Object and Pose Recognition Using Contour and Shape Information

Hugo Cornelius, Danica Kragic and Jan-Olof Eklundh
Computational Vision and Active Perception Laboratory
Centre for Autonomous Systems
Department of Numerical Analysis and Computing Science
Royal Institute of Technology
Stockholm, Sweden
Email: {hugoc, danik, joe}@nada.kth.se

***Abstract*— Object recognition and pose estimation are of significant importance for robotic visual servoing, manipulation and grasping tasks. Traditionally, contour and shape based methods have been considered as most adequate for estimating stable and feasible grasps, [1]. More recently, a new research direction has been advocated in visual servoing where image moments are used to define a suitable error function to be minimized. Compared to appearance based methods, contour and shape based approaches are also suitable for use with range sensors such as, for example, lasers.**

In this paper, we evaluate a contour based object recognition system building on the method in [2], suitable for objects of uniform color properties such as cups, cutlery, fruits etc. This system is one of the building blocks of a more complex object recognition system based both on stereo and appearance cues, [3]. The system has a significant potential both in terms of service robot and programming by demonstration tasks. Experimental evaluation shows promising results in terms of robustness to occlusion and noise.

I. Introduction

Robotic systems are becoming widely used in domestic, industrial and manufacturing settings. For service robot applications commonly operating in highly dynamic environments, robust perception is one of the key system components. Our interest in terms of robotics applications is twofold: i) development of a Programming by Demonstration system (PbD), [4], [5], [6], and ii) object manipulation and grasping for service robot applications in domestic settings, [7].

Programming by Demonstration is a promising approach to automatic robot programming since it allows the robot to learn tasks without end user programming. The end user demonstrates the task to be programmed, and a PbD system interprets the demonstration and determines the set of elementary skills that have to be evoked by the robot in order to perform the task. This is an appealing approach both in terms of service robot applications as well as industrial and manufacturing settings where the end user is not an experienced programmer. One part of our current research deals with learning of motor tasks associated with object grasping using a magnetic measurement device, [6]. Related to this, one of the complex tasks we are currently evaluating is to teach the robot how to set up a dinner table, Fig. 1. In this example, the user is picking up the objects on the table and placing them in the right order with respect to to each other. The system builds the necessary motor primitives by measuring user arm and hand motions (grasps) and relating them to the object considered, Fig. 1.

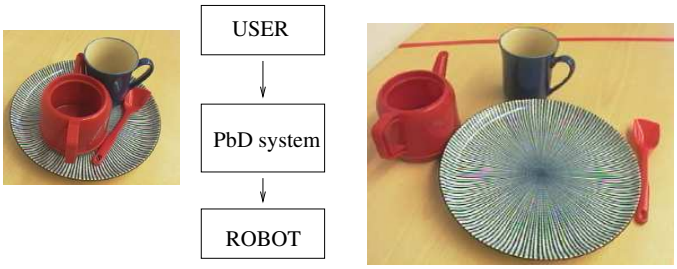

Fig. 1. An example of Programming by Demonstration scenario: setting a table.

In this paper, we consider one of the important building blocks of the PbD system: object recognition and pose estimation. In the above setting, the object manipulated by the end user first has to be recognized in order to make it possible to successfully build the task plan. After the robot has learned the task, object recognition is still required to detect and manipulate objects in future settings. Object recognition methods commonly rely on appearance or shape models. In robotic settings, contour and shape based methods have been considered as most adequate for estimating stable and feasible grasps, [1]. More recently, a new research direction has also been advocated in visual servoing where image moments are used to define a suitable error function to be minimized, [8]. Compared to appearance based methods, contour and shape based approaches are also suitable for use with range sensors such as, for example, lasers, [9].

The object recognition part of our present PbD system uses appearance properties, [10]. However, many of the objects the robot is supposed to manipulate in a general setting have weak texture. Therefore we would like to combine our appearance based object recognition algorithm with a contour based shape recognition algorithm. We believe that such a combination would result in a general and flexible object recognition system.

In this paper we have chosen to take a closer look at the contour based object recognition algorithm by Nelson and Selinger [11]. Our intention is to evaluate this method to see if we in the future can modify it or get inspiration from it to develop a new algorithm that meets our needs.

This paper is organized as follows. In Section II, the algorithm by Nelson and Selinger, as well as some other related work in contour based object recognition is reviewed. The contour extraction algorithm is presented in Section III, followed by a discussion of related issues and problems in Section IV. Extraction and representation of training data are presented in Section V. Experimental evaluation is presented in Section VI and short summary and future work in Section VII.

II. Related Work

Although generic object recognition and classification have been one of the goals of computer vision scientists since its beginnings, there are still a number of major obstacles for achieving this goal. However, in terms of the identification of known objects in different poses considering novel viewing conditions, significant progress has been made recently, [12]. The two main approaches to the problem are appearance and shape/model based methods.

Appearance based approaches represent an object in terms of several object views, commonly raw brightness images. By acquiring a set of object images or reference views, an appearance based object model is constructed. Compared to shape (model) based approaches, no explicit user provided model is necessary. In addition, such a model can also account for non-geometric object properties such as reflectance. Unfortunately, many of the objects that humans and robots have to manipulate in everyday settings, have limited appearance properties since they are uniform in color. In addition, most of the visual servoing and grasping techniques used in robotic object manipulation, rely on geometric properties of the object in order to define a suitable error function (visual servoing) or estimate stable and feasible grasps. We believe that both appearance and shape based approaches are necessary to be able to perform stable object recognition of arbitrary objects in a general setting. Since our previous work considered appearance based approaches, [10], [3], we will in this paper, consider the latter approach.

One large group of the existing methods using contour information for object recognition, assume that the object is described by one closed or sometimes possibly open contour. Among these methods are the ones that use Fourier descriptors to recognize closed contours. The basic idea is that closed 2D curves can be represented by a periodic function, and hence by Fourier descriptors. One such method is for example described in [13]. In this work, closed 2D curves are parameterized, and Fourier descriptors are used to produce a set of normalized coefficients which are invariant under affine transformations. The method is demonstrated on silhouettes of aircraft. Since the shape of airplanes are more or less planar when seen from large distances, they give rise to affine transformations when rotated in 3D. Hence, the method is ideal for this specific task.

Another method for 2D shape recognition is [14]. In this work, shape is described by a possibly open contour. Templates are stored in a database, and an unknown shape is recognized by morphing its contour to the stored templates. A quantification of the morph, invariant to similarity transformations, is used as similarity measure. A clever segmentation of the contours is used to obtain key points that are used to guide the morph. The method is successfully demonstrated on recognition of rigid objects such as tools lying on a uniform background and to cursive handwriting.

Syntactic matching of curves has also been used, for example in [15]. Here, the curve is represented by an ordered list of shape primitives, and syntactic matching between two curves is performed by dynamic programming. In this particular paper the syntactic matching is only used to align the curves. Proximity matching is then used to measure the similarity between the shapes. The method can deal with partial occlusion, and substantial deformations. Experiments matching the occluding contours of real 3D objects have been carried out, and the method has also been used to classify a large set of 2D silhouettes into classes of similar shapes. Like in [14], this method can be applied to open curves.

All three methods described above assume that the shape which is to be recognized is described by one single contour. This means that for these methods to work for object recognition, the occluding contour of the objects have to be obtained. This is in general not possible if we have non-uniform background, and clutter. Since we are looking for a method able to recognize an object lying on a table among other objects, the methods described above are of limited interest to us. For an algorithm to work in cluttered scenes, it cannot rely on the whole occluding contour to be found in one piece. Instead it must use edge fragments or unordered edge points for matching.

Template matching methods, for example [16] or matching using the Hausdorff distance [17], are able to deal with both noise and clutter. In [16], recognition during translation, rotation and scaling of the object is achieved by optimizing the alignment between the template and image by repeatedly translating, rotating and scaling the template relative to the image. The pose of the template is refined in this way until no motion can make the distance between the image and the template smaller. Because of the large risk of getting stuck in a local minimum, the procedure has to be initialized with several different translation, rotation and scaling parameters. The algorithm is difficult to make efficient if we want to allow general motion of the objects, and recognition of 3D objects is difficult since perspective transformations of these shapes are hard to model.

An interesting method which also represents the object as a collection of edge points, and which allows the templates to be deformed, is shape context [18]. In this method sampled edge points in the template and the image are described by the vectors going from the point to all other sampled edge points.

A one-to-one matching between points in the image and the template is found, and the template is then deformed to align the shapes. Although this is a very interesting approach, the method has been shown not to work well in cluttered scenes.

Carmichael and Herbert has developed a method, using local features inspired by shape context, for recognition of wiry objects in cluttered scenes [19]. The method requires many example images of typical scenes with or without the object to be recognized, and it is not invariant to scale. Furthermore, this method is designed for wiry objects made up of mainly stick-like components. Since the objects we are interested in recognizing are in general not of this kind, the method is not suitable for us. A similar method, also using local features, and also specialized in recognizing objects with long thin parts such as bikes and rackets is the method by Mikolajzcyk et al. [20]. In this work, the SIFT feature [21] has been generalized to represent the edges in a neighborhood.

The algorithm we have chosen to investigate further is a method by Nelson and Selinger [2], [11]. The method uses automatically extracted edge segments, and good results have been reported in their papers. The method can deal with moderate occlusion and cluttered backgrounds.

The first step of the algorithm is to extract edge segments from the image. The segments that are long enough are chosen as key curves. For each key curve, a fixed-size image patch is constructed. The key curve is placed at the center of the patch, and all other segments intersecting the patch are mapped into it. At each edge point, the gradient direction at that point is recorded.

When training the system, images of the object are taken from all around the object in such a way that no segments are generated by the background. For each view, patches from approximately the 30 longest edge segments are saved in the memory. In the recognition stage, patches are constructed for all segments that are longer than a certain threshold. These patches are then matched against the ones in the memory.

Matching between a model patch stored in the memory, and a candidate patch from an image is performed as follows. For each edge pixel in the model patch, one looks for an edge pixel with similar gradient direction in the corresponding neighborhood in the candidate patch. If a high enough proportion of the edge points in the model patch has a corresponding point in the candidate patch, we have a match. Patches matched to the same object and view, and whose relative positions, orientations and scales are consistent, are collected in groups. In each group, only one-to-one matching is allowed, and all matched patches in a group provide evidence for a certain object and pose. If there is one object in the image, and we want to decide what it is, the hypothesis with the highest evidence is chosen.

III. Contour Extraction

The method used for extracting the edge segments is the same as the one used by Nelson and Selinger. This is a modified version of a stick-growing algorithm presented by Nelson in [22]. The modified version of this algorithm is able to follow curved boundaries, but break segments at points with high curvature. This edge-extraction algorithm is better suited for object recognition applications than, for example the Canny edge-detector, since the curves extracted this way are less sensitive to view-point changes and changes in illumination [23].

The basic idea behind the edge-extraction algorithm is to grow edges from small seed segments by maximizing a matching score.

Gray scale images are used. First, the gradient magnitude and direction are computed, and a thinned version of the gradient image is obtained by non-maximum suppression in the direction of the gradient. The point with the largest gradient value is chosen from the thinned gradient image, and a short template stick is placed at that point, orthogonal to the gradient direction. The template stick is made up of a straight segment and two end-stop patterns. The straight segment is a Gaussian with the central point extended in one direction, and an end-stop is modeled by the difference of two Gaussians separated by two standard deviations.

The stick is grown by maximizing the correlation between the stick and the gradient image by repeatedly adjusting its position and orientation and by increasing its length. In the gradient image, only pixels where the gradient direction agrees with the direction of the model stick are considered. If the length of the stick grows beyond a certain threshold, the tips of the stick are grown independently. When the tip is grown, the orientation of the tip relative to the rest of the segment is allowed to change. By allowing this, it possible to follow curved edges, but the growth is terminated if the curvature of the edge is too high. When the growth has stopped, i.e. when a maximum correlation is reached, the edge is saved if it is longer than a certain threshold, and all pixels in the gradient image and in the thinned edge map within a certain distance of the segment are marked as used.

To get more curves in one image, the image is split into blocks of about 30×30 pixels, and at the most six curves are started in each block. A curve is allowed to grow out of its starting block and thus cover possible starting points on other blocks.

IV. Issues and Problems

The object recognition algorithm by Nelson and Selinger has proved to work well [2], [11], but it has some problems. The main problem is that the repeatability of the edge features is rather low. The edge segments extracted on an object depend not only on the viewpoint, but also very much on illumination, scale, noise and on the background. When edges are extracted from two images where an object is seen from slightly different viewpoints, some edge segments in the two images will differ a little due to the viewpoint change. Other curves, on the other hand, will have changed completely. These changes can for example be the result of merging or splitting of curves, and a curve can also have disappeared completely in one image or the other. The idea is that a high enough proportion of the curves should only undergo the first type of change and that these curves should still be possible to match.

Curves that change completely are not possible to use. Since recognition only works if enough of the segments extracted from an unidentified object are matched to corresponding segments detected in a training image, curves that change completely are a significant problem.

As mentioned above, the illumination can influence the extracted curves. Inadequate lighting commonly affects the contrast in an image and can make edges on an object invisible or hard to detect. Shadows on, or next to an object will give rise to edge segments, as well as shining parts on glossy objects. These segments do not describe the object well, since they are only the result of a certain illumination. Shadows and reflections on an object can also cause segments following the boundary of an object to break if the shadow or reflection causes the gradient induced by the boundary to change direction.

Also changes in scale will affect the curves. Although the fixed size of the patches is supposed to make them independent of scale, changes in scale will still affect the extracted edge segments that are used to construct the patches.

Another problem, which is a problem in all recognition algorithms, is incorrect matches. Although the extracted edges on an object change slightly between different images, the curves and feature patches still have to be distinct enough to not give rise to too many erroneous matches. The patches used in the method described here are meant to reduce the number of incorrect curve matches, by verifying the match in the local neighborhood. Erroneous matches will however still be present, especially for the cases when no other curves than the key curve intersect the patch.

V. Training Data

For the algorithm to work well, good training data is essential. One way to obtain good training data is by acquiring training images under ideal conditions regarding illumination and background. Another way, that we have chosen, is to do this manually. By extracting the the edge segments from the training images by hand, many of the problems mentioned above can be reduced. First of all, we make sure that only segments that belong only to the object are extracted. This way, noise due to the shadows, reflections and background is avoided. Furthermore, extracting the segments by hand gives us the possibility to make several different combinations of segments on an object. This is good, since we can then make alternative segments that are likely to be detected in different situations. For example, if there is a point with relatively high curvature on an edge, we can make one long segment going through that point, and also split that segment into two shorter ones, meeting at the point with high curvature. By doing this, it is more likely that we can match the segments we get from an object in an image although the illumination, scale and view point is different from in the training image. Consequently, by extracting training segments by hand, there is a larger probability for finding the object in an unknown image.

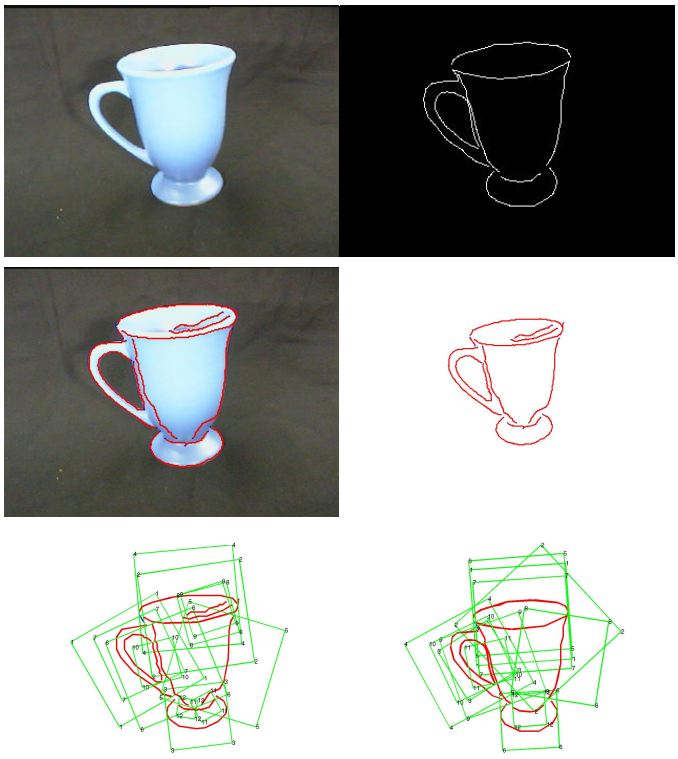

Fig. 2. The images in the first row show the original image, and the image of the edges extracted by hand. The second row shows the automatically extracted edge segments on top of the original image, and on white background. Patches obtained from the automatically extracted edge segments (the left image) and the patches obtained from the edge segments extracted by hand (the right image) are shown on the third row.

VI. Experimental Evaluation

Since the object recognition algorithm of Nelson and Selinger seems suitable for the kinds of situations we are interested in, we have chosen to test it on a set of objects we would like our robot to be able to manipulate. The main difference between our implementation and their implementation of the algorithm is that when all matches are voting for the winning hypothesis, every patch has the same importance regardless of how common the patch is. This is of course sub-optimal, but as will be shown below, the algorithm works rather well anyway.

For experimental evaluation, we have chosen seven objects: a toy car, a plastic bottle, two different cups, a teapot, a spoon and a toy animal, see Fig. 3. The objects were put on a black background and images were taken from a small tilt angle slightly above the object. Between 12 and 16 images were acquired of each object.

About eight images of every object were chosen as training images. Since the spoon and the bottle looked very similar from some directions only three images of the spoon and six images of the bottle were chosen. From the training images, edge segments were gathered manually. When choosing the segments, we tried to mimic the behavior of the automatic edge-extraction algorithm but avoiding segments from shad-

Fig. 3. Test objects used for experimental evaluation.

ows or reflections. In some places, different segments overlapping each other were chosen. This was done in places where it was not quite clear how the edge-extraction algorithm would have behaved, or where just a slight viewpoint change, or a change in illumination would alter the extracted segments. After this, patches were formed from the segments and stored.

In the recognition stage, automatically extracted edge segments were used in all experiments. The first experiment was to match all images of an object to only the training images of the same object. When doing this, every image should be matched to the most similar view in the training set. The results from this experiment were decent, see Table I.

Object	Corr. matches / No. of training images	Corr. matches / No. of images
Toy car	8 / 8	15 / 16
Bottle	6 / 6	9 / 15
Teapot	4 / 7	10 / 14
Cup 1	7 / 8	11 / 12
Cup 2	6 / 8	12 / 15
Spoon	2 / 3	12 / 15
Toy animal	8 / 8	16 / 16

TABLE I

THE TABLE SHOWS THE ALGORITHM'S ABILITY TO DISTINGUISH BETWEEN DIFFERENT VIEWS OF THE SAME OBJECT. IMAGES OF AN OBJECT HAVE BEEN MATCHED TO A NUMBER OF DIFFERENT VIEWS OF THE SAME OBJECT. IN THE FIRST COLUMN, THE SAME IMAGES HAVE BEEN USED AS TRAINING AND TEST IMAGES. IN THE SECOND COLUMN, MORE IMAGES, REPRESENTING VIEWS BETWEEN THE TRAINING VIEWS, HAVE BEEN ADDED TO THE TEST SET.

In the experiment with the toy car, all images were matched correctly, except for one view from the front, that was confused with a view from the back, see Fig. 4. The results for the bottle are the worst ones. Only nine out of 15 times is the bottle matched to the correct view. The main reason for this is that when the matched patches with consistent positions are voting for a view, we do not care about the size of the patch or how often the patch is observed. This is of course not ideal. Common features provide less evidence for a certain view, and larger patches should also be considered more important than smaller ones. In most of the examples with the bottle, the body of the bottle is captured as one single patch, and some smaller details in the image generate further patches. When a view is matched incorrectly, the large patch representing the whole bottle is most of the times not in the winning hypothesis. If matches to the larger patch representing the whole shape of the bottle would be considered more important, the correct match would probably be found more often.

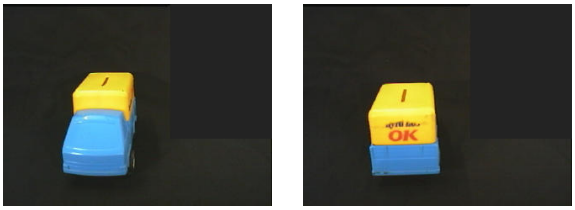

Fig. 4. Front and rear view of the toy car.

Fig. 5. Examples of relatively difficult examples with Cup 2 and the teapot. Note that the contrast between the handle and the rest of the object is poor. Also note the sharp shadow from the handle in the image of the cup.

Cup 1 is matched correctly 11 out of 12 times. The incorrect match occurred when the ear is pointing directly towards the camera. Because of insufficient contrast, the edges on the ear have not been found correctly, and the view has therefore been confused with a view where the ear is hidden behind the cup. The same has happened with Cup 2, which is correctly matched 12 out of 15 times. The teapot is matched correctly in 10 out of 14 cases. The incorrect matches has occurred when the handle and the spout are either on front of or behind the pot. The spoon is matched correctly 12 out of 15 times. The incorrect matches are due to reflections on the object, and shadows beneath it. The toy animal is always matched correctly.

In the next experiment the patches from the training images of all objects were collected in one memory. The recognition algorithm was applied to all images of all objects, and the system was forced to make a choice of what was in the image. The results from this experiment are shown in Table II. The total recognition rate in this experiment was about 87%. The most difficult object was the bottle. The algorithm only managed to recognize the bottle in three of the 15 images of the bottle. If the bottle is removed, there is just one incorrectly classified object, and the recognition rate would be close to 99%. A reason why the bottle is such a difficult object is that it has a rather simple shape, and therefore gives rise to few edge segments. The whole body of the bottle is always present on one singe patch only. Since in our implementation, every match provides the same amount of evidence, it is possible,

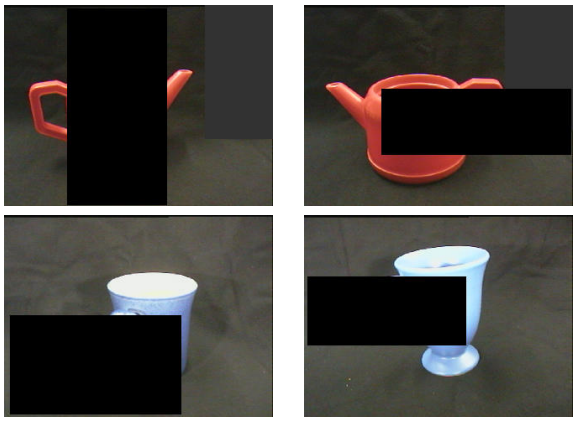

Fig. 6. Examples of partly occluded objects that were successfully recognized.

that two short straight segments provide more evidence for a random object than one single large patch with a bottle shape does for a bottle. This problem could be reduced by assigning different weights to different patches. The spoon also has a simple shape but the recognition rate for the spoon is much higher than for the bottle. This is because most of the edge segments automatically extracted on the spoon come from its boundary, and are not the result of shadows or highlights, as is the case for many edge segments detected on the bottle.

Object Name	Object Number	1	2	3	4	5	6	7
Toy car	1	15	0	1	0	0	0	0
Bottle	2	3	3	1	1	0	2	5
Teapot	3	0	0	14	0	0	0	0
Cup 1	4	0	0	0	12	0	0	0
Cup 2	5	0	0	0	0	15	0	0
Spoon	6	0	0	0	0	0	15	0
Toy animal	7	0	0	0	0	0	0	16

TABLE II

THE TABLE SHOWS THE RESULTS WHEN ALL IMAGES WERE MATCHED TO ALL TRAINING VIEWS FROM ALL OBJECTS. APART FROM THE BOTTLE, ONLY THE TOY CAR HAS BEEN CONFUSED WITH THE TEAPOT ONCE.

The algorithm's robustness to occlusion was also tested. Black rectangles representing occluded parts of the object, were inserted in a number of images and the recognition algorithm was applied. Obviously, the success of the recognition algorithm depends on if a large enough number of the segments extracted from a training image can also be found on the object in the new image. For example, it is often possible to recognize the teapot as long as the handle and the spout is visible. Some successful examples of recognition of partly occluded objects are shown in Fig. 6.

If none or too few of the segments that are extracted from a partly occluded object can be found in the memory, the recognition will fail. This can happen although a large part of the object is still visible. In Fig. 7, two unsuccessful examples are shown. In the example with the cup, the occluding rectangle is placed in such a way that it either occludes the

Fig. 7. Examples of partly occluded objects that were not recognized.

whole edge segment, or breaks it in the wrong place (a point with low curvature). None of the extracted segments looks anything like the segments from the cup in the memory. It is therefore impossible to recognize it. In the teapot image only one edge segment from the spout is the same as a segment in the memory, and this is of course not enough for recognition.

Robustness to noise was tested by generating salt-and-pepper noise in the training images. The results from this experiment were diverse for the different objects. The noise affected the extracted boundaries less if the difference in brightness between the object and the background was large. Consequently, the results are better for the bright objects than for the darker ones.

First, the color of 5000 randomly selected pixels (one pixel can be chosen more than once) were changed to black and 5000 pixels changed to white. Since the image size used was 240×320 pixels, this means that the probability that the color of a pixel is changed is just over 12%. In this experiment, the animal was always recognized, and the correct view was always found. The results were also good for Cup 2. The correct cup and view was found nine times, while the correct cup but incorrect view was chosen three times. In the remaining three examples, the top hypothesis was some view of the other cup and the correct cup was the second best hypothesis. The teapot was recognized with the correct view in about half of the examples. The results for the toy car were just a little bit better. For Cup 1, where the contrast between the object and background is relatively poor, very few useful edge segments could be found in the 12 images, and consequently, recognition always failed.

If 2500 randomly selected pixels are changed to black, and 2500 changed to white (the probability that the color of a pixel is changed is 6.3%), satisfying results are obtained for all objects but Cup 1 and the bottle. The results are summarized in Table III and Table IV. In Table IV it might seem surprising that the bottle is recognized five times although it was only recognized three times in the experiment with images without noise. The simple explanation to this is that in two of the cases, the matches are random, and it is just sheer luck that the image is matched to a bottle. Examples of the images used are shown in Fig. 8.

VII. SUMMARY AND FUTURE WORK

Object recognition and pose estimation are some of the most important building blocks of a service robot system. In this paper, an object recognition algorithm, using automatically extracted edge segments has been evaluated. The algorithm

Object	Corr. matches / No. of training images	Corr. matches / No. of images
Toy car	6 / 8	12 / 16
Bottle	3 / 6	7 / 15
Teapot	6 / 7	11 / 14
Cup 1	5 / 8	9 / 12
Cup 2	6 / 8	13 / 15
Spoon	3 / 3	12 / 15
Toy animal	8 / 8	14 / 16

TABLE III

THE TABLE SHOWS THE SAME KIND OF RESULTS AS TABLE I, BUT IN THE TEST IMAGES, 2500 RANDOMLY SELECTED PIXELS HAVE BEEN CHANGED TO WHITE AND 2500 RANDOMLY SELECTED PIXELS HAVE BEEN CHANGED TO BLACK.

Object Name	Object Number	1	2	3	4	5	6	7
Toy car	1	12	0	0	0	0	0	3
Bottle	2	4	4	1	1	0	0	5
Teapot	3	2	0	12	0	0	0	0
Cup 1	4	1	0	0	2	5	0	4
Cup 2	5	0	0	0	1	14	0	0
Spoon	6	0	0	0	0	1	10	4
Toy animal	7	0	0	0	0	0	0	16

TABLE IV

THE TABLE SHOWS THE SAME KIND OF RESULTS AS TABLE II, BUT IN THE TEST IMAGES, 2500 RANDOMLY SELECTED PIXELS HAVE BEEN CHANGED TO WHITE AND 2500 RANDOMLY SELECTED PIXELS HAVE BEEN CHANGED TO BLACK.

relies on manually extracted training data which gives the possibility of making various alternative segments on some parts of the objects. Since the automatically extracted edge segments depend on the illumination and other factors that are not related to the object, this procedure is advantageous. Providing the training data by hand, make it depend only on the shape of the object and the view point, which is desirable. Moreover, providing various alternative segments, increases the possibility of successful matching, since different segments are likely to show up depending on the illumination etc.

According to the experiments carried out, the algorithm works well and the performance is fairly good in the presence of noise or occlusion. The resistance to occlusion depends heavily on the complexity of the shape of the object. For very simple objects, like for example the bottle only giving rise to one or two long edge segments covering the whole or most of the object, occlusion causes severe problems. For more complex objects, e.g. the teapot, that generate more features that cover only smaller parts of the object, recognition can still be possible even though a large part of the object is occluded. In general, it can be concluded that for recognition to be possible, one or more parts of the object leading to segments not broken by the occluding entity, have to be visible.

Fig. 8. Examples of images used to test the algorithm's robustness to noise. In the first image 2500 randomly selected pixels have been changed to black and 2500 randomly selected pixels have been changed to white. In the second image, twice as many pixels, i.e. 5000, have been changed.

An obvious improvement to the algorithm is to apply weighted voting when the evidence gained from the different matched patches is combined. For example, a patch with a bottle shaped edge on it should provide substantially more evidence for a bottle in the image, then a patch with just a straight line on it provides for any object. The amount of evidence for an object provided by different patches should be investigated by extracting edge segments from a large set of images containing both objects and background. The resulting edge segments, should be analyzed to see which edge segments are most often found on the object, and how often these segments are found on background objects.

Another possible change, that could improve the performance of the algorithm, would be to use color instead of gray-level edges. Color edge operators have been shown to give better results with more detected edges than gray-level edge operators [24]. This would increase the probability of detecting the objects, and hopefully also prevent curves from leaving the object boundary and connecting to highlights or shadows on or behind the object.

The main advantage of the method of Nelson and Selinger is that it uses local features based on edge fragments extracted on the object. These features should be possible to detect also in cluttered scenes, where in general the whole occluding contour is not possible to obtain in one piece. Furthermore, when matching these edge fragments, we get a hypothesis about the translation, rotation and scale of the object in the image. Since the patches used are local they are also insensitive to small view-point changes, which means that with a limited number of training images taken from all around the object, it is possible to recognize 3D objects during general 3D motion.

Even though the edge segments used are easier to obtain in real images than the occluding contours of objects, the repeatability of the curves can be a problem. The points where the edge segments are broken depend not only on the shape of the object, but also on the illumination and background etc. One way to get around the need for simultaneous matching of several curves on one object, would be to try to verify or reject the hypothesis induced by each single curve match. A curve match gives an hypothesis about translation, rotation and scaling of the object and, if the match is correct, we know where in the image we would find its the other parts. One idea would be to verify the identity and position of the object with a template, but since the shape can be distorted due to a perspective transformation, different templates for different parts of the objects would have to be used. Another idea would be to verify the object by matching straight edge lines.

Straight lines can be fit to edges. The points where these lines break are more stable than for the curves, and the probability of obtaining the same lines in two images of an object is greater than the probability of obtaining the same curves. The problem using lines as features for matching is that they are not at all invariant to any transformations. However, if we have a hypothesis about the position of the object, and know where to look for the lines, matching lines could be possible. Since the shape could be distorted due to a view point change, the matching should loosely preserve the global shape of the object, but be more restrictive for lines close to each other. It would be a challenge to make such a method fast enough, but if the procedure is preceded by an attention module, which tells us in what parts of the image to look for the object, it could be achievable.

Another problem with the method, already mentioned above, arises when applying the method to objects with a simple shape, giving rise to very few and long edge segments. As described earlier, these objects are generally not possible to recognize during partial occlusion, and moreover the risk that such a long segment is not correctly detected in a scene with a cluttered background is significant. We can conclude that the method in its present form requires that the object to be detected does not to have a too simple shape.

Further future work will include combination of information from more than one view. Such information can be obtained as the robot approaches a table, or moves around it, since it will then generate images of the same scene from several viewpoints. The intention is also to build a larger system which combines two or more recognition algorithms using both shape and texture.

ACKNOWLEDGMENT

This research has been sponsored by the Foundation for Strategic Research. The foundation is gratefully acknowledged.

REFERENCES

[1] A. Bicchi and V. Kumar, "Robotic grasping and contact: A review," in *Proceedings of the IEEE International Conference on Robotics and Automation, ICRA'00*, 2000, pp. 348–353.

[2] R. Nelson and A. Selinger, "A cubist approach to object recognition," in *ICCV'98*, 1998, pp. 614–621.

[3] M. Bjorkman and D. Kragic, "Combination of foveal and peripheral vision for object recognition and pose estimation," *Proceedings. IEEE International Conference on Robotics and Automation, ICRA'04*, vol. 5, pp. 5135 – 5140, 2004.

[4] M. Kaiser and R. Dillman, "Building elementary robot skills from human demonstration," *Proceedings of the IEEE International Conference on Robotics and Automation, v. 3, pp. 2700–2705*, 1996.

[5] J. Chen and A. Zelinsky, "Programming by demonstration: removing suboptimal actions in a partially known configuration space," *Proceedings of the IEEE Intl. Conf. on Robotics and Automation (ICRA '01), vol. 4, pp. 4096–4103*, 2001.

[6] S. Ekvall and D. Kragic, "Interactive grasp learning based on human demonstration," in *Proc. IEEE/RSJ International Conference on Robotics and automation, ICRA'04*, 2004.

[7] L. Petersson, P. Jensfelt, D. Tell, M. Strandberg, D. Kragic, and H. I. Christensen, "Systems integration for real-world manipulation tasks," in *IEEE International Conference on Robotics and Automation, ICRA 2002*, vol. 3, 2002, pp. 2500 – 2505.

[8] F. Chaumette, "Image moments: a general and useful set of features for visual servoing," *IEEE Trans. on Robotics*, vol. 20 (4), 2004.

[9] G. Taylor and L. Kleeman, "Grasping unknown objects with a humanoid robot," *Australiasian Conference on Robotics and Automation*, 2002.

[10] F. H. S. Ekvall and D. Kragic, "Object recognition and pose estimation for robotic manipulation using color cooccurrence histograms," in *Proc. IEEE/RSJ International Conference Intelligent Robots and Systems, IROS'03*, 2003.

[11] A. Selinger and R. Nelson, "A perceptual grouping hierarchy for appearance-based 3d object recognition," *CVIU*, vol. 76, no. 1, pp. 83–92, October 1999.

[12] http://www.nada.kth.se/~caputo/cognitive-nipsw03.html.

[13] K. Arbter, W. E. Snyder, H. Burkhardt, and G. Hirzinger, "Application of affine-invariant fourier descriptors to recognition of 3-d objects." *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 12, no. 7, pp. 640–647, 1990.

[14] R. Singh and N. Papanikolopoulos, "Planar shape recognition by shape morphing." *Pattern Recognition*, vol. 33, no. 10, pp. 1683–1699, 2000.

[15] Y. Gdalyahu and D. Weinshall, "Flexible syntactic matching of curves and its application to automatic hierarchical classification of silhouettes." *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 21, no. 12, pp. 1312–1328, 1999.

[16] G. Borgefors, "Hierarchical chamfer matching: A parametric edge matching algorithm." *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 10, no. 6, pp. 849–865, 1988.

[17] D. P. Huttenlocher, G. A. Klanderman, and W. Rucklidge, "Comparing images using the hausdorff distance." *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 15, no. 9, pp. 850–863, 1993.

[18] S. Belongie, J. Malik, and J. Puzicha, "Shape matching and object recognition using shape contexts." *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 24, no. 4, pp. 509–522, 2002.

[19] O. T. Carmichael and M. Hebert, "Shape-based recognition of wiry objects," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 26, no. 12, pp. 1537–1552, 2004.

[20] K. Mikolajczyk, A. Zisserman, and C. Schmid, "Shape recognition with edge-based features," in *BMVC'03*, vol. 2, 2003, pp. 779–788.

[21] D. G. Lowe, "Object recognition from local scale-invariant features." in *ICCV*, 1999, pp. 1150–1157.

[22] R. Nelson, "Finding line segments by stick growing," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 16, no. 5, pp. 519–523, May 1994.

[23] A. Selinger and R. Nelson, "Improving appearance-based object recognition in cluttered backgrounds," in *ICPR'00*, 2000, pp. Vol I: 46–50.

[24] A. Koschan and M. Abidi, "Detection and classification of edges in color images," *Signal Processing Magazine, Special Issue on Color Image Processing*, vol. 22, no. 1, pp. 64–73, 2005.