2011 11th IEEE-RAS International Conference on Humanoid Robots
Bled, Slovenia, October 26-28, 2011

Segmentation and learning of unknown objects through physical interaction

David Schiebener*, Aleš Ude*†, Jun Morimoto†, Tamim Asfour‡ and Rüdiger Dillmann‡

*Jožef Stefan Institute, Dept. of Automatics, Biocybernetics and Robotics, Ljubljana, Slovenia
†Department of Brain Robot Interface, ATR Computational Neuroscience Laboratories, Kyoto, Japan
‡Karlsruhe Institute of Technology, Humanoids and Intelligence Systems Lab, Karlsruhe, Germany

Abstract—**This paper reports on a new approach for segmentation and learning of new, unknown objects with a humanoid robot. No prior knowledge about the objects or the environment is needed. The only necessary assumptions are firstly, that the object has a (partly) smooth surface that contains some distinctive visual features and secondly, that the object moves as a rigid body. The robot uses both its visual and manipulative capabilities to segment and learn unknown objects in unknown environments. The segmentation algorithm is based on pushing hypothetical objects by the robot, which provides a sufficient amount of information to distinguish the object from the background. In the case of a successful segmentation, additional features are associated with the object over several pushing-and-verification iterations. The accumulated features are used to learn the appearance of the object from multiple viewing directions. We show that the learned model, in combination with the proposed segmentation process, allows robust object recognition in cluttered scenes.**

I. Introduction

Autonomous learning of the visual appearance of unknown objects from camera images requires that the robot is able to detect and segment new objects in the acquired images. If no prior knowledge about the object and the environment is available, it is in general very difficult to segment it accurately and reliably based on visual information only. Although humans are usually very successful at this task, it is not easy to replicate the equivalent ability in artificial (passive) vision systems [1][2]. The main reason for this is that no clear and comprehensive definition for the concept "object" has been found so far. For each principle that could be used to define the concept of object, e. g. closure, connectedness, etc., counterexamples can be found. Thus in general a sufficient criterion to decide if some part of an observed scene constitutes a part of an object is not known.

Even though simple principles are not sufficient to define the concept of object, they can give hints to generate hypotheses about the existence of objects. The generated hypotheses must then be tested using stronger criteria. When a robot is not constrained to passively observing a scene, but can use its manipulation abilities to physically interact with the scene, it can observe the outcome of its own actions to provide an additional source of information. Like humans, the robot can use its (partial) control over the objects and the resulting visual input to observe - and learn about - the effects of its actions [3]. For example, moving an object can help to extract its boundaries [4]. In [5], the kinematic properties of an unknown articulated object are obtained by moving its parts.

If the robot can grasp an object it is interested in, it can move it in a controlled way. In this case, the object can be segmented reliably and its visual appearance from multiple viewing directions can be learned [6][7]. But grasping of a completely unknown, unsegmented object is in general very difficult, and in some cases it may be impossible anyway because of the size or shape of the object. A simpler alternative is to just push the object. This will result in rather uncontrolled object movements, but has been shown to be sufficient to acquire affordances of unknown objects [3].

In our previous work [8] we showed that pushing can be useful for object segmentation. Here we extend this initial work by providing a methodology to discover more candidate surfaces that give hints about the existence of the object. More importantly, we developed a new approach that allows for reliable feature accumulation across a number of different snapshots. Based on these results we developed an object recognition system, which supports both autonomous object learning and object recognition. The developed system has been tested in a number of experiments that involved both object learning and recognition.

II. Overview

Our method for learning new objects consists of the following four procedures:

- **Generation of object hypotheses:** Visual features that seem to lie on a smooth surface patch are detected and grouped together.
- **Verification by pushing:** The hypothetical object is pushed. The resulting feature motion allows to verify which features belong to the object. Additional features are added if they move concurrently.
- **Feature accumulation:** The above step can be repeated arbitrarily many times to accumulate object features from multiple viewpoints.
- **Learning of a classifier:** Since it is often difficult to reliably extract and track the same feature point across multiple views, we based our recognition system on a bag-of-features approach, which does not require that all features are tracked and matched across different views.

978-1-61284-868-6/11/$26.00 ©2011 IEEE

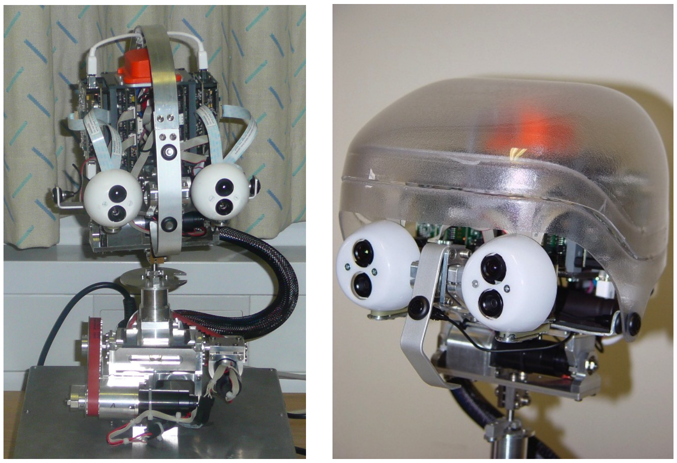

Fig. 1. The Karlsruhe Humanoid Head [9], which is equipped with two pairs of stereo cameras.

III. Hypothesis generation

The first step of our approach for segmenting and learning unknown objects is to form hypotheses about possible objects. They are generated using only the visual information that the robot perceives from its cameras (see Fig. 1). As pointed out in the introduction, the visual information may be misleading, and therefore these hypotheses can only be a starting point and must later be examined further by pushing the hypothetical object and observing the induced feature motion.

The intended scenario for our system is a household environment. Most objects in such environments consist of planar or curved surfaces. Hence it is reasonable to look for planar or cylindrical surface patches, which are mathematically simple to describe, to generate hypothesis about the existence of the objects.

We apply the Harris corner detector [10] to choose interest points that can be used both for hypothesis generation and object learning and recognition. The points determined by this detector are usually distinctive enough to allow for reliable matching in the two images from the stereo cameras. We can calculate the position of the corresponding 3-D point using the calibration of the camera pair [11]. The calibration also allows to use epipolar geometry which reduces the matching problem to a search along the epipolar line. There may still be some incorrect points due to mismatches, but they are too few to affect the hypothesis generation.

Given a set of 3-D points, our goal is to find planes and cylinders that contain as many of these points as possible. For each surface patch, we have to expect that only a rather small part of all features belongs to it. To enable the detection of surface patches among many outliers, we apply the RANSAC algorithm [12], which enables us to find the parameters defining the surface patch that contains maximal subsets of feature points belonging to the parametric surfaces. RANSAC achieves this by randomly selecting a minimal number of points, which is sufficient to calculate the parameters of the sought for surface, and then counting how many points of the whole set lie within a tolerance of the defined surface.

The plane or cylinder containing the largest number of points is added to the list of hypotheses and its points are removed from the set. RANSAC can then be run again on the remaining points. This is repeated until no surface with more than a minimal number of points can be found. The specific approaches to finding planes and cylinders using RANSAC are described in more detail in the following two subsections.

A. Plane detection

A 3-D plane is defined by the equation $ax+by+cz+d=0$ and contains all points (x,y,z) that fulfill this equation. The vector (a,b,c) is the surface normal. If it has unit length, then the above equation gives the distance of the point (x,y,z) to the plane (a,b,c,d). A plane is uniquely defined by three points that are not collinear. With this in mind, the implementation of RANSAC for planes is straightforward:

```
• repeat $N_p$ times:
    – select 3 different points at random
    – calculate the plane parameters
    – check for each point if it lies within tolerance $t_p$ of
      the plane, count the inliers
• return the parameters of the plane with maximal number
  of inliers
```

It can occur that a hypothesis extends to two or more objects which by chance contain points lying in the same plane. To avoid misled attempts of pushing in this case, we group the features of each plane using X-means clustering [13], which is a k-means based algorithm that also estimates the number of clusters. Single points that are far away from the cluster centers are discarded, because they are with high probability outliers. Sometimes a hypothesis containing a large object is accidentally divided by the above clustering process. However, this is not a serious problem for our system because the initial hypothesis will be expanded after the push (as other feature points on the object will move in unison with the initial hypothesis).

B. Cylinder detection

Finding cylinders in a point cloud is more complicated because the parameters of a cylinder can not be determined so easily from a few points on its surface. We applied the algorithm proposed in [14], which uses a 2-stage RANSAC approach, first estimating the cylinder axis and then the appropriate radius and offset from the origin for that axis.

In the first stage, the algorithm uses local surface normals to find promising candidates for possible cylinder axes. To this end, for each 3-D point a local surface normal is estimated using the point and its nearest neighbours. The set of normalized surface normals lies on the unit sphere and is called the *Gaussian image* of the points, as it is the result of applying the *Gaussian map* operation to the set of points. Points belonging to an arbitrary cylinder are mapped to a great circle on the Gaussian sphere. A great circle on the sphere is equivalent to the intersection of this sphere with a plane which passes

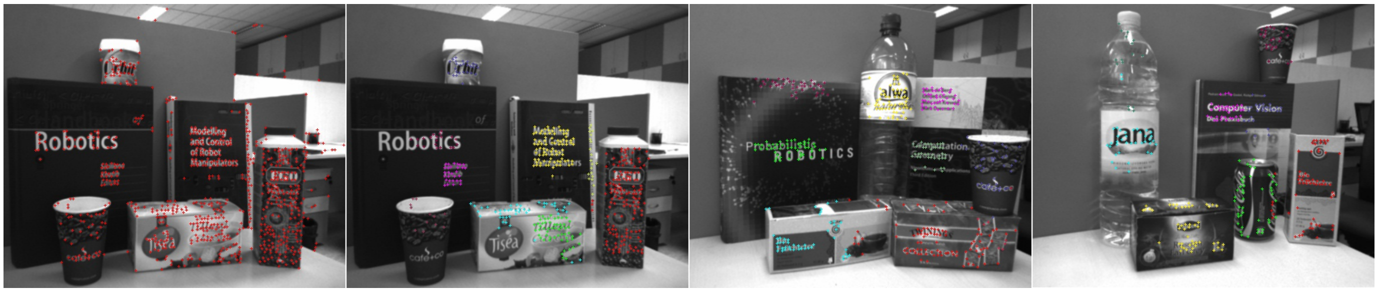

Fig. 2. Hypotheses generation: The left image shows all detected Harris interest points, the other images display the generated hypotheses for each scene. Usually, the hypotheses correspond to a textured region on an object's surface. When objects are close to each other and points on their surfaces lie on a common plane or cylinder, it may happen that these points are subsumed in one hypothesis.

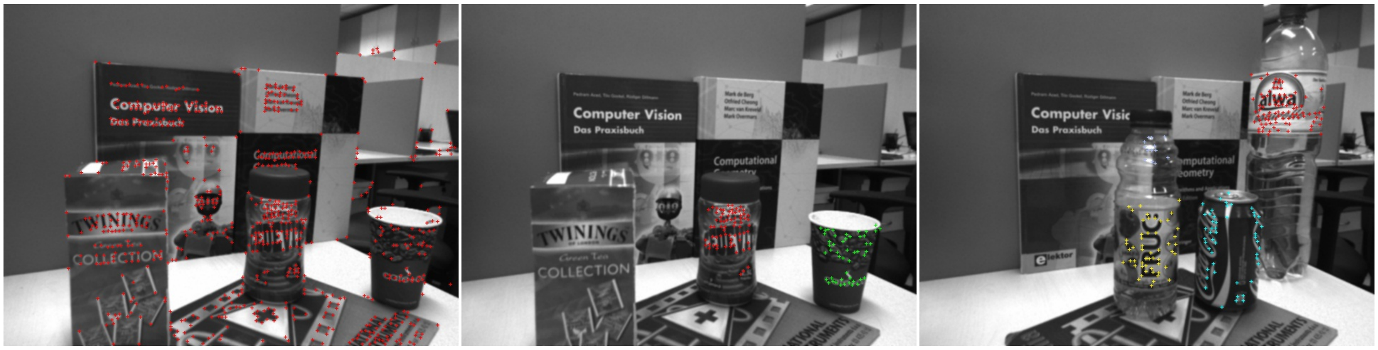

Fig. 3. Hypotheses generation for cylindrical surfaces. The left image shows all Harris interest points, the central and right images show the generated cylindrical hypotheses. Although the two objects in the central image do not have an exactly cylindrical shape, a large part of their surfaces can be captured by the cylinder hypotheses.

through its origin. Therefore, we only need to find the plane passing through the origin that contains the maximal number of points on the Gaussian sphere. This problem is identical to that of finding a plane, where one of the three sample points is always the origin. The normal of the resulting plane is the sought cylinder axis.

Once the cylinder axis has been detected, we still need to find the radius of the cylinder and its offset from the origin. This problem can be reduced to finding a 2-dimensional circle: all points are projected onto the plane orthogonal to the cylinder axis and we need to find a circle with the maximal number of points lying on it. Three non-collinear 2-D points (x_i, y_i) define a circle, its center coordinates (x_c, y_c) are given by

$$x_c = \frac{(y_3 - y_2)(x_1^2 + y_1^2) + (y_1 - y_3)(x_2^2 + y_2^2) + (y_2 - y_1)(x_3^2 + y_3^2)}{2\delta}$$

$$y_c = \frac{(x_3 - x_2)(x_1^2 + y_1^2) + (x_1 - x_3)(x_2^2 + y_2^2) + (x_2 - x_1)(x_3^2 + y_3^2)}{2\delta}$$

where

$$\delta = x_1(y_3 - y_2) + x_2(y_1 - y_3) + x_3(y_2 - y_1)$$

and the radius is simply the distance of one of these points to the center. Finding an optimal circle can therefore easily be done by another application of RANSAC. Here we need to consider only the points that contributed to the great circle on the Gaussian sphere that defines the examined cylinder axis. The radius of the resulting circle is the radius of the cylinder, and the cylinder axis passes through the center of the circle.

When the number of points lying on a cylinder candidate is being determined, only those points are accepted which would lie on the side of the cylinder that is turned towards the camera. To test if a point fulfills this criterion, we check if it lies on the correct side of the plane spanned by the cylinder axis and the vector that is orthogonal both to the cylinder axis and the viewing direction of the camera. This turned out to be very helpful for reducing the number of incorrect hypotheses because sometimes objects are arranged in a way that their sides form a half cylinder opened towards the camera. To further reduce the number of false hypotheses, only cylinders with a rather small radius are accepted, which again avoids the ”fusion” of several objects into one big cylindrical surface.

In every iteration of the outer RANSAC loop, a new possible cylinder axis is determined. After a fixed number of iterations, or when no new axis with more than a minimal support in the Gaussian sphere can be found anymore, the parameters of the cylinder with the maximal number of inliers are returned. Just like in the case of planes, we next discard all points that lie far away from the others to reduce the probability that outliers are included. In our experiments, the clustering of points belonging to one of the detected cylinders was not necessary.

IV. Hypothesis validation by pushing

Additional information need to be provided to verify or discard the generated object hypotheses. By inducing the object to move, visual features can be analyzed for coherent motion, which is a very strong evidence for deciding if they belong to the same object or not. Such information could not be obtained by passive observation. The most common assumption, which we also make, is that the object moves as a rigid body. A more general model of motion would be, for example, an articulated motion [5] or a deformable motion.

Inducing motion on the object, even if it is rather uncontrolled, resolves most of the ambiguities about object segmentation. We use simple pushing movements to verify the initial object hypotheses and to extend them to features that move coherently with the initial features. The initial hypotheses serve as a cue for promising points and directions of pushing. An obvious choice for the hypothesis on which a push is attempted is the one that contains the largest number of features because a large number of features usually result in a more robust estimation of object motion.

A necessary prerequisite for the estimation of feature point motion is to be able to match the features before and after the push. For its descriptiveness and robustness to small rotations, we use SIFT descriptors [15] to find matches of the features in the images before the push and after it. For all initial features for which a corresponding feature is found, the new 3-D positions are calculated using stereo images.

Due to occlusions or too large rotations caused by the induced object motion, some features may not be found again after the push. There may also be mismatches, especially if the object contains non-unique features. Again, RANSAC is a good choice to get a robust estimation of the object motion. The parameters of a transformation associated with the rigid body motion can be obtained from three different pairs of corresponding points before and after the push [16]. If $\mathbf{x}_o$ is the initial position of a point, then its new position $\mathbf{x}_n$ is given by the transformation $\mathbf{x}_n = \mathbf{R}\mathbf{x}_o + \mathbf{t}$, where $\mathbf{R}$ is a 3×3 rotation matrix and $\mathbf{t}$ a translation vector.

After the object has been pushed, the initial hypothesis is evaluated to confirm whether the hypothetic feature points have moved as a rigid body or not. RANSAC is applied to estimate the transformation with which most of the points of the hypothesis concur. The norm of the translation vector $\mathbf{t}$ and the angle of rotation φ, which can be calculated from $\mathbf{R}$, give a measure for the amount of motion resulting from that transformation. The hypothesis is considered confirmed if the weighted sum of $\|\mathbf{t}\|$ and φ is above a threshold. In this case, the features that moved coherently are considered validated and those who did not are discarded. The hypothesis is ignored if the estimated parameters suggest that the hypothetical features did not move. If none of the generated hypotheses moved, another attempt to push one of them is made. If at least one of the hypotheses has moved, and it still contains a sufficient number of features, we assume to have found an object whose appearance needs to be learned.

V. Object learning and recognition

To learn the appearance of the segmented object from multiple viewpoints, the object must be moved, e. g. by pushing, several times. At every step, new points are added to the hypothesis if they seem to belong to the object, and can be verified after the next push. The accumulated set of all verified points, as well as the set of only those verified points that are visible at a given instant, are admissible candidates for representing the appearance of the object. As we use SIFT descriptors for feature matching between stereo image frames, it is an obvious choice to use these features for describing the object. However, it is possible to use any other desired local descriptor at the locations of the confirmed points. Object recognition based on SIFT descriptors, especially when their spatial relationships are incorporated, has been shown to be very successful and reliable [15][17]. Another possibility is the ”bag-of-features” approach [18]. Here a so-called ”visual vocabulary” is learned first by clustering a large number of training features. When working with descriptors later, each of them is assigned to the most similar ”visual word”, i. e. cluster center. A histogram of the occurrences of each visual word on the object is calculated and stored in a database of histograms. To recognize an object, its bag-of-features histogram is calculated for the current, segmented image and matched to the histograms in the database of known objects. We use the bag-of-features approach to memorize the object appearances from different viewpoints and, as we have several histograms for each object, we can apply a k-nearest-neighbours decision for recognition.

A. Object learning

The object needs to be pushed several times to acquire snapshots from different viewpoints. This data can be used to learn a multi-view representation of a successfully segmented object. In this process, the already verified features are tracked as long as they are visible, which enables the system to estimate the underlying object motion. At every step, new Harris interest points are detected in the image, and they are added to the object model if

- they moved in unison with the object during the last pushing action, which implies that they belong to the same rigid body, or
- they lie ”inside” the object, i.e. their distance from the object center is small compared to the extent of the object.

In both cases, the new features have to be verified after the next push before they are confirmed and included in the learned object description. To estimate the object’s motion caused by the push, we use only the confirmed features.

After every push, two bag-of-features histograms of the object are created and saved. One contains all confirmed descriptors that have been accumulated up to the last push. The other histogram contains only the confirmed features that are visible after the last push. While the intent of the first histogram is to have a more comprehensive description of the object, the second one has a snapshot-like character and is

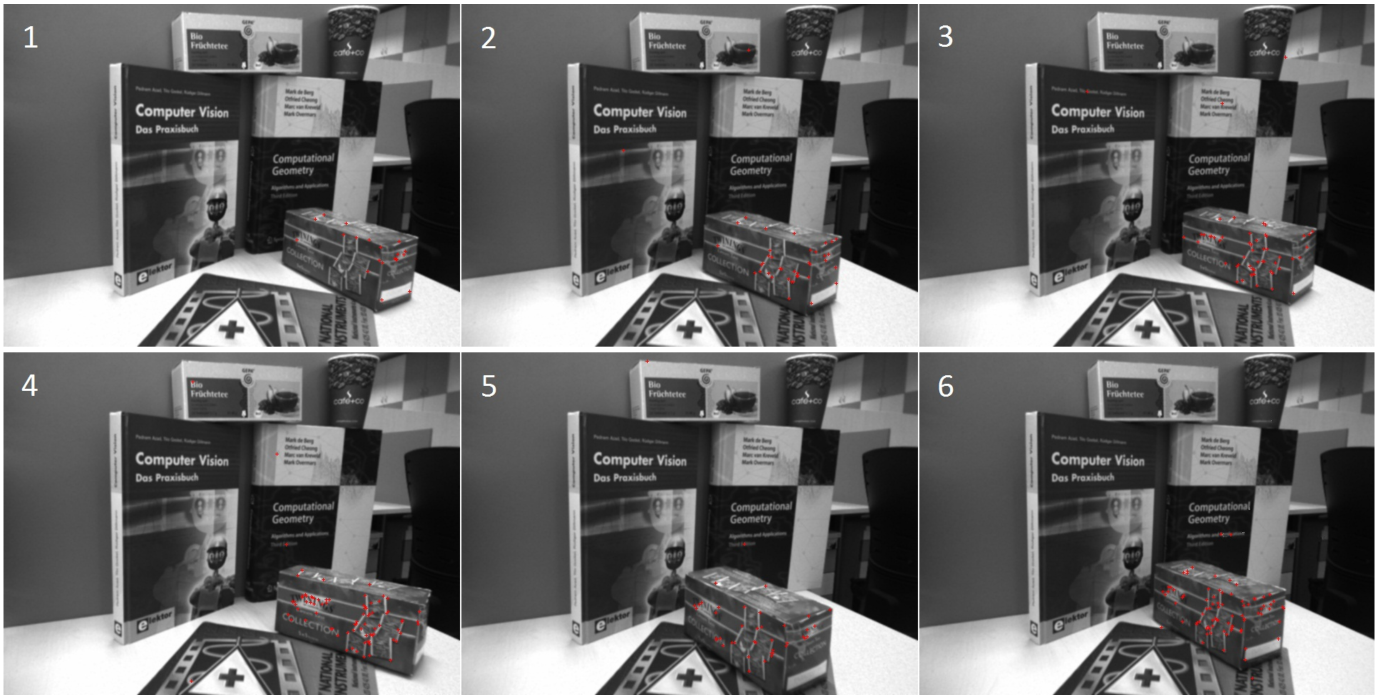

Fig. 4. An object is learned by accumulating verified feature points on its surface during repeated pushing. At each step, new candidate points are added to the object hypothesis and verified after the next push.

more specific to the appearance of the object from the current viewpoint. In our experiments, both types of histograms turned out to be helpful for recognition.

Although the SIFT descriptor is robust to minor viewpoint changes, feature matching fails once the rotation in depth becomes too large, which normally happens after a few pushes. Therefore after each push new descriptors are calculated from the current image for each of the visible, verified feature points. A new descriptor is added to the list of descriptors associated with the feature point if it is significantly different from the old descriptors.

When a confirmed feature becomes invisible, there is a possibility of a mismatch, resulting in an assignment to a point in the image that does not belong to the object. To avoid problems that may arise from such mismatches, confirmed points that do not follow the object's motion two times are not used for the motion estimation anymore. If they do not move in unison with the object four times in succession, they are discarded completely.

The learning process can be continued as long as required. Due to the uncontrolled character of the object motion, there is no guarantee that a complete description of the object will ever be obtained. Still, the chances are good that with a moderate number of pushes a large part of the possible view directions onto the object will be covered.

B. Object recognition

To recognize an object using the bag-of-features approach, its features have to be segmented in the image. Then each of them is assigned to the most similar word of the visual vocabulary and the histogram of word occurrences is calculated. Now the corresponding known object needs to be found, which can be done by comparing the current histogram with the histograms of all known objects using the χ^2 histogram distance. As several histograms of each object are available, conventional classification techniques can be applied for reliable recognition. We use a k-nearest-neighbours classifier to identify the object.

The main difficulty in recognizing objects based on the bags-of-features technique is to correctly segment the hypothetical object that needs to be recognized. If the segmentation contains only some of the object features or many features that do not belong to the object, the histogram is distorted, which makes a correct recognition improbable. Classical approaches to segmentation include feature clustering with k-means or regular and randomized windowing [19]. In our setting – since the object segmentation problem is equivalent to the one we face when learning object histograms – we can apply the same active segmentation algorithm as during the learning process. Moreover, to improve the recognition rate, we can push the object several times, which improves the quality of the segmentation by adding more features and by discarding the unstable ones.

VI. Experimental evaluation

We conducted experiments to evaluate the generation of object hypotheses in complex scenes, the segmentation and learning of unknown objects by pushing them repeatedly, and the recognition of objects using both our initial hypotheses and segmentation results that were improved by pushing the

TABLE I
QUALITY OF THE INITIAL OBJECT HYPOTHESES.

good	part of an object	wrong
50 %	39 %	11 %

TABLE II
OBJECT RECOGNITION SUCCESS RATE OF THREE EXAMPLE OBJECTS, AND THE AVERAGE OF ALL 15 OBJECTS THAT WERE LEARNED.

	init. hyp.	1 push	2 pushes	3 pushes	5 pushes
Book	57 %	54 %	77 %	85 %	90 %
Tea	65 %	77 %	91 %	93 %	97 %
Bottle	69 %	68 %	73 %	78 %	81 %
Average	68 %	65 %	79 %	86 %	92 %

object several times.

In our experiments, the number of initial hypotheses was not limited, but a hypothesis had to consist of at least 10 points. To find planes, 1000 iterations of RANSAC were performed, and the tolerance was 2 mm. With around 500 3-D points, this takes about 15 ms on a standard PC with a 2.67GHz Intel i-7 CPU. For cylinder detection, the local surface normals for the Gaussian sphere were computed by fitting a plane through each point and its 4 nearest neighbours. To find a cylinder axis in the Gaussian sphere, 500 iterations of RANSAC were executed. At most 30 different axes were evaluated, where for each axis at most 10000 RANSAC iterations were executed to find the optimal cylinder radius and offset from the origin (less iterations if there are only few candidate points). The tolerance for deciding if a point lies on a hypothetical cylinder surface was 4 mm. Finding a cylinder in a set of 500 3-D points takes about 150-200 ms. When the first (and largest) planes or cylinders are found and their points are removed, the computation time is reduced significantly. On the average it takes around 350 ms to find all hypotheses. As RANSAC can easily be parallelized, this time can be reduced considerably on a multicore CPU.

The generated hypotheses can fall into three categories of correctness: Firstly, the hypotheses can be approximately identical with an object or at least those parts of it that contain visual features. Secondly, it can contain a part of the object, which frequently happens in the case of large objects. This is acceptable because such a hypothesis still allows a successful manipulation of the underlying object. Thirdly, the hypothesis may span over more than one object. This can lead to failed manipulation attempts unless the majority of the points lie on the pushed object. We carried out a number of experiments in different complex scenes, each containing 5-8 objects that stand close together and partly occlude each other. Table I shows the quality of the hypotheses in such scenes. ”Good” means that the hypotheses approximately coincided with an object, ”part of object” indicates that they contained a part of an object, and the ”wrong” hypotheses contained parts of two or more objects. In simple scenes the hypotheses are usually correct or contain a part of a large object.

We applied our system to the learning of 15 different objects. The number of features contained in each initial object hypothesis varied strongly between the different objects. For the initial hypotheses, the numbers of features ranged from 21 to 153, the average was 53. During the learning process, after each pushing movement 20 - 150 new candidate points were added to the hypothesis, where the actual number strongly depended on the object (54 on average). The percentage of candidate points that were confirmed with the next push appeared to be approximately the same for all objects, on the average 32%. The percentage of feature points of the initial hypothesis which were validated after the first push was approximately the same.

For the evaluation of the object recognition system, in addition to the 15 test objects mentioned above, another 25 objects were learned from presegmented images. Thus the complete database contained 40 objects. We tried to recognize the learned objects in complex scenes containing 5-8 objects. For the bag-of-features, a visual vocabulary of 1000 words was learned from 50000 features that were extracted from 25 images, each containing several objects. For each object, 15-20 histograms were learned, and we used 3-nearest-neighbours classification with χ^2 distance for recognition.

For three exemplary objects and the average of all 15 tested objects, table II shows the recognition results for the initial hypotheses and after n iterations of pushing and validation. On the average, the initial hypotheses lead to a recognition rate of 68%, which also gives an idea about their usefulness for segmentation. While hypotheses that approximately contain an object (compare table I) are usually classified correctly, those hypotheses which contain only a part of an object are frequently rejected or misclassified. Hypotheses that contain two or more objects are usually rejected.

After the first push and the subsequent verification of the hypothetical feature points, the average recognition rate is 65%, which is – somewhat surprisingly – slightly lower than for the initial hypothesis. As now only the confirmed points are used for recognition, the effect of this first push was mainly to remove the features from the object hypothesis that did not move in unison with the majority of feature points, or were not found in the next image. By that, the number of features is reduced to around 32% of the size of the initial hypothesis (see above). Apparently, this affects the recognition so strongly that the positive effect of eliminating the false features is voided. But after the second push, new confirmed features are added at each iteration, and now the positive effect is significant. The recognition rate immediately rises to 79% after the second push, 86% after the third and 92% after the fifth. It finally converges to a value between 92% and 95%.

This general tendency is also visible when looking at the particular objects. As the book is frequently divided into two or three initial hypotheses, it profits significantly from the accumulation of more features from the first to the second push. The tea can be recognized very reliably, while the bottle has only very few features and is therefore more difficult to identify even with a good segmentation.

VII. Conclusion and future work

In this paper, we presented a method for the segmentation and learning of unknown objects in unstructured environments. We generate initial object hypotheses from 3-D points, which were obtained through stereo vision, by detecting planar and cylindrical surfaces amongst them. The hypotheses are then verified, corrected and extended by pushing them repeatedly. Objects are learned using bag-of-feature histograms based on the SIFT descriptors of the points belonging to the object. We have shown experimentally that the objects learned this way can later be recognized, and that the segmentation by pushing can serve as a powerful methodology for recognition in complex scenes.

One possibility to extend our method would be to allow other and more complex geometrical shapes for the initial hypothesis, like spheres, ellipsoids, superquadrics, geons etc. But since many common household objects can roughly be modeled by planes and cylinders and since the accumulation of features after the pushing movements is independent from the shape of the initial hypothesis, the benefit would probably be very limited. A more promising enhancement would be to additionally use different local descriptors. Especially the use of color information could prove to be helpful in complementing the greyscale-based SIFT descriptors. It is also an interesting question if our approach can be adapted to deal with more uniformly colored objects, e. g. by using maximally stable extremal regions (MSER) [20].

Acknowledgment

The research leading to these results has received funding from the European Communitys Seventh Framework Programme FP7/20072013 (Specific Programme Cooperation, Theme 3, Information and Communication Technologies) under grant agreement no. 270273, Xperience. It was also supported by SRBPS, MEXT. A. Ude would like to thank NICT for its support within the JAPAN TRUST International Research Cooperation Program.

References

[1] G. Kootstra, J. Ypma, and B. de Boer, Active exploration and keypoint clustering for object recognition, in: *Proc. IEEE Int. Conf. Robotics and Automation*, Pasadena, CA, 2008.

[2] G. Metta and P. Fitzpatrick, Early integration of vision and manipulation, *Adaptive Behavior*, vol. 11, no. 2., 2003.

[3] G. Metta and P. Fitzpatrick, Grounding vision through experimental manipulation, *Philosophical Transactions of the Royal Society: Mathematical, Physical and Engineering Sciences*, vol. 361, no. 1811, 2003.

[4] P. Fitzpatrick, First contact: An active vision approach to segmentation, in: *Proc. IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, Las Vegas, Nevada, 2003.

[5] D. Katz and O. Brock, Manipulating Articulated Objects With Interactive Perception, *IEEE Int. Conf. Robotics and Automation*, Pasadena, CA, 2008.

[6] A. Ude, D. Omrčen, and G. Cheng, Making object learning and recognition an active process, *Int. Journal of Humanoid Robotics*, vol. 5, no. 2, 2008.

[7] K. Welke, J. Issac, D. Schiebener, T. Asfour, and R. Dillmann, Autonomous Acquisition of Visual Multi-View Object Representations for Object Recognition on a Humanoid Robot, *IEEE Int. Conf. Robotics and Automation*, Anchorage, Alaska, 2010.

[8] E. Stergaršek Kuzmič and A. Ude, Object segmentation and learning through feature grouping and manipulation, *10th IEEE-RAS Int. Conf. Humanoid Robots*, 2010.

[9] T. Asfour, K. Welke, P. Azad, A. Ude, and R. Dillmann, The Karlsruhe Humanoid Head, in: *Proc. IEEE-RAS Int. Conf. Humanoid Robots*, Daejeon, Korea, 2008.

[10] C. Harris and M. Stephens, A combined corner and edge detector, in: *Alvey Vision Conference*, pp. 147151, 1988.

[11] R. Hartley and A. Zisserman, *Multiple View Geometry in Computer Vision*, Cambridge University Press, 2004.

[12] M. A. Fischler and R. C. Bolles, Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography, *Communications of the ACM*, vol. 24, no. 6, 1981.

[13] D. Pelleg and A. Moore, X-means: Extending K-means with Efficient Estimation of the Number of Clusters, in: *Proc. 17th Int. Conf. Machine Learning*, San Francisco, CA, 2000.

[14] T. Chaperon and F. Goulette, Extracting Cylinders in Full 3D Data Using a Random Sampling Method and the Gaussian Image, in: *Proc. Vision Modeling and Visualization Conference*, 2001.

[15] D. G. Lowe, Object recognition from local scale-invariant features, in: *Proc. Int. Conf. Computer Vision*, Corfu, Greece, 1999.

[16] B. K. P. Horn, Closed-form solution of absolute orientation using unit quaternions, in: *Journal Optical Society America A*, vol. 4, 1987.

[17] P. Azad, T. Asfour, and R. Dillmann, Stereo-based 6D object localization for grasping with humanoid robot systems, *IEEE/RSJ Int. Conf. Intelligent Robots and Systems*, 2007.

[18] G. Csurka, C. Dance, L. X. Fan, J. Willamowski, and C. Bray, Visual categorization with bags of keypoints, in: *Proc. ECCV Int. Workshop on Statistical Learning in Computer Vision*, 2004.

[19] A. Ramisa, S. Vasudevan, D. Scaramuzza, R. L. de Mántaras, and R. Siegwart, A tale of two object recognition methods for mobile robots, in: *Proc. 6th Int. Conf. Computer Vision Systems*, 2008.

[20] J. Matas, O. Chum, M. Urba, and T. Pajdla, Robust wide baseline stereo from maximally stable extremal regions, in: *Proc. British Machine Vision Conference*, 2002.