Model-Based Object Tracking Using Stereo Vision*

Romuald Ginhoux
École Nationale Supérieure
de Physique de Strasbourg
67400 Illkirch, France
ginhoux@ieee.org

Jens-Steffen Gutmann
Corporate Technology
Siemens AG
81730 Munich, Germany
gutmann@ieee.org

Abstract

Robust and accurate object recognition and tracking is a necessary pre-requisite for visual servoing tasks. While most approaches use monocular vision in order to solve this problem, this paper presents a new approach using disparity images provided by a stereo vision system from which 3-D models are reconstructed. We use a segmentation procedure based on a simple region growing process to segment the non-dense 3-D images and we compare it to a classical dense range image segmenter. Obtained regions are classified according to a simple model of the target object and multiple candidates are handled over image sequences by application of the CONDENSATION algorithm. We then use a histogram based clusterization technique to identify the target object pose for visual servoing commands. Experimental results in localizing a door-handle show an accuracy of about 10 cm for the estimated position, while being robust to clutter and noise in the sensor images.

1. Introduction

Robotic visual servoing tasks can be classified by the sensor system that is used as input source for their control loop. Lots of approaches use monocular vision and are error-prone to different lighting conditions when they aim at generating precise 3-D information of the tracked object. Other systems use proximity sensors like laser range finders which provide exact 3-D data but are usually slow and expensive. The work presented in this paper uses data from a commercial stereo vision system to build a robot servoing task. The advantage of this sensor system is that it is fast and cheap, does not suffer as much from different lighting conditions as monocular vision usually does, and that it is independent of the object’s color, e.g. it is possible to recognize an object in front of a background where both object and background have the same color (e.g. yellow door handle in front of yellow door as in our environment) as long as there is enough texture for computing disparities.

The robot’s task is to follow a moving object or to move its end-effector to a desired position with reference to an object. We use an eye-in-hand configuration on a mobile platform equipped with a 6 degree-of-freedom manipulator.

The stereo head has been fully calibrated and is used to provide disparity images from which a 3-D model of the camera environment is reconstructed. Images are segmented to find candidates for the target object and a probabilistic approach is used which allows to both localize the object among the potential candidates, and to track it in the 3-D environment through the camera motion. Finally, tracking results are used to perform a pose reconstruction for the position-based visual servoing tasks.

The originality of this project is that a stereo camera system is used as an alternative to classical laser range finders to get 3-D information about the robot environment and that 3-D localization and visual servoing are achieved thanks to a tracking method originally developed in the context of tracking of deforming curves in visual clutter.

The rest of this paper is organized as follows. The next two sections present our stereo vision system and 3-D image segmentation methods. Section 4 explains how segmented regions are classified and Section 5 shows the general framework for the stochastic localization method we use. Section 6 presents a visual servoing task to validate our tracking implementation and concludes with experimental results.

2. Stereo Vision System

Stereo vision aims at reproducing the human ability to infer information on the 3-D structure and distances of a scene from images taken from two different viewpoints [11]. Two problems have mainly to be solved: correspondence and reconstruction. Correspondence consists in determining which item in the left eye corresponds to which item in the right eye. The difference of items in retinal po-

---

*This work was partially funded by the German Federal Ministry of Education and Research (BMBF) under contract no. 01IL902D0 (Morpha).
tion is called disparity and the disparities of all pixels the disparity image. If the geometry of the stereo vision system is known, a 3-D map of the viewed scene can be reconstructed from a disparity image.

In this paper, we use the SVS Stereo Vision System which is a complete commercial solution for stereo analysis developed by SRI International [9]. It can deliver stereo frames and reconstruct 3-D information at a rate of 20 Hz. The SVS Stereo Vision System includes a calibration program to estimate the intrinsic and extrinsic camera parameters. It also computes an image warp for rectifying the left and right images to allow use of the standard stereo geometry [11]. Corresponding elements between images are found thanks to the area-correlation method where correlation is used to find the most likely correspondences between small area patches [8].

Our main task was to choose an appropriate setup for the system and to calibrate it so that it can be used in the context of 3-D visual servoing. We use the cameras at their best resolution (320x120) with the largest stereo baseline (11 cm). We also make use of lenses of small focal length (3.9 mm) since no reliable information can be computed for these filtered pixels.

![Fig. 1. Example of a stereo pair from the STH-V2 stereo head. A disparity image (in the middle) is computed from two images taken from two slightly different viewpoints.](image)

3. Range Image Segmentation

Informally, segmentation is the assignment of pixels in a given 3-D image into one of many disjoint sets such that the pixels in each set share a common property [4]. In [2], range images obtained from the SVS stereo camera are segmented into layers of near constant disparity. The segmentation results in a bunch of planes which are perpendicular to the optical axis. In [4], laser range finders furnish dense range images from which segmenters attempt to extract 3-D planes. In both cases, the segmentation procedure is mainly defined by the criterions which are used to assign each pixel to a set.

The segmentation process usually contains two steps. The first step aims at finding a seed region, which is a group of pixels or a single pixel. The second step, or region growing process, starts from the seed region and tries to assign other pixels to it to build a region. When no more pixels verifying a logical predicate can be added, a new seed region is searched to reiterate the overall process. The final output of a segmenter is an image in which each pixel is labeled with a number indicating its corresponding region.

### 3.1. “University of Bern” Segmentation

The “University of Bern” (or UB) procedure is classically used with dense 3-D range images obtained from laser range finders [4]. It is based on a scan line approximation which exploits the scan line structure of the image [7] and segments images by extracting planar surfaces. The idea is that, in the ideal case, the points on a scan line that belong to a planar surface form a straight 3-D line segment. On the other hand, all points on a straight 3-D line segment surely belong to the same planar surface.

Within UB, finding a seed region is a simple split method that recursively divides each scan line into straight line segments such that the perpendicular distance (in range units) of the points to their corresponding line segment is within a threshold $T_0$. A potential seed region for region growing is a triple of line segments on three neighboring scan lines that satisfies three conditions:

1. all three line segments have at least length $T_1$ (in range units);
2. the overlapping part of two neighboring line segments has at least $T_2\%$ of the length of each line segment;
3. every pair of neighboring points on two line segments is within a distance $T_3$ (range units).

The candidate with the largest total line segment length is chosen as the optimal seed region. In the subsequent region growing process, a line segment is added to the region if the perpendicular distance between its two end points and the plane equation of the region is within a threshold $T_4 + T_5 \times \text{size}/10000$ (range units), where size is the number of pixels of the region expanded so far. This dynamic threshold relaxes the expansion condition for very large regions. This process is repeated until no more line segments can be added, at which time a new region is started using the next best available seed region. If a region’s final size is below a threshold $T_6$ (pixels), it is discarded.

1227
We have tested this dense image segmenter on the non-dense 3-D images provided by the SVS stereo camera (see results in Section 6). It turns out that the UB method usually over-segments the stereo images which we account to the fact that our data is non-dense and we were not able to tune all of UB’s parameters accordingly. For these reasons we decided to develop our own segmentation method which is a simple region growing approach.

3.2. “Simple Growing” Segmentation

We have defined a simpler strategy which segments the image from back to front. It uses two growing criterions to successively build regions around the farthest pixel in the image. First we apply a backgrounding operation (threshold \(T_0\) set on the \(Z\) coordinates) to remove all pixels for which no reliable information can be found. These pixels correspond to the SVS-filtered areas and have actually been associated with a very large range.

Within the Simple Growing (or SG) segmentation, the seed pixel is selected as the farthest pixel that can be found in the image among the still not segmented areas. Next, in the region-growing process, all 4-connected neighboring pixels are recursively assigned to the seed region if the following two conditions are met:

1. the difference between the depth value of the current pixel and the depth value of the considered neighbor is less than a threshold \(T_1\);
2. the distance between the two pixels projected in the image plane stays larger than the depth difference of the first condition, up to a scale factor \(T_2\).

The procedure is reiterated until no more pixels can be added to the current region. Finally, if the region contains less than a minimal number of pixels (threshold \(T_3\)), it is erased and considered to belong to the background region. We then reiterate the overall procedure on the image from which all pixels of the newly built region have been removed.

We compare the ability of the two preceding segmentation procedures to work on our non-dense range images by means of the framework defined in [4]. Experimental results can be found in Section 6.

4. Object Recognition

For recognizing a certain object in the 3-D data we need to compute some significant features for each segmented region and match them to those of a user-specified object model [1]. We propose a simple feature set based correspondence method between 3-D model features and features of the segmented regions.

Let \(\mathcal{W}^k = \{w_i^k \mid i = 1, 2, \ldots , I\}\) denote the set of characteristics computed for region \(R^k\), \(k = 1, 2, \ldots , K\), and \(\mathcal{W} = \{w_i \mid i = 1, 2, \ldots , I\}\) represent the object model, with \(I \in \mathbb{N}\). The set \(\mathcal{W}\) has to be specified so that it contains the characteristics of a real-world object. We can note here that \(I = \text{card}(\mathcal{W})\) gives an idea of the complexity of the model. In our project, we only use a 3-D length and width measurement \(I = 2\) which we estimate from the shape of the regions. Two characteristics are sufficient here because the visual servoing tasks to be defined later will deal with a door-handle or any other baton-like object.

For each \(w_i^k, i = 1, 2, \ldots , I\), a similarity measure \(f_i(w)\) estimates how well the feature corresponds to the corresponding model feature. These functions are simply triangular shaped and should ensure robustness against segmentation failures and noise in the disparity images.

Then, we associate each region with a confidence level \(e^k\) that reflects how well they globally correspond to the model, in other words, that reflects the reliability of region \(R^k\) to be the target object. This represents a “global” similarity measurement which we compute as a weighted sum of the “local” similarity measurements \(f_i(w)\):

\[
e^k = \frac{\sum_{i=1}^{I} \alpha_i f_i(w^k)}{\sum_{i=1}^{I} \alpha_i} \tag{1}\]

where coefficients \(\alpha_i\) are user-defined and specify different weights to the 3-D characteristics of the model. Finally, confidence levels \(e^k, k = 1, 2, \ldots , K\), are used to compute an adaptive threshold \(h\) to determine potential candidates for the object of interest [2]. It is computed as

\[
h = E[e^k] + K_h \cdot \text{Var}[e^k] \tag{2}\]

where \(E[e^k]\) is the mean confidence level and \(\text{Var}[e^k]\) the variance of all confidence levels, \(K_h\) is a constant; in our experiments we use a value of \(K_h = 1.5\). Segmented regions with \(e^k\) greater than \(h\) become candidates for the target object and all other are discarded. This adaptive threshold is useful to avoid an explosion of the number of candidates that can occur with static thresholds.

Since we will use our classifier in the context of 3-D visual servoing, we have to specify the position and orientation (or pose) of each region, with respect to the camera frame. The position of a region is given by the coordinates \(X_G, Y_G, Z_G\) of its center of gravity, which we find as the 3-D point corresponding to the 2-D center of gravity of the region viewed as a set of pixels. Since our application will deal with an oblong object (a door handle for instance), the orientation of a region can be completely specified by two angles which can be computed using the extremity points \(M_1, M_2\) of the longest diagonal of the region:

\[
\alpha = \text{atan2}(Y_2 - Y_1, X_2 - X_1) \\
\beta = \frac{\text{atan2}(Z_2 - Z_1, \sqrt{(X_2 - X_1)^2 + (Y_2 - Y_1)^2})}{\sqrt{(X_2 - X_1)^2 + (Y_2 - Y_1)^2}} \tag{2}\]
Let \( \mu^k = (X^k, Y^k, Z^k, \alpha^k, \beta^k) \) be the vector defining the state (or pose) of region \( R^k \), \( k = 1, 2, \ldots, K \), with reference to the camera frame. It defines a five-dimensional state space and will be used in the next section. The result of the classification stage for a given image \( R \) is a subset \( \{ R^m \mid m = 1, 2, \ldots, M \text{ and } M \leq K \} \) of segmented regions \( \{ R^k \mid k = 1, 2, \ldots, K \} \) with information:

\[
R^m = \{ e^m; \mu^m \}
\tag{3}
\]

where \( e^m \) is the global similarity measurement given in (1).

This set of potential candidates usually contains the object of interest, other objects whose shape in the 3-D image has matched the model, and possible artifacts due to the disparity post-filtering effects and over-segmentation failures. These artifacts seem to have a shape that mimics the model has matched the model, and possible artifacts due to the disparity post-filtering effects and over-segmentation failures. This set of potential candidates usually contains the object of interest, other objects whose shape in the 3-D image has matched the model, and possible artifacts due to the disparity post-filtering effects and over-segmentation failures. This set of potential candidates usually contains the object of interest, other objects whose shape in the 3-D image has matched the model, and possible artifacts due to the disparity post-filtering effects and over-segmentation failures.

In order to complete the object recognition task, the target object has to be localized in the reconstructed 3-D space among all potential candidates. To achieve this, we work with image sequences and use the stochastic approach exposed in the next section.

5. Probabilistic Localization

The recognition problem is to find the right candidate among the potential hypotheses in a stream of images. We apply a probabilistic approach based on Markov localization [3] and factored sampling technique, called the conditional density propagation (CONDENSATION), which was originally developed in the context of visual tracking of curves in dense clutter [5, 6]. It allows us to perform both localization and tracking of the target object through the camera motion. The 3-D pose of the object is viewed as a stochastic variable whose density distribution is shaped by the camera motion and the classification results.

The Kalman filtering approach would be inadequate because it is based on Gaussian densities, which, being unimodal, cannot represent the simultaneous alternative hypotheses we get from the classification step. The CONDENSATION approach represents the possible interpretations by a probability distribution approximated by a set of randomly generated samples. We use a camera motion model, together with visual observations made of the potential candidates extracted from the 3-D images, to propagate the random set over time. We now present the general framework of the CONDENSATION algorithm [5, 6].

Let denote the state of the wanted object at time \( t \) by \( x_t \in \mathbb{R}^5 \), with history \( x_t = \{ x_1, \ldots, x_t \} \); similarly, let \( z_t = \{ z_1, \ldots, z_t \} \) be the history of the set \( z_t \) of candidate regions in the image. Given a Markov chain with independent observations, the conditional state-density \( p_t \) at time \( t \) is defined by

\[
p_t(x_t) \equiv p(x_t \mid z_t).
\]

This represents all information about the state of the target object at time \( t \) that is deductible from the set \( z_t \). The rule for propagation of state density [5] over time is similar to the Markov localization equations [3]:

\[
p(x_t \mid z_{t-1}) = k_t \ p(x_t \mid x_{t-1}) \ p(x_{t-1} \mid z_{t-1}), \tag{4}
\]

and \( k_t \) is a normalization constant that does not depend on \( x_t \). The CONDENSATION algorithm is based on the factored sampling technique, which is a random-sampling method for approximating a distribution \( p(x \mid z) \) evaluated as

\[
p(x \mid z) = k \ p(x \mid z) \ p(x) \text{ evaluated by a set } \{ s^{(n)}, \pi^{(n)} \} \text{ of } N \text{ samples and weights.}
\]

Extended to apply iteratively to successive images, the prior \( p(x_t) \) is replaced by a prediction density \( p(x_t \mid z_{t-1}) \) (Eq. 4). The output of an iteration is the sample-set \( \{ s_t^{(n)}, \pi_t^{(n)} \} \) which approximates the state-density or posterior \( p(x_t \mid z_t) \). The prediction density is obtained by applying a dynamical motion model \( p(x_{t+1} \mid x_t) \) to the weighted set representing the posterior of the previous time-step (Eq. 5). Weights \( \{ \pi_t^{(n)} \} \) are evaluated through the observation density \( p(x_t \mid z_t) \).

5.1. Observation Model

Our observation process is actually the classification procedure. The multiplicity of candidates reflects the presence of stereo artifacts (or clutter), so either one of the events

\[
\phi_m = \{ \text{true region is } R^m \}, \ m = 1, \ldots, M
\]

occurs, or else the target object is not visible with probability \( q = 1 - \sum_m P(\phi_m) \). Parameter \( q \) is a constant which

\footnote{Apply Bayes’ rule; \( k \) is a normalization constant independent of \( x \).}
we set to 0.25 in our experiments. The observation density is expressed as:

\[ p(x_t | \phi_m) = \frac{q_p(x_t | \text{clutter})}{\sum_{m=1}^{M} p(x_t | \phi_m)} P(\phi_m) \]  

(6)

where \( p(x_t | \phi_m) \) is the Gaussian density set on candidate \( m \) so that each mode in the observation density represents a candidate given by (3). The addend \( q_p(x_t | \text{clutter}) \) is taken as a constant [5].

5.2. Motion Model

The camera motion model \( A(t) \) estimates the transformation between the camera frame at time \( (t-1) \) and the same frame at time \( t \). It is defined as \( A(t) = T_c^{r(t-1)} \) and computed as follows:

\[ A(t) = [T_w^c(t)]^{-1} T_w^c(t-1) \]

where \( T_w^c(t) = T_w^c(t-1) \) is the homogeneous transformation matrix from the robot (or world) reference frame to the left camera frame at time \( t \). Model \( A(t) \) is used to predict the pose vector at time \( t \) for a known camera motion before any visual measurements are made. We also add a stochastic component to model the geometric and motion uncertainties and compute \( x_t \) as

\[ x_t = A(t)x_{t-1} + Bw(t) \]  

(7)

where \( w(t) \) is an independent vector of independent standard normal variables and \( B \) is a constant matrix which defines the variance of the uncertainties.

5.3. Pose Reconstruction

One remaining problem in our object tracking method using the CONDENSATION approach is how to reliably reconstruct the target pose from the sample sets in order to compute a visual servoing command. We use three different ways: mean pose, maximum likelihood (ML) and histogram clusterization.

A weighted sum of the samples can be used to estimate a mean pose, but results in an offseted and inaccurate position and orientation as long as other hypotheses are maintained by the tracking algorithm [5]. The factored sampling method also allows a simple implementation of the ML estimator [10], which returns the most weighted sample. But this estimate depends only on the last iteration of the algorithm, since the weights are updated at each time-step; it cannot therefore reflect the modification and reshaping of the sample distribution performed over the sequence. Finally, a clusterization based on histograms should permit to efficiently separate minor hypotheses to get the target object pose.

6. Results

This section presents a comparison of the two different segmentation methods and gives examples of sample distributions that show the behavior and accuracy of the implemented CONDENSATION algorithm. The stereo vision system is mounted on a mobile platform equipped with a 6 degree-of-freedom manipulator according to the eye-in-hand configuration. We define a visual servoing task using a simple “Look & Move” procedure to show the ability of our adaptation of the CONDENSATION algorithm to localize an object. It consists in localizing a door-handle to let the robot end-effector grip it or reach a desired relative position.

6.1. Comparison of Range Segmentation Methods

By using the framework from [4] we compare the ability of the UB and SG segmentation procedures to work on our non-dense range images. We first collected a sequence of test images and manually defined a ground truth (or GT) segmentation for each image by specifying the regions we would manually segment. Then, to evaluate an image, we run the segmenters and match up the resulting regions with GT to distinguish several types of region segmentation: correct detection, over-segmentation and under-segmentation. We also measure the time needed to perform the segmentation. Results are collected in Figure 3 and examples of segmented pictures are given in Figure 4.

<table>
<thead>
<tr>
<th># GT Reg.</th>
<th># Regions Detected</th>
<th># Over-Segm.</th>
<th># Under-Segm.</th>
<th>Runtime (ms)</th>
</tr>
</thead>
<tbody>
<tr>
<td>UB</td>
<td>SG</td>
<td>UB</td>
<td>SG</td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>6</td>
<td>32</td>
<td>6</td>
<td>26</td>
</tr>
<tr>
<td>2</td>
<td>7</td>
<td>37</td>
<td>11</td>
<td>30</td>
</tr>
<tr>
<td>3</td>
<td>6</td>
<td>26</td>
<td>7</td>
<td>20</td>
</tr>
<tr>
<td>4</td>
<td>8</td>
<td>27</td>
<td>14</td>
<td>19</td>
</tr>
<tr>
<td>5</td>
<td>6</td>
<td>25</td>
<td>5</td>
<td>19</td>
</tr>
<tr>
<td>6</td>
<td>6</td>
<td>31</td>
<td>6</td>
<td>25</td>
</tr>
<tr>
<td>7</td>
<td>7</td>
<td>37</td>
<td>8</td>
<td>30</td>
</tr>
</tbody>
</table>

Fig. 3. Evaluation of the UB segmenter and of the SG segmenter on 7 test images.

Figure 3 shows that our segmenter has only a very small trend to over-segment or under-segment a scene, compared to the other one. The UB method appears to globally over-segment any image: multiple regions are always found along horizontal lines, as shown in Figure 4. This is mainly due to the scan-line technique which is sensitive to discontinuities introduced by pixels with non-reliable information (e.g. a false depth measurement). The UB segmenter has been developed for dense 3-D data and does not seem to be able to work efficiently on our non-dense range images.

The time needed to segment an image mainly depends on the complexity of the scene: it increases when more data
has to be handled. This occurs naturally with more complex scenes, especially those we take with a large field of view. Although the UB method detects more regions and over-segments a lot more than our procedure, we can note that it runs quicker on our disparity images. But as the number of segmented regions will appear as a key factor in the computational load of the next image processing steps, the UB segmenter cannot be an appropriate segmenter for our stereo application. Furthermore, our segmenter only uses 2 parameters for the segmentation process which are easier to set up\(^2\) than the 6 parameters needed by the UB procedure. For these reasons, results in the remainder of this work are achieved by applying our SG segmentation algorithm.

### 6.2. Evolution of Sample Distributions

This section gives examples of sample distributions to show their evolution and the behavior of the CONDENSATION algorithm with a fixed number of $N = 10000$ samples. A sequence of 7 typical images\(^3\) is used to iterate the vision loop. The images are taken in front of a door and the aim is to localize its door-handle. The segmentation, classification and tracking steps are iterated once on each image and in loop over the image sequence.

We use Gaussian densities to model the prior distribution of the samples (step 0). As we don’t have any knowledge a priori about the pose of the target object, we choose large standard deviations for all the components (2000 mm for $X, Y, Z$ and 30\(^\circ\) for $\alpha, \beta$) so that their prior distributions cover a large scale. Figure 5 depicts the distributions of the $Z$ components of the samples after 0, 7 and 50 iterations.

Figure 6 shows some $X - Y$ and $\alpha - \beta$ distributions. Several modes can be distinguished in the histograms after 7 steps. This reflects the ability of the CONDENSATION algorithm to handle multiple hypotheses: clusters appear where factored sampling makes the samples accumulate over the image sequence. We can note that after 50 iterations the $Z$ distribution contains only one peak, and we can suppose it represents the object’s $Z$-coordinate. However, the orientation histogram is still not sharply centered around one value because minor modes persist. As a consequence, computation of a mean pose or use of the ML estimate are not appropriate to accurately find the target object pose (cf. Sec. 5.3). We have implemented a histogram clusterization technique thanks to which only the highest mode of the distribution (if there are several) is used to estimate the object pose. Results are given in the next section.

---

\(^2\)We used simple trial-and-error for determining good values.

\(^3\)For some images the target object is not classified as candidate.
6.3. Accuracy and Robustness

Reconstructed poses given by a mean-pose computation, the ML estimate, and our histogram analysis are given in Figure 7 after 7 and 50 iterations of the visual loop on the test image sequence. Results originally expressed in camera frame (cf. Figs. 5 and 6) are given in the robot frame and compared with the real door-handle pose to evaluate the accuracy of the pose reconstruction. This procedure corresponds to a simple "Look & Move" task we used to position the robot end-effector in front of a door-handle.

<table>
<thead>
<tr>
<th>Robot Frame</th>
<th>Real pose</th>
<th>ML</th>
<th>Step 7</th>
<th>histo</th>
<th>Step 50</th>
<th>histo</th>
</tr>
</thead>
<tbody>
<tr>
<td>X (mm)</td>
<td>1200</td>
<td>1275</td>
<td>1175</td>
<td>1147</td>
<td>1187</td>
<td>1117</td>
</tr>
<tr>
<td>Y (mm)</td>
<td>350</td>
<td>4</td>
<td>348</td>
<td>443</td>
<td>397</td>
<td>399</td>
</tr>
<tr>
<td>Z (mm)</td>
<td>1060</td>
<td>1339</td>
<td>1103</td>
<td>1153</td>
<td>1036</td>
<td>1107</td>
</tr>
<tr>
<td>α (deg)</td>
<td>40</td>
<td>-74</td>
<td>-59</td>
<td>65</td>
<td>50</td>
<td>-78</td>
</tr>
<tr>
<td>β (deg)</td>
<td>0</td>
<td>-53</td>
<td>11</td>
<td>-7</td>
<td>-1</td>
<td>2</td>
</tr>
</tbody>
</table>

Fig. 7. Door-handle poses reconstructed by mean-pose computation, ML estimator, and histogram cluster analysis after 7 and 50 iterations of the CONDENSATION algorithm.

Our implementation of the CONDENSATION algorithm is globally robust to sensor noise and multiple hypotheses in the disparity images since we are able to handle the sample distributions in order to localize a given object in an image sequence. Results of Figure 7 show inaccuracies of about 10 cm for the 3-D pose and an unusable value for the orientation, whatever pose-reconstruction method we use. This is certainly too much to precisely grip the object, but allows for placing the robot arm in front of the door-handle where either new measurements can be taken to improve the pose estimation or other methods can be used to precisely localize it (e.g. vision techniques using structured lighting [12]). Some inaccuracies can be attributed to the 3-D reconstruction achieved by the stereo camera and depend on the chosen stereo setup and on its calibration4. The running time of the global loop is compatible with a real-time implementation.

7. Conclusion

This paper presented a new approach for object recognition and tracking using data from a stereo vision system for visual servoing tasks. 3-D maps reconstructed from non-dense disparity images are segmented using a simple region growing approach and segmentation results are compared to those of a classical dense range segmenter. Object classification has been achieved by using model-based similarity measurements and hypotheses for the target object are associated with vectors defining their poses in the camera frame. A probabilistic localization and tracking method has been used in order to localize the wanted object. Results show that the CONDENSATION process is able to handle multiple hypotheses of a target object and that a target pose can be reconstructed over iterations on image sequences with an accuracy of about 10 cm. Our implementation has been shown to be robust to stereo noise and segmentation failures. Its running time is compatible with a closed-loop implementation to track a moving object.

Future work will deal with an improved estimation of object orientation, experiments tracking other objects than a door-handle, include comparisons with other segmentation procedures that are specific to stereo cameras, and study K-D trees for manipulation of the spatial data stored in the samples.

References


4We have already noticed errors of 5 cm in 3-D reconstruction