superkvadrikov, temveč le preverimo kota glavnima osema s sklepom povezanih superkvadrikov. Poleg tega bi lahko pri preverjanju interpretacije lahko dodatno upoštevamo ali je konfiguracija delov sploh možna.

7.3.2 Rezultati

Kot objekt za preverjanje metode smo uporabili človeške figure. Ker smo uporabljali globinske slike zajete z globinskim senzorjem, ki deluje s pomočjo strukturirane svetlobe in ima razmeroma majhno delovno območje, smo uporabili lutke. Lutke so gibljive in se jih da namestiti v različne položaje, kar smo s pridom uporabili pri eksperimentih. Z globinskim senzorjem smo zajeli globinske slike scen, te pa potem sprocesirali s SEGMENTOR-jem. Model objekta smo zgradili ročno z merjenjem ustreznih količin na lutki, pri čemer smo modelirali deset glavnih delov, glavo, trup, noge in roke. Parametre za ujemanje delov in preverjanje interpretacije smo določili na podlagi tridesetih rekonstrukcij superkvadrikov na slikah lutke.

Sistem smo preverili na dveh tipih prizor z lutko:

- prizori samo z eno lutko v različnih konfiguracijah in
- kompleksni prizori z eno ali dvema lutkama z velikim številom drugih delov.

S prvo množico testnih slik smo hoteli sistematično preveriti učinkovitost sistema za izolirane objekte. Lutko smo postavili v sedem različnih poz in za vsako zajeli osem različnih pogledov, skupaj torej 56 slik. Sistem je objekt zaznal na 39 slikah, med katerimi je bilo 24 interpretacij zelo kvalitetnih in so vsebovale povprečno 7.2 realni ujemanji. Objekta sistem ni zaznal na 17 slikah, v 9 zaradi nenavadne konfiguracije objekta, kot je bil viden z določenega zornega kota. Zaradi prekrivanja se nekateri deli niso pravilno rekonstruirali, kar je privedlo do izračunane konfiguracije delov, ki je bila kasneje v fazi prileganja superkvadrikov zavrnjena. Pri preostalih 8 primerih nezaznanega objekta je bil vzrok premalo realnih ujemanj v interpretaciji.

Delovanje sistema smo preverili tudi na 20 različnih kompleksnih prizorih, ki so poleg enega ali dveh objektov vsebovali tudi precejšnje število drugih objektov. Pri nobenem teh prizorov ni sistem zaznal objekta kjer ga ni bilo (ni bilo napačnih pozitivnih odzivov), čeprav je bilo v prizorih mnogo vsaj deloma zavajajočih konfiguracij delov.

7.4 Inicializacija 3D sledenja objektom

Prvi korak pri inicializaciji sistema za sledenje je zaznavanje objekta. Globinsko sliko prizora najprej segmentiramo da dobimo superkvadrične opise. Nato s pomočjo predlagane metode za zaznavanje objektov dobimo položaj in konfiguracijo objekta iz hipotetičnih opisov, ki se najbolje ujemajo z modelom objekta. Če objekta ne zaznamo, lahko obdelamo naslednjo sliko iz sekvence. Ko objekt zaznamo, predlagana metoda hkrati poda njegovo hipotetično pozo, ki pa jo lahko dodatno izboljšamo, da se bolje prilega globinski sliki.

7.4.1 Izboljšava kvalitete ujemanja

Ker je položaj objekta vsaj do neke mere inicializiran po tem ko je sistem zaznal objekt, lahko ta položaj uporabimo kot začetno konfiguracijo, katero nadalje prilagodimo 3D točkam iz bližine objekta. Dele objekta najprej razvrstimo glede na globino mesta, ki ga zasedajo v kinematski verigi strukture objekta. Začnemo z deli, ki nimajo pritrditve, nato deli, ki so pritrjeni na te dele, itn. Vsakemu delu (razen tistih brez pritrditve) popravimo rotacijo sklepov, ki jih povezujejo z deli nivo višje, tako da zavzemajo približno isto mesto kot preden smo premaknili dele, ki so višje na kinematski verigi. Nato poiščemo regijo bližine za del, ki je definirana kot množica 3D točk, ki so v neki razdalji do dela. Del nato prilagodimo regiji in preverimo napako prileganja za prilagojeni del ter jo primerjamo s prejšnjo. Postopek določanja bližine in prilagajanja dela ponavljamo dokler se napaka zmanjšuje ali pa je nad nekim pragom sprejemljivosti.

7.4.2 Rezultati

Da bi lahko zagotovili 3D podatke za potrebe segmentacije na eni strani in hkrati omogočili zajemanje sekvenc, smo se odločili za sistem s stereo kamero. S sistemom smo zajeli sekvence stereo slik, jih obdelali s komercialno dosegljivim sistemom za stereo kamere, da smo dobili 3D podatke o objektih na sekvencah. Metodo smo preverjali z eksperimenti s človekom kot objektom. Objekt smo modelirani z deset delnim modelom, ki je bil ročno zgrajen. Eksperimenti so pokazali, da metoda za inicializacijo deluje dokaj dobro, tako samo zaznavanje objekta kot tudi izboljšava kvalitete položaja.

7.5 Sklep

V tem delu smo najprej razvili metodo za zaznavanje 3D objektov na globinskih slikah, ki temelji na modelih s superkvadričnimi deli, kot varianto ujemanja modelov. Metoda uporablja superkvadrične opise, ki jih z globinskih slik generira SEGMENTOR. Bistvo metode je iskanje po interpretacijskem drevesu, s katerim pridemo do hipotetičnih ujemanj med deli prizora in deli modela. Predlagali smo različne tipe omejitev za primerjavo delov, kakor tudi postopek za preverjanje pravilnosti interpretacije. Metodo smo nadalje uporabili kot prvi korak pri inicializaciji sledenja. Položaj objekta, ki ga posreduje metoda za zaznavanje, smo nadalje izboljšali s pomočjo prilagajanja modela objekta direktno bližnjim regijam globinske slike. V fazi premikanja, kjer uporabimo informacijo o položaju na prejšnjih slikah za učinkovito ugotavljanje premikov, smo ravno tako uporabili prilagajanje modela objekta direktno bližnjim regijam globinske slike.

Originalni prispevki doktorske disertacije so:

- Predlagana je metoda za detekcijo in segmentacijo 3D objekta z globinskih slik s pomočjo superkvadričnih modelov, njena učinkovitost je bila preverjena z eksperimenti. Metoda temelji na interpretacijskih drevesih in predlaga različne tipe omejitev za primerjavo delov kot tudi način preverjanja interpretacije, primerne za splošno uporabo objekti s superkvadričnimi deli.
- Predlagana je metoda za inicializacijo sledenja 3D objektov na sekvencah stereo slik, ki temelji na zaznavi objekta z dodatnim korakom izboljšave položaja. Njena učinkovitost je bila eksperimentalno potrjena.

- Predlagan je način za določanje premikov strukturiranih objektov za potrebe sledenja.
- Pri problemu sledenja strukturiranih objektov so bili superkvadrični opisi uporabljeni direktno brez nekih vmesnih predstavitev.
- Do določene mere je bila eksperimentalno preverjena učinkovitost segmentacije globinskih podatkov pridobljenih s stereo kamere.

Čeprav smo se v tem delu lotili pomembnih vprašanj s področja sledenja 3D objektom, namreč zaznavanju objektov kot sredstvu za ugotavljanje položaja in inicializacije sledenje, pa se nismo dotaknili nekaterih podproblemov. Samo inicializacijo in sledenje bi morali preizkusiti na večjem številu raznovrstnih sekvenc z bolj raznovrstnimi objekti. Tako preizkušanje zahteva dolgotrajno usklajeno delovanje več raziskovalcev, ki nam za to delo ni bilo na voljo. Pri opisu metode se nismo ukvarjali z reševanjem iz napak pri sledenju. Za uspešno reševanje bi morali definirati postopek za ugotavljanje kvalitete zaznave objekta.

Bibliography

Baker, H. H. and Binford, T. O.: 1981, Depth from edge and intensity based stereo, in *Proceedings of the seventh IJCAI*, Vol. 1, pp 631–636, Vancouver, BC

Barr, A. H.: 1981, Superquadrics and angle-preserving transformations, *IEEE Computer Graphics and Applications* pp 11–23

Barron, C. and Kakadiaris, I.: 2000, Estimating anthropometry and pose from a single image, in *IEEE International Conference on Computer Vision and Pattern Recognition*, pp 669–676

Biederman, I.: 1985, Human image understanding: Recent research and a theory, *Computer Vision, Graphics, and Image Processing* **32**, 29–73

Bolles, R. C. and Horaud, P.: 1986, 3dpo: A three-dimensional part orientation system, *International Journal of Robotic Research* **5(3)**, 3–26

Bregler, C. and Malik, J.: 1998, Tracking people with twists and exponential maps, in *IEEE International Conference on Computer Vision and Pattern Recognition*, pp 8–15

Chella, A., Frixione, M., and Gaglio, S.: 2000, Understanding dynamic scenes, *Artificial Intelligence* **123**, 89–132

Chen, L. H., Liu, Y. T., and Liao, H. Y.: 1997, Similarity measure for superquadrics, in *IEEE Proceedings Vision, Image and Signal Processing*, Vol. 144(4), pp 237–243

Delamarre, Q. and Faugeras, O.: 1999, 3d articulated models and multi-view tracking with silhouettes, *International Conference on Computer Vision* pp 716–721, Corfu, Greece

Deutscher, J., Blake, A., and I., R.: 2000, Articulated body motion capture by annealed particle filtering, in *IEEE International Conference on Computer Vision and Pattern Recognition*, Vol. 2, pp 126–133

Dickinson, S. J., Pentland, A. P., and Rosenfeld, A.: 1992, From volumes to views: An approach to 3-d object recognition, *CVGIP: Image Understanding* **55(2)**, 130–154

Drummond, T. and Cipolla, R.: 2001, Real-time tracking of highly articulated structures in the presence of noisy measurements, in *ICCV*, Vol. 2, pp 315–320

Filova, V., Solina, F., and Lenarčič, J.: 1998, Automatic reconstruction of 3d human arm motion from a monocular image sequence, *Machine Vision and Applications* **10**, 223–231

Gavrila, D. and Davis, L.: 1996, 3d model based tracking of humans in action: A multiview approach, *IEEE International Conference on Computer Vision and Pattern Recognition* pp 73–80, San Francisco, CA

Grimson, W. E. L.: 1990, *Object Recognition by Computer*, MIT Press, Cambridge (MA)

Isard, M. and Blake, A.: 1998, Condensation - conditional density propagation for visual tracking, *International Journal of Computer Vision* **29(1)**, 5–28

Jaklič, A.: 1997, *Construction of CAD Models from Range Images*, doctoral dissertation, University of Ljubljana, Faculty of Electrical Engineering and Computer Science

Jaklič, A. and Solina, F.: 2003, Moments of superellipsoids and their application to range image registration, *IEEE Transactions on Society, Man and Cybernetics–Part B: Cybernetics* **33(4)**, 648–657

Jaklič, A., Leonardis, A., and Solina, F.: 2000, *Segmentation and Recovery of Superquadrics*, Kluwer Academic Publishers, Dordrecht

Jojic, N. and Huang, T. S.: 2000, Computer vision and graphics techniques for modeling dressed humans, in A. Leonardis, F. Solina, and R. Bajcsy (eds.), *The confluence of computer vision and computer graphics*, pp 179–200, Kluwer, Dordrecht

Jones, D. G. and Malik, J.: 1992, A computational framework for determining stereo correspondence from a set of linear spatial filters, in *European Conference on Computer Vision*, pp 395–410

Kim, W. Y. and Kak, A. C.: 1991, 3-d object recognition using bipartite matching embedded in discrete relaxation, *IEEE Transactions on Pattern Analysis and Machine Intelligence* **13(3)**, 224–251

Krivic, J. and Solina, F.: 2001, Superquadric-based object recognition, *9th International Conference on Computer Analysis of Images and Patterns CAIP* pp 134–141, Warsaw, Poland

Krivic, J. and Solina, F.: 2002, Part-level object recognition, *4th IEEE Region 8 International symposium on Video / image processing and multimedia communications* pp 339–344, Zadar, Croatia

Krivic, J. and Solina, F.: 2004, Part-level object recognition using superquadrics, *Journal of Computer Vision and Image Understanding* **95(1)**, 105–126

Leonardis, A.: 1996, *Image analysis using parametric models: model-recovery and model-selection*, doctoral dissertation, University of Ljubljana, Faculty of Electrical Engineering and Computer Science

Leonardis, A., Gupta, A., and Bajcsy, R.: 1995, Segmentation of range images as the search for geometric parametric models, *International Journal of Computer Vision* **14**, 253–277

Leonardis, A., Jaklič, A., and Solina, F.: 1997, Superquadrics for segmentation and modeling range data, *IEEE Transactions on Pattern Recognition and Machine Intelligence* **19(11)**, 1289–1295

Nevatia, R. and Binford, T.: 1977, Description and recognition of curved objects, *Artificial Intelligence* **38**, 77–98

Pentland, A. P.: 1986, Perceptual organization and the representation of natural form, *Artificial Intelligence* **28**, 293–331

PGR: 2006, *Point Grey Research, Inc.*, http://www.ptgrey.com

Plaenkers, R. and Fua, P.: 2002, Model-based silhouette extraction for accurate people tracking, in *European Conference on Computer Vision*, Vol. 2, pp 325–339

Plankers, R. and Fua, P.: 2001, Articulated soft objects for video-based body modeling, *IEEE International Conference on Computer Vision* pp 394–401

Raja, N. S. and Jain, A. K.: 1992, Recognizing geons from superquadrics fitted to range data, *Image and Vision Computing* pp 179–190

Rosales, R., Athitsos, V., and Sclaroff, S.: 2001a, 3d hand pose estimation using specialized mappings, in *International Conference on Computer Vision*, Vol. 1, pp 378–387

Rosales, R., Siddiqui, M., Alon, J., and Sclaroff, S.: 2001b, Estimating 3d body pose using uncalibrated cameras, in *IEEE International Conference on Computer Vision and Pattern Recognition*, Vol. 1, pp 821–827

Sidenbladh, H., Black, M. J., and Fleet, D. J.: 2000, Stochastic tracking of 3d human figures using 2d image motion, in *6th European Conference on Computer Vision*, Vol. 2, pp 702–718

Sigal, L., Bhatia, S., Roth, S., Black, M., and Isard, M.: 2004, Tracking loose-limbed people, in *IEEE International Conference on Computer Vision and Pattern Recognition*, Vol. 1, pp 421–428

Skočaj, D. and Leonardis, A.: 2000, Acquiring range images of objects with non-uniform albedo using high-dynamic scale radiance maps, in *Proceedings of ICPR'00*, pp 778–781

Sminchisescu, C. and Triggs, B.: 2003, Estimating articulated human motion with covariance scaled sampling, *International Journal of Robotics Research* **22(6)**, 371–393

Solina, F. and Bajcsy, R.: 1990, Recovery of parametric models from range images: The case for superquadrics with global deformations, *IEEE Transactions on Pattern Analysis and Machine Intelligence* **12**, 131–147

Solina, F. and Leonardis, A.: 1998, Proper scale for modeling visual data, *Image and Vision Computing* **16(2)**, 89–98

Solina, F., Leonardis, A., Jaklič, A., and Kverh, B.: 1998, Reverse engineering by means of range image interpretation, in P. Kopacek and D. Noe (eds.), *Intelligent Assembly and Disassembly, A Proceedings volume from the IFAC workshop*, pp 153–158, Pergamon, Oxford, UK, Bled, Slovenia

Taylor, C.: 2000, Reconstruction of articulated objects from point correspondences in a single uncalibrated image, *Journal of Computer Vision and Image Understanding* **80(3)**, 349–363

Wachter, S. and Nagel, H. H.: 1999, Tracking persons in monocular image sequences, *Journal of Computer Vision and Image Understanding* **74(3)**, 174–192

Zhang, Y. and Kambhamettu, C.: 2002, 3d head tracking under partial occlusion, *Pattern Recognition* **35**, 1545–1557

Izjava

Izjavljam, da sem doktorsko disertacijo izdelal samostojno pod mentorstvom prof. dr. Franca Soline.