Human Pose and Motion Estimation with a Three Dimensional Articulated Object Model

Yoshinari Kameda

1999

Abstract

In this research, a method of pose and motion estimation from ocellus image is proposed for human body.

It is widely demanded to observe human body with an ocellus camera and to estimate the pose and the motion of the human body at the fields of construction for man-machine interface, tele-operation, computer graphics manufacture, construction of absorbed virtual space and higher dimension video compression.

We first propose a method of pose and motion estimation by using an articulated model for one silhouette image.

This method of pose estimation is a fundamental technology throughout this research. It considers geometrical information which is extracted only from one silhouette image.

The articulated model which corresponds to a target object is constructed beforehand in the computer. The pose of the human body is provided by joint angle parameters of this articulated model. Although the matching evaluation function can be defined in consideration of all the variables about the entire human body, analysis of the function is difficult and consumes much calculation cost. Therefore, the method of achieving the pose and motion estimation of the entire human body on an image plane was proposed in this thesis by doing the matching evaluation between silhouette region and the projection region of the metamere in each metamere node of the articulated model.

The metamere node is evaluated only once and its joint angle is determined by our method which achieves the pose and motion estimation. We introduce a pose decision tree in the articulated model to determined order of the evaluation of the metamere node.

However, this partial evaluation with the pose decision tree occasionally cause miss-match according to relationship between location of the root node of the articulated model and a pose of the articulated object in the image.

This can be detected by comparing the estimated pose of the articulated model with entire silhouette region. Hence, we realize an accurate pose estimation method by backtracking in the pose decision tree when the miss-match is found by the detection done at the end of the pose estimation procedure so as to avoid falling into local optimal solution.

A silhouette region is an image feature that can provide shape of the human body without knowledge of the human body. However, it is enumerated not to be able to detect self-occlusion in the silhouette region as a defect of using it. Therefore, at the time of the processing of a certain pose estimation, we propose to exclude projection region from the evaluation, which corresponded to metamere nodes of which the joint angle have been determined already. We call this excluded region an gnawed region. It is shown to avoid falling easily into local solution of the joint angle according to the pose decision tree, and to obtain better solution with our method by the experimental result.

When we take a picture of motion of human body by having fixed the camera, the image sequence which contains silhouette image by the background difference at each frame is called a motion silhouette. We realize motion estimation of human body by estimating pose of the

human body for silhouette region at each frame. A characteristic of our method is to assume the moment of inertia to each metamere. As a result, a metamere in the silhouette region can be tracked even if it is under self-occlusion in the human body. If the transition of the joint angle parameter is linear in the motion under the self-occlusion, the location of the metamere can be forecasted by the moment of inertia. Moreover, the range of the range of the joint angle parameter is forecasted based on an assumption that a physical limitation exists at addition and subtraction speed of the rotation movement. We can reduce the calculation cost by this forecast of the movable range of the joint angle parameters.

In the motion silhouette, it is possible to evaluate which part of the human body is moving and which part is fixed according to observation for movement of the silhouette region in the frame. Hence we expand the motion estimation method previously proposed and propose to estimate only the metamere nodes which correspond to the parts of the human body. This method reduces the calculation cost for match evaluation because it doesn't process the metamere nodes of fixed parts. We introduce double difference method to extract moving image region from a continuous frame in the motion silhouette and the match evaluation only of the metamere node which can be projected to the detected area is done.

Though this thesis, we present pose and motion estimation method with a three dimensional articulated model for human body observed by ocellus camera. We first discussed the matching method for one picture with pose decision tree, and then expanded the method to motion silhouette on introducing moment of inertia and double difference region.

Contents

Chapter 1

Introduction

In this research, a method of pose and motion estimation from ocellus image is proposed for human body. The pose and motion estimation of human body is a difficult problem because it may change its pose in contrast with solid objects.

It is widely demanded to observe human body with ocellus camera and to recognize pose and motion of the human body in the many social fields such as construction of man-machine interface, tele-operation, computer graphics manufacture, and immersive virtual space and the high level video compression like Mpeg-4. Various vision based researches have been done to apply these social situations [1] [2] [3] [4] . Although good pose and motion estimation results can be obtained with multiple cameras [5] [6] [7], it may not be expected to set multiple cameras in common situations where pose and motion estimation are needed.

In the researches by which the pose is estimated for human body, the pose is estimated by describing the human body by an articulated model though the shape of the human body is not specified in many cases or they use the incomplete functional human model [8]. Because they don't give explicit definition of pose for human body in their researches, a quantitative evaluation to each technique is not obtained.

For example, some researches aim to understand human motion at symbol level directly from image sequences [9] [10] [11] [12] [13] [14], or from image sequences with simple motion model [15] [16]. These are useful if the final purpose is to understand symbolic motion like "shaking head right and left" or "waving hand", but these approaches cannot be applied to three dimensional motion understanding. In some social applications, it is strongly required to process pose and motion estimation in video-rate [17] [18] [19] [20], hence they degrade articulated model representation and as a result they cannot recover the motion precisely in the three dimensional space.

On the contrary, we introduce an articulated model by which the shape can be spatially reproduced and propose the pose and motion estimation method which uses it.

In this research, we have to obtain a solution that adjusts the articulated model which involves three dimensional information to an image which involves two dimensional information under geometrical condition of the camera projection. In that case, silhouette region is used as an image feature because it can be extracted easily even if light condition varies and without knowledge concerning the human body.

We discuss what pose and motion estimation means by describing deformation of the human body that our articulated model can represent in the following section. Moreover, we define silhouette image used as an image feature in our method.

1.1 Human Pose and Motion with Articulated Model

Let us consider 3D characteristics of our human body in PoseAndMotion. Our body anatomically consists of several different kinds of organs. They are bones, muscles, skins, internal organs, fat, and so on. When we change our pose, we make various joints change their status by activating muscles. The three dimensional shape of our body against our bones is geometrically deformed everywhere at pose change in the sense that the muscles are involved in our body under the skin and over the bones. The shape change of muscles is the motive power of our shape deformation. However, this deformation is generally quite small at many persons and so from now on we put it out of consideration.

Considering about shape deformation with the bones, it is thought that the relation of the bones provides human pose and motion. Bone relationships are classified into three types and all the three types can approximate a kind of rotation relation.

From the other aspect, deformable objects not limited from a geometrical viewpoint to human body are to be defined that a point is seemed to be moved from another point on the object when it is being deformed. In other words, a deformable object is an object which changes the location of the points on the object in the local coordinate system of the object respectively in its deformation.

In addition, deformable objects are often classified into two kinds. One is a free deformation object in which the movement of the points is all independent on deformation. The other is functional deformation object surface of which can be expressed by certain function which expresses pose by its control parameters. For example, super-quadric function expression is classified into this kind. An articulated object is also classified into the latter kind as its special form. In the articulated object, its three dimensional shape is uniquely decided by using several number of parameters which control deformation. Considering about the human body, the function rotates the parts of the human body according to its control parameters. These parameters are called joint angle parameters. One of the features of the articulated object is that it is very hard to analyze the function that defines the overall shape such as differentiating, forecasting its behaviour, and so on.

In this research, our articulated model is expressed by the graph structure that the metamere nodes are arranged on tree structure. It is considered that pose is estimated if all the joint angles between metamere nodes are determined uniquely. In general, this research can be caught as a case of the problem to obtain optimum value of this tens of the joint angles because human body is provided by over ten metamere nodes and tens of joint angle parameters.

1.2 Camera Projection and Silhouette Image

As image feature and camera model are strongly related to each other, the camera model is discussed at first in this section before explaining what silhouette is.

We assume orthographic projection on taking a picture of human body by ocellus camera in this research.

Generally, most of the cameras are to be modeled by perspective projection. however, difference between the perspective projection and orthographic projection become obvious only when the target object comes closer to the camera focus and so we don't reduce generality of our research even if we reject such situations.

An object contour shape is used as an image feature in ocellus image in this research.

Fundamental image features like pixel brightness, its color can be obtained from an image, and also, edges and regions can be extracted from the fundamental image features. Several researches have been done to extract image features especially for human beings [21] [22] [23]

[24] [25] [26] [27] [28], but they don't consider robustness against light environment variation and unlimited pose change of the human body.

A projection region of the human body is observed on the image when taking a picture of the human body. The image features which can be extracted from the projection region are brightness information in the region, edge information in the region, and its object contour shape.

It is difficult to state brightness information is commonly useful on considering various environments of pose and motion estimation because the brightness could be shifted or changed according to the light condition, reflection rate distribution of the human surface, CCD color characteristic of the camera and all these conditions should be estimated or controlled in order to reconstruct the shape of the object only from brightness information. Edge information is expected to be robust against the light condition and is useful to specify the border-lines between the metameres, but is also difficult to be extracted because false edges are found in the projection region by textures on the human surface. It is difficult to achieve a high reliability on edge extraction even if rejection of the false edges is theoretically possible by tracking the transition of of the brightness information and analyzing texture throughout the change of the human body. On the contrary, the object contour can be obtained only if the camera projection model is specified and location relation between the camera and the object is fixed. The object contour can be easily extracted by acquiring background image beforehand. In other words, the knowledge for the human body is not needed in extracting the object contour. This means that the object contour shape can be used as an absolute constraint in pose and motion estimation. Strictly speaking, there is possibility that the precise object contour cannot be obtained due to the color resembleness between the human body and the background scene. However, it is easy to evade such situations.

Therefore, weak perspective projection is assumed as camera model and the object contour of the projection region is used as an image feature in this research. This object contour shape is called a silhouette in this text.

One of the focus points of this research is that how much pose and motion estimation can be done with the articulated model from only silhouette information. Our proposed method arranges the articulated model which has high flexibility in three dimensional space to the silhouette which is robustly obtained but gives limited clue to the pose of the human body.

Researches that adopt three dimensional articulated model have been proposed [29] [30], but those are intended to estimate only two dimensional movement with input images. Some other researches [31] [32] [33] [34] [35] [36] [37] [38] [39] achieved to estimate pose and motion with three dimensional articulated model. However, the models they prepared have only joint angle information and don't have explicit shape information. Therefore, it is hard to evaluate pose correspondence in three dimensional measurement against input data.

The rest of this thesis is as follows: In Chapter 2, the basic pose estimation method for one silhouette image is presented. Strategies for traversing data structure in an articulated model are also discussed in the chapter. In Chapter 3, a region based estimation method on determining the pose of the model is proposed. Improved traversing rules are also shown in this chapter. In Chapter 4, we integrate both region and edge image features for estimating the pose of the human body. In Chapter 5, the proposed pose estimation method is expanded to apply motion estimation for silhouette image sequences by introducing inertia of the human body into the model. In Chapter 6, it is shown that motion estimation can be executed with less computation cost by using more precise motion region information which is extracted by double-difference image processing. This paper is summarized in Chapter 7.

Chapter 2

Pose Estimation Based on Overlapped Region Evaluation

In this chapter, we propose new model matching algorithms to estimate the pose of an articulated object from only one silhouette image and reveal the relationship between the silhouette and the ability of our algorithms.

In order to generalize the machine vision system much more, it is necessary to cover the variety of the objects existing in our 3D real world. Therefore, we take up the articulated objects which are an assembly of solid parts and whose deformations are fully controlled by their joints. We propose *Pose Decision Tree* to describe the the model of the articulated object and make the system easy to handle the deformation of the model.

In the matching process, an important problem is to select the kinds of image clues. Though many researches have applied edges as the image clue, they might not be stable because the statistical characteristics of the input image vary largely. So far we utilize the silhouette information instead of edges.

On taking up the silhouette as the input information and utilizing the model expressed by the *Pose Decision Tree*, we propose two matching algorithms, *Order Independent Strategy (OIS)* and *Order Independent Strategy (ODS)*, to estimate the pose of the articulated object. The former one has the ability of partially parallel processing and the latter one gives a good estimated pose similar to the original one.

We have applied these two algorithms for a real human hand and human body. The resultant poses are quite similar to the original ones. Since the experiments for the real object involves various factors and it is difficult to know the pure performance of our proposed algorithms, we also have experimented on the computer-generated silhouette images for the hand and have shown that the average difference between the region projected by the estimated pose and the original silhouette region becomes $1090.39mm^2$ in OIS and $855.96mm^2$ in ODS.

It is an interesting question whether the silhouette image is sufficient to estimate the complete 3D pose of the object or not. If not, to what extent would the model matching algorithm be able to estimate the pose with only one silhouette? We also discuss this issue and indicate the relationship between the ability of the algorithm and the silhouette information. We believe this would be a useful guide to other researchers.

2.1 Introduction

3D model-based vision is one of the most important paradigms in computer vision. This is because there is a lot of needs to extract various information such as its location and its pose from a image in a situation where the object itself is already known. Especially, the pose estimation of articulated objects such as a human hand and a human body is a fundamental technique to convey a person's will to computers, because human beings usually use fingers or a gesture to express it.

Though there are many non solid objects in our real 3D world, most of the researches have intended to estimate a rotation or a location of an solid object in 3D world. Hence a more flexible framework which covers a wide variety of objects is needed.

Most of non solid objects are articulated objects [40] [41] [42] . The articulated objects consist of several solid parts and the parts are connected each other at joints. Examples are human bodies and most of animals. As human bodies commonly play a important role in the domain of man machine communication, it is worth estimating their pose.

There are two important points in this paper. One is to show how to make a model for the articulated objects and estimate a pose of the model from one silhouette image. The other is to show the limitation of using only one silhouette image in 3D model-matching. We assume only one image is given to our system.

On using a model that is suitable to an articulated object, we should pay attention to the point that a primary difference between solid objects and articulated ones is deformability. The deformation is fully specified by the motion of the joints that follow some object-dependent formulas. Therefore, the pose of the model in our system should be controlled by parameters in the same way as the articulated object.

There are many fundamental methods to extract a shape from an image shape from shading, contour, texture, and so on. Hence our research situates in the domain of man machine interface, we could control the environment in taking an input image. In this occasion, amongst these clues a contour is thought to be easiest to extract and none the less most stable against noise and variation of light conditions. We use only a silhouette of the object and a silhouette keeps less information than a gray-scaled image, therefore the other purpose of this paper is to evaluate whether or not the silhouette has enough information to reconstruct 3D information with 3D model-based vision system. Some researches [43] [44] [45] have been done for the same purpose, but their models are in two dimensional representation and hence are not able to represent three dimensional pose.

We make clear the background of this paper in section 2.2. Section 2.3 provides a definition and a description of the articulated objects and shows an algorithm to estimate a pose of the articulated object by determining the values of model's parameters. In section 2.4, two applications are shown. We choose a human hand and a human body as real objects. Discussion about limitation of using a silhouette is given in section 2.5. At last, in section 2.6, we give the conclusions of this paper and mention related future works.

2.2 Related Works

There have been many researches on the model matching processing in the field of machine vision. However, these researches have too much constraints. One of them is that the objects are solid and furthermore that they are represented by cylinders [46] or are constructed by some straight segments [47, 48]. Though some of the researches have accomplished to reduce the constraints in model-based vision [49, 50], they still have the constraint that the object cannot change their shape.

There are many objects that are not solid in the real 3D world, and in some cases, they play an important role in our society. Examples are human hands or human bodies. Because a man usually shows his intention by hands or gestures, it might be useful for a computer to recognize the pose of them. These objects are an assembly of solid parts and are called "**articulated objects**."

In this situation, it is important for the system to estimate its 3D pose rather than its location. Some of the parts may be occluded by the other parts (we call this state "self occlusion"), which makes the pose estimation much more difficult. Horowitz and Pentland applied the Kalman filter and extracted the shape of each part shown in the image, but they did not consider the possibility of self occlusions [51]. Shakunaga used straight segments on the surface of an object to estimate the poses of the parts. His algorithm works well when the segment is seen in the image, but it has the same weakness in the case of self occlusions [52].

Some researches are restricted to deal with the human body [53, 54, 55, 56] or the hand[57, 58, 59]. They impose the peculiar constraints, such that each part must be represented by a cylinder so that they can use "ribbons" in the matching process, or that self occlusions are restricted or have been solved previously. Due to this, the generality of the studies is considered more or less lost.

So, there seems to exist a need of the new model matching algorithm that can cope with the self occlusions and estimate the pose of the articulated object without any special constraints. From this point, we propose the new algorithms which can cover the various kinds of articulated objects. We also propose the *Pose Decision Tree* description to describe the articulated objects in our system.

In the previous researches, it is also noted that most of them use the edge information on their model matching processing. Though the edges are easy to use for the system, the system does not extract them in a stable way because the results of the extracted edges depend on the parameters in the edge extraction algorithm and it is difficult to optimize the parameters with the given image. Instead, our proposed model-based vision system handle the silhouette image. As the silhouette image depends on only the camera parameters and could be extracted stably, it is a good clue to the pose estimation.

2.3 3D Shape Estimation with Pose Decision Tree

For the pose estimation of the articulated object, we take a silhouette as input information because it is relatively accessible in our constraints (introduced in 2.3.1) and stable against noise. One of the purposes of this paper is to estimate a pose of the articulated object from only one silhouette.

In this section, we first introduce the constraints, then describe the way of modeling the articulated objects using what we call "*Form Decision Tree.*" The last part includes a pose estimation algorithm for the articulated objects.

2.3.1 Constraints

The constraints introduced in our research are derived from the demands that we want to achieve our purpose straight forward without bad influences – such as the identification of the model or the estimation of the camera parameters – which are not related with the purpose. The constraints are classified into three groups, one for the articulated objects, one for the image acquisition environment, and the one for the input data to the system.

At first, we must make clear the kind of the articulated objects we handle. The constraints about the articulated objects are shown below:

- The articulated objects have several solid parts and their parts are connected at the joints, as we have mentioned before.
- They change their shape under certain formulas. A deformation of the articulated object is fully controlled by the status of its joints, i.e. the values of the parameters which are

attached to the joint.

- The solid parts can be represented by a tree graph, that means they do not form a loop. At most one joint exists between two solid parts.
- Regarding joints, we only consider rotation joints and not sliding joints, so they have at most three parameters for each.

These constraints don't reduce the generality of our model because the real articulated objects seldom have their solid parts form a loop and have sliding joints. For example, animals such as human beings and robot hands satisfy these constraints.

Our 3D model-based vision system estimates a pose of the articulated object from only one silhouette. In the silhouette image, the target object must appears completely. Self occlusions are not allowed. As Our research does not intend to recognize the target object from a scene, no other objects must not be in the image.

At last, the constraints about input information are presented here. The system knows the shape information of its solid parts and the information about its joints in advance. The information about the joints includes the relationship between parts, number of freedoms and the ranges of angle to rotate. Camera parameters specified on taking an original image are also known in advance. The system then accepts a silhouette and estimate a pose of the target articulated object.

2.3.2 Modeling with Pose Decision Tree (PDT)

The model for the articulated object consists of the solid parts and the joints between them. As the solid parts do not change their shape, values of the joint parameters affect the deformation of the model, in other words, the joint parameters control the pose of the model completely.

In the 3D model-matching vision system, one of the important problems is how to describe a model in a computer, which is convenient to the model matching. As the articulated object can change its shape in various and complicated way, its description for model-matching should be hierarchical in order to proceed with the process step by step[60]. Fortunately, most of the structures of the articulated objects are modeled in a tree graph, so we describe the model using a tree structure. We call this model description "*Pose Decision Tree* (PDT)."

In the tree, each node represents a solid part and has its identification number. Each arc represents a joint and also has its identification number. A link's identification number is the same as the identification number of the descendent node in the PDT.

Because a node in PDT represents a solid part of the model, it holds static attributes. They are :

< *Node Attributes* >

1. Identification number of this node.
2. Identification number of the descendent link. This is the same as the number of this node's identification number. (See Figure 2.1 for explanation.)
3. Geometric shape information of the solid part represented in this node's local right-hand coordinate system. The actual point of the joint toward ancestors is set to be at the origin of this coordinate system. We call this point a "*connection point*." In Figure 2.2, we show an example of node i.
4. The coordinate values of the *connection point* in the ancestor node's coordinate system.

Exceptionally, the root node doesn't have Attribute no.2 and no.4. The root node has special attributes that are used to determine the location and the pose of the root node in the 3D real world.

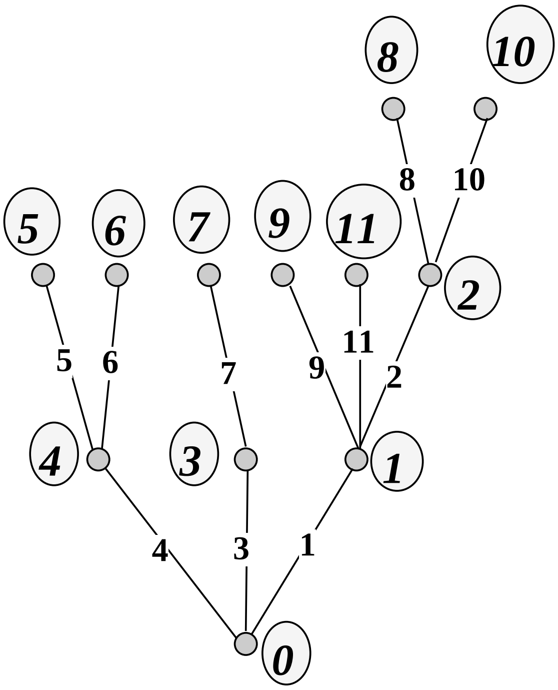


Figure 2.1: Example of PDT

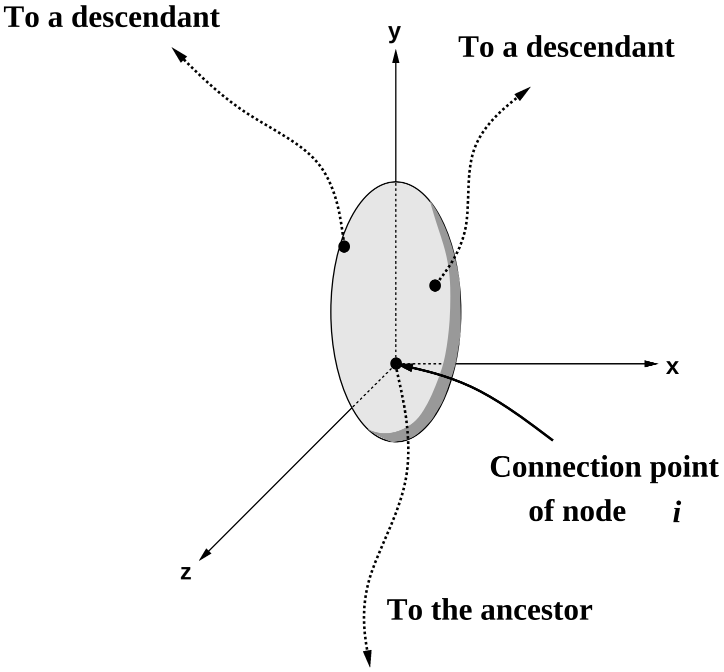


Figure 2.2: Attributes of node i in PDT

Each arc in PDT represents a joint of the model. In addition to the static attributes, it has dynamic attributes that reflect a deformation of the model. As mentioned before, only rotation joints are allowed and their values determine the deformation. Rotation axes are X-axis, Y-axis, and Z-axis.

< *Arc Attributes* >

1. Static : Angles of a priori rotation that express distortion between two successive solid parts.
2. Static : Selection of the axes. This determines which axis is to be used.
3. Static : Order of the rotation axes. This is necessary because the pose that is formed by turning around X-axis first and then around Y-axis is generally different from a pose that is formed by turning first around Y-axis and then X-axis.
4. Static : Ranges of the rotation angles which specify the lower and upper bound of rotation at each axis.
5. Dynamic : Current rotation angles of joint parameters. Only the parameters in use is allowed to exist.

From the viewpoint of kinematics of the model, its deformation process using the PDT proceeds as follows. The world coordinate system represents the 3D real world and let $\mathbf{X}_w = (x_w, y_w, z_w, 1)^t$ as the notation of a position in the world. A position in the local coordinate system of the node i is denoted by $\mathbf{X}_i$, where i is the identification number of the node. A matrix $T_{rt(i),i}$ is derived from the attribute no.4 of the node i where each x_i, y_i, z_i means the replacement along each axis and the function $rt(i)$ returns the identification number of the ancestral node i.

$$T_{rt(i),i} = \begin{pmatrix} 1 & 0 & 0 & x_i \\ 0 & 1 & 0 & y_i \\ 0 & 0 & 1 & z_i \\ 0 & 0 & 0 & 1 \end{pmatrix} \tag{2.1}$$

Another four by four matrix $R_{rt(i),i}$ is derived from Attribute no.1. Applying this matrix to the coordinates in the node i is to apply the distorted rotations between the solid parts in the beginning.

A matrix $V^R_{rt(i),i}$ is derived from Attribute no.3 and no.5 of the arc i. $V^{R(x)}_{rt(i),i}$ means the rotation transformation around the X-axis in the local coordinate system of node no.i and similarly, $V^{R(y)}_{rt(i),i}$ and $V^{R(z)}_{rt(i),i}$ are around local Y-axis and Z-axis respectively. From now on we call these parameters "*joint parameters.*" Their values represent a pose of the model. On calculating $V^R_{rt(i),i}$, the order of applying $V^{R(x)}_{rt(i),i}, V^{R(y)}_{rt(i),i}, V^{R(z)}_{rt(i),i}$ is controlled by Attribute no.3 of arc i. For example, if the order of the arc i is specified by the rotation around X-axis followed by the rotation around Z-axis and Y-axis, $V^R_{rt(i),i}$ is :

$$V^R_{rt(i),i} = V^{R(y)}_{rt(i),i} V^{R(z)}_{rt(i),i} V^{R(x)}_{rt(i),i} \tag{2.2}$$

Concerning to the special attributes in the root node r, let $V^T_{w,r}$ and $V^R_{w,r}$ be the transformation matrices for the placement and rotation of the root node against the world coordinate system respectively.

Suppose we take a node i and assume the root node r. (See Figure 2.3.) In this PDT, the arc between node j and $j-1$ is numbered as j. A position $\mathbf{X}_i$ is converted into $\mathbf{X}_j$ in a following manner:

1. Let $k \leftarrow i$.

2. Obtain $\mathbf{X}_{rt(k)}$ from Equation (2.3).

$$\mathbf{X}_{rt(k)} = T_{rt(k),k} R_{rt(k),k} V^{R}_{rt(k),k} \mathbf{X}_k \tag{2.3}$$

3. Change $k \leftarrow rt(k)$. Go to step 2 if $k \neq r$.

4. Finally $\mathbf{X}_w$ is obtained from Equation (2.4).

$$\mathbf{X}_w = V^{T}_{w,r} V^{R}_{w,r} \mathbf{X}_r \tag{2.4}$$

To change the pose of the model, apply the process to every node in the PDT. The nodes which are close to the root node must be estimated earlier than the farther ones.

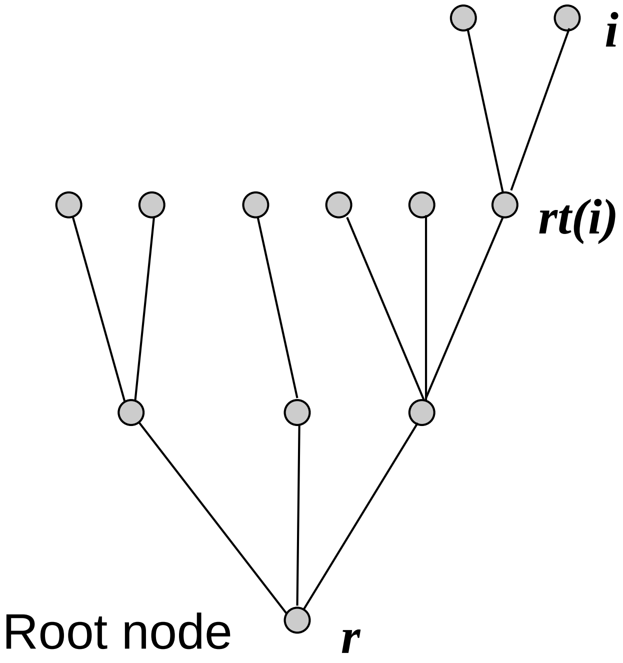


Figure 2.3: Model deformation in PDT

2.3.3 Estimation Algorithm

The 3D model-based vision system for the articulated object should solve two problems. One is to solve the location of the object in the 3D real world and the other is to solve the pose of the object. Solving the former one is equal to find the best values for the special attributes of the root node and the latter one is equal to determine the values for the *joint parameters* that make the projected region of the model best matching to the image.

Based on the constraints and the model described in the PDT, we describe how to estimate a pose of the articulated object from only one silhouette image.

This algorithm consists of two parts. First, to determine a pose of the solid part corresponding to the root node in a model-dependent way. Second, to determine the poses of the other parts sequentially in a certain order. We propose two algorithms one of which is possible to proceed partially in parallel.

Step 1 : Model Matching On Root Node

In the first step of the algorithm, the system determines the location and the pose of the solid part corresponding to the root node in the PDT. The location and the pose is specified through the special attributes $V^T_{w,r}$ and $V^R_{w,r}$ of the root node k.

To execute this step, the information of the image plane on which the silhouette exists in the 3D real world is required. It can be taken from the constraints mentioned in Section 2.3.1. As we have not determined any *joint parameters* at this step , the system should calculate $V^T_{w,r}$ and $V^R_{w,r}$ with only one solid part which represents the root node in PDT. It analyzes the difference between the projected image of the solid part and the silhouette previously given, and modifies $V^T_{w,r}$ and $V^R_{w,r}$ until they become well matched.

Because the algorithm of this step depends on the target real object, details are discussed in section 2.4.1 and 2.4.2.

Step 2 : Model Matching On Other Nodes

Here we propose two algorithms which determine all the poses of the remaining solid parts other than the root solid part one by one with overlap information between the silhouette and the projected area of the parts.

In the 3D model-based vision system, it is important how to lead the matching process so that the system prevents the process from failing in local minima and reduces calculation cost. For these purposes, matching indicators play an important role. As for a solid object, only location and rotation parameters in 3D real world, actually there are six parameters, are enough to be taken as matching indicators because the specification of these values directly identifies the model's pose. But in case of an articulated object, there are many *joint parameters* in the model. In addition, only one silhouette image is given to the system. Therefore we define a new matching indicator E. This indicator means the amount of Exclusive-OR area between the silhouette and the projected region of the model.

$$E = \sum_{\mathbf{i} \in D} S(\mathbf{i}) \oplus P(\mathbf{i}) \tag{2.5}$$

Where a closed region D is a area clipped by camera parameters in the image plane and $\mathbf{i}$ is a point in D. A function $S(\mathbf{i})$ returns a value 1 when $\mathbf{i}$ is in the silhouette and a value 0 out of it. $P(\mathbf{i})$ is a projection function and returns a value 1 when the model is projected onto the position $\mathbf{i}$ and otherwise returns a value 0. If E becomes 0 at the end of the matching process, the matching-process is considered to be completely attained. From the viewpoint of handling the 3D world, there can be a mismatch even if E becomes 0. (See Figure 2.4 for example.) This problem comes from the constraint that we give only one silhouette image to our 3D model-based vision system. It will be discussed in Section 2.5.2 in details.

To bring E gradually into 0, a modification of Equation (2.5) is needed. Analysis of the right hand in Equation (2.5) leads to the decomposition of the algorithm. First, this can be written as

$$E = E_{out} + E_{in} \tag{2.6}$$

where E_{out} and E_{in} are in the forms

$$E_{out} = \sum_{\mathbf{i} \in D} (1 - S(\mathbf{i})) \cdot P(\mathbf{i}) \tag{2.7}$$

$$E_{in} = \sum_{\mathbf{i} \in D} S(\mathbf{i}) \cdot (1 - P(\mathbf{i})) \tag{2.8}$$

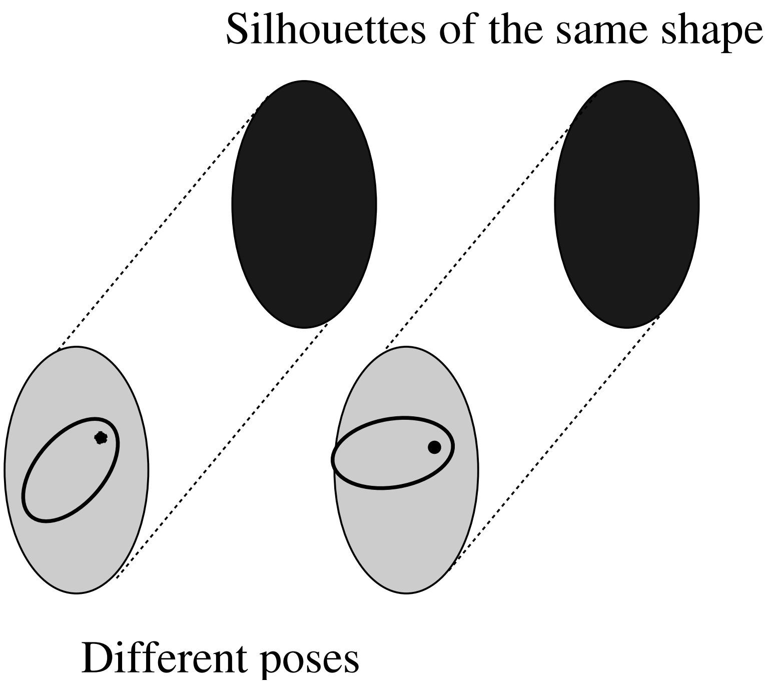


Figure 2.4: Same silhouettes of different poses

Minimizing E_{out} prevents the model's projection in the image plane from going out of the silhouette region. It works as if it could make the projection push into the silhouette region. Therefore, while we call the former constraint (2.7) "*outer pressure condition*," we call the latter constraint (2.8) "*inner pressure condition*" which makes the model matching fill the silhouette region with the projected region completely.

Both (2.7) and (2.8) are not enough to implement the actual matching process because the model's pose is determined from the root node towards the leaf nodes. So these conditions should be modified into another form in which the calculation of E can be decomposed into sub-modules that are carried out only based on the information of the node under investigation and the nodes already processed. Though there are many ways to deform them, we take care of the in-dependency of each term and change them into :

$$\begin{aligned} E_{out} &= \sum_{\mathbf{i}\in D}\{(1-S(\mathbf{i}))\cdot\sum_n P_n(\mathbf{i})\} \\ &= \sum_n\sum_{\mathbf{i}\in D}\{(1-S(\mathbf{i}))\cdot P_n(\mathbf{i})\} \qquad (2.9) \\ E_{in} &= \sum_{\mathbf{i}\in D}\{S(\mathbf{i})\cdot(1-\sum_n P_n(\mathbf{i}))\} \\ &= \prod_n\sum_{\mathbf{i}\in D}\{S(\mathbf{i})\cdot(1-P_n(\mathbf{i}))\} \qquad (2.10) \end{aligned}$$

where a subscript n applies all the node's identification numbers in the PDT and the function $P_n(\mathbf{i})$ returns a value 1 if the projection of the node n covers the position $\mathbf{i}$ in the image plane and a value 0 otherwise. As the operators $\sum_n$ and $\prod_n$ related with the identification numbers are moved outward of the right items, each $m_i^{out} = \sum_{\mathbf{i}\in D}\{(1-S(\mathbf{i}))\cdot P_n(\mathbf{i})\}$ and $m_i^{in} = \sum_{\mathbf{i}\in D}\{S(\mathbf{i})\cdot(1-P_n(\mathbf{i}))\}$ can be calculated independently of the other nodes.

If the matching process comes to a certain node i, $V^R_{rt(i),i}$ can be determined to make $m_i^{out}+m_i^{in}$ minimum without considering the nodes that do not exist on the path to the root node in the PDT. Hence this algorithm is described as follows.

< *Order Independent Strategy (OIS)* >

1. Let F be a set of nodes including only the root node in the PDT. The nodes which are next to the root node are in the set N. Other nodes are included in the set R.
2. Choose one node from N arbitrarily. Let the identification number of this node be i. Move the node i from N into F. The nodes which are next to the node i in R are moved into N. (See Figure 2.5.)
3. Find the values for $V^R_{rt(i),i}$ which make $m_i^{out} + m_i^{in}$ minimum. You should pay attention to the point that all the *joint parameters* on the path from the root node to node i have been determined before.
4. Until N becomes empty, go to step 2.

When all the nodes in the the PDT have processed, the pose of the model is determined. It is obvious that the order of the node choice at step 2 in this algorithm makes no effect on the result, so this algorithm is independent of the order of the processing. The algorithm also makes it possible to proceed partially in parallel. When a node is taken to be matched at step 3, it is never affected from the results of its brother nodes or children nodes in the PDT. So the same number of the processes as that of the nodes in N can be executed in parallel.

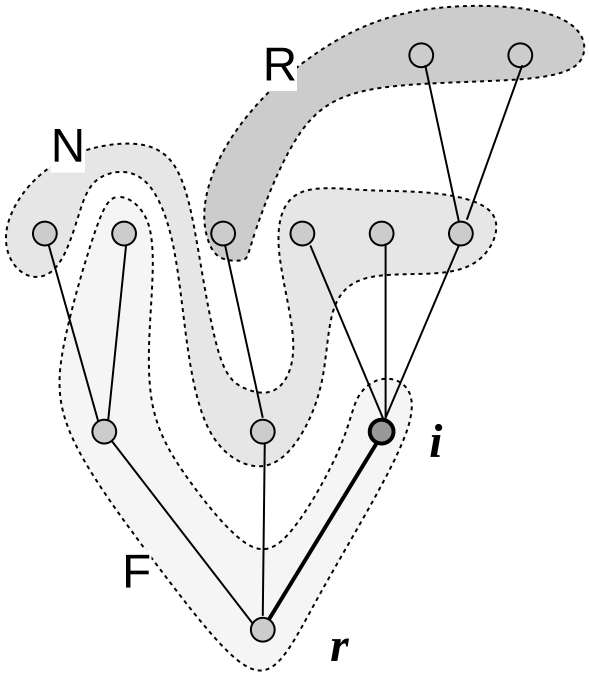


Figure 2.5: Status in the matching process

If several *joint parameters* can not be determined at step 3, they are marked as "*non-determined*" *joint parameters*. This happens when $m_i^{out} + m_i^{in}$ is constant regardless of the variation of the *joint parameters*. In this case, the system also marks the descendent arcs as *non-determined*, and the values of these parameters are set arbitrarily provided that the projections of the descendent solid parts do not make E worse.

The OIS algorithm has a tendency to rotate the solid part under consideration to cover the silhouette region as much as possible with the projected region. This is conspicuous when the projected region is completely included in the silhouette region regardless of the variation of the rotation angles. We call the part projecting the region in this situation "*complete occluded part.*"

Unfortunately, the OIS algorithm has several disadvantages. One is derived from the approximation of (2.7) and (2.8) into (2.9) and (2.10). The matching process is accomplished by reducing the sum of (2.9) and (2.10) to 0. The problem is that minimizing E' does not guarantee the minimum of E.

$$\begin{aligned} \min E &= \min(E_{out} + E_{in}) = \min \sum_i (m_i^{out} + m_i^{in}) \\ \min E' &= \sum_i \{\min(m_i^{out} + m_i^{in})\} \end{aligned} \tag{2.11}$$

This is because of the characteristic of the PDT, i.e. each pose of the solid parts must be determined in a direction from the root to the leaves. If we intend to avoid the inequality between E and E', we must deal with all the *joint parameters* in E at the same time. It is not a realistic way from the viewpoint of calculation cost. On the contrary to that defect, the calculation cost of the OIS algorithm is suppressed because the OIS algorithm proceeds along just as the same way of determining the pose of the model in the PDT,

Another disadvantage is that this algorithm does not always cover the silhouette region completely by the projection region of the model. In the result region of the projection there may be a "spot" on the image plane where many solid parts are projected, or a "spot" where no solid part is projected. To avoid the occurrence of such a spot, we introduce the next algorithm.

< *Order Dependent Strategy (ODS)* >

1. Let a set F include the root node in the PDT. Nodes which are next to the root node are in a set N. Other nodes are included in a set R.

2. Choose one node from N arbitrarily. Let this node's identification number be i. Move the node i from N into F. Nodes which are next to the node i in R are moved into N.

3. Find values for $V^R_{rt(i),i}$ which make $m_i^{\prime out} + m_i^{\prime in}$ minimum. $m_i^{\prime out}$ and $m_i^{\prime in}$ are given as follows:

$$m_i^{\prime out} = \sum_{\mathbf{i}\in D} \{(1 - S(\mathbf{i})) \cdot (1 - Q(\mathbf{i})) \cdot P_n(\mathbf{i})\} \tag{2.12}$$

$$m_i^{\prime in} = \sum_{\mathbf{i}\in D} \{S(\mathbf{i}) \cdot (1 - Q(\mathbf{i})) \cdot (1 - P_n(\mathbf{i}))\} \tag{2.13}$$

Where the function Q is :

$$Q(\mathbf{i}) = \begin{cases} 1 & \text{when at least one solid part has already been} \\ & \text{projected onto the point } i. \\ 0 & \text{otherwise.} \end{cases} \tag{2.14}$$

4. Mark data of point **i** where the point is covered by the projected area of nodes in F for all **i** in the image plane.

5. Until N and R become empty, go to step 2.

Although the result of the algorithm is influenced by the order to choose the nodes in step 2 and cannot be processed in parallel, it has a tendency to cover all the silhouette region by the region projected by the solid parts in the model. By using the function Q, the system knows the accumulated region which is the union of the regions projected by the nodes which have been processed. Due to the rejection of the accumulated region from the silhouette region, the system could cover the yet non-covered silhouette area with the projected region of the solid object under investigation by adjusting the *joint parameters*.

The ODS algorithm shows a quite different behaviour when it comes across the *complete occluded part*. According to the step 3, it moves the part in order that the projected region covers the remaining region as much as possible where $S(\mathbf{i}) \cdot (1 - Q(\mathbf{i}))$ returns a value 1. In the case the accumulated region where the function Q returns a value 1 has already covered most of the silhouette region, it is better for the system not to cover the remaining region with the projected region rather because such a covering over the remaining region sometimes fails in making a gap between the projected region and the silhouette region, and as a result, it comes to increase E. More discussion will be shown in Section 2.4.2.

2.4 Applications

We apply the two algorithms and show experimental results for real objects. The target object must satisfy the constraints in Section 2.3.1 and more over, it should be available in man machine interface. We choose a human hand and a human body as the target articulated objects since they are qualified completely for these requirements. Both of them consist of many solid parts and joints, and the estimation of their pose is difficult for the conventional methods being used for solid objects. In addition, because a gesture which signifies a person's intention is usually represented by his hands or body, we think they are good examples for our experiments.

In this section, we make clear the constraints and information given to the 3D model-based vision system in detail and explain the first step of our algorithm which was briefly described previously in Section 2.3.3. Several experimental results are also shown for the human hand and the human body, respectively.

2.4.1 Hand

We pick up a human hand as the target object for the first experiment of our research. Because a man usually shows his intention by hands, it might be useful to estimate the pose of the hands in the field of man machine interface. Several researches have been done to recognize the pose of the hands, but their methods only recognize the pose as a symbol [57, 58, 59]. In these researches the system should be reconstructed if the set of the symbols that were represented by poses of the hand were modified. Therefore, these are not general. On the contrary, our system simply finds out the values of the *joint parameters* which make the estimated pose fit well to the image. On applying our system, we can divide the hand recognition system into two modules, the former one is our system and the latter one is the symbol classification system. In this case, since the latter system simply accepts the resultant *joint parameters* and has no need to deal with image processing, it can be consisted for general purpose easily.

Constraints

The object is the bare right hand of a man. It includes the fingers, the palm, and the forearm. The geometric shape information of the hand is fully measured and utilized by the system.

We introduce the following constraints in the experiment.

- No ornament is attached to the hand.

- Concerning its pose, the wrist and other joints in the hand can be bent arbitrary in the extent the hand can be normally allowed. The rotation around the forearm axis is also allowed.

- On acquiring the image, the forearm axis must be perpendicular to the camera direction which means the direction from the camera lens center to the object. So the forearm axis lies on the plane which is perpendicular to the camera direction.

- The distance between the camera and the plane on which the forearm axis lies is known to the system. All the other camera parameters are also known to the system.

- The forearm goes out of the image frame halfway between the wrist and the elbow. The image should include the fingers and the palm perfectly, but self occlusions are allowed.

- The background of the image is black colored and no other objects are taken together into the image. The image is firstly taken as a gray scaled image, but it is converted into a binary image where the pixel in the silhouette region has a value 1 and in the background region does a value 0.

Hand Model

Here we construct a model for the hand.

According to anatomy, it is natural to think that one bone corresponds to only one solid part. The hand including the forearm consists of 29 bones. Some of them are considered to be tied firmly. For example, though the palm part contains 12 bones, the joints between them can be bent very little. The forearm part consists of 2 bones, which can not change their relative location. Therefore, less than 29 parts is sufficient to make the hand model. In this paper, we apply a model consisted of 17 parts: three parts for each finger, one for the palm and one for the forearm. Consequently, the PDT has 17 nodes and 16 arcs.

The problem when we represent the model in the PDT is to decide which part corresponds to the root node. The location of part corresponding to the root node must be extracted accurately because aberration of the location may fails to accomplish the pose estimation at the later step in the model matching algorithm. We choose the forearm part as the root node because the forearm is a relatively larger part than the others and because it is placed perpendicular to the camera direction.

Figure 2.6 shows the solid parts of the hand model and Figure 2.7 shows its PDT graph structure. Table 2.1 shows the range of rotation angles (Arc Attribute no.4) of the arcs. In the table, $w^{R(A)}$ means the range of the rotation around A-axis in the local coordinate system of the node, the number of which is denoted in the left most column in the row. Values are expressed in degree. A column with a bar means no rotation is allowed around the axis.

The actual geometric shapes of the solid parts corresponding to the fingers and the forearm are described by elliptic cylinders at the top and the bottom of which hemispheres are attached.

Pose Estimation of the Root Node

In the first step of the model matching process, the system extracts the direction and the location of the forearm axis in accordance with the constraints shown in Section 2.4.1.

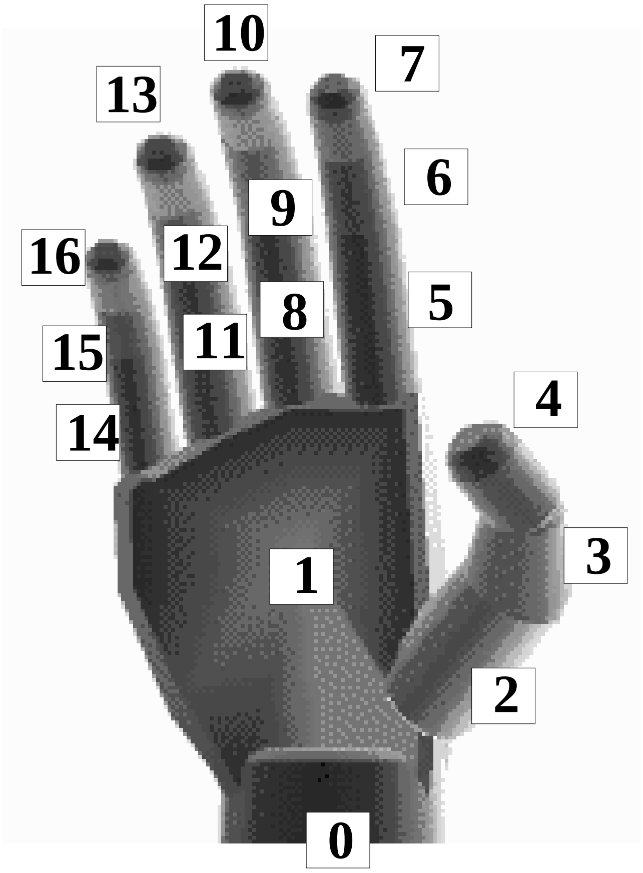


Figure 2.6: Hand model

Table 2.1: Rotation range of the hand joints

arc	$w^{R(X)}$[deg]	$w^{R(Y)}$[deg]	$w^{R(Z)}$[deg]
1	85	-	40
2	55	-	85
3	-	-	50
4	-	-	90
5	90	-	20
6	114	-	-
7	76	-	-
8	90	-	20
9	114	-	-
10	76	-	-
11	90	-	15
12	114	-	-
13	76	-	-
14	90	-	20
15	114	-	-
16	76	-	-

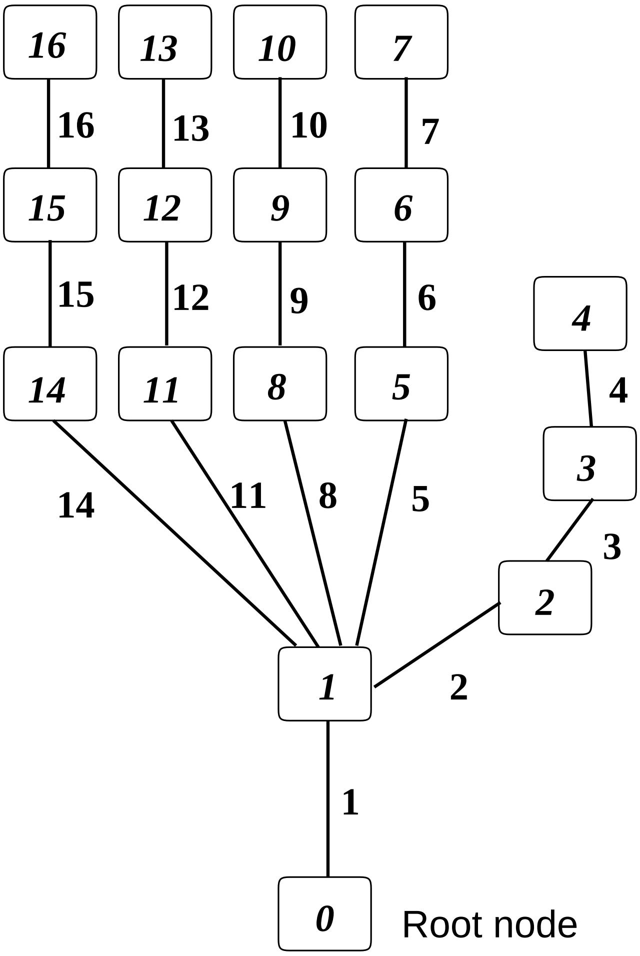


Figure 2.7: PDT of the hand model

We pay attention to the geometric shape of the forearm part. It is represented by an elliptical cylinder. As it is located in the way its axis and the camera direction become orthogonal, a pair of lines, that are located on the surface of the elliptical cylinder to which the line from the camera center to the object tangents, correspond to a pair of straight segment pair in the silhouette contour and are parallel with the forearm axis. Therefore the middle line between two straight segments which start from the boundary of the image frame should be the projected line of the forearm axis (Figure 2.8).

With the location of the middle line in the image plane, the system can calculate the 3D location of the forearm axis according to the camera parameters.

Then the system needs to specify the rotation angle around the forearm axis. Let an elliptic cylinder be projected on a region whose width is $\overline{AB}$ in the image plane (Figure 2.9 <1>). The image plane is represented by the line on the right side. There are three other possible poses which shares the same cylinder axis (denoted as a black dot). The models in those poses are projected in the width $\overline{AB}$ (Figure 2.9 <2>,<3>,<4>). Since the system can not distinguish these four cases, the system takes all of them as the root pose candidates.

In the second step, the system executes the model matching algorithm introduced in Section 2.3.3 for every root pose candidate. In the final step, the system takes one of the resultant poses of which E is the minimum among them.

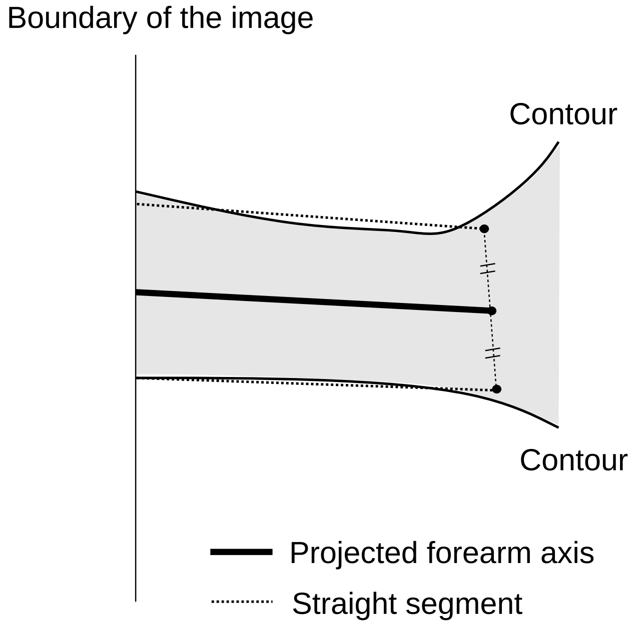


Figure 2.8: Location of the forearm axis

Experimental Results

The two algorithm OIS and ODS is executed on 21 sample images. 8 of them is a size of 580 pixels by 420 pixels and 13 of them is a size of 600 pixels by 400 pixels. The distance between the camera and the hand is about $1500mm$. Because of the distance and the camera parameters, we adopt orthogonal projection as the projection algorithm. As a result, one pixel becomes a square of $0.5mm$ by $0.5mm$.

Unfortunately, it is difficult to verify the matching degree in the 3D world because we have no further information occluded by the front part with only one image and unable to reconstruct the 3D pose completely. The matching degree in this paper is therefore examined by the matching indicator E. Table 2.2 shows the averages of E of the resultant poses. The difference between the result of OIS and that of ODS is relatively small ($296mm^2$). As OIS has an advantage of the partially parallel processing, it is said that the OIS algorithm is the better of the two in terms of the processing time. On the other hand, if the resultant pose is required to resemble the shape of the silhouette region, ODS is better since it generates the better resultant pose. This is because it has a tendency to cover the whole silhouette region with the projected region of the hand model.

Table 2.2: Averages of E for the hand

	OIS	ODS
$E[mm^2]$	3340	3044
$\frac{E}{Silhouette}$	0.215	0.196

Area average of the silhouettes = 15548.7 $[mm^2]$.

Among them, five experimental results are shown in Figures 2.10, 2.11, 2.12, 2.13 and 2.14. Here these images are reversal. In each figure, (a) is the original gray scaled image and (b)

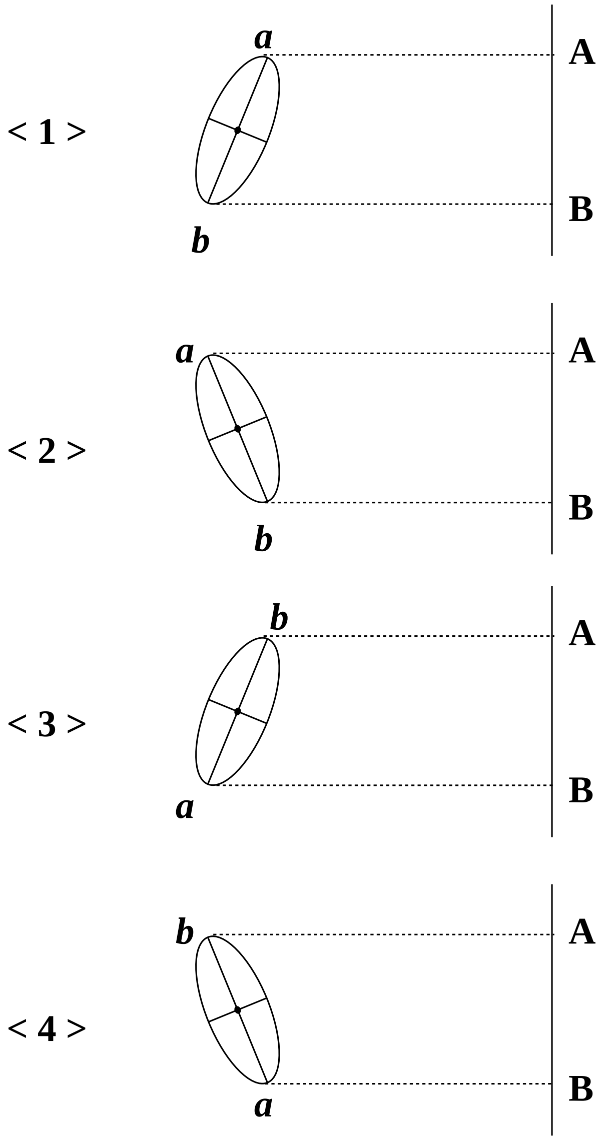
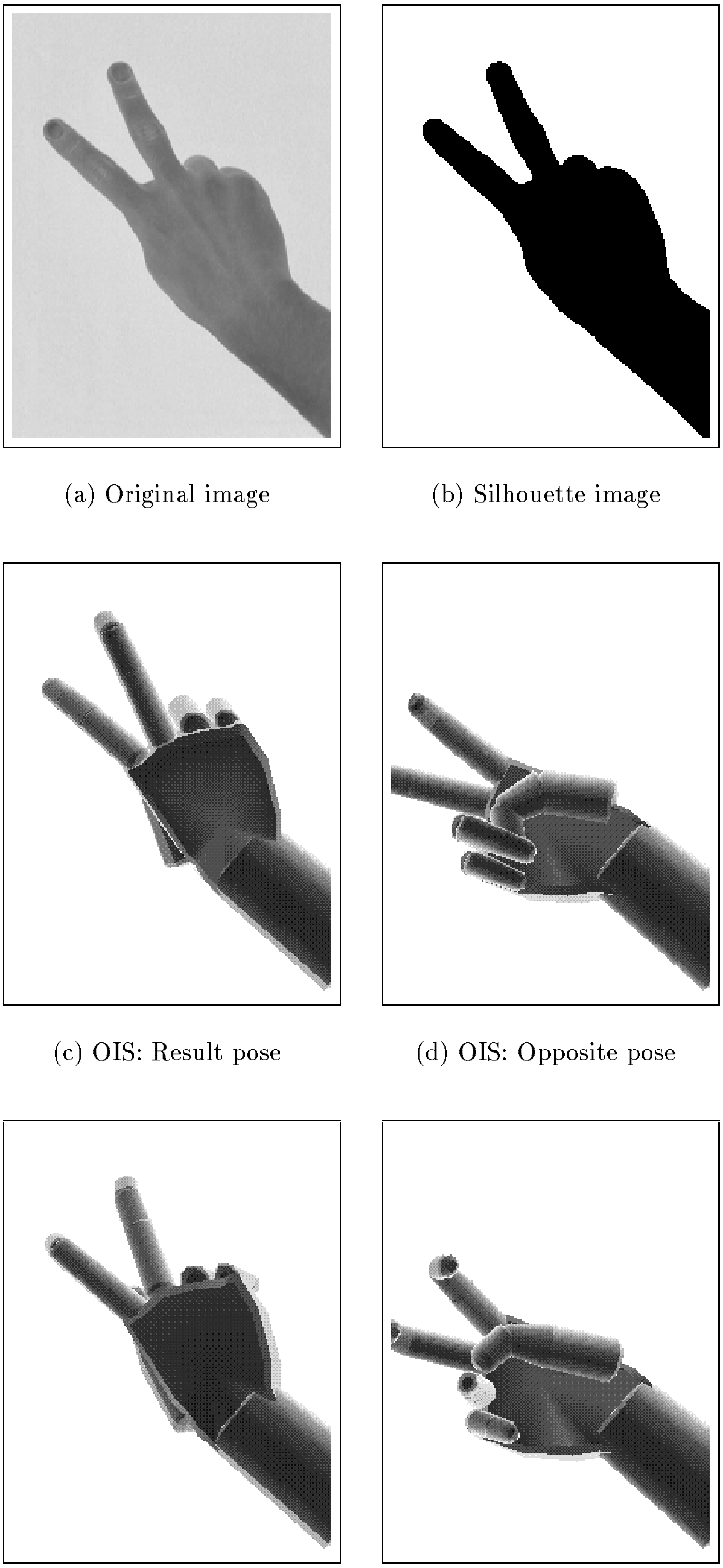


Figure 2.9: Projection of an elliptic cylinder

is the converted silhouette image, which is given to the system. The OIS algorithm estimates the pose and the result is shown in (c). In (d), the model is rotated 180^o around the forearm axis from the pose in (c). (e) and (f) are similar to (c) and (d), respectively, except for the point that ODS generates them. The resultant pose of ODS in (e) is more accurate than that of OIS in (c) as an estimation using the silhouette image (b).

In some cases, OIS and ODS show different results as are shown in these figures. These differences are derived from the existence of the *complete occluded parts*. (See (d) and (f) of Figure 2.10 for example). Though the poses of the third finger and the little finger are quite different, those fingers make little effects on the matching indicator E. Because the opposite side of the occluded region can not be seen at all, we have no further information for judging which is right.



(a) Original image

(b) Silhouette image

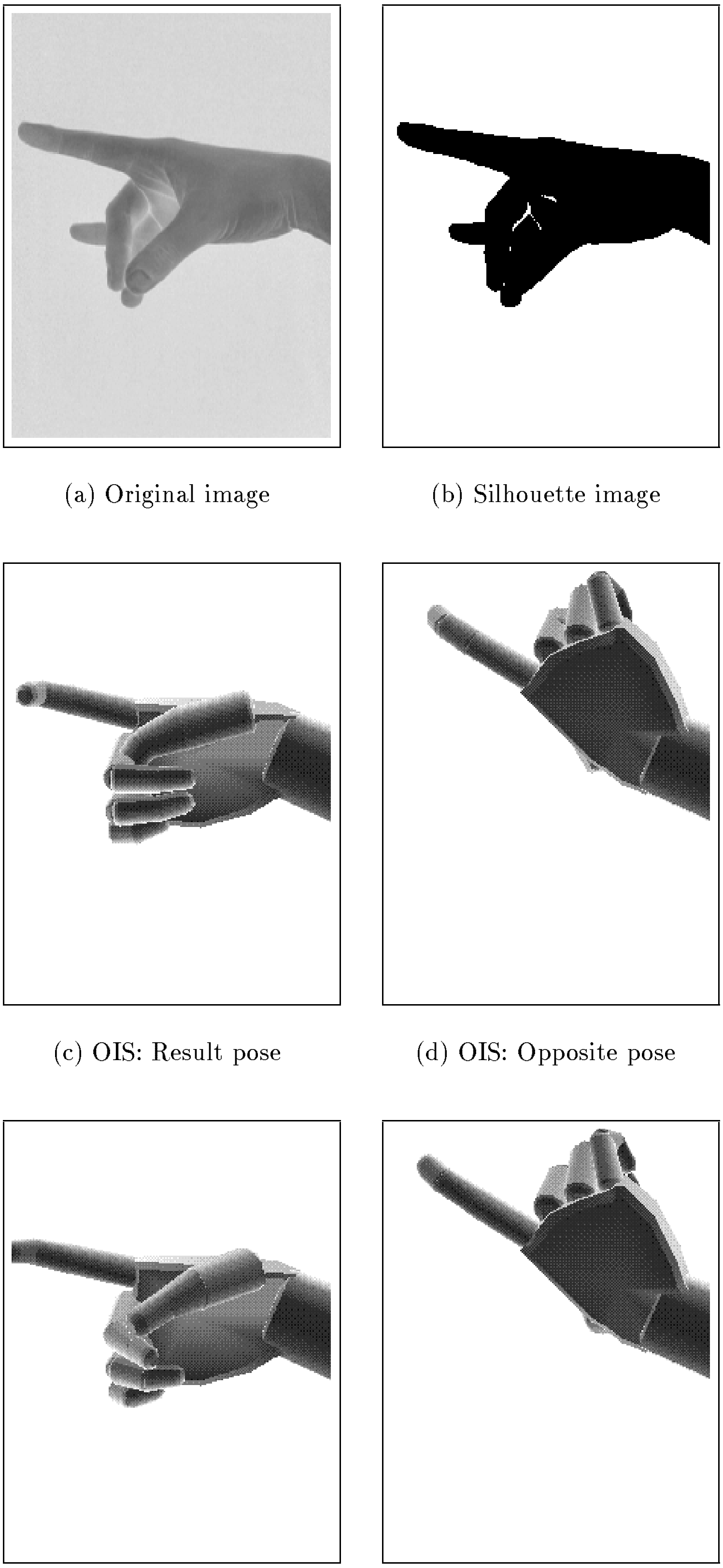
(c) OIS: Result pose

(d) OIS: Opposite pose

(e) ODS: Result pose

(f) ODS: Opposite pose

Figure 2.10: Hand experiment no.1



(a) Original image

(b) Silhouette image

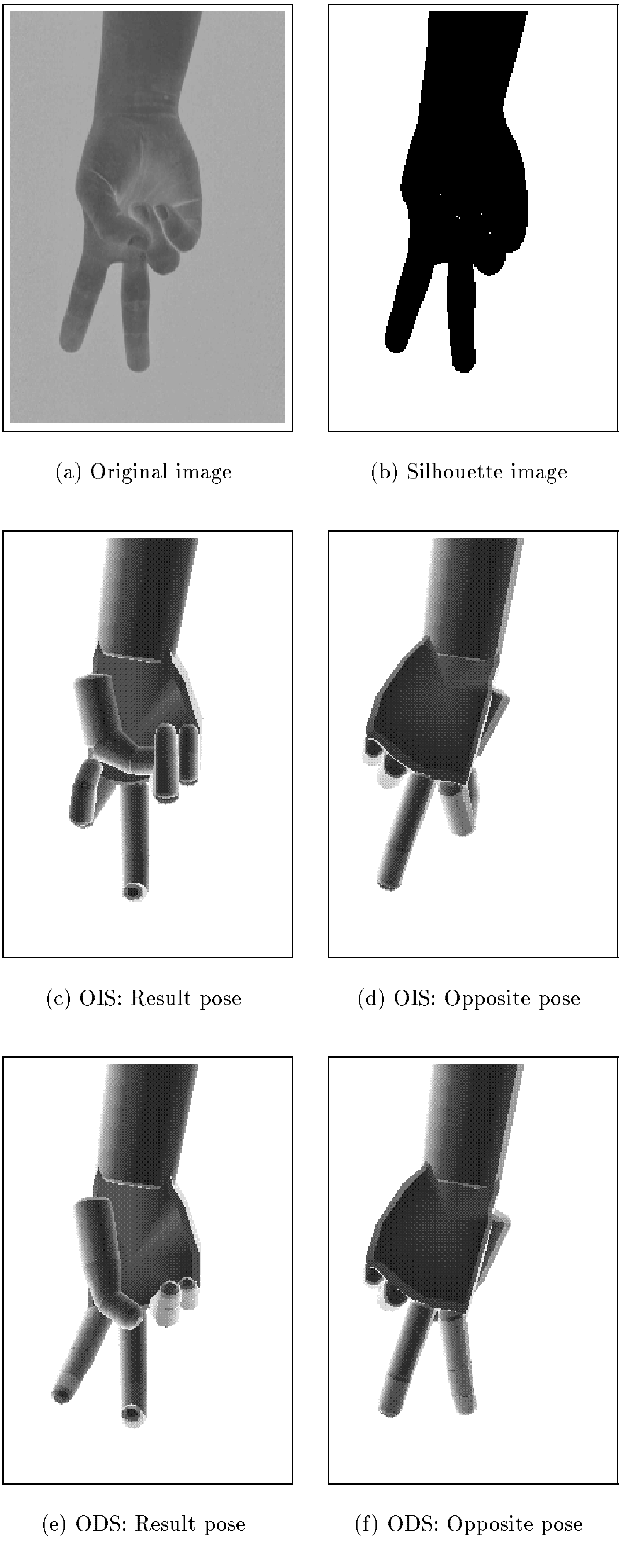
(c) OIS: Result pose

(d) OIS: Opposite pose

(e) ODS: Result pose

(f) ODS: Opposite pose

Figure 2.11: Hand experiment no.2



(a) Original image

(b) Silhouette image

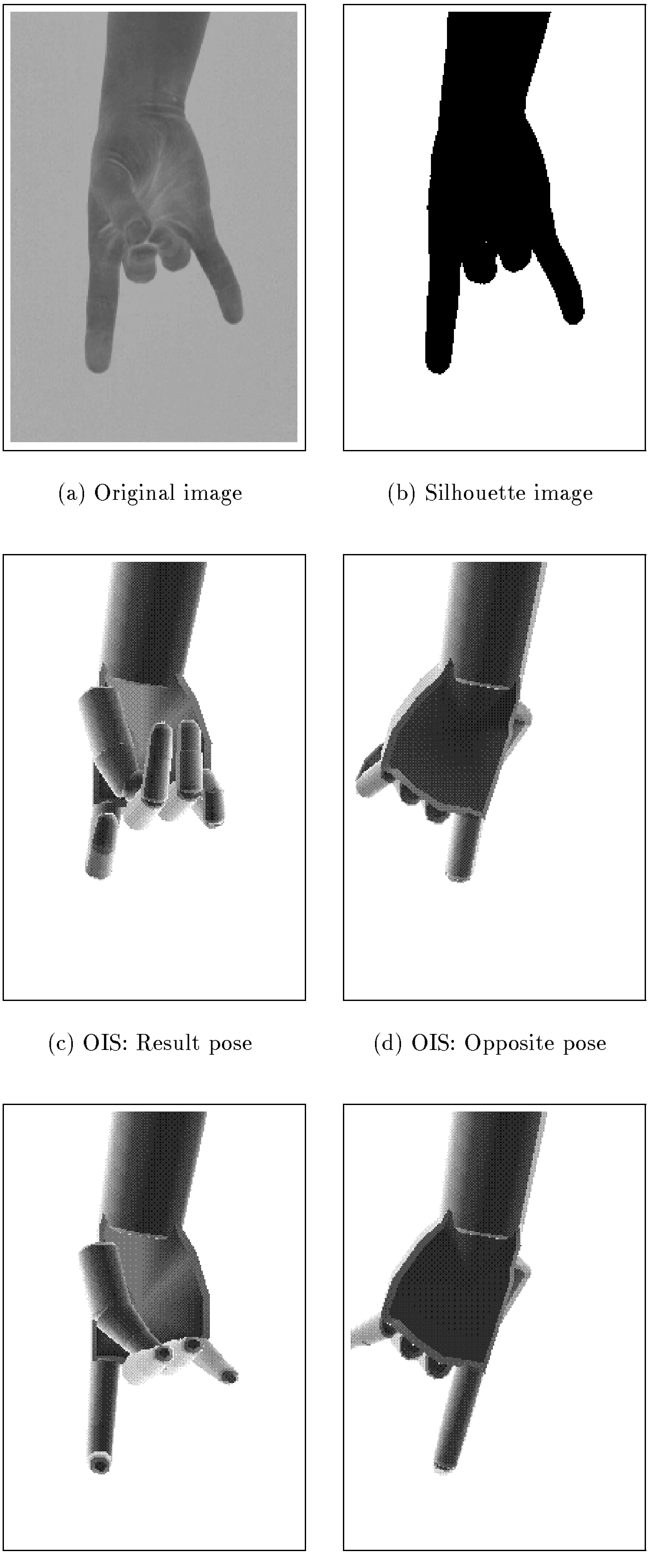
(c) OIS: Result pose

(d) OIS: Opposite pose

(e) ODS: Result pose

(f) ODS: Opposite pose

Figure 2.12: Hand experiment no.3



(a) Original image

(b) Silhouette image

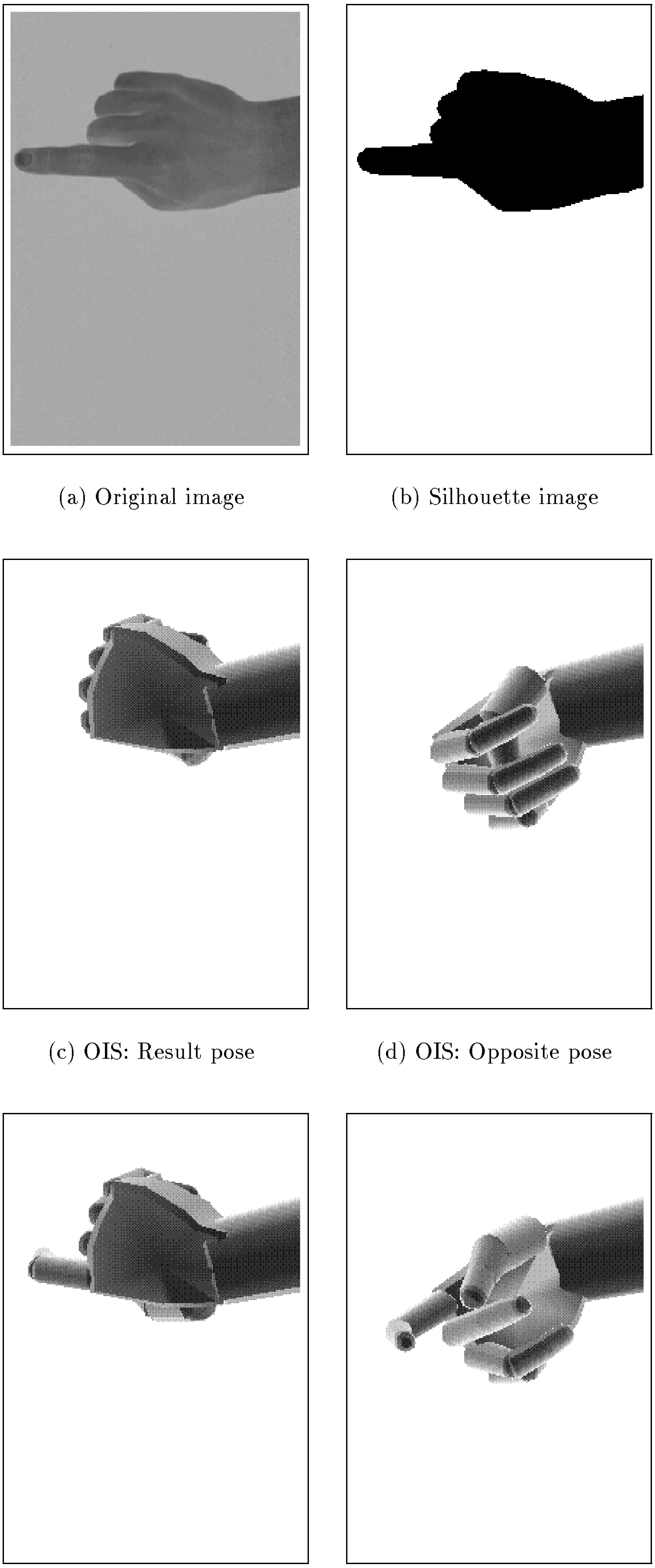
(c) OIS: Result pose

(d) OIS: Opposite pose

(e) ODS: Result pose

(f) ODS: Opposite pose

Figure 2.13: Hand experiment no.4



(a) Original image

(b) Silhouette image

(c) OIS: Result pose

(d) OIS: Opposite pose

(e) ODS: Result pose

(f) ODS: Opposite pose

Figure 2.14: Hand experiment no.5

2.4.2 Human Body

In the second experiment, we take up a human body. A human body is a typical articulated object and yet more difficult to estimate its pose from its image. However, since a man expresses his intention not only by his hands or words but also by his gesture in his daily life, the method of estimating the pose of the human body will be an essential technique for a new man machine interface of computers.

Constraints

Here, the target articulated object is the whole body of a woman. The geometric shape information of the body is obtained by measuring a woman with a 3D range sensor.

- The body wears no ornaments and no deformable things. Only the clothes that are fitted tightly to the body is allowed. To remove the deformability of the hair, a bandage covers it.
- All the joints are allowed to be bent as far as in a normal extent. Both sides of the neck part should be seen from the camera. Except for this, self occlusions are allowed.
- The neck axis, which is the same as that of the spinal cord, is perpendicular to the floor. From the camera, both sides of the neck must be seen.
- The camera is set to be horizontal. The distance from the camera to the body and all the camera parameters are given to the system as a priori information.
- The whole of the body must be seen and no other object should not be seen in the image.
- The color of the background of the image is black. The image is taken as a gray scaled image, and then it is converted into a binary image where a pixel in the silhouette region returns a value 1.

In our experiment, we take images in the situation in which the woman is walking.

Human Body Model

Similar to the hand model, it is natural to think that one bone corresponds to only one solid part. Although the human body consists of more than one hundred bones, many of them are tied firmly or too small to be consider in the model matching. For example, a hand consists of 27 bones but each one is too small compared with the other parts such as arms or legs. Thus, it is not reasonable to handle both the hands and other large parts together.

We divide the human body into 17 solid parts. The division is made at the joints which can be bent to a large extent and their adjacent parts are relatively large. The parts are a head, a neck, two upper arms, two forearms, two hands, a chest, a waist, two thighs, two legs and two foots. As a result, there 17 nodes and 16 arcs in the PDT.

As we have mentioned, the choice of the root node affects much on the later model-matching process. We choose the neck part for the root node so that the system does not misplace the object. The constraints guarantee the extraction of the two segments corresponding to both sides of the neck part and the system can easily estimate the location of the neck.

The geometric shape information of the solid parts and their identification numbers in the PDT are specified as shown in Figure 2.15 and its graph structure is shown in Figure 2.16. Table 2.3 shows the ranges of rotation angles (Arc Attribute no.4) of the arcs. Values are expressed in degree. Node 0 represents the root node which corresponds to the neck part. The root node (node 0) has two descendants, one is node 1 and the other is node 2, which represent the head and the chest, respectively.

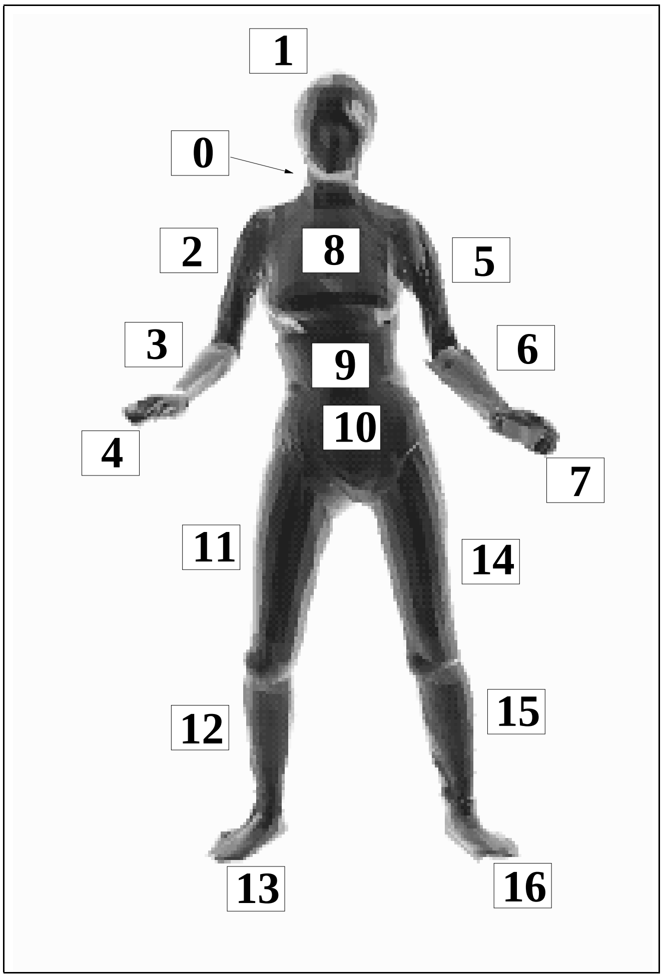


Figure 2.15: Human body model

The actual geometric shape information of the solid parts have been obtained by the range data and is expressed in the patch-modeling description. The shape of the neck part is similar to a trapezoidal cone.

Unfortunately, the shape information near the connection points is not so accurate because the data is obtained from the woman in certain pose and no information is available on bending the joints. We attach an ellipsoid between two solid parts to interpolate its shape.

Pose Estimation of The Root Node

In the first step of the model matching process, the system estimates the 3D location of the solid part corresponding to the root node in accordance with the constraints shown in Section 2.4.2.

As Many good researches [61] [62] [63] [64] [65] [66] [67] [68] [69] [70] have been proposed to extract human face from an image, we don't concentrate on how to locate the root node in the case it is head part.

However, considering about silhouette region, it is reasonable to select the neck part for the root node of the PDT because its location can be easily estimated. The algorithm to estimate the location of the neck part is executed in the following manner.

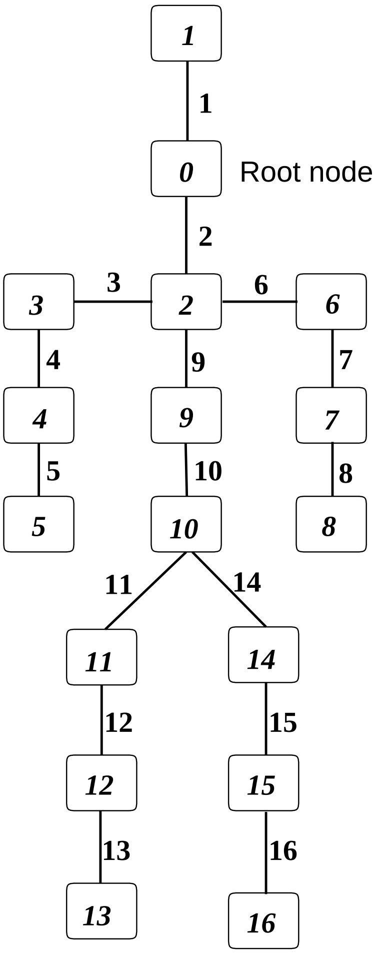


Figure 2.16: PDT of the body model

From the silhouette contour the system finds two segments which correspond to the sides of the neck part. Searching from the top of the contour, the system extracts the two segments which are straight and the distance of which are almost the same as the diameter of the neck part. (See Figure 2.17.)

Second, it calculates the center line of the two segments. Because the neck part is perpendicular to the floor, the center line corresponds to the projected neck axis.

Then, calculate the 3D location of the neck axis in accordance with the camera parameters.

Unfortunately, it is difficult to determine the rotation angle around the neck axis only by the projection of the neck part. The system calculates the angle so that the projection of the neck part fits the silhouette region best, but its resultant pose does not usually match the projected region of the real neck. To cope with the problem, the joints to the head and the chest part are allowed to rotate widely around the neck axis. See the column $w^{R(Y)}$ at the arc 1 and arc 2 in Table 2.3. This allows the system manage to cope with the location failure in estimating the pose of the neck part in the second step of our algorithm.

Experimental Results

We have experimented with two algorithm, OIS and ODS, for 15 sample images. The size of the images is 320 pixels by 360 pixels. The distance from the camera to the body is about $400cm$. One pixel becomes a square of $4.43mm$ by $4.43mm$.

Table 2.3: Rotation range of the body joints

arc	$w^{R(X)}$[deg]	$w^{R(Y)}$[deg]	$w^{R(Z)}$[deg]
1	40	180	30
2	20	360	20
3	200	-	110
4	160	110	-
5	40	180	85
6	180	-	110
7	160	110	-
8	40	180	85
9	20	40	20
10	40	100	60
11	110	70	40
12	120	-	-
13	80	40	70
14	110	70	40
15	120	-	-
16	80	40	70

As in the case of the hand, we use the matching indicator E for the evaluation of the matching result. Table 2.4 shows the average E of the resultant poses. The difference between OIS and ODS is $21095mm^2$. Whereas in the experiment for the hand ODS improves the result E only by 8.83% compared with that of OIS, here for the human body it improves the result by 21.81% compared with that of ODS. The reason why the ODS algorithm has the advantage against the OIS algorithm is considered as follows. The silhouettes for the human body have quite large, long and narrow regions sticking outwards. The OIS algorithm hardly covers those sticking regions whereas the ODS algorithm covers them many times. Suppose the situation

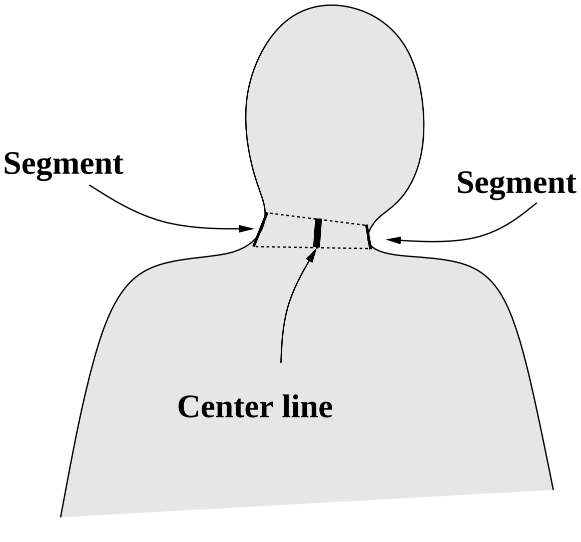


Figure 2.17: Location of the neck axis

that the solid part i under consideration in step 3 of the OIS algorithm is located near the base of the sticking-out region. In many cases, the system sets it inner part of the silhouette region. If the system set the node i to cover the sticking-out region, there might be a possibility of more or less increasing $m_i^{min} + m_i^{out}$ even if its projected region fits well with the sticking-out region, because there is still a little difference between two regions. On the other hand, in the same situation the ODS algorithm can cover the sticking-out region well only if the accumulated region, where the function Q returns a value 1, has already covered most of the silhouette region. Since the accumulated region is considered to be eliminated from the silhouette region in the ODS algorithm, the pose covering the sticking-out region makes $m_i^{min} + m_i^{out}$ smaller. Even if there exists not a little gap between its projected area and the sticking-out region, it does not prevent the ODS algorithm from covering the sticking-out region. Thus ODS works much better than OIS in the case of the pose estimation of the human body.

Table 2.4: Averages of E for the body

	OIS	ODS
$E[mm^2]$	96581	75486
$\frac{E}{Silhouette}$	0.298	0.233

Area average of the silhouettes = 324503 $[mm^2]$.

Five of the experimental results are also shown in Figures 2.18, 2.19, 2.20, 2.21 and 2.22. All of these figures are reversal. Similar to Section 2.4.1, (a) shows the original gray scaled image and (b) is the converted silhouette image from (a). The resultant poses of the OIS and ODS algorithm are shown in (c) and (e), respectively. We also show the images which are taken from the opposite side of the resultant poses, such that (d) corresponds to (c) and (f) to (e).

In these experimental results, the advantage of the ODS becomes more obvious in the case of dealing with a *complete occluded part* in comparison with those of the hand. On that case, the OIS algorithm tends to cover the silhouette region as much as possible with its projected region. As the solid parts are similar to cylinders, the OIS algorithm may also said to have a tendency to lay down the *complete occluded part* in the image plane. However, as the silhouette region is generally long and narrow in our experiments, the OIS algorithm often fails in having the descendant part sticked out from the silhouette region. For example, see Figure 2.18. The OIS places the left upper arm in the way its elbow comes behind the head and as a result, the descendent forearm and hand parts are misplaced completely, whereas the ODS algorithm avoids that.

2.5 Discussion

In the section 2.4, we showed the experimental results of which the matching indicator E was small but not reduced to 0. The 3D model based vision system using only one silhouette image has three issues related one another.

1. Is there any factor that improves the matching algorithm in order to estimate the better pose in our framework ?
2. Is it possible for the 3D model based vision to decrease E into 0 from only one silhouette image ?

3. How much extent does the resultant pose, of which E is 0, resemble the real pose in the sense of 3D measurement ?

We take up the human hand as the typical articulated object and discuss these issues in a practical way. To argue the first issue, a supplemental experiment is necessary and is described in Section 2.5.1. The second and third issues are discussed together in Section 2.5.2.

2.5.1 Verification of OIS and ODS

In this section, we want to disclose the ability of the OIS and ODS algorithms proposed in Section 2.3.3. There are two factors that are not related with the essence of those algorithms and so far we should remove in the experiment.

One of the reasons why the results of the experiment shown in the section 2.4 could not reduce their E into 0 is considered to be derived from the geometrical inconsistency between the real hand and the hand model in the system.

As we have mentioned in Section 2.4.1, each of the solid parts except for the palm are represented by an elliptic cylinder attached by two hemispheres on both ends. In this modeling method, it is difficult to represent muscles between the parts. Though they occupy not a large part of the hand, yet they surely exist. We could avoid this disadvantage by adding the deformable parts representing the muscles between the solid parts, since it is relatively easy to modify our algorithms to cope with the deformable parts which are controlled by the *joint parameters*.

Nonetheless, there exists the geometric inconsistency between each of the solid parts and its corresponding real part. This may be yielded by the lack of the ability of the modeling methods, or the errors in measuring the real hand might cause the inconsistency. The important point is that there certainly remains the inconsistency as far as we take up the real articulated object as the target.

There is another issue that makes E worse. It comes from the error of locating the forearm part that corresponds to the root node. It is difficult for the system to recover the influence yielded by by the error on the root node. Since we are interested in only the abilities of OIS and ODS which are the second step of our pose estimation algorithm, we should reject the root node estimation process at the first step in the experiment.

To remove these two factors and make clear the ability of our algorithm not disturbed by them, we use the computer generated image as an input image and fix the location of the solid part corresponding to the root node in the experiment.

The target object is represented in the image generated by computer graphics, and the corresponding object is used as the model in the matching process. The actual target object is the hand model described in the section 2.4.1.

In generating the silhouette image, we fix the location of the forearm axis in order to remove the location error at the root node and skip the first step of the algorithm.

We generated 1000 sample images whose poses are determined arbitrary. We applied the second step of our algorithms to these images. The values of the matching indicator E of the OIS and ODS algorithm are shown in Figure 2.23. In this graph, the horizontal direction denotes the amount of E and the vertical direction denotes the number of the samples. The averages of E in both experiments are shown in Table 2.5. The average E of ODS is $234.43mm^2$, which is smaller than that of OIS. The difference of the results between OIS and ODS is shown in Figure 2.24. In this graph, the vertical direction denotes the difference of $E_{OIS} - E_{ODS}$ at each sample. The ODS algorithm works better than the OIS algorithm on 661 samples.

Since we remove the factors that are not related with the essence of the OIS and the ODS algorithms, these results reveal the capability of the algorithms. This experiment indicates

Table 2.5: Averages of E for CG hand

	OIS	ODS
$E[mm^2]$	1090.39	855.96
$\frac{E}{Silhouette}$	0.0883	0.0693

Area average of the silhouettes = 12351.1 $[mm^2]$.

that, in general, the ODS algorithm can make the matching indicator E better than that of the OIS algorithm.

In this experiment, many results have quite a small values of E. Figure 2.25 and Figure 2.26 are examples of the best resultant poses. In each figure, (a) is the original computer generated image and is converted into the silhouette image (b). (c) is a resultant pose of the OIS algorithm and (d) is that of the ODS algorithm.

However, some of the results show that our algorithms fail in reducing E into 0. The algorithms are not able to make E small in a few results. This is salient on the OIS algorithm, but both of them show the similar tendency. Two of the worst cases are shown in Figure 2.27 and Figure 2.28. The main reason why the system fails in estimating the pose for the image (b) is due to the failure of determining the palm part location, in other words, the system fails to determine the values of the *joint parameters* between the forearm part and the palm part. See the silhouette image (b) in Figure 2.27 as an example. In this example, the projected region of the thumb part (node 2 in the PDT) is located adjacent to the projected region of the palm part (node 1 in the PDT). In this case, the system fails to determine the *joint parameters* of the palm part because it does not know where is the projected region of the palm part nor where is that of the thumb part or others at that time. To avoid this kind of failure, the system should precedes to evaluate its descendent part, in this example the thumb part (node 2), and know which region is projected by which part. This method needs much more calculation. Instead, we think the system should backtrack and re-determine the *joint parameters* only in the case that E becomes relatively worse, since the system achieves the pose estimation well for most of the silhouette images.

2.5.2 Limitation of Pose Estimation for One Silhouette Image

We proposed the model matching algorithms which use only one silhouette image in Section 2.3 and discussed the ability of the algorithms in Section 2.5.1. However, one question may arise whether or not only one silhouette is enough for the the 3D model based vision to reduce the resultant E into 0 even if the system has an accurate model for the target articulated object.

This question is closely related with the issue whether the pose having E equal to 0 is always similar to the real pose in the 3D world or not. If it is not, having E reduced into 0 is not sufficient to estimate a 3D pose.

Unfortunately, the answer is negative. Because the opposite side of the silhouette can not be seen to the system and it is impossible to know the complete state of the occlusion in the silhouette, the system can not estimate the poses of the occluded parts precisely similar to the original one. Figure 2.10 is a good example. With only looking the image (b) the system is not able to know the precise state of the third finger and/or the little finger. They may be stretched or bent, and both estimations are completely fine as far as they are the estimation of (b) and the system can not judge which is correct.

Consequently, there may be a possibility that one silhouette corresponds to several estimated poses which have the resultant value $E = 0$ and there may be an another possibility that it resembles many confusing silhouettes that are produced by other real poses. These relationships are considerably complicated and are difficult to be divided into the separated issues.

In this section, we argue these issues together and make clear the limitation of the pose estimation using only one silhouette image.

Relationship between Silhouette and Pose

We should have a measure to evaluate the similarity among silhouette images. We define a distance between two silhouette images which we call "*Xor Distance.*" The *xor distance D* between the silhouette image s and the silhouette image t is defined as

$$X(s,t) = \sum_{\mathbf{i} \in D} \{S_s(\mathbf{i}) \oplus S_t(\mathbf{i})\} \tag{2.15}$$

where D means a clipped region by the camera parameters and $S_k(\mathbf{i})$ returns a value 1 when the point $\mathbf{i}$ is included in the silhouette region of the silhouette image k. This metric can be defined only when both silhouette images are taken with the same camera parameters.

Suppose there are three silhouette images s, t and u. Obviously, the following inequality is derived from Equation (2.15).

$$0 \leq X(s,u) \leq X(s,t) + X(t,u) \tag{2.16}$$

We also define a set of the silhouette images in which the shapes of the silhouettes are considered to be similar. The set $G(\delta)$, which contains the silhouette images similar to one another in the extent of the *xor distance* δ, is defined as

$$G(\delta) = \{k | X(k,s) \leq \delta \ for \ \exists s \in G(\delta)\} \tag{2.17}$$

where k and s means a silhouette image. We call $G(\delta)$ the *silhouette group* within the *xor distance* δ.

Suppose the set $G(\delta)$ have n silhouette images. Because of the inequality (2.16), some of the *xor distances* between the pair of silhouette images in $G(\delta)$ become larger than δ. However, the worst *xor distance* in $G(\delta)$ is bounded under $n \cdot \delta$ and we obtained the conclusion that the maximum *xor distance* in $G(\delta)$ seldom reaches the worst value through the experiment. Hence this characteristic does not matter because $G(\delta)$ contains similar silhouette images in the extent of δ, so long as δ is not so large.

Then, the next problem is to measure the extent of the similarity between the 3D poses whose projected silhouettes are grouped in the same *silhouette group* from the viewpoint of the 3D model matching. The formulation of this measurement is very difficult because there is no standard that specifies the "similarity." Whereas it is easy to answer whether the two poses are same or not, it is quite a difficult task to describe the extent of their similarity.

Since the deformation of the object is fully controlled by the *joint parameters*, the measurement of the similarity is considered to be calculated by them. The amount of the contribution of each *joint parameter* to the deformation differs in terms of the characteristics of the solid part which is directly controlled by the *joint parameter*. Thus, using the PDT description we define "*pose similarity value*" L for the *silhouette group* $G(\delta)$ as

$$L(G(\delta)) = \sum_{j} \{w_j \cdot D(j)\} \tag{2.18}$$

where j means a *joint parameter* and

$$w_j = \frac{cp(arc(j))}{CP} \cdot \frac{v(affected\_node(j))}{V} \qquad (2.19)$$

$$CP = \frac{1}{A}\sum_a cp(a) \qquad (2.20)$$

$$D(j) = \{\frac{1}{m}\sum_{k\in G(\delta)} (J(k,j) - E(j))^2\}^{1/2} \qquad (2.21)$$

$$E(j) = \frac{1}{m}\sum_{k\in G(\delta)} J(k,j) \qquad (2.22)$$

In Equation (2.19), $arc(j)$ returns the identification number of the arc which includes the *joint parameter* j and $affected\_node(j)$ returns the identification number of the node which is the descendant of $arc(j)$. The function $cp(a)$ returns the Euclid distance between the ancestral *connection point* which indicates the origin of the local coordinate system of the ancestral node a and the center of the balance of the solid part corresponding to node a. The function $v(n)$ brings back the volume of the solid part corresponding to the node n. V means the volume of the whole of the object. In Equation (2.20), a takes all the arc identification numbers in the PDT and A means the number of the arcs. $J(k,j)$ in Equation (2.21) and Equation (2.22) returns the value of the *joint parameter* j of the object which projects its silhouette on the image k, and m is a number of the silhouette images in $G(\delta)$.

The *pose similarity value* L becomes small if the poses, whose projected silhouettes are grouped in $G(\delta)$, are three dimensionally similar to one another.

We also define the average of $L(G(\delta))$ for the certain object. This is calculated in a following manner. First, generate the all the poses of the object and project each of them into an image. The images are classified into the image sets each of which corresponds to a *silhouette group* of the *xor distance* δ. Then, the *pose similarity value* is calculated in each set. Finally, the average is obtained by dividing their sum by the number of the sets.

Ability of the Model Matching Algorithm

The definition of the *xor distance* X is quite similar to that of the matching indicator E. Actually, E is equal to the *xor distance* between the original silhouette image and the projected image of the estimated pose. Therefore, we can conclude the relationship between silhouettes and the ability of the model matching algorithm in the following way.

> Suppose the value of the average $L(G(\delta))$ is given to the 3D model matching system which uses only one silhouette image as an input. The system can not extract further information from the silhouette image in order to estimate a more precise pose if the resultant matching indicator E is smaller than δ.

This is because the estimated pose is similar to the poses whose projected silhouettes belong to the certain *silhouette group* of *xor distance* δ. Of course, there might be a possibility that the *silhouette group* the estimated pose belongs to differs from the *silhouette group* the original pose belongs to, and so we can use δ only as an index. However, it is true that the system should at least estimate a pose of which E is smaller than δ.

As a result, it is natural to consider that the evaluation of the model matching algorithm depends on the *pose similarity value* which is given from the outside.

We have calculated practically the averages of $L(G(\delta))$ for the actual hand object on varying δ. Though the whole of the pose should be generated and projected to calculate them, it is impossible to prepare huge number of poses because of the amount of the computation. Hence we generate the hand poses in every 20 degrees for all the *joint parameters* in the hand model and also reduce the rotation range slightly in order to keep the number of poses small. As a result, we take 23328 poses into consideration.

The relationship between the *xor distance* and the *pose similarity value* is visualized in Figure 2.29. There, the horizontal direction means the *xor distance* in mm^2 and the vertical direction means the *pose similarity value* in *degree*. Figure 2.30 shows the the average number of the silhouette images included in one *silhouette group* at certain *xor distance*. The number grows rapidly as the *xor distance* increases.

Now let us consider the ODS algorithm. Suppose we want to obtain the estimated pose in the extent of the *pose similarity value* 5.0 *degrees*. By looking at Figure 2.29 and Figure 2.30, we find out that the corresponding *xor distance* is $432mm^2$ and the *silhouette group* which includes the resultant pose might contain 2.0 poses in average. In this case , from Figure 2.23 the ODS algorithm satisfies the *xor distance* $432mm^2$ in the percentage of 43.3%. So it can be said that the ODS algorithm uses the complete information of the silhouette images about half of the pose estimations.

With this value, the ODS algorithm is able to know the matching process is falling in the local minima. Hence it can eliminate the matching process under investigation by watching the amount of the matching indicator E in the halfway of the pose estimation. This might be a good utilization of the *pose similarity value*.

2.6 Conclusion

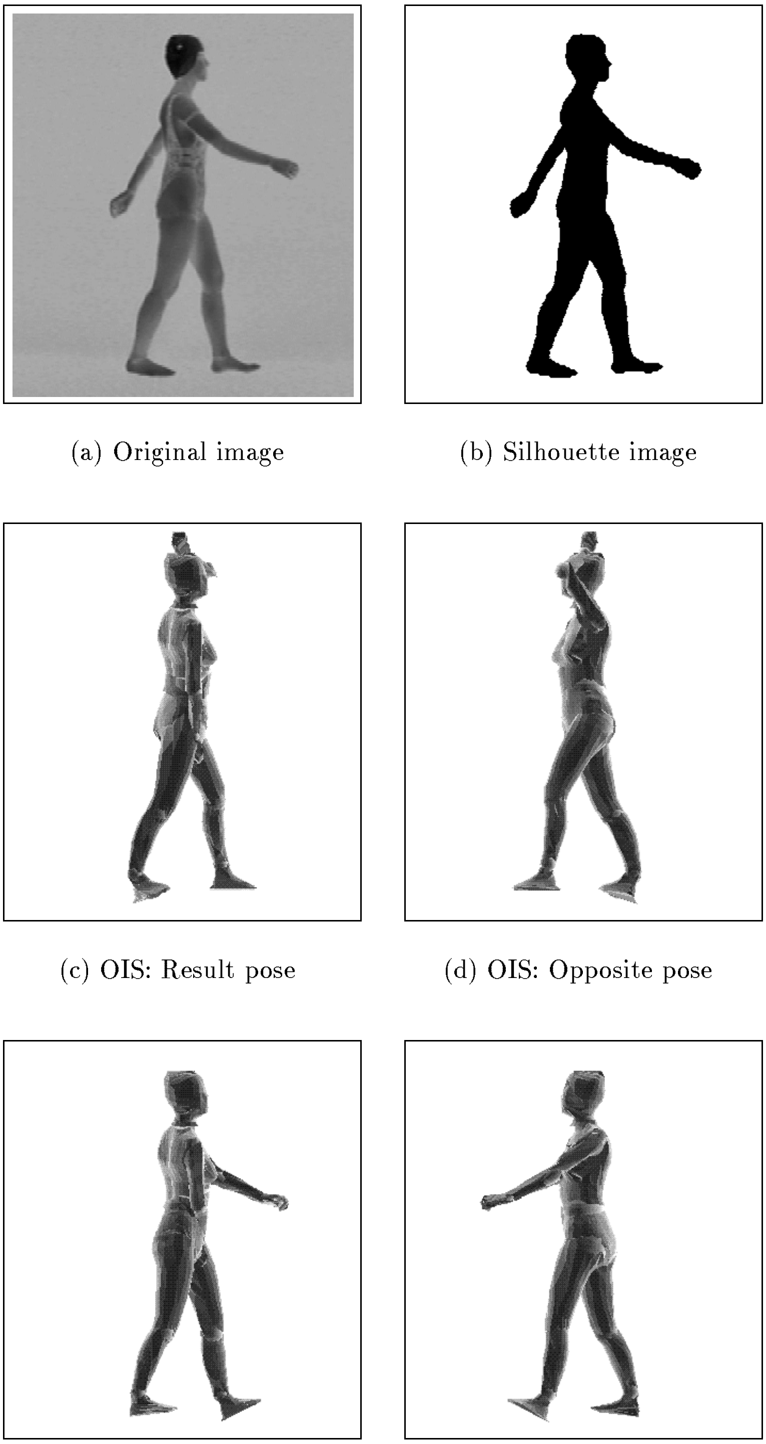
We have described the 3D model-based vision system which uses only one silhouette image as an input. For the system, we have proposed the Pose Decision Tree description to represent the model of the articulated objects. The system can control the deformation of the articulated object easily by using the *joint parameters* in the PDT.

We have also proposed the two algorithms, *Order Independent Strategy* and *Order Dependent Strategy*, to estimate the pose of the object in the system. While the OIS algorithm has the advantage of the processing speed according to its partially parallel processing, the ODS algorithm gives the better pose estimation than OIS in general.

We have experimented these two algorithms for the real hand and the human body. The results validate the usefulness of these algorithms. Furthermore, in order to explain the ability of these algorithms, we have introduced the computer generated images as input silhouette images. We took the hand object as the target and experimented on both algorithms. As a result, the average E of the OIS becomes $1090.39mm^2$ and that of ODS does $855.96mm^2$.

Because our research locates relatively in a new paradigm, we should also study the evaluation method for the model-based vision using a silhouette. For this purpose, we have proposed the *xor distance* and the *pose similarity value*. With these measures we have revealed the ability which the 3D model-based vision system should accomplish. We have also suggested the way to check the estimation failure by using the *pose similarity value*.

We have presented that our proposed algorithms estimate the pose well with only the area information of the difference between the silhouette region and the projected region. To improve the algorithms, it could be good to use the shape information of the silhouette contour such as curvature, or to change the way of unfolding Equation (2.5) so that the algorithm can deal with more than one parts at a time.



(a) Original image

(b) Silhouette image

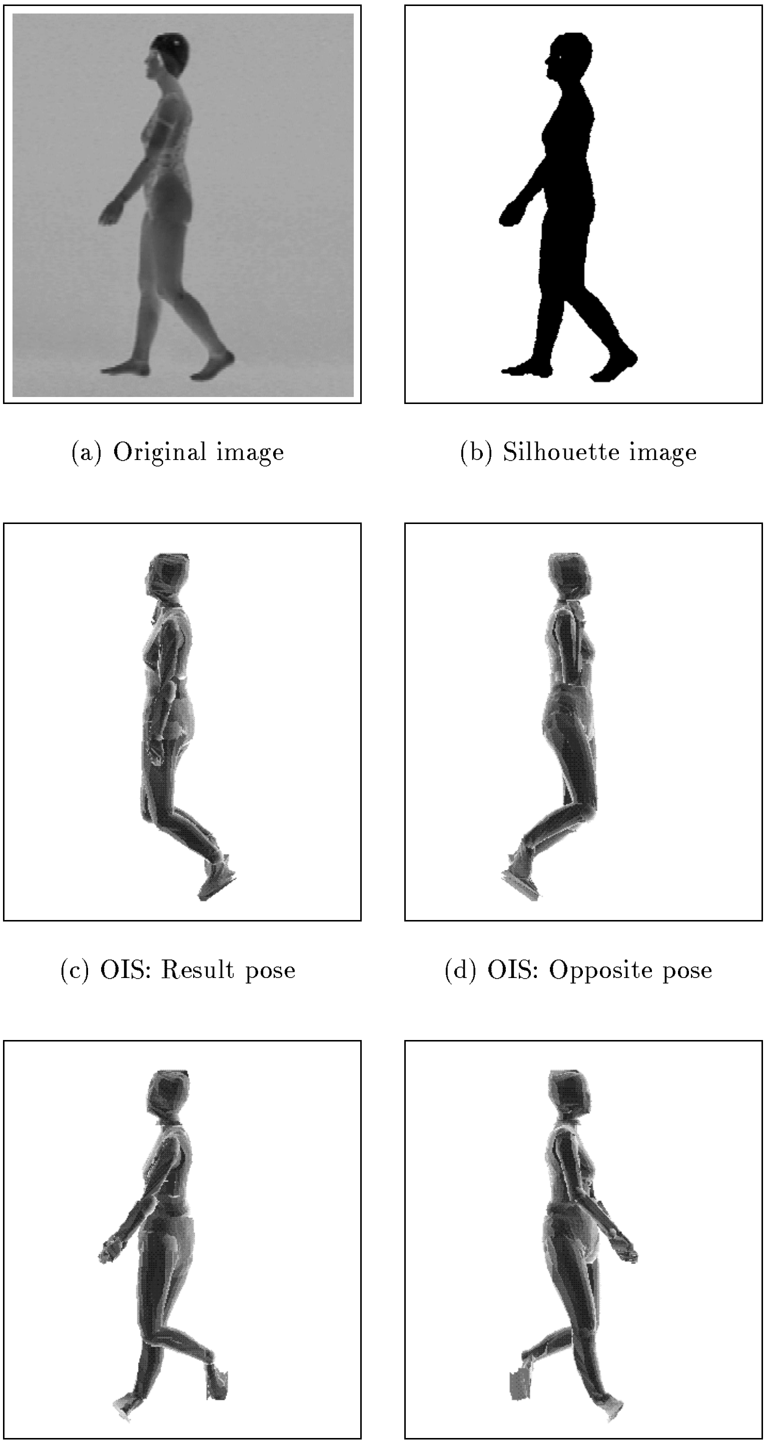
(c) OIS: Result pose

(d) OIS: Opposite pose

(e) ODS: Result pose

(f) ODS: Opposite pose

Figure 2.18: Human body experiment no.1



(a) Original image

(b) Silhouette image

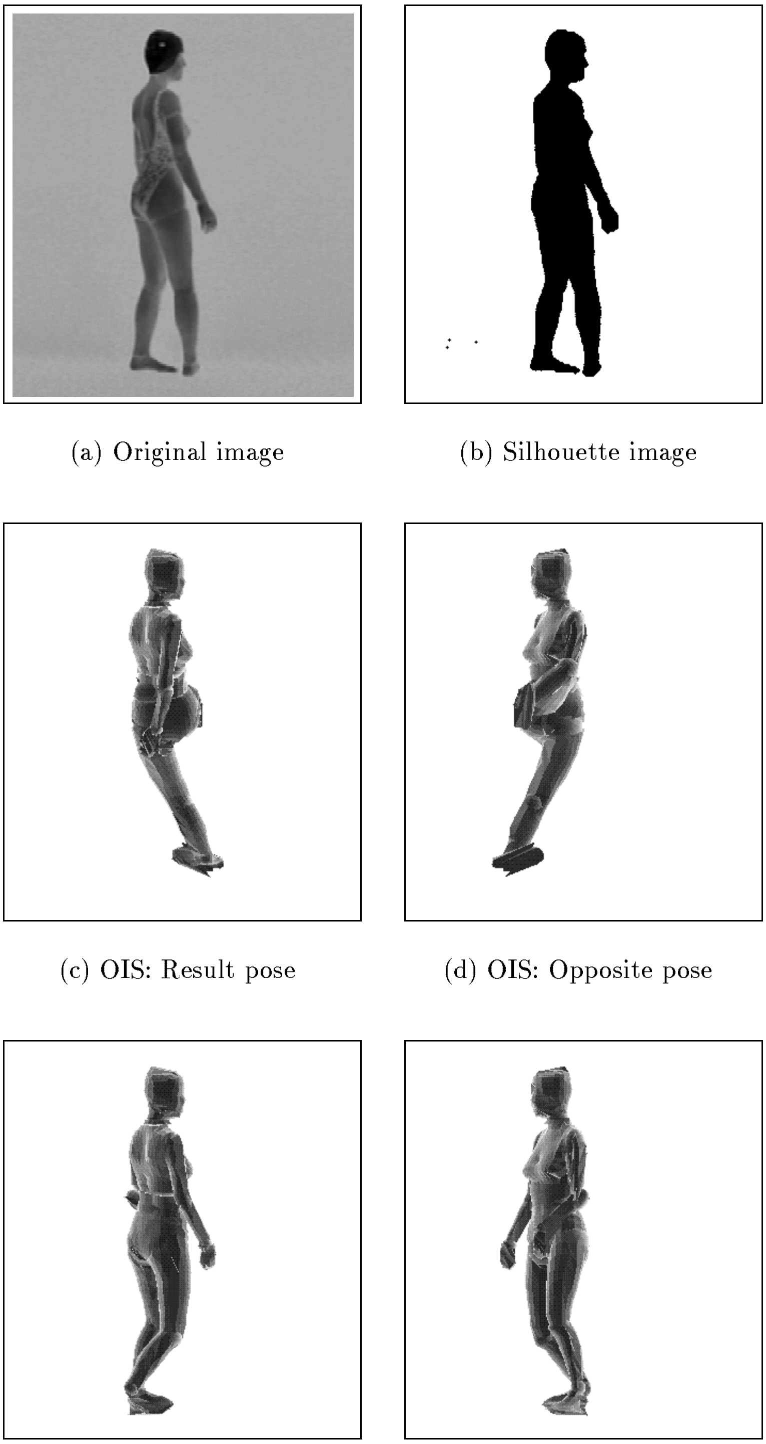
(c) OIS: Result pose

(d) OIS: Opposite pose

(e) ODS: Result pose

(f) ODS: Opposite pose

Figure 2.19: Human body experiment no.2



(a) Original image

(b) Silhouette image

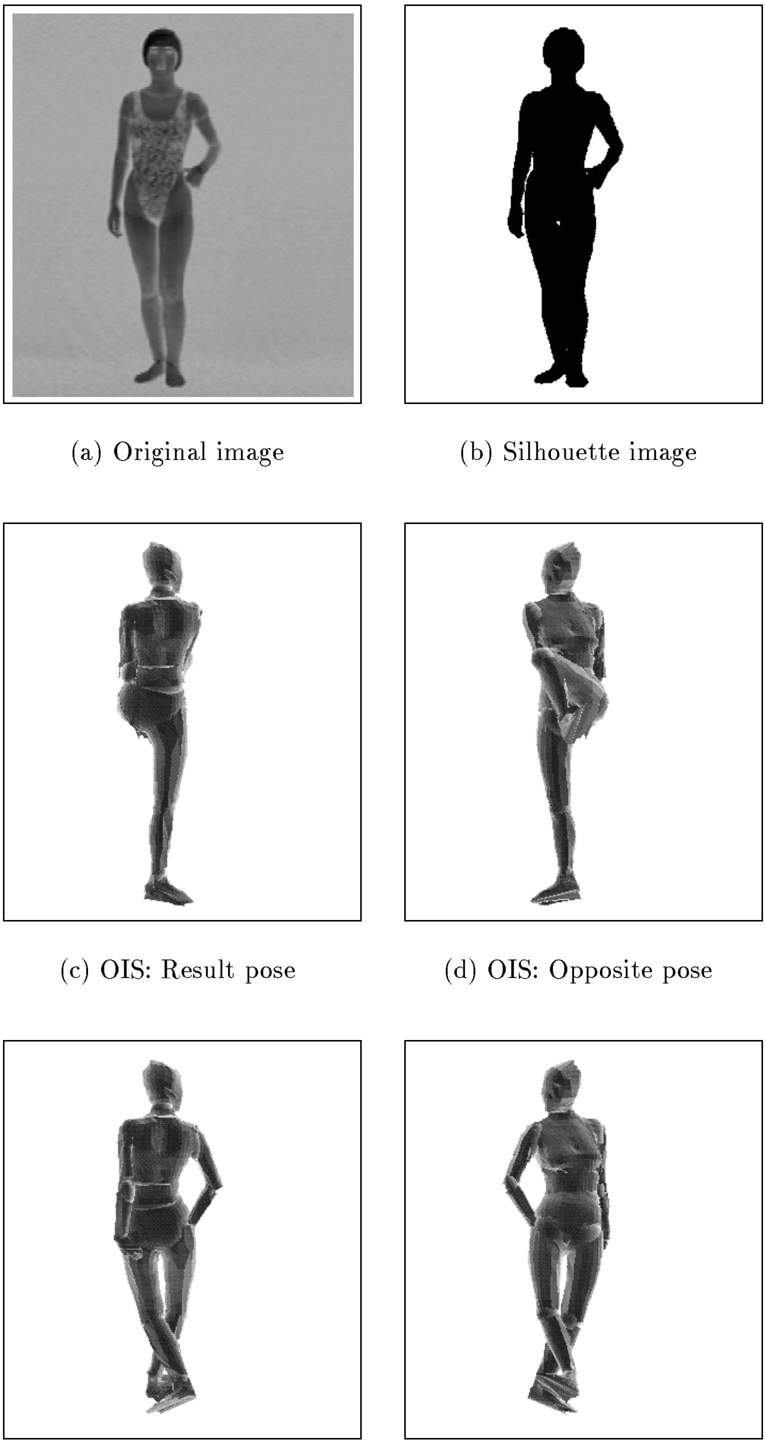
(c) OIS: Result pose

(d) OIS: Opposite pose

(e) ODS: Result pose

(f) ODS: Opposite pose

Figure 2.20: Human body experiment no.3



(a) Original image

(b) Silhouette image

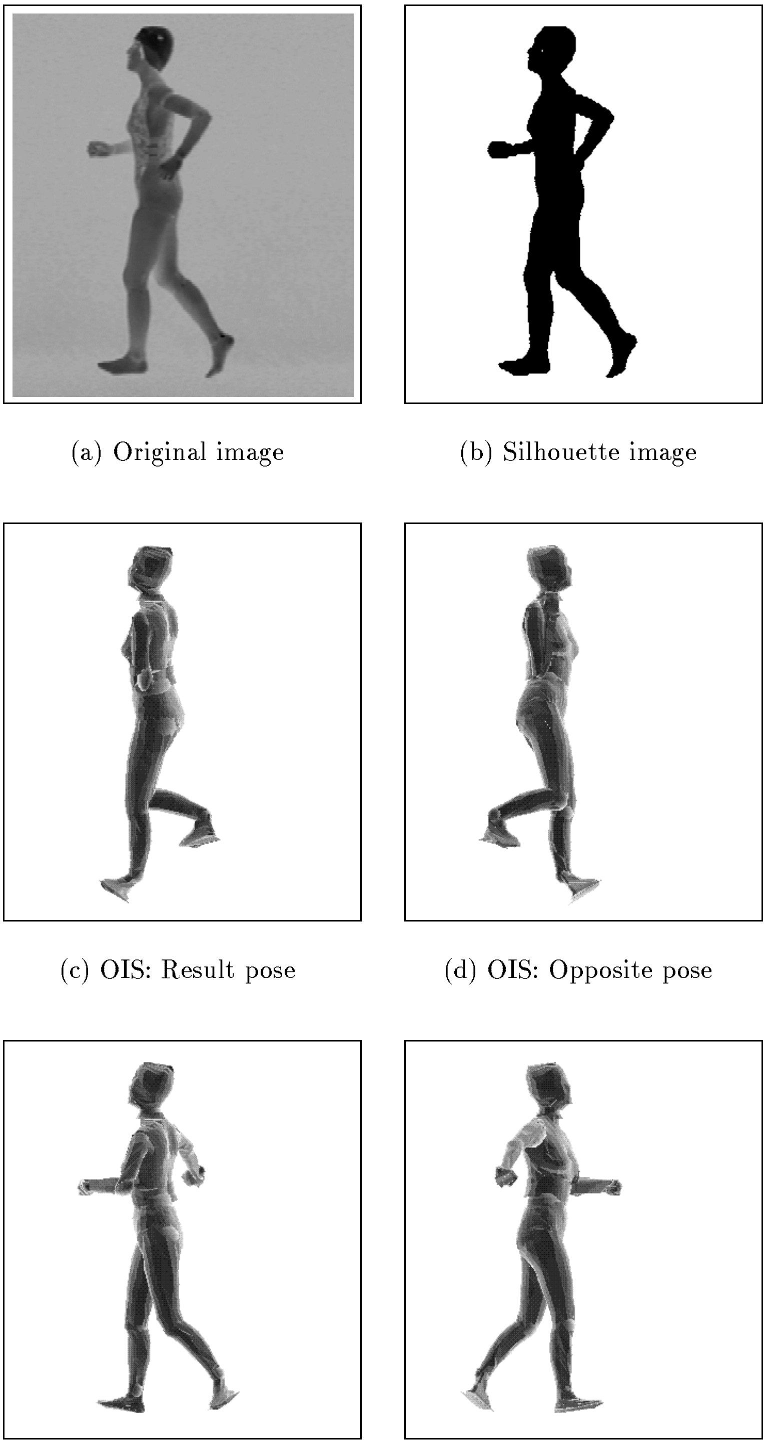
(c) OIS: Result pose

(d) OIS: Opposite pose

(e) ODS: Result pose

(f) ODS: Opposite pose

Figure 2.21: Human body experiment no.4



(a) Original image

(b) Silhouette image

(c) OIS: Result pose

(d) OIS: Opposite pose

(e) ODS: Result pose

(f) ODS: Opposite pose

Figure 2.22: Human body experiment no.5

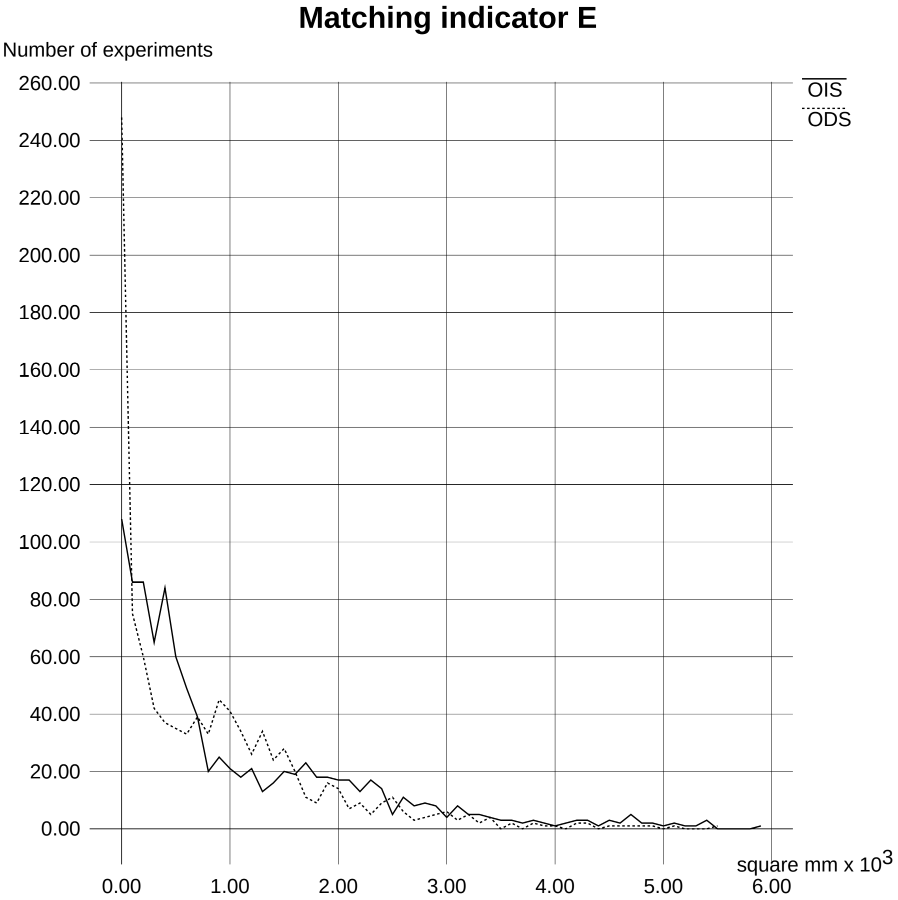


Figure 2.23: CG hand : Results of pose estimation

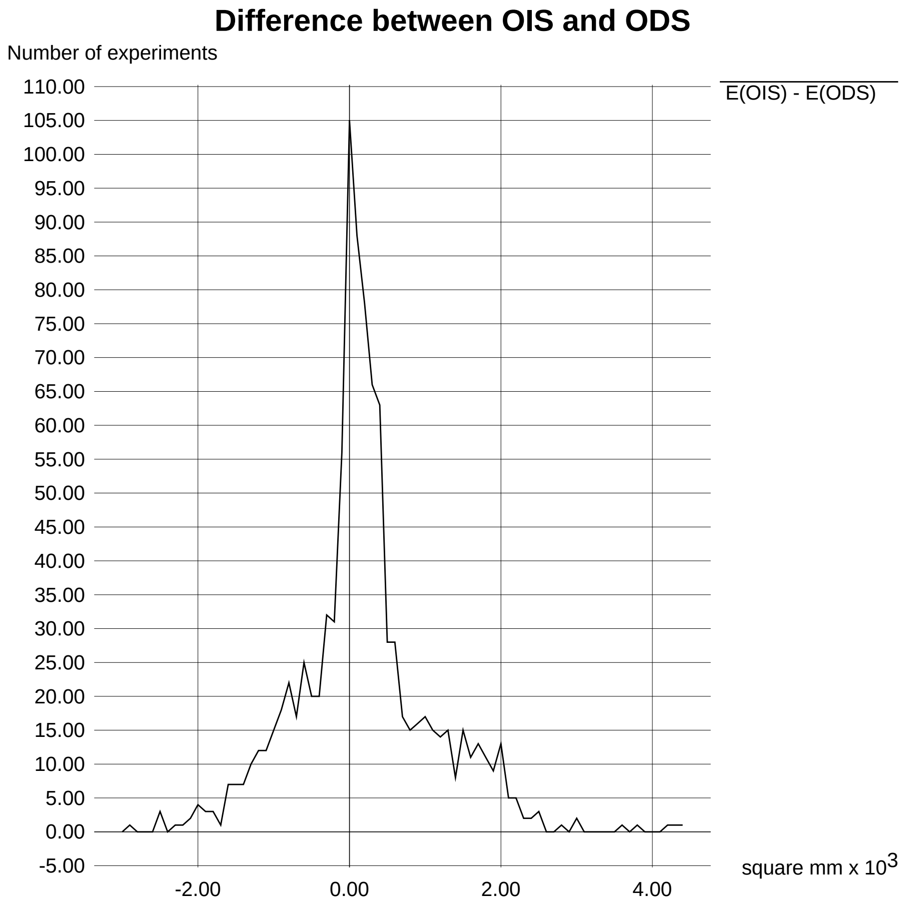
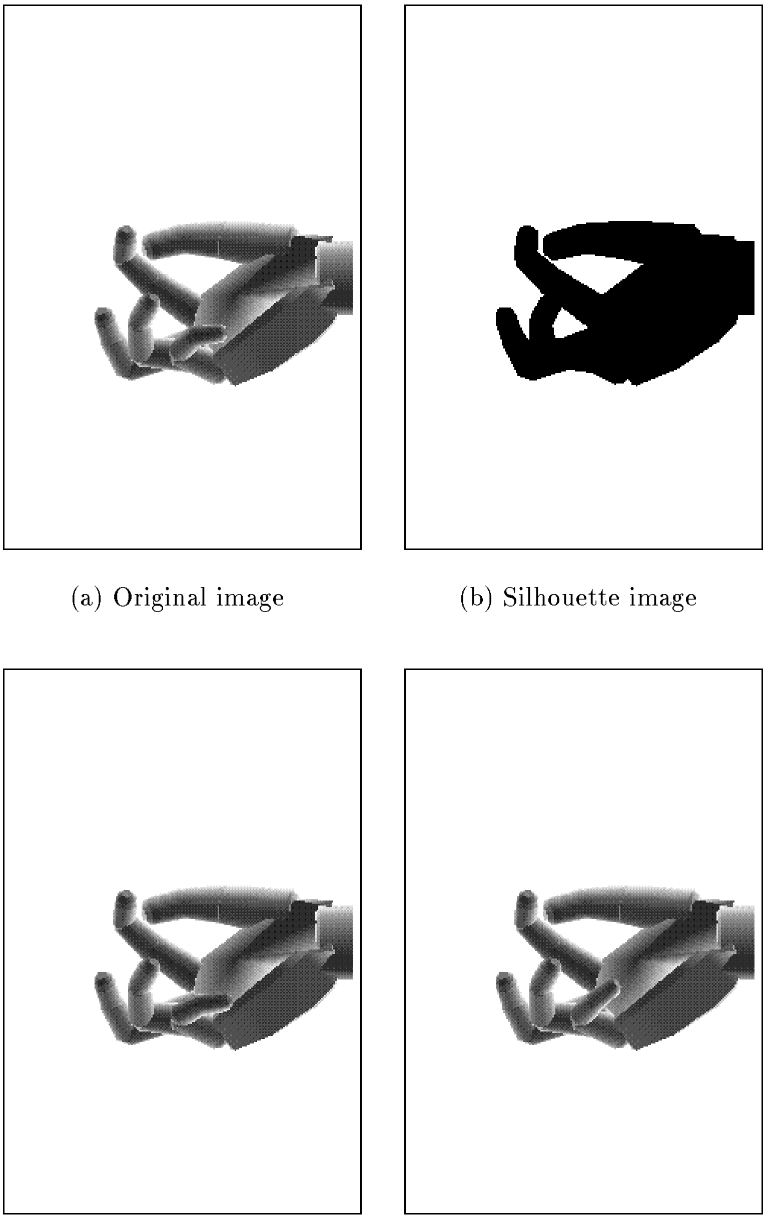


Figure 2.24: CG hand : Difference between OIS and ODS



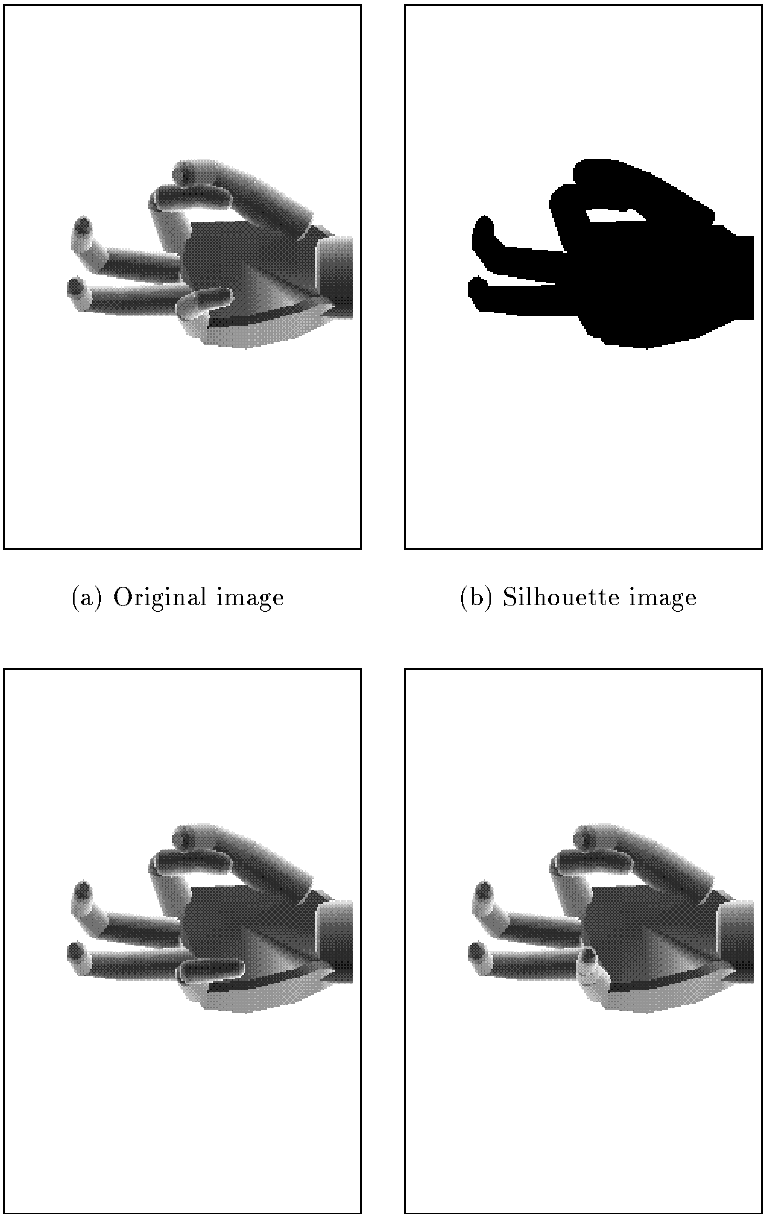
(a) Original image

(b) Silhouette image

(c) OIS: Result pose

(d) ODS: Result pose

Figure 2.25: CG hand : Best sample no.1



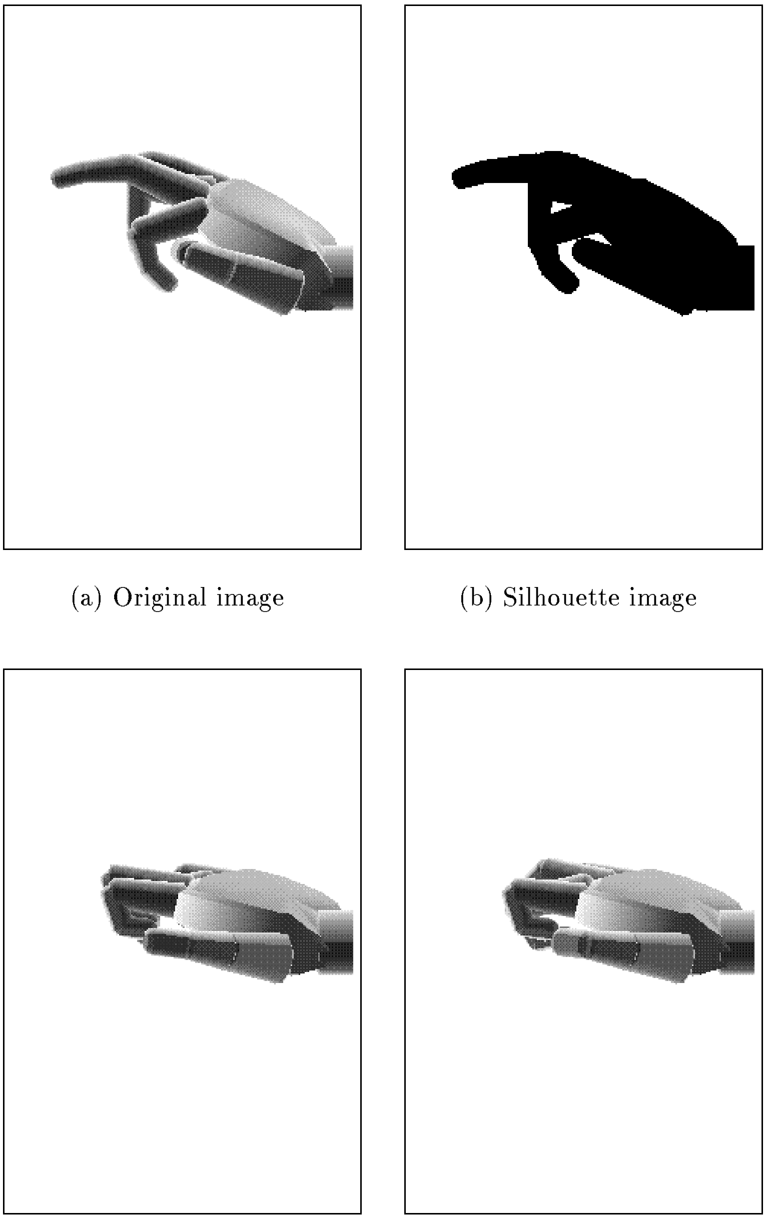
(a) Original image

(b) Silhouette image

(c) OIS: Result pose

(d) ODS: Result pose

Figure 2.26: CG hand : Best sample no.2



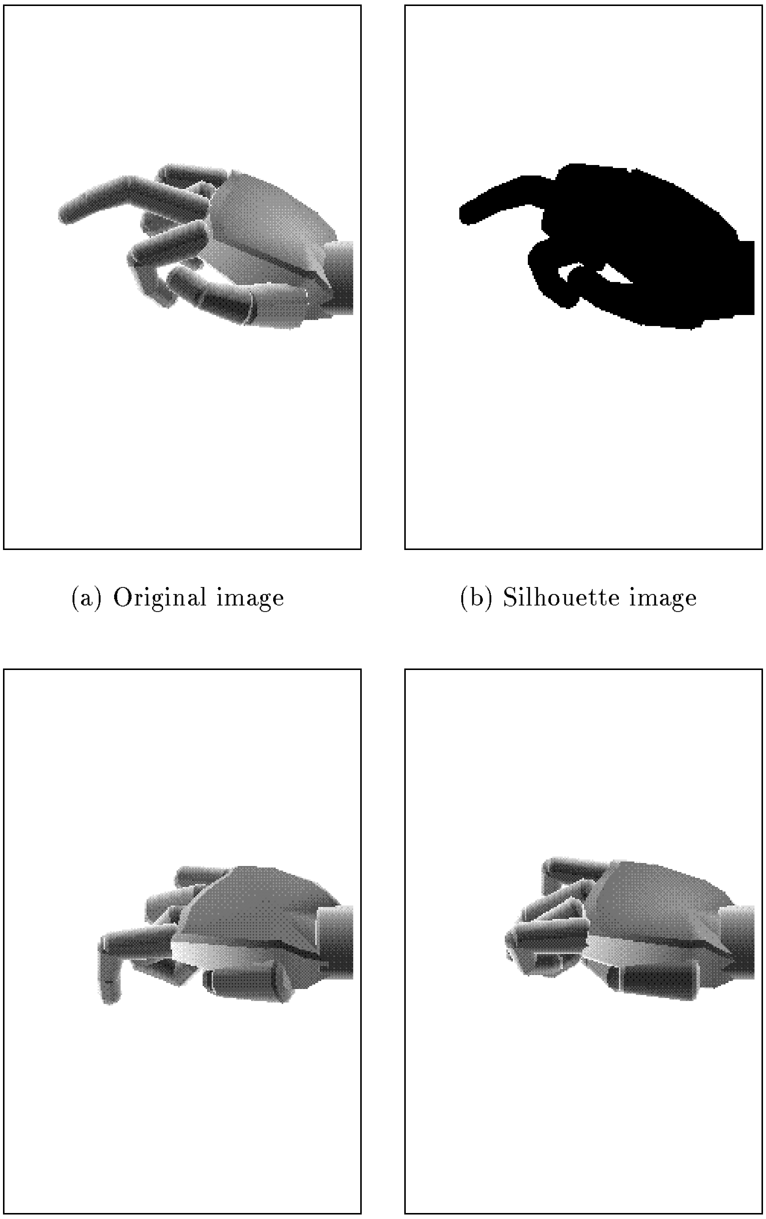
(a) Original image

(b) Silhouette image

(c) OIS: Result pose

(d) ODS: Result pose

Figure 2.27: CG hand : Worst sample no.1



(a) Original image

(b) Silhouette image

(c) OIS: Result pose

(d) ODS: Result pose

Figure 2.28: CG hand : Worst sample no.2

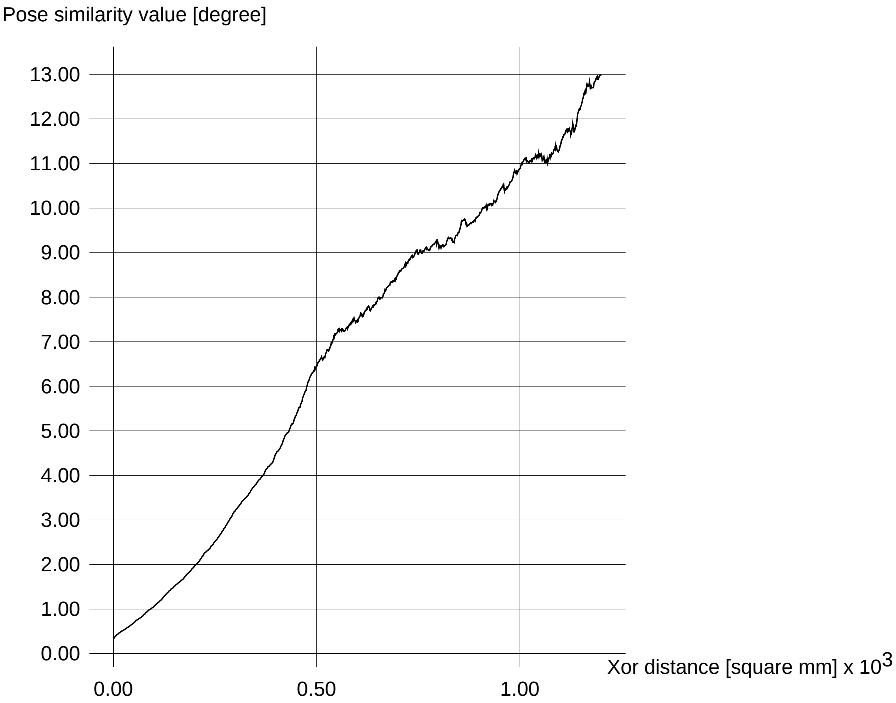


Figure 2.29: Relationship between *xor distance* and *pose similarity value*

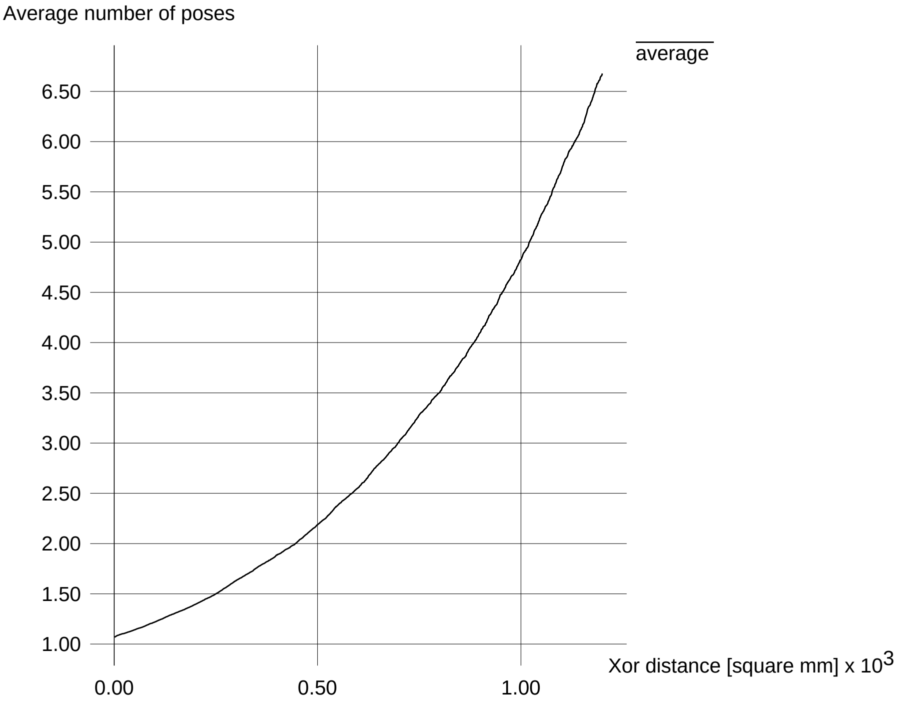


Figure 2.30: Relationship between *xor distance* and the average of the number contained in one silhouette group

Chapter 3

Pose Estimation with Tree Traverse

We have propose a model based pose estimation method for an articulated object from an image in the previous chapter. In this chapter, we discuss the way of finding the resultant pose to improve the estimation accuracy and propose a modified pose estimation method. We prepare corresponding articulated object model in advance all so in this chapter. Our method can estimate human pose without heuristic knowledge in calculating inverse kinematics and in matching procedure at the condition that the model has same functionality as the object. We assume an articulated object is considered to be formed by tree structure of which node corresponds to a body part, and the parts connects to each other by joints. With this assumption, a pose can be defined by a set of all the joint angles in the human body. We implemented our method and evaluate its ability and features with an experiment by simulation experiment on synthesized images, and also conducted an experiment on real images where various poses of human body are photographed.

3.1 Backgrounds

Pose estimation of articulated objects has been required in the some computer applications like human machine communication. We consider a pose as a status of joint angles that indicates bending degree of body parts.

For example, VR reconstruction including human shaped agents needs to estimate the whole human body pose. Freehand tele-operation requires the recognition of finger pose and movement. As the human machine interface, pose estimation could be useful in expressing user's behaviour numerically.

A previous research proposed ribbon feature in estimating a human pose. They consider that many parts in the human body can be approximated by a kind of cylinder, and they turn their attention to the geometric feature of cylinder that projected shape of the cylinders has two parallel lines in orthogonal camera projection. They extract the ribbons from images and infer the three dimensional location of the cylinders in the real space.

Kurakake at el. proposed to estimate a pose of an articulated object by extracting ribbons from edges in an image without defining explicit human functional model, but movement of body parts are limited on a plane perpendicular to the camera direction.

Yunipan at el.[57] proposed the method to estimate a pose of human arm. They first extract skin region with skin color information from an image and then determine whether the extracted region is human arm or not by its shape. the three dimensional pose is inferred by referring the location relationship among the arm regions in the image plane. Like Kuratake's research, the proposed method has limited ability of 2D pose estimation on the image plane.

In these approaches, it is impossible to estimate human pose in thee dimensional world because they don't consider the cubic human pose or motion in their proposed methods. On the contrary, direct 3D measurement approaches have been applied to estimate not only pose and motion but also three dimensional shape of articulated objects. One of the most conventional methods based on image processing is to attach color markers on the human bodies. Apart from image processing, a wear type measurement device such as data-glove and data-suit may be used for three dimensional pose estimation. Unfortunately, these kinds of approaches are not acceptable in many social applications because the equipment of markers or devices on human body cause much stress at the users. We cannot make users wear the markers or devices if they stay in remote places or in hard environment. Therefore, we have to consider a pose estimation method which does not rely on special equipment on a human body.

Alternative approach is to understand images taken by a video camera instead of having the persons wear special equipment. Among various image understanding techniques, model matching method is the most applicable for three dimensional pose estimation of human body. However, conventional methods involves heuristic knowledge in their matching procedure implementation. In other words, some special features extracted from an image is considered to be related with a certain part of the human body model, and they don't mention why the feature should be assigned to the part of the model. For example, many researches assumes that rectangle shaped regions and ribbons should make a match with a arms or fingers in their models, but they don't verify that relation would be acceptable or not [71, 72].

For example, Kimoto at el. represents a human body by a stick model and place each stick three dimensionally by referring the core lines extracted from the images, and achieves the pose estimation of the stick model[54]. On the other hand, Yunipan at el extract the rectangle shaped region effectively because they introduce MRF and utilize the information in their human model, but the model are limited to move only in a plane perpendicular to the camera direction[73].

Several researchers don't estimate shape of the regions in an image. Ishii at el focus on skin color region and estimate a human pose by stereo vision. In this method, however, the correspondence between skin regions and parts in their proposed model has been resolved in advance[74].

Shimada at el proposed a estimation method for human fingers[75], in which they put an heuristic knowledge that sticked out region are to correspond finger tips in their finger model.

We can get an advantage on realizing fast and effective matching procedure for pose estimation if we include the correspondence between image features and models in matching procedure. But, at the same time, we have to admit that this kind of method is difficult to apply various situations.

Therefore, this paper aims a generic formulation and solution that achieves pose estimation which is independent from heuristic knowledge at matching procedure and from the kinds of the objects. To realize this purpose, we use fully defined functional articulated object model. We consider it is not needed to extract complicated image features as the functionality and the shape comes closer to a real object. With this relief, we propose a pose estimation method which use silhouette image feature.

In section3.2, we describe an articulated object and define a corresponding model. In section3.3 we mention why take up silhouette image as input information and discuss its features. We make clear the goal of the approach in Section3.4 and explain the process in our method. Thereafter some experiments were conducted and we discuss validity of this method with the results in Section3.6. At last, in Section3.7, we put concluding remarks in this chapter.

3.2 Articulated Object and Its Model

In this paper, a pose of an articulated object is uniquely determined if and only if joint angles are fixed.

On modeling the articulated object, we make several conditions shown below. These conditions are common natures of most kinds of articulated objects, and so these don't weaken generality of our research.

- A joint is expressed with combination of at most three joint rotation angles.
- A part which belongs to the articulated object is connected with each other at the joints.
- Each part has solid shape.
- Shape around a joint is determined only by joint angles of the joint.
- The parts are arranged in tree shape structure.

Our articulated object model is represented by a tree data structure. We call this data structure a pose decision tree. Each node in the pose decision tree corresponds to one body part in the articulated object and holds shape information and relation information of a joint which locates between the part and its parent part. Table3.1 shows the attributes of the information in the node. A local coordinate system of node i is translated so that its origin comes to joint connection point in a coordinate system of the parent part. Joint rotation is performed in a order of rotation axes $\mathbf{a}_k$ $(k = 0, 1, 2)$. The rotation angle $a_i(k)$ never exceeds the rotational range of the joint.

Table 3.1: Information attributes of a node in a pose decision tree

Node ID	$i, i \geq 0$
Parent Node ID	p_i
Shape	Surface of node i is represented by patches (in local coordinate system of node i)
Joint connection point	A point defined in a local coordinate system of the parent part
Rotation axis	$A_i(k), 0 \leq k < n_i$ (n_i is a number of axes)
Joint angle	$a_i(k), 0 \leq k < n_i$ (n_i is a number of axes)
Rotational range	$r\_low_i(k) \leq a_i(k) \leq r\_high_i(k)$

A joint connection point of root node r represents location of the root part in the world coordinate system. We represent location of the articulated object model in the world coordinate system by that of the point and its rotation by rotation angles $bfa_r = \{a_r(k) | 0 \leq k < n_r\}$.

Whereas the conventional methods approximate body parts by cylinders or its extended representation, our approach adopts patch description which can represent more complicated shape and can approximate the object more precisely.

Figure 3.1 and Figure 3.2 show an shape representation example of an articulated object model. Figure 3.1 describes a part of human chest. Light grey colored patches are

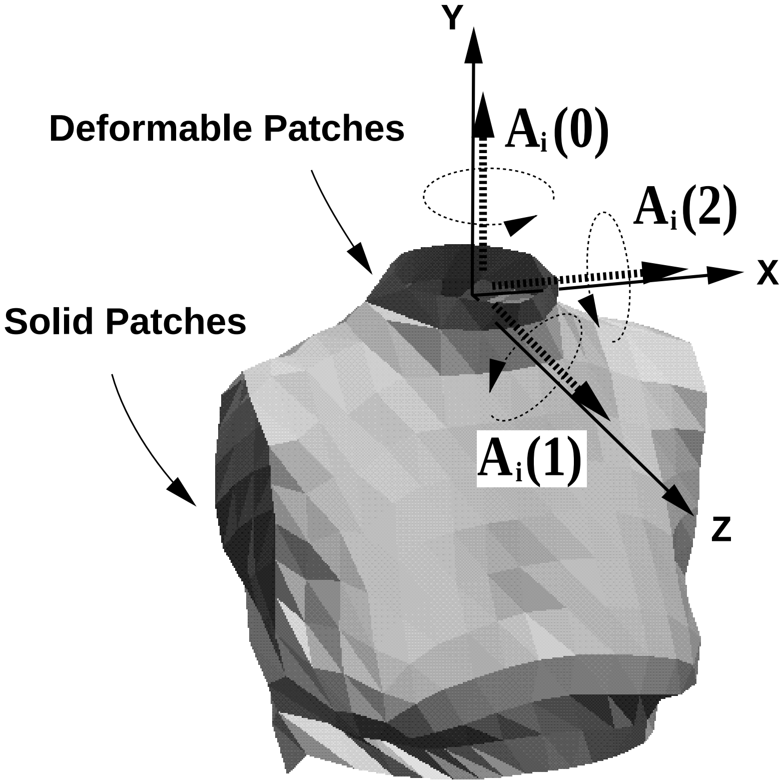


Figure 3.1: A node corresponding to chest

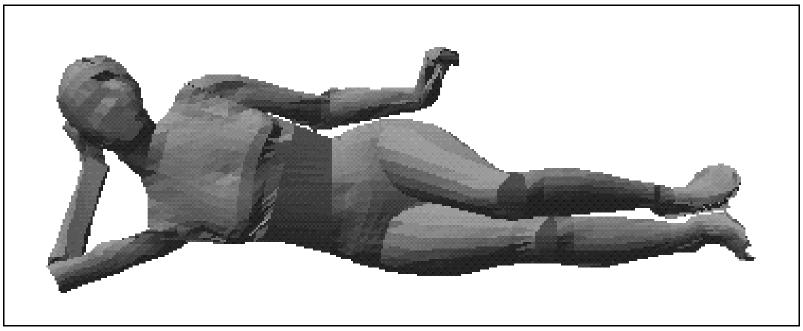


Figure 3.2: A female model

3.3 Input Image

In the pose estimation procedure, it should be examined whether an articulated object exists or not at first, and if it found, the pose will be estimated after the location in the image is extracted. However, as our main discussion concentrates on its pose ant not on its discrimination of the existence, we assume that types of the articulated object and its functionality and structure are known to us in advance, and we also assume that camera parameters used on taking an image is fixed and given to the system, and that the location of the object is also given before pose estimation procedure begins. In other words, the location of the joint connection point at the root node in the articulated model is given in advance. As the model is formed in tree data structure, we can change the root node easily in the model because any node in the tree data structure could be a root node and other nodes be rearranged to form a tree again. Therefore, we take up a node which corresponds to a body part of which location could be most easily determined, and set the node as the root node in the model.

In this paper, we don't consider existence of other moving objects in the image except for the target articulated object.

3.4 Problem Definition of Pose Estimation

As we use two dimensional information such as silhouette image as a clue to estimate a pose of a target object in this research, evaluation of pose estimation also should be done by the dimensional image features.

Suppose a joint angle of node i is represented by $\mathbf{a}_i = \{a_i(k) | 0 \leq k < n_i\}$, a pose of an articulated object model could be described by $\{\mathbf{a}_i\}$. We denote a silhouette region in an input image as S, and a region which is obtained by projecting shape of node i as $P_i(\mathbf{a}_i)$. We define $f(A)$ as a function that calculate area of a region A. The pose estimation could be defined finding a solution that minimizes following formula.

$$J[\{\mathbf{a}_i\}] = f\left(\left(\bigcup_i P_i(\mathbf{a}_i)\right) \oplus S\right) \tag{3.1}$$

In the formula, $\oplus$ is exclusive-or operator between two regions.

If difference of the functionality and solid shape between the target object and the model, minimum value of $J[\{\mathbf{a}\}]$ should be zero. A pose which makes $J[\{\mathbf{a}\}]$ zero is considered to be fitted with the pose in the image. There are two situations for joint angle set $\{\mathbf{a}_i\}$ which makes J zero. In the first situation, only one joint angle set gives the same pose as that in the image, and the other doesn't so even if it makes J zero. In the other situation, several joint angle sets make J zero although their poses doesn't the same as the original pose except for one joint angle set which is the same as the original one. In the latter situation, it implies that some of $\mathbf{a}_i$ could be changed without changing $\bigcup_i P_i(\mathbf{a}_i)$ on the image plane. This is one of the reason that makes three dimensional pose estimation difficult and is called occlusion problem in computer vision.

In the following pose estimation procedure, we stop the procedure at the point where $J[\{\mathbf{a}_i\}]$ comes to minimum value because shape of an object and that of the model are not always the same in practical experiments.

3.5 Pose Estimation

In our research, we take up silhouette images as an input image. We can extract two kinds of image features from silhouette images. One is area information and the other is contour information. These two features are to be used in evaluating the resembleness between projected image of the model and the input image.

It is predicted that a matching evaluation method based on contour information shows good results and realizes good pose estimation if the shape of the model is almost the same as that of the object in the input image. However, this method has too much sensitiveness against a slight gap occurred when the contour of the model is different from the original one. By this reason, we don't use contour information. Area information doesn't have over-sensitiveness against the shape difference, so we consider an evaluation method with the area calculation.

In our matching procedure, we succeed in reducing processing cost of the matching because we use sub-matching procedure according to the tree data structure of the articulated object model. This is effective against a normal approach that handles all the joint angle parameters equally and modify them simultaneously. Since we divide the procedure, there may be a possibility that estimated joint angles fall into local minima. We avoid this local minima by introducing two stage process in the sub-procedure. Convergence of searching the optimal joint angles is defined by the time $J[\{\mathbf{a}_i\}]$ comes to minimal value. We propose to lead the joint angles to the minimal value by using our evaluation definition of the matching between the silhouette region and the body parts in the model, yet this approach doesn't guarantee the convergence to the optimal joint angles.

The matching procedure we propose here is divided into two major sub-procedures. The first procedure is to determine the joint angles of the each node in the pose decision tree sequentially, and the second procedure is to find nodes to be rearranged for getting more appropriate approximation of the pose from the starting point which is given by the result of the first procedure. We explain these two procedures in detail in the following sections.

3.5.1 Pose Estimation Procedure

The pose estimation procedure starts at the root node in the pose decision tree and traverses the tree towards the leaf nodes.

Let $\mathcal{F}$ a node set of pose estimated nodes, and $\mathcal{N}$ a node set that contains nodes which have not been processed yet and of which parent node belongs to $\mathcal{F}$. We denote a node set of reminded nodes $\mathcal{R}$. $\mathcal{F}$ is considered as a set of "fixed" nodes, and $\mathcal{N}$ as a set of "next estimation" nodes, and $\mathcal{R}$ as a set of "reminded" nodes.

At the beginning of the procedure, $\mathcal{N}$ contains only the root node and other nodes belong to $\mathcal{R}$.

The joint angles of the nodes are determined sequentially in the procedure. The procedure will end when all the nodes belong to the node $\mathcal{F}$. Figure 3.3 shows a flowchart of this procedure.

1. **Node Selection**
   Node i with the fewest number of steps to root node is selected in $\mathcal{N}$. If $\mathcal{N} = \emptyset$, the process ends because the equation indicates the pose is fixed.

2. **Calculation of Uncovered Region**
   When all nodes which belong to $\mathcal{F}$ are projected, their region generated on the image is called covered region C, and it is expressed by expression (3.2). In addition, uncoating area in which the covered region C is excluded from S is denoted by S'. S' is expressed by equation (3.3).

$$C = \bigcup_{i\in\mathcal{F}} P_i \tag{3.2}$$

$$S' = S \cap \overline{C} \tag{3.3}$$

3. **Generation of Joint Angle Candidate (JAC)**
   To determine the joint angle of node i, it is necessary to examine the relation between P_i and S. Generally, because P_i changes according to the joint angle, it is thought to express P_i as a function of the joint angle and to process it analytically. However, it is almost impossible to do so because there is no limit in the geometry of each node in this research. Therefore, a sampling method for a movable range of the joint of node i by constant intervals is adopted instead. That is, after P_i of each sampled joint angle is generated, then sampled joint angles are selected which may protrude outside of S but the protruded area smaller than shape error limit area r, and we call the them joint

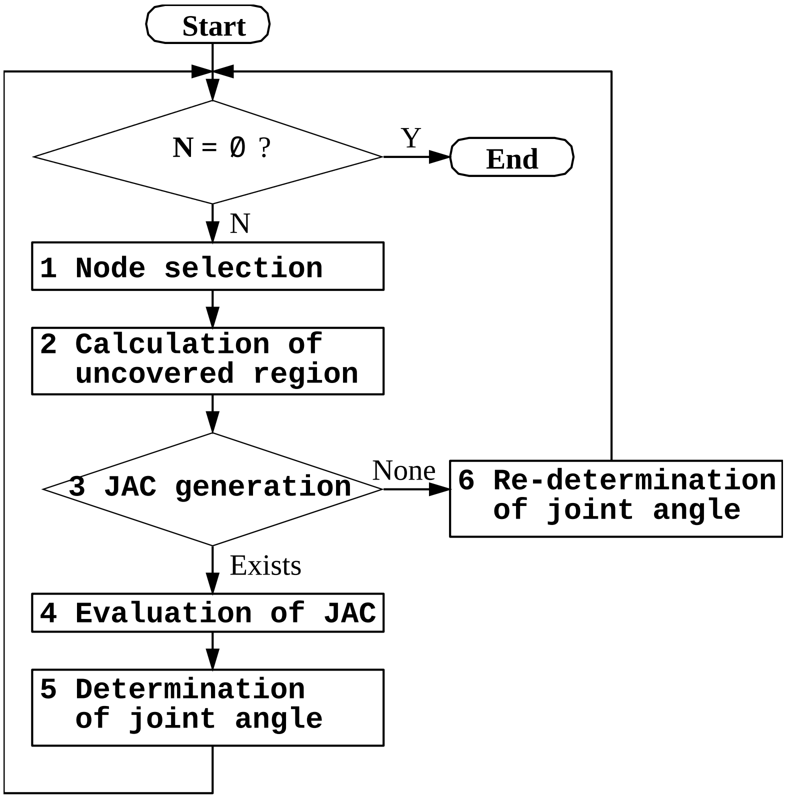

Start
N = Ø ?
Y
End
N
1 Node selection
2 Calculation of uncovered region
3 JAC generation
None
6 Re-determination of joint angle
Exists
4 Evaluation of JAC
5 Determination of joint angle

Figure 3.3: Pose estimation procedure

angle candidate $\mathbf{JAC}_i(t)$. t means this is the tth candidate of the joint angle of node i. $\mathbf{JAC}_i(t)$ is a vector which holds the joint angles $a_i(k)$, $0 \leq k < n_i$ as its element.

$$\mathbf{JAC}_i(t) \;\; where \;\; f\Big(P_i\big(\mathbf{JAC}_i(t)\big) \cap \overline{S}\Big) \leq r \tag{3.4}$$

If any joint angle candidate is not generated, go to Process 6.

4. **Evaluation of JAC**
An evaluation shown in expression (3.5) is done at each joint angle candidate $\mathbf{JAC}_i(t)$. (Refer to Figure 3.4).

$$e_i(t) = f\Big(S' \cap P_i\big(\mathbf{JAC}_i(t)\big)\Big) \tag{3.5}$$

$e_i(t)$ comes to larger and better as the projected region P_i of node i covers larger region over the uncoating area S'. Note that the value is not affected by area size where P_i and covered region C is overlapped. Expression (3.5) is defined from an intention that does not care the size of the overlapped area between P_i and the covered region C but let P_i cover the uncoating area S' as much as possible.

A snapshot of pose estimation procedure which traverses in the pose decision tree at process 4 is displayed at Figure 3.5.

5. **Determination of Joint Angles**
$\mathbf{JAC}_i(t)$ is sequentially arranged from the one with a large value of $e_i(t)$. In this way , a

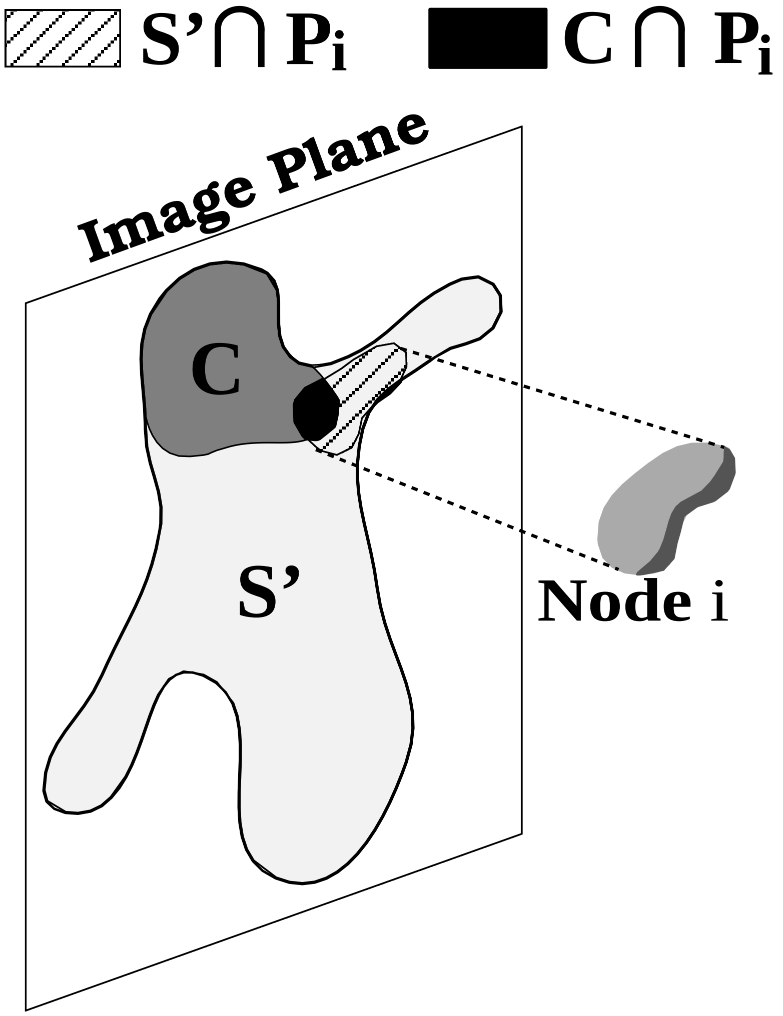


Figure 3.4: Evaluation of a Joint Angle Candidate(JAC) on an image plane

list is made. At this time, t at each joint angle candidate is renumbered according to the list. The first joint angle candidate $\mathbf{JAC}_i(1)$ becomes the joint angle adopted at node i. Node i is moved to $\mathcal{F}$ and its children nodes are moved to $\mathcal{N}$. Go back to process 1.

6. **Re-determination of Joint Angles**
The reason why no joint angle candidates are generated on node i is that incorrect joint angle at any of the nodes in root side has been adopted. Hence, the tree is traced from node i to root node and the process finds the first node j which has the unadopted joint angle candidates. On this node j, u_j shows the order in the list of the joint angle Candidate for whom the process is being adopted now. The joint angle candidate $\mathbf{JAC}_j(u_j)$ adopted by node j is now rejected and the next best joint angle candidate $\mathbf{JAC}_j(u_j + 1)$ is adopted newly. Children nodes of node j is moved to $\mathcal{N}$ again and if they have their children nodes, the grandchildren are moved to $\mathcal{R}$ again. Go back to process 1.

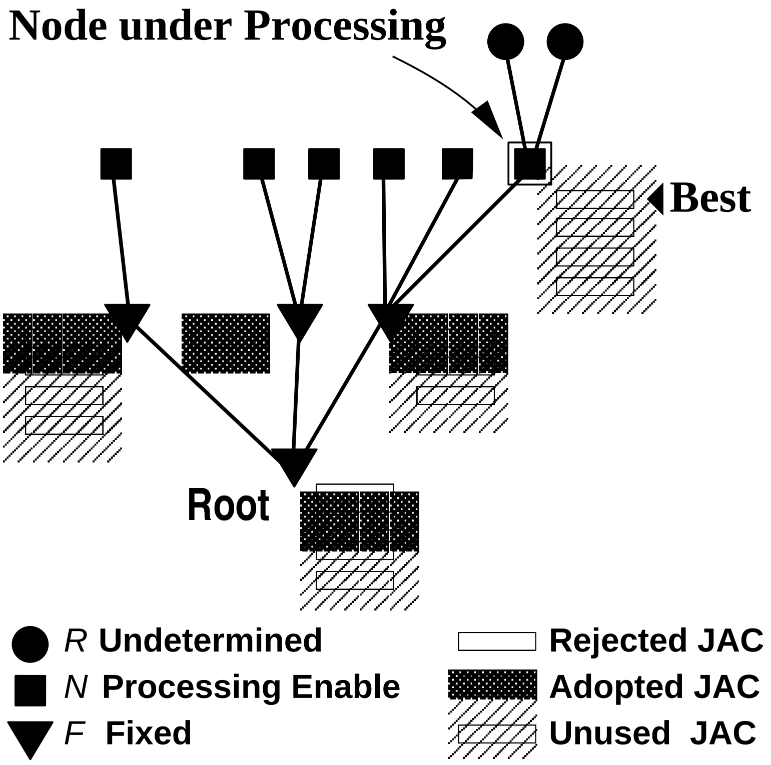


Figure 3.5: A status of a pose decision tree under pose estimation processing

3.5.2 Pose Re-estimation Process

On performing the process described in process 1, a certain pose estimation result is to be obtained when $\mathcal{N} = \emptyset$. Ideally, all poses which minimize $J[\{\mathbf{a}_i\}]$ should be obtained at the end. However, it requires huge calculation cost of matching to obtain the poses. For instance, it requires 5.7×10^8 times matching evaluation at worst with 20 degree sampling interval with a model shown in Figure 3.2 and Table 3.2. Therefore, in the following subsections, we propose two incremental estimation methods that refines a pose which has been obtained once in the previous estimation process so as to find joint angles that minimize $J[\{\mathbf{a}_i\}]$. One is concerning leaf nodes and the other is concerning non-leaf nodes in the pose decision tree.

Combination of Leaf Nodes

When the plural un-adopted joint angle candidates as the leaf node are remaining, it searches combination of those joint angle candidates that gives better matching score of $J[\{\mathbf{a}_i\}]$. If it is found, the pose comes to be the final pose estimation result. When the projection region of two or more leaf nodes overlaps mutually, the improvement of the pose estimation by this processing is effective.

Maximum Exposed Node

When the joint angle is modified at a non-leaf node, it is indispensable to re-estimate all nodes of the descendant joint angles. Therefore, the plan to select the non-leaf node for adjusting its joint angles to improve match result largely influences the calculation cost of the match processing.

Table 3.2: Movable range of joint angles in the human model (unit: *degree*)

Node	Axis	Min	Max	Node	Axis	Min	Max	Node	Axis	Min	Max
Head	Y	-90	90	Right hand	Y	-90	90	Pelvis	Y	-50	50
	Z	-30	30		Z	-45	40		Z	-30	30
	X	-30	30		X	-10	30		X	-20	20
Chest	Y	-60	60	Left upper-arm	X	-180	20	Right thigh	Y	-50	20
	Z	-30	30		Z	-20	90		Z	-40	0
	X	-30	30	Left forearm	Y	-90	20		X	-100	10
Right upper-arm	X	-180	20		X	-160	0	Right calf	X	0	120
	Z	-90	20	Left hand	Y	-90	90	Right foot	X	-40	40
Right forearm	Y	-20	90		Z	-40	45		Y	-40	0
	X	-160	0		X	-10	30		Z	-20	50
Left thigh	Y	-20	50	Left calf	X	0	120	Left foot	X	-40	40
	Z	0	40						Y	0	40
	X	-100	10						Z	-50	20

In this paper, the node with few concealment relation to other nodes is priorly processed by thinking that it is necessary to determine the joint angle earlier than other nodes.

First of all, a set $\mathcal{A}$ of non-leaf nodes which have un-adopted joint angle candidates is calculated. It is assumed that node i is chosen from $\mathcal{A}$ and non-concealment area R_i shown by Expression (3.6) is estimated. This is a part of silhouette region where projection region of all the nodes in the pose decision tree except node i is removed.

Node i is projected onto the non-concealment area R_i according to the joint angle candidate $\mathbf{JAC}_i(t)(t \geq u_i)$, and a value of covered area $cover_i(t)$ is estimated by equation (3.7) for each joint angle candidate. The covered area is estimated for all the nodes in $\mathcal{A}$, and the node m and its joint angle candidate $\mathbf{JAC}_m(t_m)$ is selected which gives the maximum vale of the covered area as defined by formulation (3.8). This node m is called the maximum exposure node.

$$
\begin{aligned}
R_i &= \Big(\bigcup_{k \neq i} P_k\big(\mathbf{JAC}_k(u_k)\big)\Big) \cap S && (3.6)\\
cover_i(t) &= f\Big(R_i \cap P_i\big(\mathbf{JAC}_i(t)\big)\Big) && (3.7)\\
\mathbf{JAC}_m(t_m) & \quad where \quad \max_{m \in \mathcal{A}}\Big(\max_{t_m \geq u_m}\left(cover_m(t_m)\right)\Big) && (3.8)
\end{aligned}
$$

If $t_m = u_m$, after node is eliminated from $\mathcal{A}$, the maximum exposure node and its joint angle candidate are selected again. If $t_m > u_m$, u_m at the maximum exposure node m is set to t_m and its joint angle is modified to $\mathbf{JAC}_m(t_m)$.

It is necessary to re-determine the joint angle of the descendant node of the maximum exposure node m again by the change of the joint angle at the node m. Therefore, the children nodes of the node m belongs to $\mathcal{N}$ and their descendant nodes are moved to $\mathcal{R}$. After that, the method described in foregoing paragraph 3.5.1 is executed again.

If any node does not exist in $\mathcal{A}$, it becomes all pose estimation processing terminations.

3.6 Experiments and Discussion

The experimental results are shown and are discussed about the method of pose estimation to the silhouette image proposed by this paper. The former experiments are done on synthesized images with CG, and the latter experiments are done on real-life object with real image.

3.6.1 Experiment and Discussion on CG Synthesized Images

In the experiment to the CG synthesis image, we took up a female human body as target object. First of all, 219 silhouette images are synthesized with CG by using a female human body model as input images pose estimation process. The joint angle was generated at random by 20 degree sampling interval. It was assumed that location of the joint connection point was already given. The size of the image is 700 pixels in both width and height. As the camera condition, 1.0cm in the real world corresponds to the length of 2.26 pixels.

A movable range of the joint angle of the articulated model used in the experiment is shown in Table 3.2. Figure 3.6 shows the pose of the model when all joint angles are 0. all the local coordinate axes of each node are aligned so as to let X axis go right against the camera, Y axis go top, and Z axis direct to the camera position when all the joint angles are 0. The display order of each axis at each node in Table 3.2 shows the rotation order. The upper axis rotates first, then the second axis rotates if it exists, and the third does. In this articulated model, the head part corresponds to the root node. It is considered that the head is most easily extracted in case of free pose in this research and that is why the head part is selected as the root node.

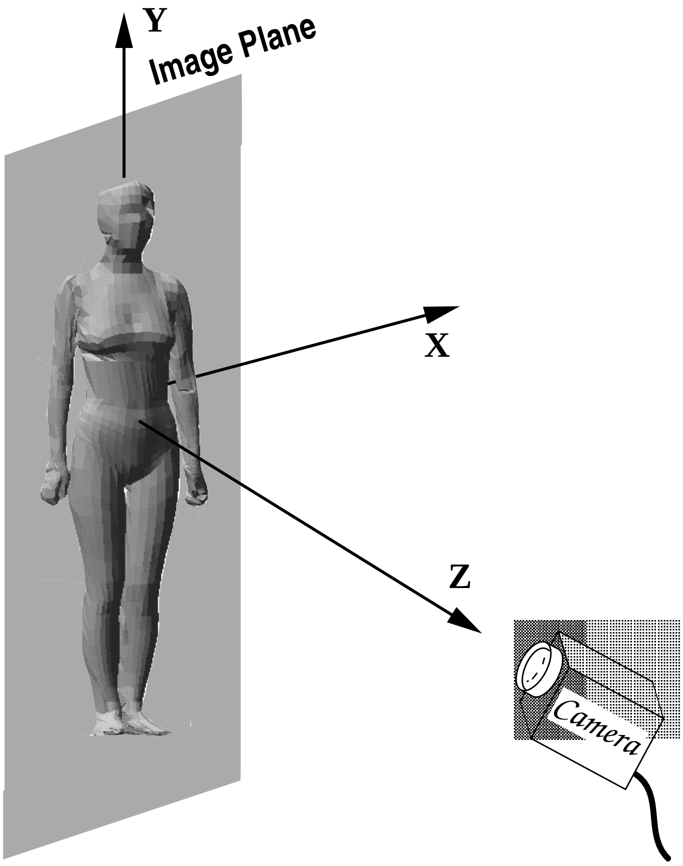


Figure 3.6: Relationship between the uplight model and axes of coordinates

In the pose estimation experiment, the sampling interval for the generation of the joint angle candidates in our system was set to 20 degrees. Moreover, the number of the joint angle candidate is limited to 5 pieces or less. That is, the system lists only five high-ranking candidates of evaluation value $e_i(t)$ when the joint angle candidate is evaluated.

Because the projection region of each node will be included completely in the silhouette region if the shape error limit area is assumed to be $r = 0$ in the CG synthesis image experiment. it is predicted that the pose estimation becomes successful. This is the same as putting assumption that geometry of the articulated model is quite equal to the actual articulated object. However, it is sometimes impossible to give the system to the geometry of the articulated model which is equal to target articulated object completely. Therefore, we provided shape error limit area r in 200 pixels in the experiment. This corresponds to $39cm^2$ and 12.6% of the average projection region of the articulated model.

In this research, the pose estimation is to be minimize $J[\{\mathbf{a}_i\}]$. E_{total} defined by Expression (3.9) is used to normalize and to evaluate the difference of the size of $f(S)$ at each pose here.

$$E_{total} = \frac{J[\{\mathbf{a}_i\}]}{f(S)} \tag{3.9}$$

The average evaluation value of 219 examples is 2.27%. Complete matches of S and $P = \bigcup_i(P_i(\mathbf{a}_i))$ were 135 examples. It indicated that pose estimation succeeded at 61.6% examples. Among these, as for the 101 example, all the joint angles were the same. The several samples are enumerated to Figure 3.7. When the pose estimation is processed, the silhouette image is input to the system though the display is a grey image. These are poses in which self occlusion (Concealment) is rarely observed. On the other hand, S and P are not corresponding and therefore their poses are not the same at the remaining 34 examples. As described in section 3.4, the reason is that pose that satisfies $J[\{\mathbf{a}_i\}] = 0$ is not unique in these examples. Concretely, there are two causes.

One is self occlusion and this is not solved as long as ocellus image is used. Such case is found in 14 examples. Two examples are shown in Figure 3.8. The upper row is an original image and the middle image shows a side-view snapshot of the original pose. The lower image shows a side-view snapshot of the estimated pose.

Another reason is a lack of edge information in the silhouette region. As for this, there were 22 examples. Three examples are shown in Figure 3.9. Original pose is at the left and the right is the estimated result. If edge information in the silhouette region is used in the future research, these examples are considered to be estimated accurately.

The joint angle candidate $\mathbf{JAC}_i(t)$ with comparatively high evaluation value $e_i(t)$ by un-adopted joint angle candidates are left for the pose estimation result when it is not fitted well to the original pose. Therefore, if edge information and other observation information from other camera directions are introduced at a post-processing of this method, the information of the un-adopted candidates is considered to be useful on selecting appropriate node and its joint angle candidate among them.

At 75 examples, the final estimated pose is improved by the pose re-estimation processing and the value of E_{total} comes better than the first estimation result. This shows that local optimal solution was evaded by the pose re-estimation processing. In Figure 3.10, the original pose is shown at the lower image and the system first failed to estimate the pose as shown at the upper image. Starting with the first fully determined pose, our system succeeded in estimating the pose completely by pose re-estimation processing. The number of average processed node, that is, the number of executing processing 3, is 26.2 nodes. Regarding the calculation cost of the match processing, because the minimum of the number of execution is 15, the figure

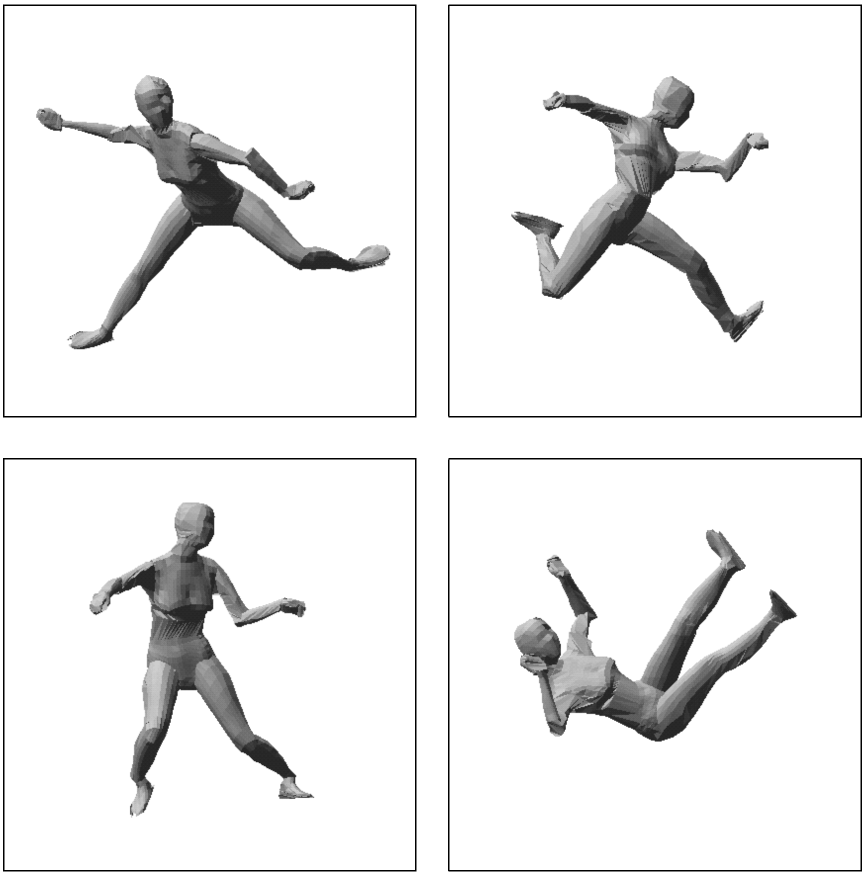


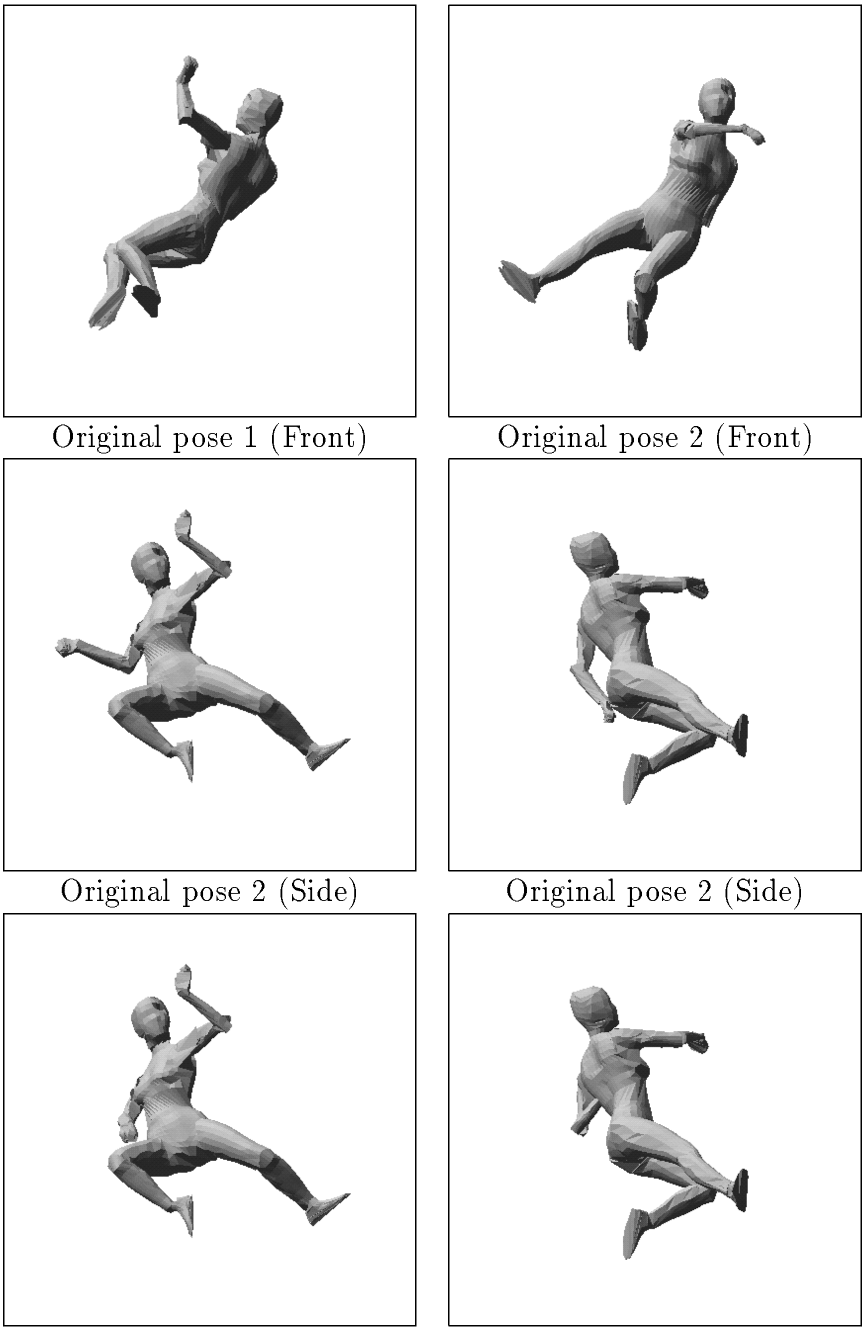
Figure 3.7: Succeeded examples of coincident poses

is quite closer to the minimum number of execution. The distribution of the number of the processed node is shown in Figure 3.11. Remaining average except the seven worst examples becomes 20.1 nodes.

By the way, one reason why some results are not estimated correctly is that the number of the joint angle candidates is limited to at most 5 candidates. One example is shown in Figure 3.12. In this example, a lot of joint angle candidates is generated at the chest node because the chest and the right forearm are located close to each other. Therefore, the complete match between S and P is not obtained because the five high ranks of the joint angle candidates do not contain the correct joint angle. Although it is possible to improve the rate of succeeded estimation by increasing the number of the joint angle candidates in the list, the trade-off becomes a problem because the calculation cost is going to be larger in the exponential order.

3.6.2 Experiment and Discussion on Real-life Objects

An example result of pose estimation is shown in Figure 3.13 for female human body. The camera took an image in which actual female human body was snapped. The woman articulated model is used. The location of the joint connection point is placed at a right ankle root part and set 10 degree as the sampling interval of the joint angle candidates. The reason that the foot part is selected as the root node in the articulated model is because it touches the ground and is easy to extract the location in the image in this situation. The articulated model used is constructed based on the range data of the female human body by [76]. The accuracy of the pose estimation result drops in some measure compared with the experiment



Original pose 1 (Front)

Original pose 2 (Front)

Original pose 2 (Side)

Original pose 2 (Side)

Estimated pose 1 (Side)

Estimated pose 2 (Side)

Figure 3.8: Occluded examples fitted to silhouettes

result of the CG image because setting the joint connection point are not accurate in that articulated model. The image is also rougher than the CG experiment, size of which is 320 pixels in width and 360 pixels in height. We similarly experimented on the 6 example and the average of E_{total} was 22.3% against the average of the silhouette region. It is shown that this method is effective for human body pose estimation.

Moreover, 3 experiments were done for estimating pose of a person's left hand as an example of applying this method to other articulated object. One example is shown in Figure 3.14. Root node of the articulated model is set to a left hand forearm part because it is thought that the location of the forearm could be extracted easily when the hand is targeted. The location

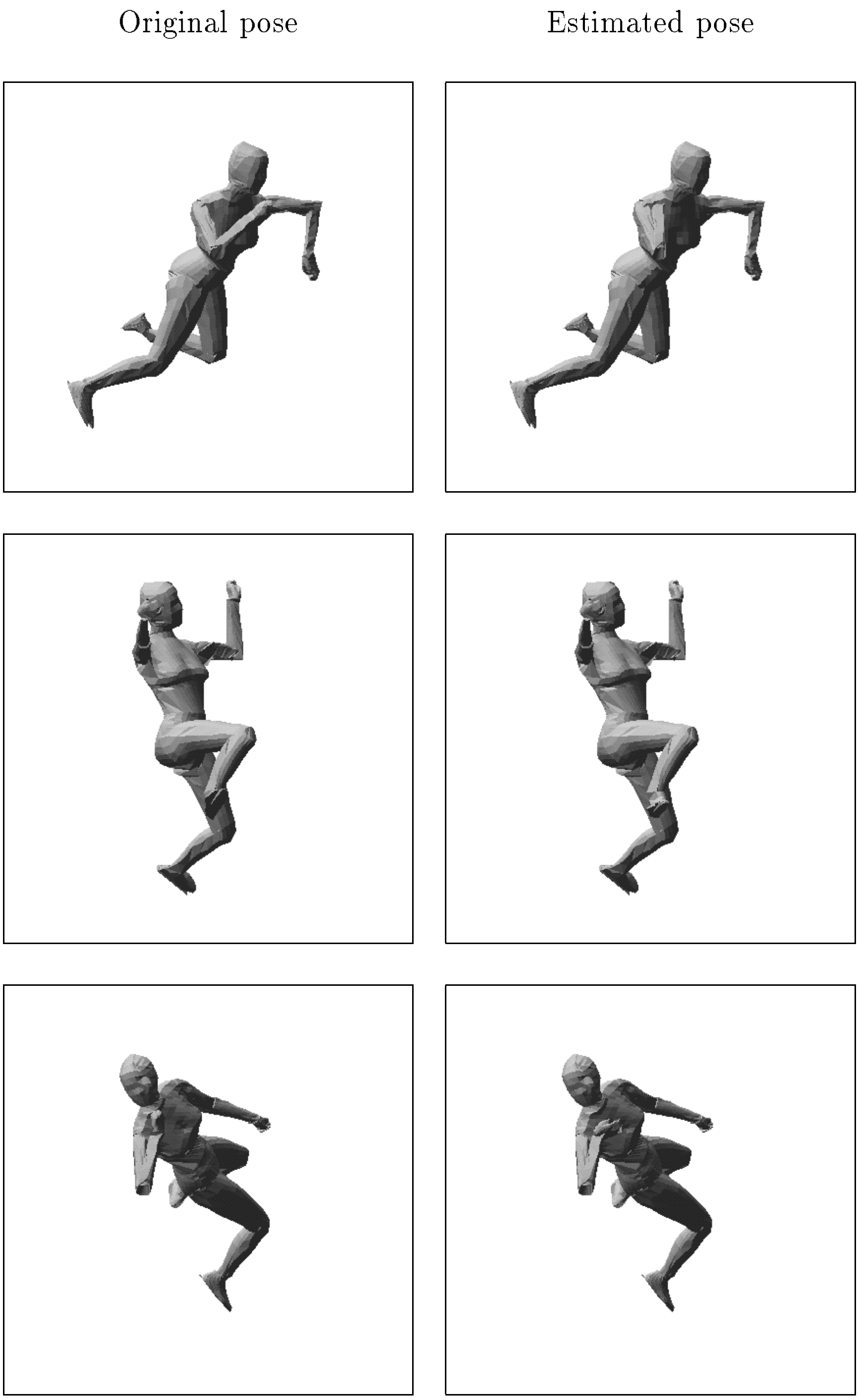
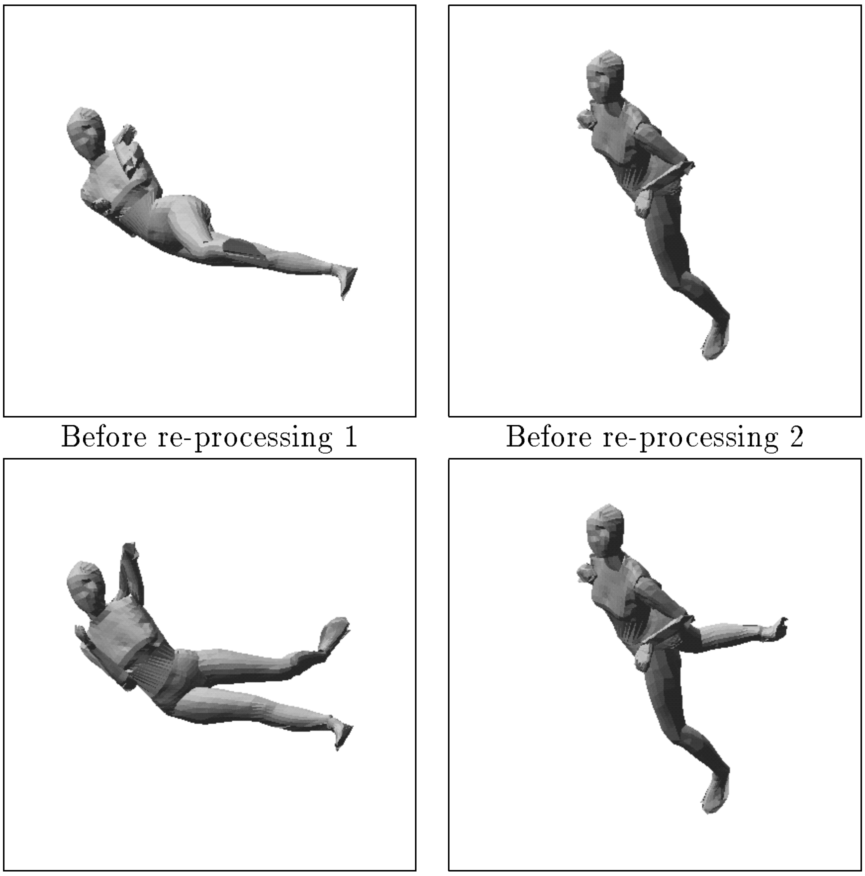


Figure 3.9: Examples fitted to silhouettes under lack of edge information

is assumed to be already-known in the experiment. The size of the image is 448 pixel in width and 400 pixels elements in height. Because the geometry of the articulated model and that of the Hand are different in this experiment, they don't make match completely in the images. The average of E_{total} by the 3 example was 17.2%.

It was proved to be able to estimate the pose of the articulated object by our method with these experiments on the real-life objects.



Before re-processing 1

Before re-processing 2

Final result 1

Final result 2

Figure 3.10: Improved examples

3.7 Conclusion

In this chapter, we proposed the method of estimating the pose of the articulated object framed by one silhouette image using the articulated model. Because this method neither uses heuristic knowledge in image processing nor inverse kinematics approach, it is applicable in the pose estimation for various kinds of articulated objects. We clarified the ability of the method with CG image experiments. The limitation of using one silhouette for pose estimation is also discussed in this chapter. In addition, the experiments on real-life human body and human hand were conducted and showed pose estimation results.

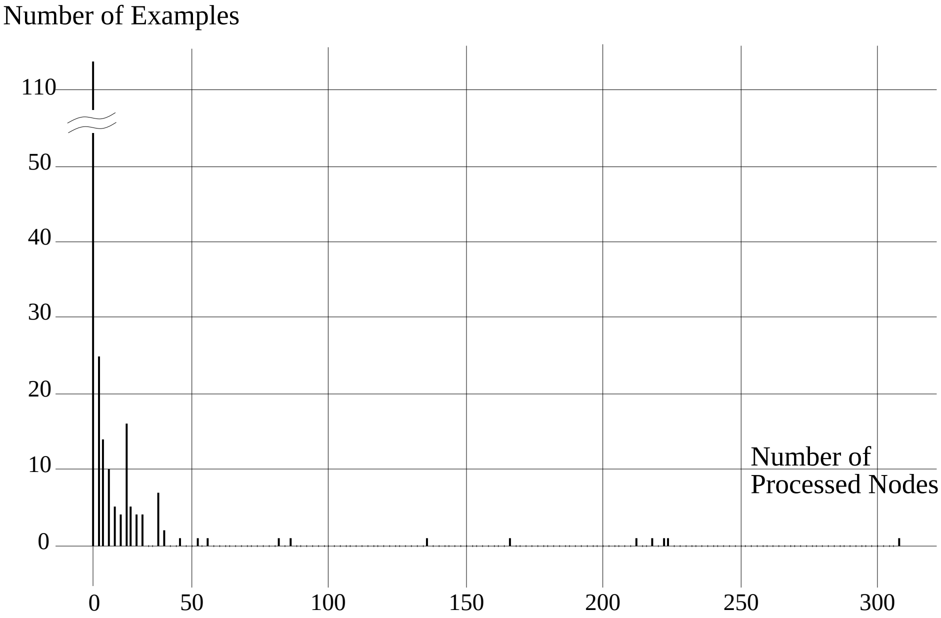


Figure 3.11: Number of processed nodes

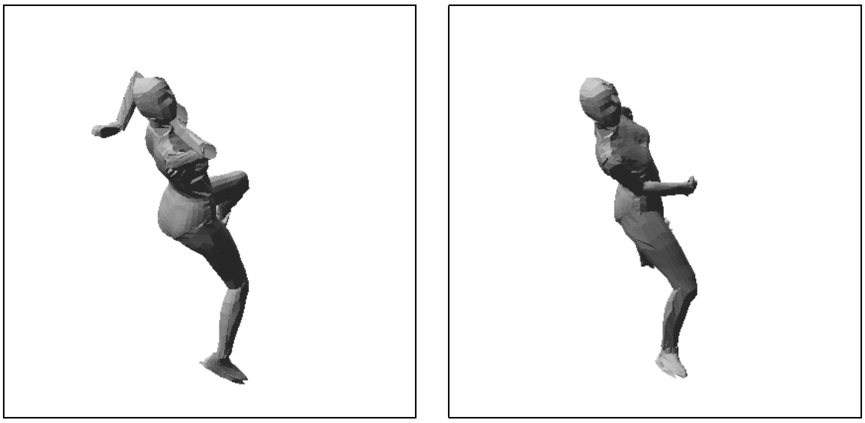
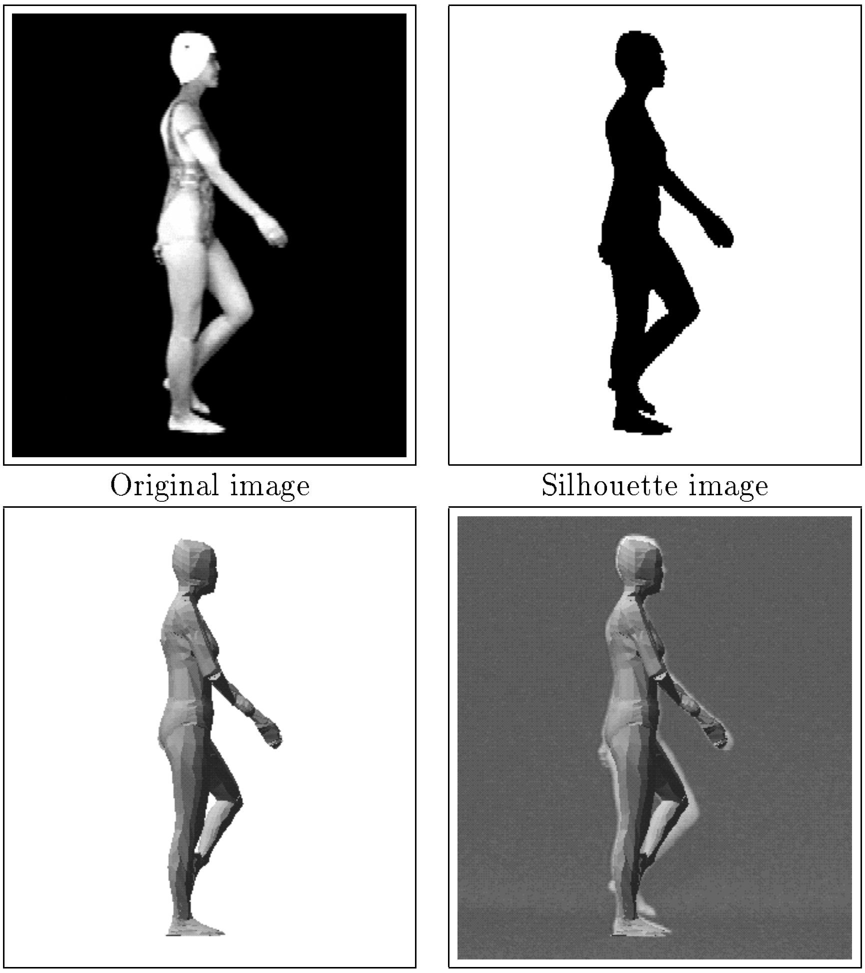


Figure 3.12: An example misfitted to silhouette



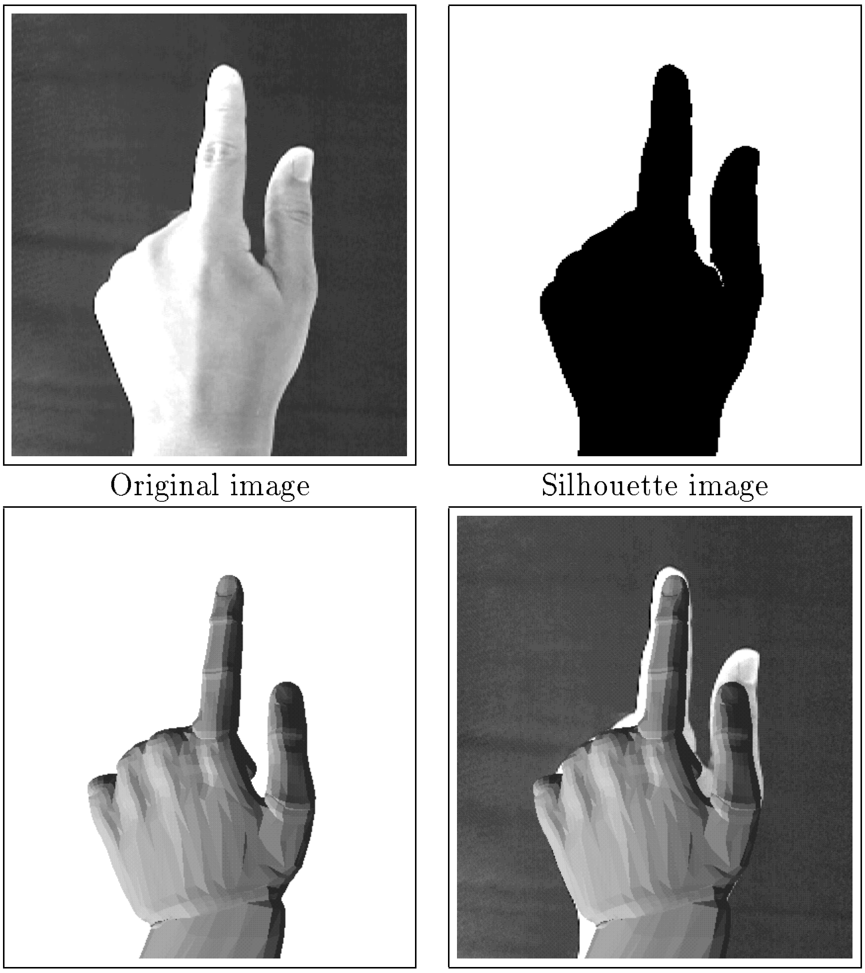
Original image

Silhouette image

Estimated pose

Imposed image

Figure 3.13: A pose estimation example for female human body



Original image

Silhouette image

Estimated pose

Imposed image

Figure 3.14: A pose estimation example of a hand

Chapter 4

Pose Estimation Based on Edge and Gnawed-Region Evaluation

In this chapter, we introduce "gnawed region" on evaluating matching score at each node to make each body node to cover the silhouette region more precisely. It assumes that the object's shape information and the camera calibration parameters have been given to the system in advance. We first show a edge based method which uses a contour of the silhouette and then apply it to the pose estimation of a real-life object. The method presented here has been tested on several silhouettes, including both computer generated and real-life humans.

4.1 Introduction

Recognizing a pose of articulated objects including human beings could be very useful. It could be applied to develop a non-contact input device in exercise physiology, free hand pointing device in man machine communication, and could also be a starting point towards body-language recognition.

Several model-matching methods have been proposed where the strategy depends too much upon the object of estimation, and the information about the model is often only implicitly implemented in their algorithm[55, 56, 54]. If the model is implicitly defined in the algorithm, it is difficult to know which part of the algorithm uses the model information. In such a case, the method could hardly be applied to other target objects. We perfectly separate the model definition from the algorithm.

In the model-matching method, the selection of image clues is an important issue. Much research has applied edges as the image clue[55, 56, 54]. However, edge detection has a tendency to extract a large amount of noise and it is hard to find out the desired edges. On the contrary, silhouette extraction has the advantage that it rarely involves noise though a silhouette has less information than edges. We use the silhouette information in our model.

We explain our articulated object model and the contour based method in Section 4.2, and apply it to a real-life object in Section 4.3. Experimental results are shown in each section.

4.2 Contour Based Method

4.2.1 Model

An articulated object under consideration is defined as consisting of several solid parts which are arranged in a tree graph structure. Each part holds its shape information by a patch-modeling method so that we can cope with the wide variety of geometric shapes.

Each part in the model also holds the joint information that connects the part to the parent part. The joint has at most three axes around which to rotate the part, and at each axis the range of the rotation angle is defined. Only the root part has the location information in the 3D world, assumed to be given to the system in advance. Therefore, to estimate the pose of the articulated object is equivalent to determine all the rotation angle values in the model.

4.2.2 Algorithm

In this section, we describe a contour based algorithm to estimate a pose of the articulated object. This algorithm uses the definition of the model, but does not use the values in the model so we can easily change the target object without modifying the algorithm.

Each part in the model is taken up one by one and its rotation angles are determined based on the overlap relationship between the contour of the silhouette and that of the projected region on the image plane.

Suppose a set $\mathcal{F}$ has the parts whose rotation angles have been determined and a set $\mathcal{N}$ has the parts whose parent part belongs to $\mathcal{F}$. The rest parts belong to a set $\mathcal{R}$. At the beginning $\mathcal{N}$ only contains the root part and all other parts are in $\mathcal{R}$.

1. **Selection of the Part**
   Select a Part i in $\mathcal{N}$. If there is no part in $\mathcal{N}$, the algorithm terminates. Make a new candidate-list for Part i. A candidate in the candidate-list represents a possible pose of Part i, and has at most three rotation angle values which are quantized by a certain unit interval. The unit size defines the resolution of the pose estimation. A candidate can not have values that make the projected region of Part i stray out from the silhouette. If there exist no candidates in the candidate-list, go to Step 3.

2. **Estimation of the angles**
   To find the best pose of Part i, the system measures the length of the contour where the contour of the silhouette overlaps with that of the projected region for each candidate in the candidate-list. The candidate with the largest overlap is adopted as the estimated result. The rotation angles are fixed to the values of the candidate, and then it is removed from the candidate-list. Move Part i from $\mathcal{N}$ to $\mathcal{F}$, and the children of Part i in $\mathcal{R}$ are moved to $\mathcal{N}$. Go to Step 1.

3. **Backtracking**
   Backtrack from Part i to the root part until Part j, which keeps at least one candidate in the candidate-list, is found. Move all of its children descendants into $\mathcal{R}$. For Part j, execute the same algorithm as Step 2. Go to Step 1.

Since it is not clear what kind of criterion is necessary to select the part in Step 1, our method here selects it arbitrarily. We are currently investigating this problem.

4.2.3 Experimental Results

There are two evaluation criteria for the pose estimation in our method. One is whether the projected region of the estimated pose coincides with the silhouette given to the system, and the other is whether the estimated rotation angles are same as those of the original pose. Since we use silhouettes as input information, there may occur cases whose results qualify the former criterion but not the latter one.

To evaluate the performance of our proposed method, we conducted an experiment with computer generated silhouette images of a human body. The model we used consisted of 17 parts where the root part corresponded to the head. The size of the silhouettes was 500 pixels by 500 pixels and the target object was scaled to 4.42 *mm* per pixel. The unit for generating candidates in Step 1 was set to 20 degree.

We experimented with 1,843 cases and in 1,107 cases the estimated results qualified the former criterion. Among them, 847 cases qualified the latter criterion. In the remaining 260 cases, the estimated results contained at least one part whose rotation angles were not determined uniquely. Table 4.1 shows the number of ambiguous cases.

Table 4.1: Number of ambiguous cases

Number of Parts	Ambiguous Cases
1	172
2	71
3	14
4	3

Figure 4.1 shows two cases which qualify both criteria. Figure 4.2 shows a case which qualifies only the former one. In this case, the pose of the left arm which consists of three parts could not be determined uniquely.

There were 966 cases in which Step 3 was not used (that means the method executed Step 1 just 17 times). Concerning the rest 141 cases, the method could clear the former criterion because the backtrack in Step 3 was applied. Figure 4.3 represents the distribution of the executed times of Step 1 in these cases.

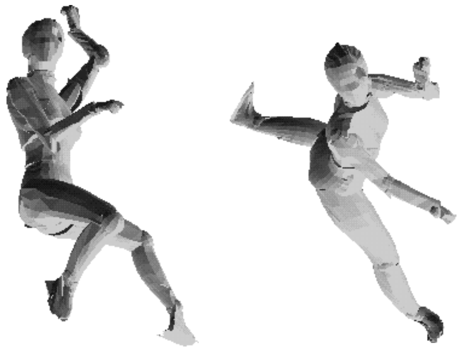


Figure 4.1: Uniquely estimated cases

736 cases did not satisfy the former criterion in this experiment. Figure 4.4 shows the distribution of the silhouette areas not covered by the projected regions. These failures occur because the method does not examine the uncovered silhouette area during the process progression.

4.3 Application to a Real-Life Object

Our proposed method shows its applicability through the experiment for computer generated images. However, when we take up a real-life object as the target a problem arises: the

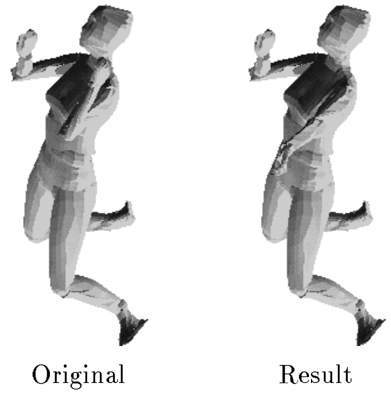


Figure 4.2: Ambiguously estimated case

geometric features of the model might not be precisely the same as those of the real-life object.

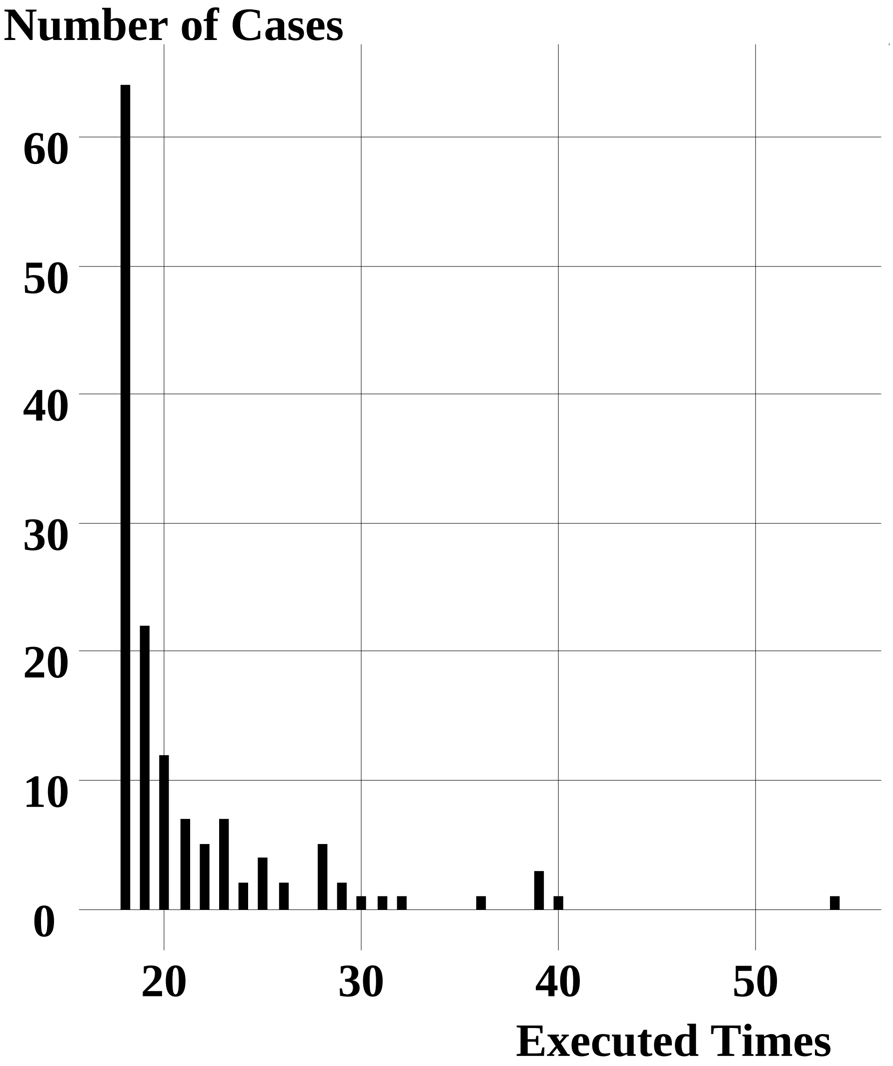


Figure 4.3: Executed times of step 1

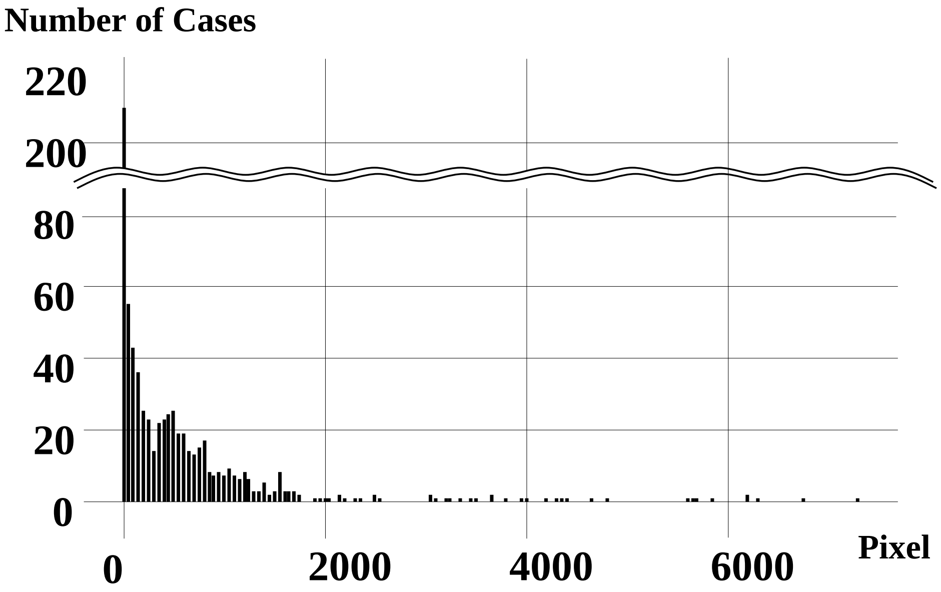


Figure 4.4: Uncovered silhouette area

Since the method previously proposed refers only the contour information and therefore is sensitive to the noise on the contour, it is essential to extend the method to overcome this problem. The method is modified on two points.

1. In Step 1, generated candidates must not project Part i in a way that more than a certain amount of the projected region strays from the silhouette. The amount should be determined according to the geometric unlikeness of Part i to the target real-life object.

2. In Step 2, the system first makes a "gnawed image." It is an image in which the silhouette contour is removed from the given image and in which the regions projected by the parts in $\mathcal{F}$ are also removed. Then, the system measures the exclusive OR area between the silhouette on the gnawed image and the projected region for each candidate. The candidate which makes the smallest exclusive OR area is selected as the estimation of the rotation angles of Part i.

As an example, suppose the model is the same as that used in the previous section and consider the situation when $\mathcal{F}$ contains the head, neck, and breast part. The gnawed image at that time is shown in Figure 4.5. The bright gray colored contour corresponds to the removed contour and the dark gray colored region corresponds to the removed projection region. Exclusive OR calculation is worked out only in white or black colored regions. The dark gray colored region has the role, like Step 2 in the previous proposed method, of attracting the edge of the projected region to the silhouette contour. In addition, since the bright gray colored contour is out of consideration on counting the exclusive OR area, the system acquires a tendency to rotate the parts in order to cover as much of the black colored region as possible.

We have implemented the modified method and tested several cases for an image of a woman. In this experiment, the resolution of the pose estimation is set to 20 degree and silhouette images are 320 pixels by 360 pixels. One of the results is shown in Figure 4.6. The estimated pose is quite similar to the target woman's pose in the original image. We show another case in Figure 4.7, where the breast part is posed in a slightly wrong way. As a result, the estimation of the right arm failed. The reason for this type of failure is considered to be the method that does not examine uncovered silhouette regions during the process. However, it would be very expensive to compute a prediction for the part to be processed next, so that

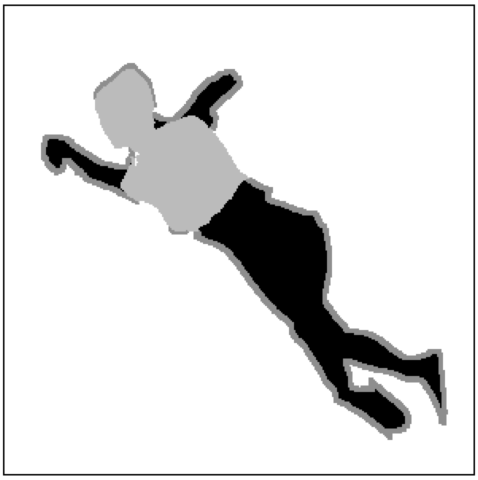
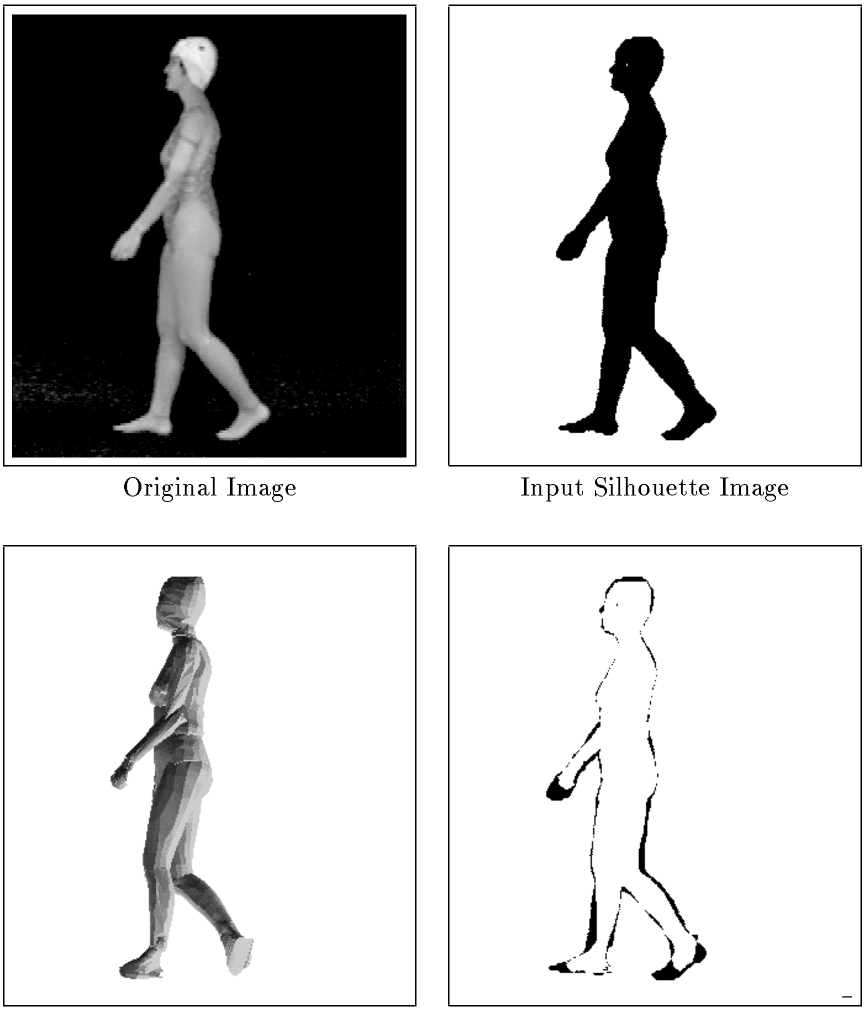


Figure 4.5: Gnawed image

it can reduce regions not covered in Step 1. This problem is left for further study.



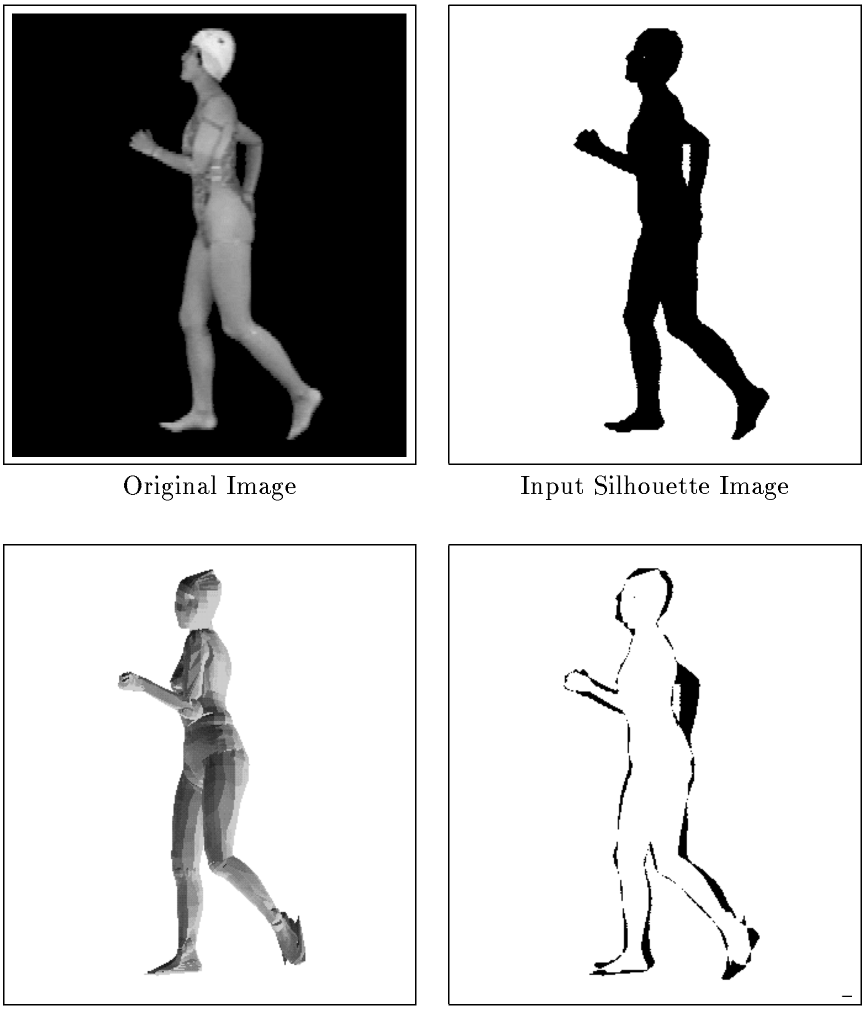
Original Image

Input Silhouette Image

Result

Exclusive OR

Figure 4.6: A case of real-life human



Original Image

Input Silhouette Image

Result

Exclusive OR

Figure 4.7: Another case of real-life human

4.4 Conclusion

We proposed a new model-matching method to estimate the pose of an articulated object from only one silhouette image. We have separated the model and the algorithm clearly. The matching method refers the contour relationship. We have showed its availability throughout the experiment for computer generated images. We also applied it for a real-life human by introducing the gnawed image and presented several experimental results.

If the method can control the order of the parts to be processed with some reasonable criteria, it will not only reduce the computational amount but also resolve the uncovered silhouette region problem. This issue will be studied in the future.

Chapter 5

Silhouette Based Motion Estimation

We move onto motion estimation from sequence of silhouette images from this chapter. Here, a model-matching method to estimate a motion of an human body is proposed. We assume that shape information of the human body and camera calibration parameters have been given to the system in advance. In the algorithm, parts which consists of the human body are classified into 3 modes: moving, stationary, and occlusion mode. When a part is in moving mode, the system matches the part's projection with a difference image. When it is in stationary mode, the system doesn't use the image. When it is in occlusion mode, the system predicts the motion from its locus and doesn't use the image.

5.1 Introduction

Motion estimation which aims to handle deformable objects is one of the important research in image understanding. Among the deformable objects, a human body has a significant value to be estimated because it is very useful to recognize a user's intension and instruction.

Recognition of motions of a human body is a challenging problem in computer vision. Rohr[77], Bharatkumar[31], Attwood[78] have contributed to solve this problem. Many researches use a stick or cylinder model and edges in images. However, human bodies are not stick in a strict sense. Moreover, edges in images usually contains not only useful ones but also many meaningless ones. Haritaoglu at el.[79] proposed to use difference region information to track articulated human body although they didn't define explicit articulated object model.

In this paper, we propose a human motion estimation method using a difference image sequence. A difference image is obtained from continuous two images which are taken by a fixed camera. Our method is based on a model matching method. A human model consists of 15 nodes each of which represents a part of a human body and has a joint attribute. In this paper, motion estimation is defined to estimate the joint angles in the model. In each frame of an image sequence, our model matching algorithm takes a node of the model one by one. We classify a status of the nodes into three modes; moving mode, stationary mode, and occlusion mode. A node is labeled with one of three modes by considering the relationship between the difference image and the projected region of the node on the image plane. We calculate joint angles according to the modes of the nodes at each frame. If a node is in the occlusion mode, the images cannot be used. To cope with such situation, we introduce inertia constraint such that the parts in a human body are generally rotated at a constant speed.

We assume that an image sequence does not include any moving object except for a human body and the camera calibration is done in advance and that the initial pose of the human body model is obtained from an other method such as [80][81].

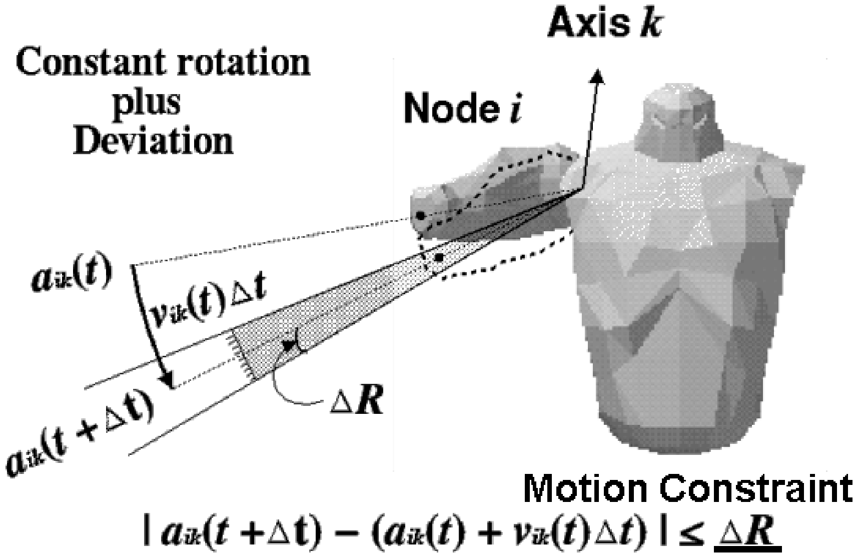

Axis k
Constant rotation plus Deviation
Node i
$a_{ik}(t)$
$v_{ik}(t)\Delta t$
$a_{ik}(t+\Delta t)$
ΔR
Motion Constraint
$| a_{ik}(t+\Delta t) - (a_{ik}(t) + v_{ik}(t)\Delta t) | \leq \Delta R$

Figure 5.1: Joint rotation at one node

5.2 Model

A human body is represented by an articulated object model. The model consists of 15 nodes. Each node corresponds to a part of human body. The nodes are connected to each other in a tree graph so that one node includes one joint that has at most 3 rotational axes. Joint rotational angles of the node i at time t are expressed by a vector $\mathbf{a}_i(t) = \{a_{ik}(t)\}$. The subscript k means the rotational axis of the joint rotational angle. The minimum and maximum value of $a_{ik}(t)$ are defined so that a human model prohibit unrealistic motion.

The motion is said to be estimated if the angles of all the joints are calculated for every frame in an image sequence.

5.3 Joint Rotation Considering Inertia

We assume that human motion generally undergoes inertia. In other words, we assume that a joint rotates with a constant speed if there is no factors to change the speed. As we consider one node at a time on evaluating joint angles and mass of the node does not vary during the motion, we can ignore the mass term. In the real 3D world, the rotational speed of a joint may change to some extent. So, we use the formulation below to predict a joint rotational angle at time $t + \Delta t$.

$$|a_{ik}(t + \Delta t) - (a_{ik}(t) + v_{ik}(t)\Delta t)| \leq \Delta R \tag{5.1}$$

$v_{ik}(t)$ means the speed around the axis k of the node i at time t. The maximum rotational speed deviation ΔR can be determined depending on a kind of motion. This inequality is illustrated in Figure 5.1.

5.4 Node Mode and Difference Image

The points of our method are summarized into two issues; a node mode and segmentation of a difference image.

When the process estimates joint angles of a node the process decides which to use by its node mode, the difference image information or the joint angle prediction. A difference image is segmented into 4 types.

5.4.1 Node Mode

Suppose you observe a motion of a human body from the camera viewpoint. Human body parts can be classified into 3 modes: moving mode, stationary mode, and occlusion mode. To correspond these modes with the model, we classify the nodes in the model in the same way. The characteristics of these node modes are described below.

1. Moving Mode
   A part corresponding to the node labeled the motion mode can be seen from the camera. A difference image is available to estimate the joint angles of the node.

2. Stationary Mode
   A part corresponding to the node labeled the stationary mode can be seen from the camera. As the node is not moving, there is no area of it in a difference image.

3. Occlusion Mode
   A node in this mode can not be seen from the camera. In this case, we cannot get any information about the joint rotational angles from the images. So, we apply the inertia assumption to predict the joint angles.

Let $\mathcal{M}_t$ be a node set at time t where the nodes are in the motion mode, and $\mathcal{S}_t$ be a set of the stationary mode nodes and $\mathcal{O}_t$ be a set of the occlusion mode nodes.

5.4.2 Difference Image

Let $\mathbf{I}_{t_n}$ be an image of the frame n which is taken at time t_n and the images are taken with the time interval Δt. A pixel value s located at $\mathbf{x}$ at time t_n is defined as below.

$$s(t_n, \mathbf{x}) = |p(\mathbf{I}_{t_n}, \mathbf{x}) - p(\mathbf{I}_{t_{n-1}}, \mathbf{x})| \tag{5.2}$$

$p(\mathbf{I}_t, \mathbf{x})$ means the pixel value at $\mathbf{x}$ in the image $\mathbf{I}_t$. A difference image is segmented into two regions according to $s(t_n, \mathbf{x})$.

- Stationary Region $\mathbf{S}_{t_n} = \{\mathbf{x} | s(t_n, \mathbf{x}) = 0\}$

  This region consists of the projections of the nodes in $\mathcal{S}_{t_n}$ and a background.

- Moving Region $\mathbf{M}_{t_n} = \{\mathbf{x} | s(t_n, \mathbf{x}) \neq 0\}$

  Moving region $\mathbf{M}_{t_n}$ consists of three regions; generated moving region $\mathbf{G}_{t_n}$, continuous moving region $\mathbf{K}_{t_n}$, and vanishing moving region $\mathbf{V}_{t_n}$. These three regions are defined below. In the formulations, $\mathbf{P}(\mathcal{A})$ means union of the projections of all the nodes in the node set $\mathcal{A}$ onto the image plane.

$$\begin{aligned}
\mathbf{G}_{t_n} &= \mathbf{P}(\mathcal{S}_{t_{n-1}}) \cap \mathbf{P}(\mathcal{M}_{t_n}) && (5.3)\\
\mathbf{K}_{t_n} &= \mathbf{P}(\mathcal{M}_{t_{n-1}}) \cap \mathbf{P}(\mathcal{M}_{t_n}) && (5.4)\\
\mathbf{V}_{t_n} &= \mathbf{P}(\mathcal{M}_{t_{n-1}}) \cap \mathbf{P}(\mathcal{S}_{t_n}) && (5.5)
\end{aligned}$$

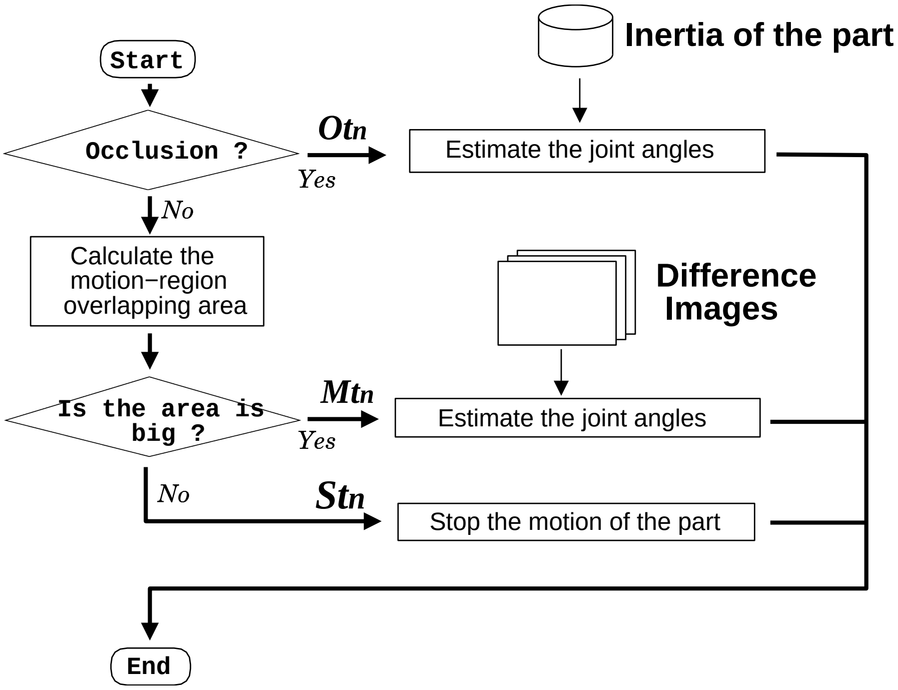

Start
Occlusion ?
Otn
Yes
Inertia of the part
Estimate the joint angles
No
Calculate the motion−region overlapping area
Difference Images
Is the area is big ?
Mtn
Yes
Estimate the joint angles
No
Stn
Stop the motion of the part
End

Figure 5.2: Flowchart of processing a node at each frame

5.5 Algorithm

This section describes the model matching algorithm we proposed. It utilizes the node modes and takes the relationship between the difference images and the projections of the human model into consideration. The process deals with a node one by one at each frame. The flowchart of the matching algorithm for a node is as shown in Figure 5.2.

Suppose the process comes to the frame n.

First, check whether the current node is in the occlusion mode or not. If it is in the occlusion mode, the joint angle is estimated according to the prediction using the inertia discussed before.

Then, search $\hat{\mathbf{a}}_p(t_n)$ which maximizes the area size $F[\mathbf{a}_p(t_n)]$ by varying the joint angle $\mathbf{a}_p(t_n)$. If the difference image is the first one in the image sequence (that means $n = 2$), $F[\mathbf{a}_p(t_n)]$ is formulated as the equation (5.6). Otherwise, it is defined as the equation (5.7).

$$F[\mathbf{a}_p(t_2)] = f\left(\left(\mathbf{G}_{t_2} \oplus \mathbf{T}(\mathbf{a}_p(t_2))\right) \cap \overline{\left(\mathbf{H}(\{\mathbf{a}(t_1)\}) \cup \mathbf{C}_{t_2}\right)}\right) \tag{5.6}$$

$$F[\mathbf{a}_p(t_n)] = f\left(\left(\mathbf{M}_{t_n} \oplus \mathbf{T}(\mathbf{a}_p(t_n))\right) \cap \overline{\mathbf{C}_{t_n}}\right) \tag{5.7}$$

In the formulations, $\mathbf{T}(\mathbf{a}_p(t_n))$ indicates the region projected by the node p at time t_n. $\mathbf{C}_{t_n}$ is an union region of the projections of the nodes that have been estimated at time t_n. And $\mathbf{H}(\{\mathbf{a}(t_1)\})$ is a projection of the human model at time t_1 that is given to the system in advance. $\oplus$ is an exclusive-or operator between two binary regions.

If $F[\hat{\mathbf{a}}_p(t_n)]$ is larger than the threshold value μ, the node p is determined to be moving, otherwise stationary. The threshold μ is introduced to remove the influence of the noise occurred in the calculation of the difference. If the node is determined to be in the moving mode, the joint angles of the node is set to $\hat{\mathbf{a}}_p(t_n)$. If the node is determined to be in the stationary mode, the joint angles are set to the predicted ones discussed in Section 5.3 and its rotational speed is reset to zero so that it stands still.

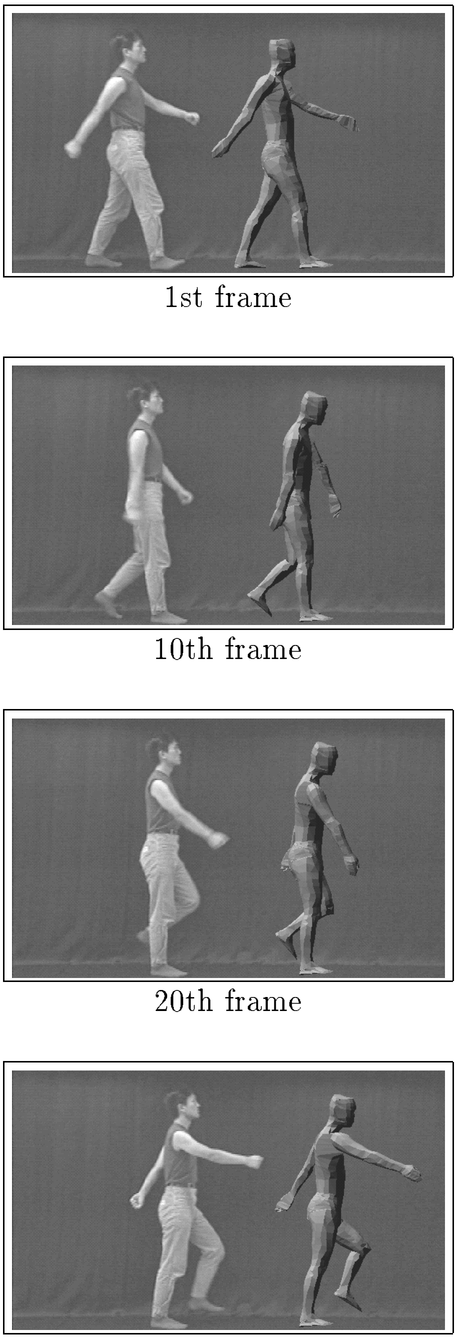
After processing all the nodes in the model at one frame, the model matching algorithm goes to the next frame and continues the process until it comes to the end of the image sequence.

5.6 Experiment

We applied our method to an image sequence with a male walking. The image size 600 by 360 pixels and the sequence includes 40 frames. ΔR is set to 6 degree and μ is $2,500$ pixels. The estimated result is shown in Figure 5.3. The result motion is displayed 100 cm right to the actual result location for the readers. Here you can see that our method keeps tracking the parts in spite of occlusions. As our method uses a complete human model and can obtain the joint angles from the image sequence, we can show the estimation result from any viewpoint. See Figure 5.4.

5.7 Conclusion

In this paper, we propose a human motion estimation method using a difference image sequence. We classify a status of nodes in the model into three modes; moving mode, stationary mode, and occlusion mode. A node is labeled with one of three modes by considering the relationship between the difference image and the projected region of the part on the image plane. The process calculates joint angles by referring the modes of the nodes at each frame. Through the experiment, we showed that our method can keep tracking the parts of the human in spite of the occlusions.



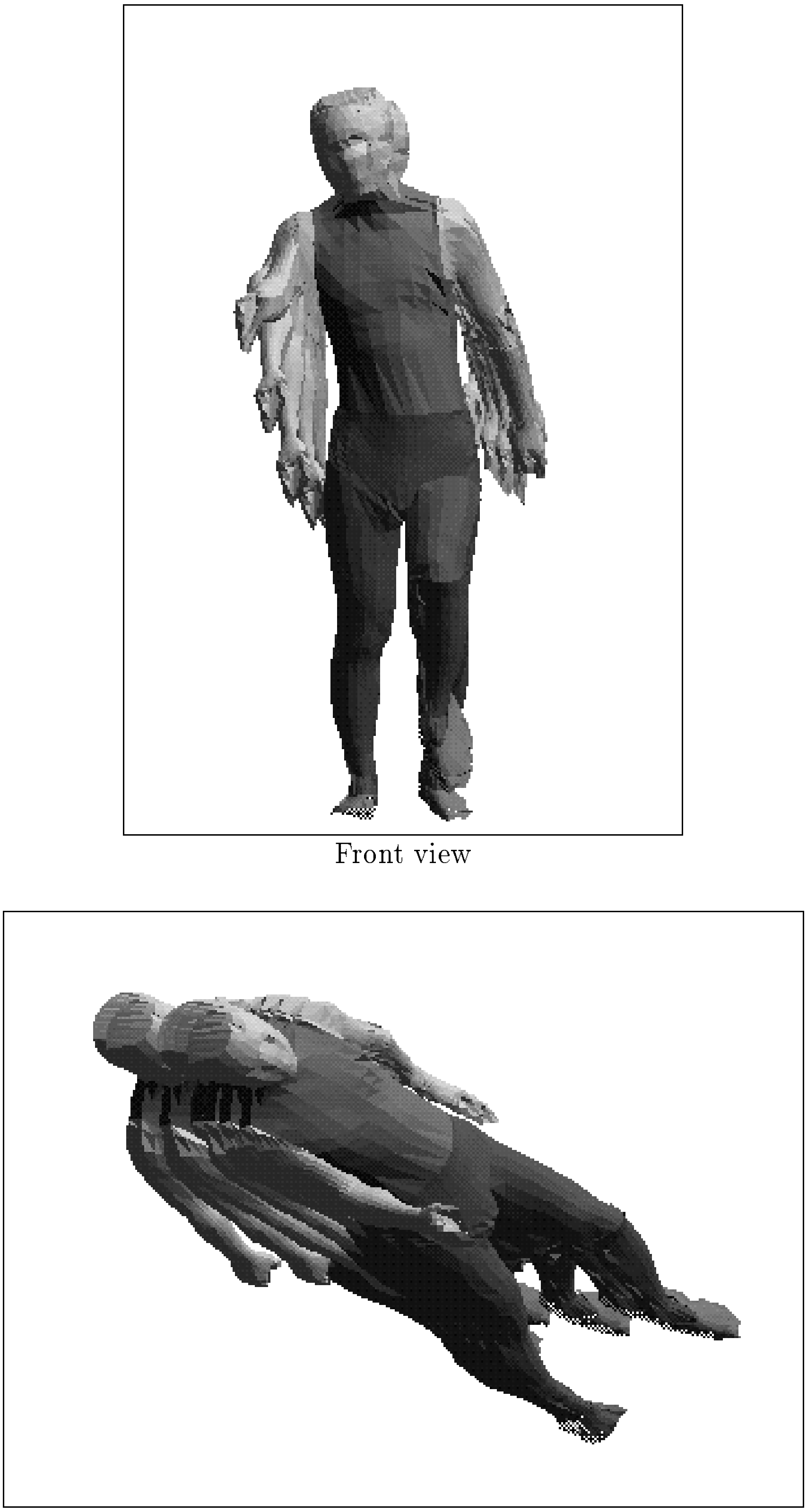
1st frame

10th frame

20th frame

30th frame

Figure 5.3: Result



Front view

Slant view

Figure 5.4: Result from another viewpoint

Chapter 6

Difference Based Motion Estimation

We have introduced inertia on motion estimation and have described our method in the previous chapter. However, the image feature referred in the method is a simple difference. In this chapter, we introduce double difference image that is obtained from silhouette sequence and refine our motion estimation method with it.

As it is no need to estimate a motion of a person when he stays still, we propose to use a *double-difference image* to get motion information from the video frames and estimate a pose partially on the region where motion feature is detected. A double-difference image is obtained by an AND operation between successive two difference images. We implemented a motion estimation system that executes model matching on a double-difference image.

6.1 Introduction

Automatic input and analysis of human motion have attracted attention from virtual reality and human interface researchers. The analysis by sensors or special devices were implemented and users find it useful these days. However, such special devices tend to impose a burden to measured persons and are not common in our daily life.

Computer Vision, especially using one camera, becomes main technical trend for human motion estimation and many researchers have contributed to it [31], [77], [78], [30], [41]. We have proposed to estimate human motion based on precise human shape model where input video frames are binarized in advance [82], [81].

On estimating a human motion by observing video frames, it is important what kind of features to be extracted from the video frames. We have concentrated on silhouette images in our previous research [81], but it is not easy to obtain the silhouette when the background or light conditions changes. In this paper, we propose to use difference images, because the difference operation derives motion information of the object in the images. If the human body does not move the difference value becomes zero, which means it is not necessary to estimate its motion.

Our method introduces a model matching method. A human body model consists of several solid objects each of which represents a part of a human body and has a joint attribute. Here, motion estimation is defined to estimate the value of the joint angles in the model.

We conducted the experiments on the real video frames. The results show that our method can keep tracking the motion of the human body.

6.2 Double-difference image

We propose to use a *double-difference image* to get motion regions from video frames and estimate a pose partially on the region where the motion regions is detected. The motion region is a region where a pixel value changes. A double-difference image is obtained by AND operation between successive two difference images.

It is assumed that the video frames do not include any moving object except for a human body.

We make a double-difference image from three successive frames in an video stream(Figure 6.1). First, we generate two difference images from corresponding two successive images ($t-1$ and t, t and $t+1$). Then we binarize the difference images and execute AND operation on these two images. We call a resultant binary image a *double-difference image*.

As a double-difference image is a product of two difference images, it tends to include isolated noise pixels. These pixels disturbs motion estimation described later in Section 6.4. Therefore, each 4 by 4 pixels in the double-difference image is grouped into one square block. A block is marked true if more than half of the pixels in the block is true. This process not only prevents noise but also reduces the computation cost in the image processing. Then the system removes isolated blocks to get rid of slight changes in the video frames. The motion regions consists of the pixels whose value is true in the remained blocks.

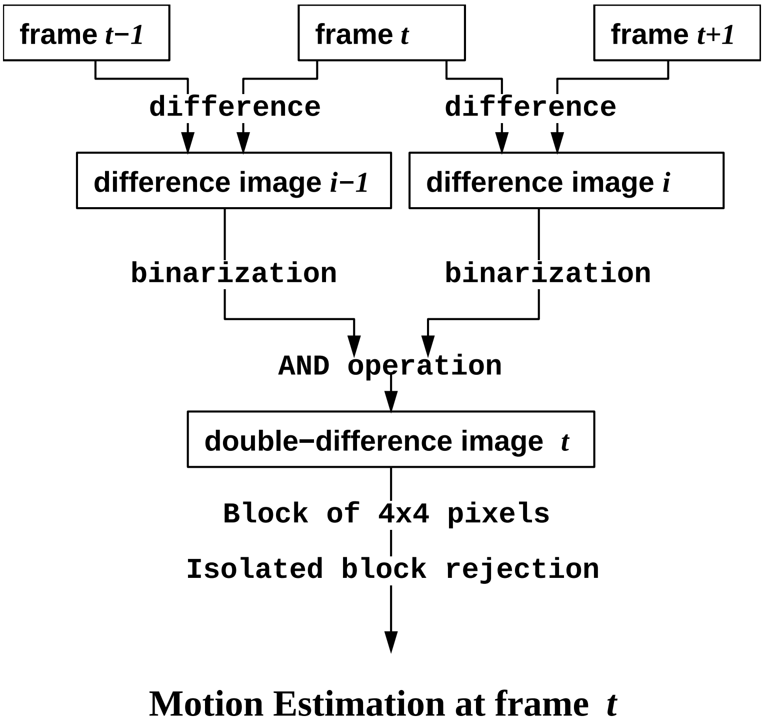


Figure 6.1: Double-difference image generation

A double-difference image has two good features. One is that motion regions on the double-difference image keeps the shape of the human body at time t. Regions on a normal difference image do not express the shape of the object because it is a mixture of the object shape on the image plane at time $t-1$ and that at time t. For example, consider a rectangle object transition in Figure 6.2. In the left, extracted shape is a combined contour (thick line) of that of time $t-1$ and time t. The right figure shows a double-difference image and the AND operation keeps the original shape at time t.

The other feature is that it is easy to detect whether the current frame contains motion information or not. If motion regions on a double-difference image are small or do not exist, it indicates that the human body stands still and it is no need to estimate the pose in that frame(Figure 6.3).

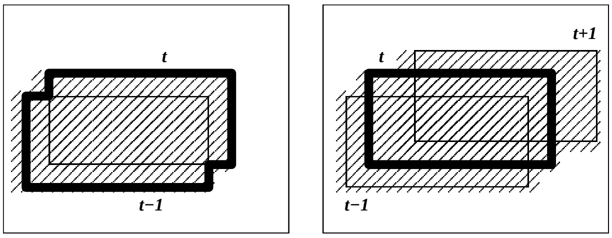


Figure 6.2: Extracted shape on difference images

6.3 Human Model

A human body is represented by an articulated object model which consists of several solid body objects. A solid object corresponds to a part of the human body and has a joint to the adjacent body object. The body objects are connected to each other in a tree structure. According to the tree graph structure, pose of one body object is defined directly by the joint angles that is a attribute of the joint connecting it to a parent body object in the tree. A joint has at most 3 axes to rotate. Joint angles of the body object i at time t are expressed by a vector $\mathbf{a}_i(t) = \{a_{ik}(t)\}$. The subscript k means the axis of the joint rotation. The minimum and maximum joint angle are defined so that a human model does not come to an unrealistic pose. Figure 6.4 shows a human body model consisting of nine parts. The pelvis corresponds to the root body object. Black dots on the model represents the joints. For example, a joint of the head body part locates near the mouth in the figure. Dotted lines denotes the tree structure of this model.

In this way, the motion is defined by the time varied angles of all the joints for a certain period.

6.4 Motion Estimation Algorithm

We assume that the camera calibration is done in advance and that the initial pose of the human body model is obtained from the method we have proposed in previous paper [81].

Suppose now is time t. As shown in Figure 6.3, if the double-difference image at time t contains few motion regions (white frames in the figure), the joint values are all fixed as those in the previous frame.

On the other hand, if the double-difference image t contains motion regions(gray frames in the figure), it implies the human body has changed its pose. In this case, the model matching algorithm processes a body object in the model one by one, from the root position to the leafs of the tree structure in the model. We classify a status of the body objects into three modes; *moving mode*, *stationary mode*, and *occlusion mode*. Each body object is labeled with one of the three modes by observing the relationship between the motion regions and the projected region of the body object on the image plane. We calculate joint angles according to the

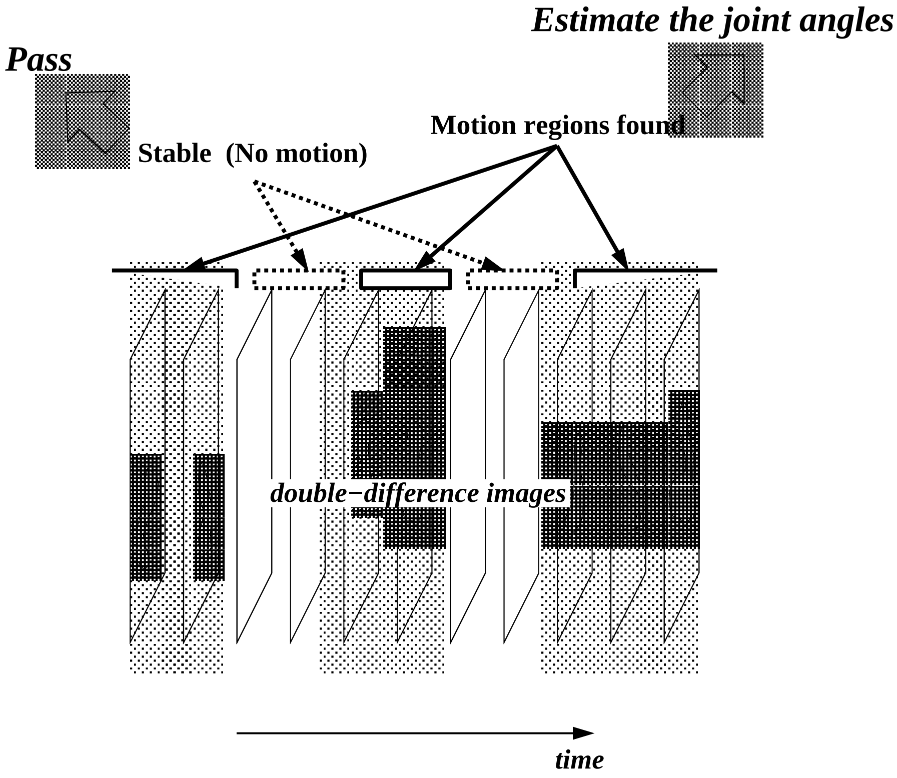


Figure 6.3: Frame skip

modes of the body objects at each frame(Figure 6.5). If a body object is hidden by other body objects, it is determined to be in the occlusion mode and the image plane is not referred. To cope with such situation, we introduce inertia constraint such that the body objects in a human body are generally rotated at a constant speed[82].

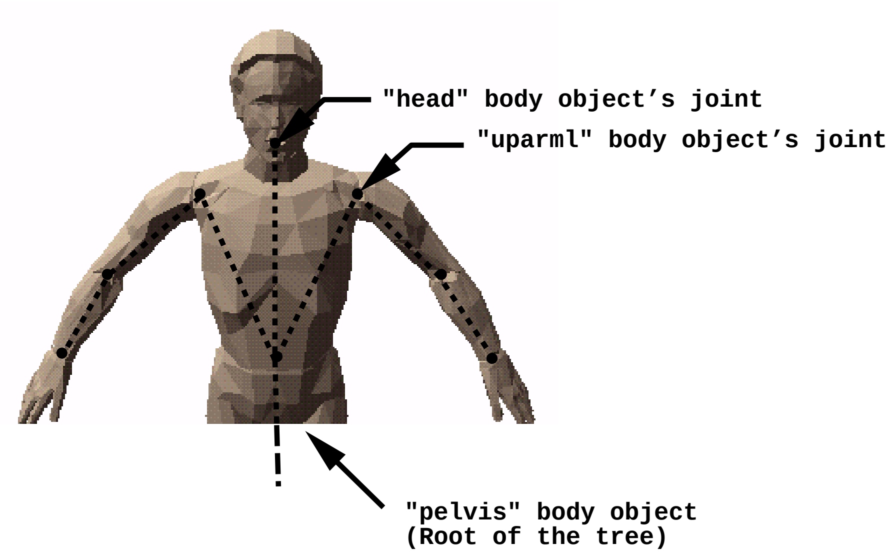


Figure 6.4: Human body model

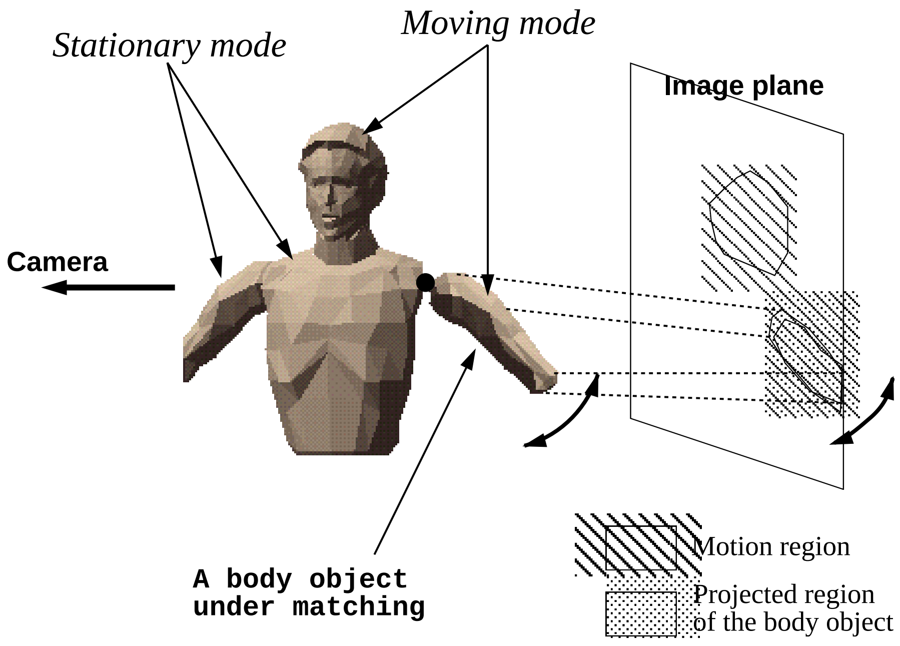


Figure 6.5: A body object under matching process

6.5 Experiment

We implemented this method and applied it to real video frames. The human model used is shown in Figure 6.4. The video camera is set on the top of the desk bookshelf and focus on a person sitting in front of the desk. Camera position, camera direction, focus length are measured in advance. We constructed this system on SGI Indy (R4400 200MHz).

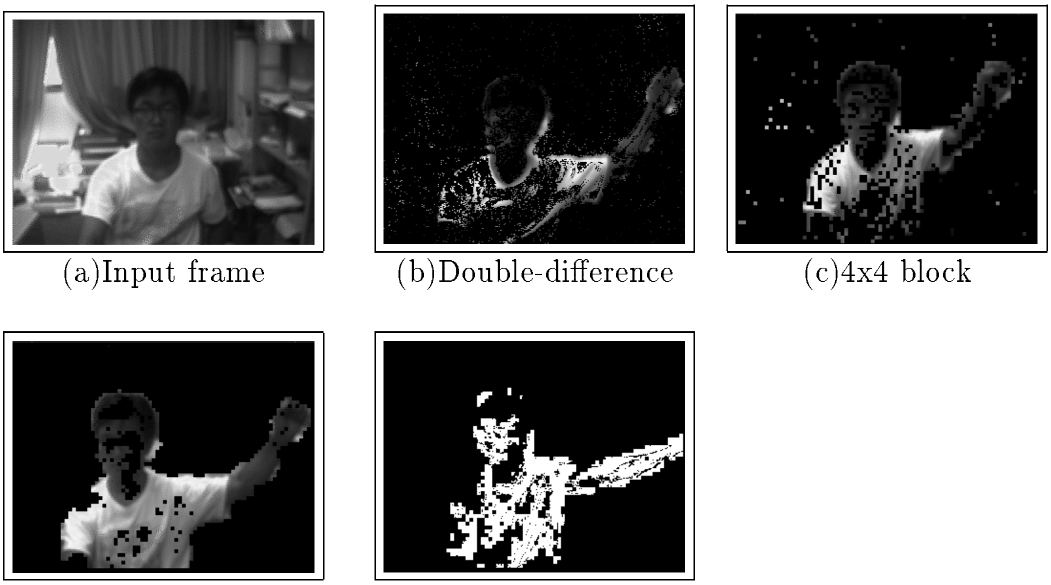
Figure 6.6 shows an example of the image processing. Figure (a) to (e) were taken at different moment. Figure(b) includes many noise pixels and they are rejected through (c), (d). In Figure (b), (c), (d), the pixels whose value is true in the double difference image express the original pixel value in the input frame. Final output Figure(e) passed to the motion estimation system includes small regions because each pixel in the motion regions is shown there.

The system generated 4.40 frames per second in average when the input video frame size was set to 320 pixels by 240 pixels and without motion estimation process. This frame rate depends on the number of motion regions in the video frame. See Table 6.1.

In the motion estimation experiments, we assumed that maximum rotation speed at each joint is 10 degrees per second. The experiments succeeded in simple exercises like waving hands. But there may exists difficulties in the present system. One of the reasons of the estimation failures is that one video frame consists of two fields. Since the current system merges two fields in one frame, shape of the motion regions do not reflect the correct shapes. Another reason of the estimation failures is that a body object once covers the incorrect motion regions, there is no way to update the joint angles to the right values.

Through the experiment, the average rate of the stable double difference images for all the frames comes to 37.2%. This rate also depend on what the person is working in the office which cause the rate to be high.

The current system can process only 1.1 frames per second in average. Speed-up and computation reduction is our future work.



(a)Input frame (b)Double-difference (c)4x4 block

(d)Isolated block rejection (e)Motion regions

Figure 6.6: Image processing

Table 6.1: Frame rate for image process

	Average	Max	Min
frames/s	4.40	4.83	3.88

6.6 Conclusion

We have proposed a human motion estimation method from three successive video frames. As it is no need to estimate a motion of a person when he stays still, we propose to use a *double-difference image* to get motion information from video frames and estimate his motion only if the motion regions exists on the double-difference images. We implemented a motion estimation system that executes model matching on a double-difference image and showed its abilitiy.

Chapter 7

Concluding Remarks

In this thesis, we proposed a method of pose and motion estimation from ocellus image for human body.

In this research, we have to obtain a solution that adjusts the articulated model which involves three dimensional information to an image which involves two dimensional information under geometrical condition of the camera projection. In that case, silhouette region is used as an image feature because it can be extracted easily even if light condition varies and without knowledge concerning the human body.

We discussed what pose and motion estimation means by describing deformation of the human body that our articulated model can represent.

We first propose a method of pose and motion estimation by using an articulated model for one silhouette image in Chapter 2. The pose of the human body is provided by joint angle parameter of this articulated model. Although the matching evaluation function can be defined in consideration of all the variables about the entire human body, analysis of the function is difficult and consumes much calculation cost. Therefore, the method of achieving the pose and motion estimation of the entire human body on an image plane was proposed in this thesis by doing the matching evaluation between silhouette region and the projection region of the metamere in each metamere node of the articulated model.

In Chapter 3, we proposed the improved pose estimation method by referring the pose decision tree. The partial evaluation with the pose decision tree occasionally cause miss-match according to relationship between location of a root node of the articulated model and a pose of the articulated object in the image. This can be detected by comparing the estimated pose of the articulated model with entire silhouette region. Hence, we realized an accurate pose estimation method by backtracking in the pose decision tree when the miss-match is found by the detection done at the end of the pose estimation procedure so as to avoid falling into local optimal solution.

In Chapter 4, we introduced gnawed region on pose estimation. A silhouette region is an image feature that can provide shape of the human body without knowledge of the human body. However, it is enumerated not to be able to detect self-occlusion in the silhouette region as a defect of using it. Therefore, at the time of the processing of a certain pose estimation, we propose to exclude projection region from the evaluation, which corresponded to metamere nodes of which the joint angle have been determined already. It was shown to avoid falling easily into local solution of the joint angle according to the pose decision tree, and to obtain better solution with our method by the experimental result.

We expanded the pose estimation method to motion estimation in Chapter 5. We realized motion estimation of human body by estimating pose of the human body for silhouette region at each frame. The characteristic of our method is to assume the moment of inertia to each metamere. As a result, a metamere in the silhouette region can be tracked even if it is under self-occlusion in the human body.

And at Chapter 6, we proposed an improved motion estimation method which evaluates overlapped area between the metamere and double difference region as matching score. We expanded the motion estimation method previously proposed and propose to estimate only the metamere nodes which correspond to the parts of the human body. This method reduces the calculation cost for match evaluation because it doesn't process the metamere nodes of fixed parts.

As we have not discussed effects of three dimensional shape difference between the articulated model and the target human body, we should evaluate relationship between the transformation degree of each metamere in the articulated model and the target metamere. The motion estimation method should be expanded for complicated human motion as our future work.

Bibliography

[1] V. I. Pavlovic, R. Sharma, and T. S. Huang: "Gesturel interface to a visual computing environment for molecular biologists", Proc. of the 2nd Int. Conf. on Automatic Face and Gesture Recognition, pp. 30–35 (1996).

[2] C. J. Cohen, L. Conway and D. Koditschek: "Dynamical system representation, generation, and recognition of basic oscillatory motion gestures", Proc. of the 2nd Int. Conf. on Automatic Face and Gesture Recognition, pp. 60–65 (1996).

[3] N. H. Goodard: "Incremental model-based discrimination of articulated movement from motion features", Proc. of the workshop on Motion of Non-rigid and Articulated Objects, pp. 89–94 (1994).

[4] G. S. Pingali, Y. Jean and I. Carlbom: "Real time tracking for enhanced tennis broadcasts ", Proc. of Computer Vision and Pattern Recognition'98, pp. 260–265 (1998).

[5] L. Gencalves, E. D. Bernardo and P. Perona: "Reach out and touch space (motion learning)", Proc. of the 3rd Int. Conf. on Automatic Face and Gesture Recognition, pp. 234–239 (1998).

[6] Q. Delamarre and O. Faugeras: "Finding pose of hand in video images: a stereo-based approach", Proc. of the 3rd Int. Conf. on Automatic Face and Gesture Recognition, pp. 585–590 (1998).

[7] C. Bregler and J. Malik: "Tracking people with twists and exponential maps", Proc. of Computer Vision and Pattern Recognition'98, pp. 8–15 (1998).

[8] J. K. Aggarwal, W. L. Q. Cai and B. Sabata: "Articulated and elastic non-rigid motion: A review", Proc. of the workshop on Motion of Non-rigid and Articulated Objects, pp. 2–14 (1994).

[9] S. Naoya, S. Seki and R. Oka: "A theoretical consideration of pattern space trajectory", Proc. of the 2nd Int. Conf. on Automatic Face and Gesture Recognition, pp. 72–77 (1996).

[10] F. K. Quek and M. Zhao: "Inductive learning in hand pose recognition", Proc. of the 2nd Int. Conf. on Automatic Face and Gesture Recognition, pp. 78–83 (1996).

[11] W. T. Freeman, K. Tanaka, J. Ohta and K.kyuma: "Computer vision for computer games ", Proc. of the 2nd Int. Conf. on Automatic Face and Gesture Recognition, pp. 100–105 (1996).

[12] K. Takahashi, S.Seki, H.Kojima and R. Oka: "Recognition of dexterous manipulations from the time-varying images", Proc. of the workshop on Motion of Non-rigid and Articulated Objects, pp. 23–28 (1994).

[13] R. Cutler and M. Turk: "View based interpretation of real-time optical flow for gesture recognition", Proc. of the 3rd Int. Conf. on Automatic Face and Gesture Recognition, pp. 416–421 (1998).

[14] T. Kurita and S. Hayamizu: "Gesture recognition using hlac features of parcor images and hmm based recognizer", Proc. of the 3rd Int. Conf. on Automatic Face and Gesture Recognition, pp. 422–427 (1998).

[15] J. Triesch and C. von der Malsburg: "Robust classification of hand postures against complex backgrounds", Proc. of the 2nd Int. Conf. on Automatic Face and Gesture Recognition, pp. 170–175 (1996).

[16] K.-H. Jo, Y. Kuno and Y. Shirai: "Manipulative hand gesture recognition using task knowledge for human computer interaction", Proc. of the 3rd Int. Conf. on Automatic Face and Gesture Recognition, pp. 468–473 (1998).

[17] C. Wren, A. Azarbayejani, T. Darrell and A. Pentland: "Pfinder: Real-time tracking of the human body", Proc. of the 2nd Int. Conf. on Automatic Face and Gesture Recognition, pp. 51–56 (1996).

[18] T. Heap and D. Hogg: "Towards 3d hand tracking using a deformable model", Proc. of the 2nd Int. Conf. on Automatic Face and Gesture Recognition, pp. 140–145 (1996).

[19] H. Hienz, K. Grobel and G. Offner: "Real-time hand-arm motion analysis using a single video camera", Proc. of the 2nd Int. Conf. on Automatic Face and Gesture Recognition, pp. 323–327 (1996).

[20] H. A. Rowley, S. Baluja and T. Kanada: "Rotation invariant neural network-based face detection", Proc. of Computer Vision and Pattern Recognition'98, pp. 38–44 (1998).

[21] L. W. Campbell, D. A. Becker, A. Azarbayejani, A. F. Bobick and A. Pentland: "Invariant features for 3-d gesture recognition", Proc. of the 2nd Int. Conf. on Automatic Face and Gesture Recognition, pp. 157–162 (1996).

[22] D. W. Jacobs and C. Chennubholta: "Segmenting independently moving, noisy points", Proc. of the workshop on Motion of Non-rigid and Articulated Objects, pp. 96–103 (1994).

[23] T. Y. Tian and M. Shah: "A general approach for detecting 3d motion and structure of multiple objects from image trajectories", Proc. of the workshop on Motion of Non-rigid and Articulated Objects, pp. 110–115 (1994).

[24] M. Yamada, K. Ebihara and J. Ohya: "A new robust real-time method for extracting human silhouettes from color images", Proc. of the 3rd Int. Conf. on Automatic Face and Gesture Recognition, pp. 528–533 (1998).

[25] S. C. Zhu: "Stochastic computation of medial axis in markov random fields", Proc. of Computer Vision and Pattern Recognition'98, pp. 72–79 (1998).

[26] J. Ma and N. Ahuja: "Dense shape and motion from region correspondences by factorization", Proc. of Computer Vision and Pattern Recognition'98, pp. 219–224 (1998).

[27] X. Feng and P. Perona: "Scene segmentation from 3d motion", Proc. of Computer Vision and Pattern Recognition'98, pp. 225–231 (1998).

[28] Y. Gdalyahu and D. Weinshall: "Automatic hierarchical classification of silhouettes of 3d objects", Proc. of Computer Vision and Pattern Recognition'98, pp. 787–793 (1998).

[29] D. Geiger and T.-L. Liu: "Recognizing articulated objects with information theoretic methods", Proc. of the 2nd Int. Conf. on Automatic Face and Gesture Recognition, pp. 45–50 (1996).

[30] J. M. Rehg and T. Kanade: "Digideyes: Vision-based hand tracking for human computer interaction", Proc. of the workshop on Motion of Non-rigid and Articulated Objects, pp. 16–22 (1994).

[31] A. Bharatkumar, K. E. Daigle, M. G. Pandy, Q. Cai and J. K. Aggarwal: "Lower limb kinematics of human walking with the medial axis transformation", Proc. of the workshop on Motion of Non-rigid and Articulated Objects, pp. 70–76 (1994).

[32] Y. Hel-Or and M. Werman: "Recognition and localization of articulated objects", Proc. of the workshop on Motion of Non-rigid and Articulated Objects, pp. 116–123 (1994).

[33] R. J. Holt, A. N. Netravali, T. S. Huang and R. J. Qian: "Determining articulated motion from perspective views: A decomposition approach", Proc. of the workshop on Motion of Non-rigid and Articulated Objects, pp. 126–137 (1994).

[34] C. R. Wren and A. P. Pentland: "Dynamic models of human motion", Proc. of the 3rd Int. Conf. on Automatic Face and Gesture Recognition, pp. 2–27 (1998).

[35] N. Shimada, Y. Shirai, Y. Kuno and J. Miura: "Hand gesture estimation and model refinement using monocular camera – ambiguity limitation by inequality constraints", Proc. of the 3rd Int. Conf. on Automatic Face and Gesture Recognition, pp. 268–273 (1998).

[36] J.-M. Chung and N. Ohnishi: "Cue circles: Image feature for measuring 3-d motion of articulated objects using sequential image pair", Proc. of the 3rd Int. Conf. on Automatic Face and Gesture Recognition, pp. 474–479 (1998).

[37] M. Yamamoto, T. Kondo, T. Yamagiwa and K. Yamanaka: "Skill recognition", Proc. of the 3rd Int. Conf. on Automatic Face and Gesture Recognition, pp. 604–609 (1998).

[38] M. Yamamoto, A. Sato, S. Kawada, T. Kondo and Y. Osaki: "Incremental tracking of human actions from multiple views", Proc. of Computer Vision and Pattern Recognition'98, pp. 2–7 (1998).

[39] D. D. Morris and J. M. Rehg: "Singularity analysis for articulated object tracking", Proc. of Computer Vision and Pattern Recognition'98, pp. 289–296 (1998).

[40] V. Burdin, C. Roux, E. Stindel and C. Lefevre: "Study of 3-d human movements: Influence of the forearm bone morphology on the magnitude of the prosupination motion", Proc. of the workshop on Motion of Non-rigid and Articulated Objects, pp. 29–34 (1994).

[41] F. J. Perales and J. Torres: "A system for human motion matching between synthetic and real images based on a biomechanic graphical model", Proc. of the workshop on Motion of Non-rigid and Articulated Objects, pp. 83–88 (1994).

[42] C. W. Gear: "Feature grouping in moving objects", Proc. of the workshop on Motion of Non-rigid and Articulated Objects, pp. 214–219 (1994).

[43] S. X.Ju, M. J. Black and Y. Yacoob: “Cardboard people: A parameterized model of articulated image motion”, Proc. of the 2nd Int. Conf. on Automatic Face and Gesture Recognition, pp. 38–44 (1996).

[44] I. A. Kakadiaris, D. Metaxas and R. Bajcsy: “Active motion-based segmentation of human body outlines”, Proc. of the workshop on Motion of Non-rigid and Articulated Objects, pp. 50–56 (1994).

[45] S. M. Yi Li and H. Lu: “A multiscale morphological method for human posture recognition ”, Proc. of the 3rd Int. Conf. on Automatic Face and Gesture Recognition, pp. 56–61 (1998).

[46] R. A. Brooks: “Model-based three-dimensional interpretations of two-dimensional images ”, IEEE Transactions on Pattern Recognition and Machine Intelligence, **5**, 2, pp. 140–150 (1983).

[47] R. Horaud: “New methods for matching 3-d objects with single perspective views”, IEEE Transactions on Pattern Recognition and Machine Intelligence, **9**, 3, pp. 401–412 (1987).

[48] M. Dhome, M. Richetin, J.-T. Lapresté and G. Rives: “Determination of the attitude of 3-d objects from a single perspective view”, IEEE Transactions on Pattern Analysis and Machine Intelligence, **11**, 12, pp. 1265–1278 (1989).

[49] D. G. Lowe: “Stabilized solution for 3-d model parameters”, First European Conference on Computer Vision Proceedings, pp. 408–412 (1990).

[50] J. Ponce, A. Hoogs and D. J. Kreigman: “On using cad models to compute the pose of curved 3d objects”, CVGIP Image Understanding, **55**, 2, pp. 184–197 (1992).

[51] B. Horowitz and A. Pentland: “Recovery of non-rigid motion and structure”, Proceedings IEEE Computer Society Conference on Computer Vision and Pattern Recognition, pp. 325–330 (1991).

[52] T. Shakunaga: “Pose estimation of jointed structures”, Proceedings IEEE Computer Society Conference on Computer Vision and Pattern Recognition, pp. 566–572 (1991).

[53] S. Kuratake and R. Nevatia: “Description and tracking of moving articulated objects”, Proceedings 11th IAPR International Conference on Pattern Recognition Vol.1 Conference A-i Computer Vision and Applications, pp. 491–495 (1992).

[54] T. Kimoto, A. Kajigaya and Y. Yasuda: “A method of analyzing a human walker from monocular moving pictures based on stick models”, Transactions of Institute of Electronics, Information and Communication Engineers, **J74-D-II**, 3, pp. 376–387 (1991).

[55] A. C. Downston and H. Drouet: “Model-based image analysis for unconstrained human upperbody motion”, International Conference on Image Processing and its Applications, pp. 274–277 (1992).

[56] C. I. Attwood, G. D. Sullivan and K. Baker: “Model-based recognition of human posture using single synthetic images”, Proceedings of the Fifth Alvey Vision Conference, pp. 25–30 (1989).

[57] K. Yunibhand, H. Kinoshita and Y. Sakai: “A hand recognition method for visual language processing”, Transactions of Institute of Electronics, Information and Communication Engineers, **J75-D-II**, 9, pp. 1489–1497 (1992).

[58] Y. Nagashima, T. Onodera, H. Nagashima, M. Terauchi and G. Ohwa: "A study of recognition method of japanese finger spelling", Technical Report of Human Communication Engineering of Institute of Electronics, Information and Communication Engineers, **92**, 210, pp. 23–30 (1992).

[59] K. Ishibuchi, H. Takemura and F. Kishino: "Real-time hand shape recognition using pipeline image processor", Technical Report of Human Communication Engineering of Institute of Electronics, Information and Communication Engineers, **92**, 26, pp. 19–24 (1992).

[60] D. Marr: "Vision", W. H. Freeman and Company (1982).

[61] H. P. Graf, E. Cosatto, D. Gibbon, M. Kocheisen and E. Petajan: "Multi-model system for locating heads and features", Proc. of the 2nd Int. Conf. on Automatic Face and Gesture Recognition (1996).

[62] T. Horprasert, Y. Yacoob and L. S. Davis: "Computing 3-d head orientation from a monocular image sequence", Proc. of the 2nd Int. Conf. on Automatic Face and Gesture Recognition, pp. 242–247 (1996).

[63] K. C. You and R. Cipolla: "Detection of human faces under scale, orientation and viewpoint variations", Proc. of the 2nd Int. Conf. on Automatic Face and Gesture Recognition, pp. 295–300 (1996).

[64] K. HOTTA, T. KURITA and T. MISHIMA: "Scale invariant face detection method using higher-order local autocorrelation features extracted from log-polar image", Proc. of the 3rd Int. Conf. on Automatic Face and Gesture Recognition, pp. 70–75 (1998).

[65] Q. Chen, H. Wu, T. Fukumoto and M. Yachida: "3d head pose estimation without feature tracking", Proc. of the 3rd Int. Conf. on Automatic Face and Gesture Recognition, pp. 88–93 (1998).

[66] I. Shimizu, Z. Zhang, S. Akamatsu and K. Deguchi: "Head pose determination from one image using a generic model", Proc. of the 3rd Int. Conf. on Automatic Face and Gesture Recognition, pp. 100–105 (1998).

[67] J.-C. Terrillon, M. David and S. Akamatsu: "Automatic detection of human faces in natural scene images by use of a skin color model and of invariant moments", Proc. of the 3rd Int. Conf. on Automatic Face and Gesture Recognition, pp. 112–117 (1998).

[68] Q. B. Sun, W. M. Huang and J. K. Wu: "Face detection based on color and local symmetry information", Proc. of the 3rd Int. Conf. on Automatic Face and Gesture Recognition, pp. 130–135 (1998).

[69] E. Elagin, J. Steffens and H. Neven: "Automatic pose estimation system for human faces based on bunch graph matching technology", Proc. of the 3rd Int. Conf. on Automatic Face and Gesture Recognition, pp. 136–141 (1998).

[70] A. Utsumi, H. Mori, J. Ohya and M. Yachida: "Multiple-human tracking using multiple cameras", Proc. of the 3rd Int. Conf. on Automatic Face and Gesture Recognition, pp. 498–503 (1998).

[71] M. K. Leung and Y. Yang: "A region based approach for human body motion analysis", Pattern Recognition, **20**, 3, pp. 321–339 (1987).

[72] W. Long and Y. Yang: "Log-tracker: An attribute-based approach to tracking human body motion", International Journal of Pattern Recognition and Artificial Intelligence, **5**, 3, pp. 439–458 (1991).

[73] K. Yunibhand, H. Kinoshita and Y. Sakai: "Hand gesture recognition using stick figure model", Transactions of Institute of Electronics, Information and Communication Engineers, **J77-D-II**, 1, pp. 51–60 (1994).

[74] H. Ishii, K. Mochizuki and F. Kishino: "A motion recognition method from stereo images for human image synthesis", Transactions of Institute of Electronics, Information and Communication Engineers, **J76-D-II**, 8, pp. 1805–1812 (1993).

[75] N. Shimada, Y. Shirai and Y. Kuno: "Hand pose estimation from monocular image sequence using 3-dimensional model", Technical Report of IEICE, **PRU94-4**, 94-50, pp. 25–32 (1994).

[76] Y. Sakaguchi, M. Minoh and K. Ikeda: "Party: Grid generation for human body and paper pattern by the geometrical constraints method", Transactions of Institute of Electronics, Information and Communication Engineers, **J77-D-II**, 11, pp. 2210–2219 (1994).

[77] K. Rohr: "Incremental recognition of pedestrians from image sequences", Proc. of the 1993 IEEE Computer Vision and Pattern Recognition, pp. 8–13 (1993).

[78] C. I. Attwood, G. D. Sullivan and K. Baker: "Model-based recognition of human posture using single synthetic images", Proceedings of the Fifth Alvey Vision Conference, pp. 25–30 (1989).

[79] I. Haritaoglu, D. Harwood and L. S. Davis: "w^4: Who? when? where? what? a real time system for detecting and tracking people", Proc. of the 3rd Int. Conf. on Automatic Face and Gesture Recognition, pp. 222–227 (1998).

[80] Y. Kameda, M. Minoh and K. Ikeda: "Three dimensional pose estimation of an articulated object from its silhouette image", Proceedings of First Asian Conference on Computer Vision, pp. 612–615 (1993).

[81] Y. Kameda, M. Minoh and K. Ikeda: "A pose estimation method for an articulated object from its silhouette image", Trans. of the Institute of Electronics, Information and Communication Engineers, **J79-D-II**, 1, pp. 26–35 (1995).

[82] Y. Kameda, M. Minoh and K. Ikeda: "Three dimensional motion estimation of a human body using a difference image sequence", Proceedings of Second Asian Conference on Computer Vision, **II**, pp. 181–185 (1995).

Acknowledgements

I would like to express sincere gratitude to Professor IKEDA Katsuo of Kyoto University. He gave me the opportunity of this research, and has been giving me continuous guidance, interesting suggestions, intensive criticism and kind encouragements during the research.

I would like to appreciate Professor MINOH Michihiko of Kyoto University for his invaluable suggestions, accurate criticism, and encouragements.

I also acknowledge interesting comments that I received from Assistant Professor AMANO Akira and Assistant Professor HIROSE Shouichi of Kyoto University, and Dr. Yoshiyuki Sakaguchi of TOYOBO.

Thanks are also due to all the members of Professor Ikeda's Laboratory and Professor Minoh's Laboratory for their enthusiastic discussions and supports through this research.

List of Publications by the Author

Major Publications

1. KAMEDA Yoshinari, MINOH Michihiko, IKEDA Katsuo, “Three Dimensional Pose Estimation of an Articulated Object from its Silhouette Image,” Proceedings of Asian Conference on Computer Vision ’93, pp.612-615, 1993.

2. KAMEDA Yoshinari, MINOH Michihiko, IKEDA Katsuo, “Three Dimensional Motion Estimation of a Human Body Using a Difference Image Sequence,” Proceedings of Second Asian Conference on Computer Vision ’95, pp.181-185.

3. KAMEDA Yoshinari, MINOH Michihiko, IKEDA Katsuo, “A Pose Estimation Method for an Articulated Object from its Silhouette Image,” The Transactions of the IEICE D-II, J79-D-II, No.1, pp.26-35, 1996.

4. OGINO Tomotaka, KAMEDA Yoshinari, SAKAGUCHI Yoshiyuki, MINOH Michihiko, IKEDA Katso “A Collision Detection Method for Interacting with Virtual Weaven Cloth, ” Proceedings of International Conference on Virtual Systems and Multimedia ’96, pp.129-134, 1996.

5. KAMEDA Yoshinari, MINOH Michihiko, “A Human Motion Estimation Method using 3-successive video frames,” Proceedings of International Conference on Virtual Systems and Multimedia’96, pp.135-140, 1996.

6. Yoshinari Kameda, Takeo Taoda, Michihiko Minoh, “High Speed 3D Reconstruction by Video Image Pipeline Processing and Division of Spatio-Temporal Space,” Proceedings of MVA’98 IAPR Workshop on Machine Vision Applications, pp.406–409, 1998.

7. Yoshinari Kameda, Takeo Taoda, Koh Kakusho, Michihiko Minoh, “High Speed 3D Reconstruction by Pipeline Video Image Processing and Division of Spatio-Temporal Space,” IPSJ Journal, Vol.40, No.1, pp.13–22, 1999.

8. Yoshinari Kameda, Hideaki Miyazaki, and Michihiko Minoh, “A Live Video Imaging for Multiple Users,” Proceedings of International Conference on Multimedia Computing and Systems (ICMCS’99), 1999 (To be published).

Technical Reports

1. KAMEDA Yoshinari, Minoh Michihiko, IKEDA Katsuo, “A 3-D Shape Presumption Of A Human Hand Using A Silhouette,” Meeting on Image Recognition and Understanding MIRU’92, IPSJ Symposium Series Vol.92, No.3, pp.239-246, 1992.

2. KAMEDA Yoshinari, MINOH Michihiko, IKEDA Katsuo, "A Human Motion Recognition Method Using Difference Images," Technical Report of IEICE, Vol.95, No.165, PRU95-98, pp.115-120, 1995.

3. YAGI Keisuke, FUJIKAWA Kenji, KAMEDA Yoshinari, SENDA Shuji, MUKUNOKI Masayuki, MINOH Michihiko, IKEDA Katsuo, "PLS: A Plan for Polymorphic Learning Space," Technical Report of IEICE, Vol.95, No.183, MVE95-38, pp.23-30, 1995.

4. Takuro Sakiyama, Yoshinari Kameda, Michihiko Minoh, Katsuo Ikeda, "Facial Image Extraction of Lecture Attendants in an Image Sequence using Template Matching," Meeting on Image Recognition and Understanding MIRU'96, I, pp.13-18, 1996.

5. Koji IMAO, Yoshinari KAMEDA, Michihiko MINOH, Katsuo IKEDA, "A Deformation Method of a Three Dimensional Shape Model For Representing a Personal Human Body Shape," Proceedings of the 2nd Intelligence Information Symposium, pp.183-190, 1996.

6. Jun KITAWAKI, Yoshinari KAMEDA, Koh KAKUSHO, Michihiko MINOH, Katsuo IKEDA, "Primitive Operations for Interactive Design of Clothes in Virtual Space," IPSJ SIG Notes HI, Vol.97, No.107, 97-HI-75, pp.39-44, 1997.

7. Yoshinari Kameda,Takeo Taoda,Koh Kakusho,Michihiko Minoh, "Pipeline Video Image Processing and Division of Spatio-Temporal Space for High Speed 3D Reconstruction," IPSJ SIG Notes DPS, Vol.98, No.55, 98-DPS-89, pp.85-90, 1998.

8. KUBODA Hidekazu, KAMEDA Yoshinari, MINOH Michihiko, "Realization of Eye-contact Dialogue in Profile on Tele-conference," Technical Report of IEICE, Vol.98, No.128, HIP98-8, pp.55-62, 1998.

9. Hideyuki Miyazaki, Kentaro Kichiyoshi, Yoshinari Kameda, Michihiko Minoh, "A Real-time Method of Making Lecture Video Using Multiple Cameras," Meeting on Image Recognition and Understanding MIRU'98, IPSJ Symposium Series Vol.98, No.10, I, pp.123-128, 1998.

10. Yoshinari Kameda, Michihiko Minoh, "Video Image Generation of 3D Lecture Space by Interpreting Dynamic Situation," Meeting on Image Recognition and Understanding MIRU'98, IPSJ Symposium Series Vol.98, No.10, I, pp.371-376, 1998.

Convention Records

1. KAMEDA Yoshinari, AMANO Akira, Minoh Michihiko, IKEDA Katsuo, "A 3D-Model Matching with Energy Functions for Hand Images," Proceedings of the 43th National Convention IPSJ, Vol.2, pp.389-390, 1991.

2. KAMEDA Yoshinari, Minoh Michihiko, IKEDA Katsuo, "A 3D Shape Recognition Strategy For a Hand Using its Silhouette," Proceedings of the 45th National Conference IPSJ, Vol.2, pp.255-256, 1992.

3. Yoshinari KAMEDA, Michihiko MINOH, and Katsuo IKEDA, "Pose Estimation Ability for an Articulated Object from its Silhouette," Proceedings, of the 47th National Conference IPSJ, Vol.2, pp.151-152, 1993.

4. KAMEDA Yoshinari, Minoh Michihiko, IKEDA Katsuo, “A 3-D Pose Estimation of an Articulated Object from its Silhouette Image,” The 21th Annual Conference and Visual Computing’93 of the Institute of Image Electronics Engineers of Japan, pp.133-136, 1993.

5. KAMEDA Yoshinari, MINOH Michihiko, IKEDA Katsuo, “A Human Motion Estimation Method Using a Silhouette Image Sequence,” Proceedings of the 1994 IEICE Fall Conference, D-355, 1994.

6. HITOMI Kojiro, KAMEDA Yoshinari, MINOH Michihiko, IKEDA Katsuo, “A Pose Estimation Method Based on Constraint Satisfaction Image Processing,” Proceedings of Annual Conference of IIEEJ, pp.57-58, 1994.

7. IMAO Koji, KAMEDA Yoshinari, MINOH Michihiko, IKEDA Katsuo, “A 3D Model Transformation Method Fitted to A Silhouette Image,” Proceedings of Annual Conference of IIEEJ, pp.105-106, 1994.

8. KAMEDA Yoshinari, MINOH Michihiko, IKEDA Katsuo, “A Human Motion Estimation Method Using a Model — Introduction of Inertia —,” Proceedings of the 1995 IEICE General Conference, D-658, 1995.

9. OGINO Tomotaka, KAMEDA Yoshinari , SAKAGUCHI Yoshiyuki, MINOH Michihiko, IKEDA Kastuo, “A Fast Collision Detection Method for Handling Virtual Cloth,” Proceedings of the 53th National Conference IPSJ, Vol.4, pp.53-54, 1996.

10. KAMEDA Yoshinari, MINOH Michihiko, “A Human Motion Tracking Method Using Double Difference Technique,” Proceedings of Conference of Kansai District Union of Electric Related Institutes, S12-3:S64, 1996.

11. KITAWAKI Jun, KAMEDA Yoshinari, IKEDA Katsuo, MINOH Michihiko, “Recognition for Manipulating Cloth in Virtual Environment,” Proceedings of the 1997 Information and Systems Society Conference of IEICE, A-16-6, 1997.

12. KAMEDA Yoshinari, KICHIYOSHI Kentaro, MINOH Michihiko, “A Method of Transferring Video Using Multiple Cameras in A Lecture Room,” The Proceedings of the 2nd Image Media Processing Symposium, pp.13-14, 1997.

13. Michihiko Minoh , Yoshinari Kameda, “Three Dimensional Model Based Interpretation of Dynamic Situation in a Lecture Room,” Proceedings of First International Workshop on Cooperative Distributed Vision, pp.177-194, 1997.

14. Michihiko MINOH, Yoshinari KAMEDA, “Human Shape Measurement Based on A 3D Model,” The 40th Union Conference on Automatic Control, SS23, pp.67-70, 1997.

15. Michihiko MINOH, Yoshinari KAMEDA, “A method for Taking and Processing Several Videos in a Classroom,” Proceedings of Classroom Vision Symposium (CRV’98), pp.43-50, 1998.

16. IIYAMA Masaaki, KAMEDA Yoshinari, MINOH Michihiko, “Pose Estimation of Human Body on Voxel Data Considering Interference with Movable Area of Human Model Parts, ” Proceedings of the 1998 Information and Systems Society Conference of IEICE, D-12-85, pp.307, 1998.

17. Jun TAKADA, Yoshinari KAMEDA, Michihiko Minoh, “A Method to Attend Multiple Meetings by Audio Media Control According to Users’ Intention,” Proceedings of the 57th National Conference IPSJ, Vol.4, pp.49–50, 1998.

18. Michihiko Minoh, Yoshinari Kameda, “Imaging a 3D Lecture Room by Interpreting its Dynamic Situation,” Proceedings of Second International Workshop on Cooperative Distributed Vision, pp.243–264, 1998.

19. Norihiro Wada, Yoshinari Kameda, Koh Kakusho, Michihiko Minoh, “Visual and Force Feedback Implementation for Manipulating Virtual Cloth,” Proceedings of the 167th Technical Meeting of IIEEJ, Vol.98-05, pp.25-31, 1998.

Other Publications

1. KAMEDA Yoshinari, MINOH Michihiko, IKEDA Katsuo, “A Pose Estimation Method for an Articulated Object from its Silhouette Image,” Image Labo Vol.7, No.8, pp.1-4, 1996.

Awards

1. Best Paper Award for Young Researchers of the 43th National Convention of IPSJ