2012 IEEE International Conference on Robotics and Automation
RiverCentre, Saint Paul, Minnesota, USA
May 14-18, 2012

Integrating surface-based hypotheses and manipulation for autonomous segmentation and learning of object representations

Aleš Ude, David Schiebener, Norikazu Sugimoto, and Jun Morimoto

Abstract— **Learning about new objects that a robot sees for the first time is a difficult problem because it is not clear how to define the concept of object in general terms. In this paper we consider as objects those physical entities that are comprised of features which move consistently when the robot acts upon them. Among the possible actions that a robot could apply to a hypothetical object, pushing seems to be the most suitable one due to its relative simplicity and general applicability. We propose a methodology to generate and apply pushing actions to hypothetical objects. A probing push causes visual features to move, which enables the robot to either confirm or reject the initial hypothesis about existence of the object. Furthermore, the robot can discriminate the object from the background and accumulate visual features that are useful for training of state of the art statistical classifiers such as bag of features.**

I. INTRODUCTION

Statistical approaches to object recognition and categorization have received a lot of attention by the computer vision community in recent years. Excellent performance and state-of-the-art results have been achieved with methods such as bag-of-features, which represent an image as a collection of local feature points [2], [25]. However, the bag-of-features methods do not have a built-in ability to segment objects from the background [13]. This can significantly reduce the performance of object recognition, especially if the object image covers only a small portion of the whole image. Designing a reliable and general object segmentation system that works in many different environments and under varying lighting conditions is an extremely difficult problem, but is a necessary component of an autonomous robot. While statistical learning can overcome some of these problems, it typically requires the robot to acquire and process many training images. This is not an option for an autonomous robot, which needs to have the ability to expand its library of objects as quickly as possible to be able to operate in unstructured and uncontrolled environments.

In this paper we propose to overcome the problem of identifying and learning new objects by exploiting the manipulation capabilities of a humanoid robot like the one in Fig. 1. If object manipulability is taken into account, it is much easier to define the concept of object than when only visual characteristics are used [3]. Based on the concept of object manipulability, we can define objects as physical entities that are manipulable by the robot and whose features move in unison when the robot manipulates them. Such characteristics were also used by Gibson [7] to define the concept of object and were exploited for figure-ground segmentation in a number of previous works [5], [10], [11], [12], [14], [21], [22]. While some of these works assume that the object has been first grasped [11], [12], [22], others do allow for simpler actions such as pushing [5], [10], [14], [21] (also called poking, nudging). Although pushing results in a less controlled motion of the pushed object than manipulation after grasping, probing pushing actions are much easier to generate than actions that assume grasping.

In this paper we present a new methodology to generate probing pushes necessary to confirm or reject the initial object hypotheses and techniques for segmenting and learning of unknown objects. Based on 3-D points obtained from local features, regular surface patches and point clusters are detected to form initial object hypotheses. These hypotheses are then validated by the robot as it attempts to push the hypothetical objects. We utilize linear, autonomous dynamic systems to generate the probing pushes. The induced motion provides sufficient cues for distinguishing the pushed object from its environment. After the existence of an object has been confirmed, it is pushed repeatedly to segment and accumulate the features that move in unison with it. We demonstrate that these features enable reliable object learning and recognition. The developed method requires no prior knowledge about the object or the environment, the only

Fig. 1. Humanoid robot CB-i touching an object placed on the table. It has an active visual system, which on the one hand improves the object fixation capabilities, but on the other hand reduces the accuracy of 3-D vision.

A. Ude and D. Schiebener are with Jožef Stefan Institute, Department of Automatics, Biocybernetics, and Robotics, Jamova 39, Ljubljana, Slovenia ales.ude@ijs.si, david.schiebener@ijs.si

J. Morimoto, H. Sugimoto, and A. Ude are with ATR Computational Neuroscience Laboratories, Department of Brain Robot Interface, Kyoto, Japan xmorimo@atr.jp, xsugi@atr.jp, aude@atr.jp

978-1-4673-1405-3/12/$31.00 ©2012 IEEE

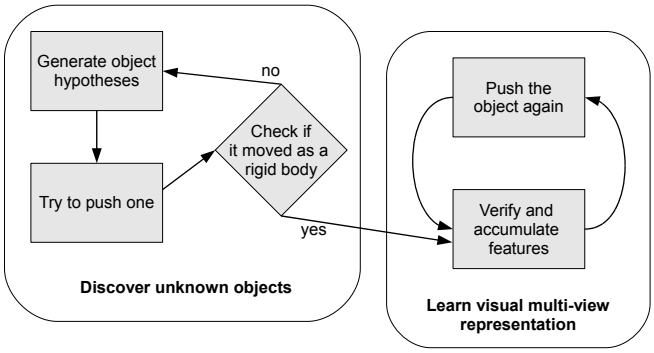

Fig. 2. The system diagram. Essentially, the system consists of two phases; object discovery and object learning / recognition. In the object discovery phase, new hypotheses are generated until one is confirmed through pushing. At this point the system switches to the object learning / recognition phase. In this phase, the robot continuously pushes the hypothetical object, which allows the system to discriminate the object from the background and to accumulate new data for training if in learning mode or evaluate the previously learned classifier if in recognition mode.

necessary assumptions are that the object contains some distinctive visual features and moves as a rigid body. The complete system overview is shown in Fig. 2.

II. Searching for Objects

For the generation of initial object hypotheses we use visual information obtained from stereo cameras of the humanoid robot. In particular, we determine 3-D points within the field of view using stereo calibration on an active camera system [23]. Like in our previous work [21], we use the Harris interest point detector [8] to find points that allow robust stereo matching. These salient points are mostly located in highly textured parts of the image. In our current system, we additionally use color-based maximally stable extremal regions (MSER) [16], [6] as a second type of interest points to complement the Harris interest points in image regions with less texture.

For both Harris interest points and color MSERs, we perform stereo matching using epipolar geometry. In this way we obtain a set of 3-D points, which are usually very reliable and accurate when calculated from Harris interest points, but somewhat less precise, although still mostly useful, when determined from color MSERs. If an object has large untextured areas on its surface, the hypothesis generation benefits significantly from the use of MSERs in addition to the Harris interest points, as can be seen in Fig. 4.

Amongst these 3-D points we look for possible objects. The criteria we use for the initial calculation of object hypotheses are smoothness of surface patches and local proximity of subsets of the detected points. As we consider smooth surface patches to be a more reliable hint about the underlying structure, we search for them first. Planar, spherical and cylindrical surface patches are detected amongst the points using RANSAC [4]. This algorithm repeatedly chooses a random subset of 3-D feature points, calculates the parameters of the considered kind of surface from them, counts how many points of the overall set lie

Fig. 3. Initial hypotheses that were generated for a number of typical household objects. Crosses of the same color belong to the same hypothesis.

Fig. 4. The left image shows hypotheses generated using only Harris interest points. In the right image, color MSERs are used in addition to them, which enables us to recover the surfaces more completely.

within a tolerance of that surface, and returns the best found parameters. It is a robust statistical method and is therefore well suited to detect structures that contain only a small portion of 3-D feature points, which is usually the case in our scenario, especially when there are several objects in the field of view. Details about the detection of planes, spheres and cylinders are given in Sec. II-A and II-B. If none such surfaces are found, the system simply uses localized groups of features to generate initial hypotheses. Even this simple proximity criterion has proved to be quite successful in our real experiments.

From each of the hereby generated hypotheses we remove the points that are far away from the hypothesis' center compared to the extent of the region enclosed by them, as there is a high risk that such feature points are outliers. To avoid subsuming several objects into one hypothesis, we apply X-means [17] to each hypothesis and divide it if that seems appropriate. By doing so, we might create several hypotheses lying on the same object, but that is not a serious problem. All features that belong to the object will later be added again to the hypothesis if they move in unison with it when the object is pushed (see section III-A).

We search for all three considered kinds of surface patches simultaneously and keep the hypothesis that contains the maximum number of feature points, which are then removed from the complete feature point set. This process is repeated with the remaining points until no surface containing more than a minimum number of feature points is found. Finally, we apply X-means clustering algorithm to the remaining points, and the resulting clusters are added as hypotheses if they contain enough points and have a high points-per-volume ratio. With this last step, we can detect objects that contain a cluster of interest points and/or color MSERs, which do not lie on any of the considered smooth surfaces.

A. Planes and spheres

A plane in 3-D space is uniquely defined by three non-collinear points $\mathbf{x}_1, \mathbf{x}_2, \mathbf{x}_3$. Its normal $\mathbf{n}$ can be calculated from these three non-collinear points as $\mathbf{n} = (\mathbf{x}_2 - \mathbf{x}_1) \times (\mathbf{x}_3 - \mathbf{x}_1)$. The plane is then given by the equation $\mathbf{n}^T\mathbf{x} + d = 0$, with $d = -\mathbf{n}^T\mathbf{x}_1$. This equation must be fulfilled by all points $\mathbf{x}$ lying on the plane.

A sphere is uniquely defined by four non-coplanar points $\mathbf{x}_1, \mathbf{x}_2, \mathbf{x}_3, \mathbf{x}_4$. Its parameters can be calculated in a closed form, too. The center $\mathbf{c}$ and radius r of the sphere can be calculated from the points by solving the determinant equation $|\mathbf{M}| = 0$, where

$$\mathbf{M} = \begin{bmatrix} \mathbf{x}^T\mathbf{x} & \mathbf{x}^T & 1 \\ \mathbf{x}_1^T\mathbf{x}_1 & \mathbf{x}_1^T & 1 \\ \mathbf{x}_2^T\mathbf{x}_2 & \mathbf{x}_2^T & 1 \\ \mathbf{x}_3^T\mathbf{x}_3 & \mathbf{x}_3^T & 1 \\ \mathbf{x}_4^T\mathbf{x}_4 & \mathbf{x}_4^T & 1 \end{bmatrix}. \quad (1)$$

Let $\mathbf{M}_{ij}$ denote the submatrix of $\mathbf{M}$ formed by leaving away row i and column j. The solution is given by

$$\mathbf{c} = \begin{bmatrix} 0.5\frac{|\mathbf{M}_{12}|}{|\mathbf{M}_{11}|} \\ -0.5\frac{|\mathbf{M}_{13}|}{|\mathbf{M}_{11}|} \\ 0.5\frac{|\mathbf{M}_{14}|}{|\mathbf{M}_{11}|} \end{bmatrix}, \quad (2)$$

$$r = \mathbf{c}^T\mathbf{c} - \frac{|\mathbf{M}_{15}|}{|\mathbf{M}_{11}|}. \quad (3)$$

If $|\mathbf{M}_{11}| = 0$, the four points are coplanar and there is no solution.

We can find a plane or a sphere in the 3-D point set using RANSAC, where the following steps are repeated N_p times:

- select 3 (4) points at random,
- calculate the parameters of the plane (sphere) defined by these points,
- count how many of the points from the set lie on the plane (sphere).

The plane (sphere) with the maximum number of inliers is then returned.

B. Cylinders

The detection of cylinders within a point set 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 [1], which uses a 2-stage RANSAC approach, first estimating the cylinder axis and then the appropriate radius and offset from the origin of that axis.

Promising candidates for the cylinder axis can be found by analyzing local surface normals. They are calculated from all points and their nearest neighbors and, once normalized, all lie on a unit sphere. A cylinder amongst the point set corresponds to a great circle on the unit sphere. Such great circles are equivalent to the intersection of the sphere with a plane through its origin. Consequently, using RANSAC, the great circle with a maximum number of inliers can be found by testing the great circles defined by the plane through two randomly chosen normals and the origin. The normal of the optimal plane is chosen as the candidate cylinder axis for the next step.

For a given cylinder axis, its offset and the cylinder radius can be determined easily because this problem can be reduced to finding a two dimensional circle. All 3-D points whose local surface normals contributed to the great circle are projected onto the plane orthogonal to the cylinder axis. Using RANSAC again, the circle with the maximum number of points lying on it can be found, exploiting the fact that three non-collinear 2-D points (x_i, y_i) define a circle. Its center coordinates (x_c, y_c) are given by

$$x_c = \frac{(y_3 - y_2)(x_1^2 + y_1^2) + (y_1 - y_3)(x_2^2 + y_2^2)}{2\delta} + \frac{(y_2 - y_1)(x_3^2 + y_3^2)}{2\delta}, \quad (4)$$

$$y_c = \frac{(x_3 - x_2)(x_1^2 + y_1^2) + (x_1 - x_3)(x_2^2 + y_2^2)}{2\delta} + \frac{(x_2 - x_1)(x_3^2 + y_3^2)}{2\delta}, \quad (5)$$

where

$$\delta = x_1(y_3 - y_2) + x_2(y_1 - y_3) + x_3(y_2 - y_1), \quad (6)$$

and the radius is simply the distance of one of these points to the center. The radius of the resulting circle is the radius of the cylinder, and the cylinder axis passes through the center of the circle.

In every iteration of the outer RANSAC loop, a new possible cylinder axis is determined that has to be different from the axes which have already been tested. After a fixed number of iterations, or when no promising new axis can be found anymore, the parameters of the cylinder with the maximum number of inliers are returned.

III. Confirming the Hypotheses by Pushing

The initial object hypothesis includes information about the hypothetical object position, which can be used to generate a probing pushing movement. However, on a robot with many degrees of freedom and an active eye system like in Fig. 1, we cannot rely on the system being accurately calibrated. Even though we account for the eye configurations when calculating stereo triangulation [23], the calculated locations are still rather inaccurate in the robot's body frame.

To improve the accuracy of the probing pushing movements, we included a learning component into our system.

Training is done by moving a robot to a number of locations on the table, on which the robot should look for new objects. We place an object that our vision system can easily localize at a location where the robot hand touches it. Thus, we acquire the following data

$$\{\mathbf{x}_i, \mathbf{y}_i\}_{i=1}^{N}, \tag{7}$$

where $\mathbf{x}_i$ is the position of the object as estimated by the robot's visual system and $\mathbf{y}_i$ are the associated joint angles specifying the robot configuration, including the configuration of its eyes. In our experiments we placed the robot arm at N different locations on a regular grid. To avoid the need for using the robot's inverse kinematics, we estimate function

$$\mathbf{F} : \mathbf{x} \mapsto \mathbf{y}, \tag{8}$$

where $\mathbf{x} \in \mathbb{R}^3$, $\mathbf{y} \in \mathbb{R}^D$, and D is the number of robot degrees of freedom relevant for the task. We applied Gaussian process regression (GPR) [19], which is a state-of-the-art statistical function approximation method, to estimate this function. Given a new desired hand position $\mathbf{x}^* \in \mathbb{R}^3$, the training data $\{\mathbf{x}_i, \mathbf{y}_i\}$, and writing $\mathbf{y}^j = [y_1^j, \ldots, y_N^j]^T$, $j = 1, \ldots, D$, $\mathbf{X} = [\mathbf{x}_1, \ldots, \mathbf{x}_N]$, the associated robot joint configuration $\mathbf{y}^* = [y_1^*, \ldots, y_D^*]^T$ can be estimated as

$$y_j^* = \mathbf{K}_j(\mathbf{x}^*, \mathbf{X})[\mathbf{K}_j(\mathbf{X}, \mathbf{X}) + \sigma_{j,\mathrm{n}}^2 \mathbf{I}]^{-1} \mathbf{y}^j. \tag{9}$$

The coefficients of matrix $\mathbf{K}_j$ are defined as

$$(\mathbf{K}_j(\mathbf{X}', \mathbf{X}''))_{k,l} = \mathrm{k}_j(\mathbf{x}'_k, \mathbf{x}''_l), \tag{10}$$

where k_j is the selected real kernel function (see below) and $\mathbf{X}' = [\mathbf{x}'_1, \ldots \mathbf{x}'_{N'}]$, $\mathbf{X}'' = [\mathbf{x}''_1, \ldots \mathbf{x}''_{N''}]$. Thus $\mathbf{K}_j(\mathbf{x}^*, \mathbf{X}) \in \mathbb{R}^{1 \times N}$ and $\mathbf{K}_j(\mathbf{X}, \mathbf{X}) \in \mathbb{R}^{N \times N}$. The variance of the predicted values can be estimated as

$$\begin{aligned} \mathrm{cov}(y_j^*) = \; & \mathbf{K}_j(\mathbf{x}^*, \mathbf{x}^*) - \\ & \mathbf{K}_j(\mathbf{x}^*, \mathbf{X})[\mathbf{K}_j(\mathbf{X}, \mathbf{X}) + \sigma_{j,\mathrm{n}}^2 \mathbf{I}]^{-1} \mathbf{K}_j(\mathbf{X}, \mathbf{x}^*). \end{aligned}$$

One commonly used kernel function is

$$\mathrm{k}_j(\mathbf{x}', \mathbf{x}'') = \sigma_{j,\mathrm{f}}^2 \sum_{i=1}^{3} \exp\left(-\frac{1}{2} \frac{(x_i - x'_i)^2}{l_{j,i}^2}\right), \tag{11}$$

which results in a Bayesian regression model with an infinite number of basis functions. The parameters $\{\sigma_{j,\mathrm{f}}, \sigma_{j,\mathrm{n}}, l_{j,1}, l_{j,2}, l_{j,3}\}_{j=1}^{D}$ are called hyperparameters and need to be estimated by an off-line nonlinear optimization process. See [19] for more details.

An active eye system is crucial for reliable object fixation with a humanoid robot. On the other hand, error in the estimated positions increases when the eyes are active [23]. Fig. 5 shows that correction by Gaussian process regression can successfully cancel out a part of the estimation error, which enables the robot to touch an object even though the estimated 3-D object positions are fairly inaccurate.

To generate a pushing movement, we first estimate the center point of all 3-D features included in the initial object hypothesis. A probing push can be started from a position sufficiently displaced from this central position. This displacement is generated along a vector parallel to the table with a randomly selected direction. The end position is chosen to be on the other side of the object along the selected pushing vector through the center point.

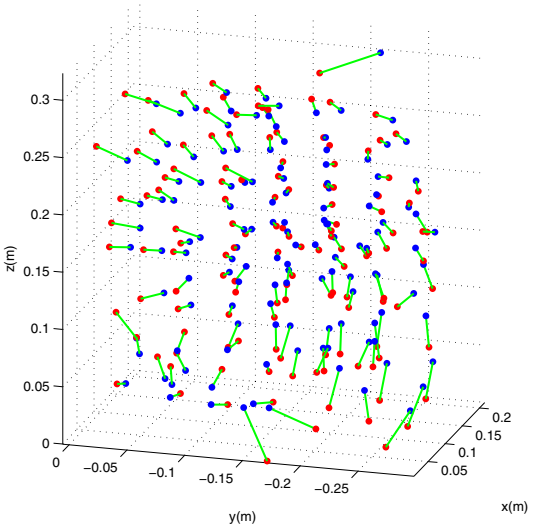

Fig. 5. Blue dots show the robot hand positions estimated by vision, whereas the red dots show the positions calculated by forward kinematics from joint configurations, which were estimated by Gaussian process regression using formula (9). There is a significant systematic error, which is illustrated by green lines.

The simplest way to compute a pushing movement is to generate a straight line between the two end-points and to move the robot hand along this straight line parametrized by time, using function (8) instead of the standard forward kinematics. However, a time-parametrized movement along the straight line is not always suitable for movements in unstructured environments, which are often perturbed and need to be adapted with respect to sensory signals. We therefore decided to generate pushing movements using a discrete pattern generator based on autonomous dynamic systems. The application of dynamic systems as policy primitives is closely related to the idea of motor pattern generators in neurobiology [20]. While general discrete arm movements require the introduction of a nonlinear component like for example the one introduced in [9], this was not necessary for the generation of probing pushing movements. We employed the following linear system to generate the desired point-to-point movements

$$\tau \dot{r} = \alpha_g(g - r) \tag{12}$$

$$\tau \dot{z} = \alpha_z(\beta_z(r - y) - z), \tag{13}$$

$$\tau \dot{y} = z. \tag{14}$$

Here y is one of the degrees of freedom that define the robot configuration $\mathbf{y}$ from Eq. (8), and z and r are auxiliary variables. It is easy to show that the above system is critically damped and that it has a unique attractor point at g for $\alpha_z = 4\beta_z > 0, \alpha_g > 0, \tau > 0$. System (12) – (14) is suitable for the generation of probing pushes because it is guaranteed to converge to g in a smooth manner regardless

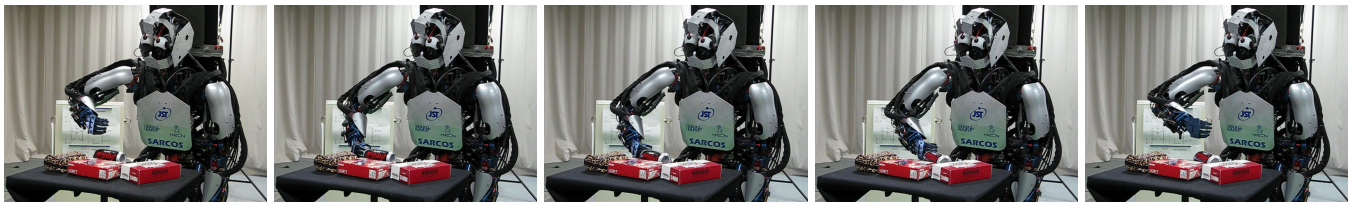

Fig. 6. A successful probing push. The robot starts at the position above the object, moves to the starting position for pushing, applies the probing pushing movement, and withdraws to the position above the object. All movements are generated using linear, autonomous dynamic systems. After the push the robot removes the arm from the viewfield to allow for unobstructed acquisition of the object image.

of the starting position and perturbations. In addition, the speed of movement can be modulated with parameter τ and even if the end configuration g is changed on the fly, the movement remains smooth up to the second order.

We generate a probing pushing behavior by executing a sequence of five dynamic systems (12) – (14), which result in the following movements

- Relocate the hand from its initial position to the position above the starting point for the pushing movement (leftmost image in Fig. 6).
- Move the hand towards the initial position for pushing (second image left in Fig. 6).
- Move the hand from the initial to the end position calculated as described above, thus generating the probing push (from second to fourth image in Fig. 6).
- Move the hand to a position above the end position for pushing (rightmost image in Fig. 6).
- Move the arm away from the viewfield of the robot.

The resulting probing movements are also shown in the video that accompanies this paper. With such a sequence of movements we reduce the possibility that the robot bumps into entities that are not included in the initial hypothesis, which reduces the danger of damaging the robot. Note that to generate such movements, we need to train two functions (8); one to convert 3-D positions above the table and the second to convert 3-D positions on the table into robot configurations. In our experiments we acquired such data by kinesthetic guiding, where the robot arm was lead to a number of positions on and above the table, simultaneously estimating the resulting hand positions by active vision and saving the associated joints as sensed by proprioception.

In theory, a Cartesian straight line movement is more appropriate for probing pushes than a movement generated by a discrete dynamic system. However, since unknown objects cannot be located precisely and because of vision errors, it is not surprising that we observed no performance differences in our object learning experiments when we compared the proposed system with the pushing movements along straight-lines in Cartesian space. Note also that it is possible to utilize dynamic systems to generate Cartesian straight line movements by introducing a nonlinear component into system (12) – (14), like for example proposed in in [9]. The advantage of doing this compared to straightforward time parametrization is that nonlinear dynamic movement primitives retain all positive properties of system (12) – (14) with respect to the movement modulation and robustness against perturbations. As explained above, it was not necessary to follow this route in our experiments.

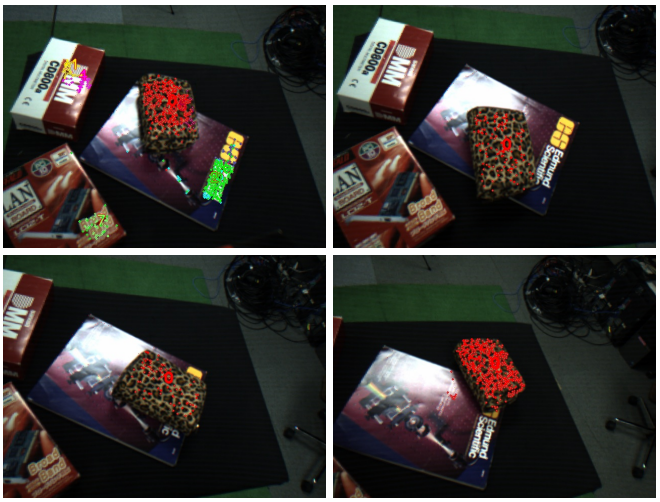

Fig. 7. The extracted features as seen from the robot eyes. The upper left image shows the initial object hypotheses. Hypothesis 0, which contains the largest number of feature points, was selected to generate the initial push. The upper right image shows the confirmed object feature points after the push. The robot then continues pushing the object to acquire more object snapshots from different viewpoints. Note that the head and eyes are active to ensure that the object remains within the robot's viewfield. The number of extracted features can vary considerably from snapshot to snapshot. The acquired feature points are used to train a bag-of-features classifier.

A. Hypothesis Validation

After an object has been pushed, the Harris interest points and color MSERs have to be detected in the new camera images to verify if one of the hypotheses has moved. To match the interest points, we use the SIFT descriptor [15], which has proven to be descriptive and robust to small transformations. For the color MSERs, we use a rating calculated from the ratio of the length of the two principal axes of the region, and the average hue and saturation of the pixels belonging to it.

Since the robot arm very often occludes at least a part of the object during the execution of the pushing action, we do not attempt to track the detected feature points frame by frame. Instead, the correspondences are determined after the probing push has finished. Like before we apply RANSAC [4] to match the features before and after the push. In this case RANSAC uses rigid body motion as the underlying

model that explains the data within the matching process.

As the SIFT descriptors are sensitive to changes in scale and rotations in depth, we associate several descriptors with each point. After a push, we add descriptors at three different scales to the points that have been confirmed. When the number of descriptors associated with a point grows above a certain limit, we apply a k-means clustering to reduce it to the half of that limit. In this way the points can be tracked with high reliability, especially when descriptors from different viewing angles have already been accumulated.

IV. Object Learning and Recognition

The validated object hypothesis can be extended in the course of several push-and-verification steps, by adding new feature points that move consistently with the object or lie within its extent, and are verified or discarded after each push. As the object becomes visible from different directions, its visual appearance can be learned from multiple viewpoints. Features get out of sight when the object is rotated, in which case they are either simply not found or they are mismatched to different feature points. To prevent problems that would arise from mismatched features, a validated feature point that does not move in unison with the hypothesis is not used for the estimation of motion at the next step, and if it does so twice, it is completely discarded.

To encode the visual appearance of an object, we create a bag-of-features model (BoF) as introduced in [2], which is a histogram of the occurrences of feature descriptors that are assigned to clusters learned from a large number of training features. We create the BoF model using SIFT descriptors of the verified feature points belonging to the object hypothesis. To include color information, we do not directly use color MSERs, but instead create a saturation-weighted hue histogram [24] within the ellipse spanned by the principal axes of the set of confirmed interest points and MSER centers. The BoF model and the hue histogram together form an object descriptor that incorporates both local greyscale descriptors of salient points and global color information.

After each push and subsequent validation of the points and MSERs belonging to the hypothesis, two object descriptors are saved. One is created using all validated features that have been accumulated so far, with the intent to obtain a comprehensive description of the object. The other uses only those validated features which are visible at that instant, thus having a snapshot-like character. Depending on the number of pushes, several descriptors are created and saved for each object that needs to be learned.

For object recognition, the descriptor of the considered hypothesis is calculated and compared to the stored descriptors of known objects. As a distance measure between the two descriptors, we use the weighted sum of normalized χ^2 histogram distances of the BoF model and the hue histogram. Both histogram distances are normalized individually by dividing them by the average distance of the hypothesis to all stored histograms. For recognition, we then apply a k-nearest-neighbors decision.

TABLE I

Object recovery rate after motion in depth

distance ratio	1.2x	1.3x	1.4x	1.5x	1.6x
recovery rate	100 %	100 %	91 %	54 %	4 %

TABLE II

Object recovery rate after rotation

rotation angle	20°	30°	40°	50°	60°
recovery rate	100 %	100 %	83 %	56 %	11 %

TABLE III

Object recognition rate for the initial hypotheses and after a few pushes

init. hyp.	1 push	2 pushes	3 pushes
77 %	86 %	96 %	98 %

The performance of bag-of-features based recognition strongly depends on the successful segmentation of the object that needs to be recognized. The segmentation problem is often resolved by statistical feature clustering and by regular or randomized windowing [18]. As the segmentation problem is identical to the one that we face during the learning process, we use our hypothesis generation and active segmentation approach also to support recognition. By pushing the object several times, we achieve very high recognition rates due to the highly accurate segmentation.

A. Experimental Results

Since pushing induces a rather uncontrolled object motion, it is of crucial importance for the success of the learning process that the robot does not loose track of the object. The SIFT descriptor is sensitive to large changes in scale and rotations in depth, therefore large translations in the direction of the camera axis or significant rotations may be harmful, while a translation in the image plane causes no problems. Table I shows with which reliability the object is recovered after a motion along the camera axis. Enlarging the distance from the camera by a certain factor causes a scale change of the same factor. As can be seen, moving the object over a distance of up to 30% of its distance to the camera is unproblematic, above that value there is an increasing risk of loosing track. In practice this means that even for a rather small object-camera distance of 50 cm, a translation of 15 cm is safe.

Greater peril arises from rotations in depth. Table II shows the sensitivity of our approach to such transformations. While a change in orientation of the object of up to 30° is not a big problem, larger rotations may lead to the object not being recovered after the push. Therefore, if the pushing strategy is designed with the intent to reveal different sides of the object, it is safer to execute many small rotations instead of a few large ones.

To perform pushing, the system assumes that the object is placed on a planar surface. Different pushing movements would need to be implemented for different surfaces. The

system does not rely on any particular arrangement of the objects on the planar surface or on a particular type of object motion; it is normally successful as long as the object moves in a different way than any other object in the scene.

To test the usefulness of the obtained object representation for recognition, we learned the appearance of 25 objects from different viewing directions (20 histograms for each object). As recognition is based on a bag-of features model and on a global hue histogram of the object, it is necessary to first segment the object. Then the BoF and hue histogram are calculated, and a 3-nearest neighbors decision based on the χ^2 histogram distance to the known objects is made.

To evaluate the performance of the recognition system, we tested using our initial hypothesis generation (see Sec. II) as well as the validated hypotheses after the probing pushes. Table III shows the recognition accuracy for the initial hypotheses and for confirmed hypotheses after $1-3$ pushes. As can be seen, a combination of the greyscale-based BoF and hue histogram allows for very reliable recognition. In the process of iterative pushing and verification, false features are discarded and an increasingly complete object representation is obtained, which leads to nearly error-free recognition after a few pushes.

V. Conclusion

Previously developed systems based on pushing make different assumptions and use different technologies for segmentation than ours. For example, the approach proposed by Kenney et al. [10] relies on background models, which is a problem for fully active systems like humanoid robots. Li and Kleeman [14] based their approach on symmetry detection. On the other hand, our approach uses robust statistics for segmentation and makes as little assumptions as possible about the objects in the scene.

While in this paper we focused on autonomous acquisition of object models, our system allows the accumulation of knowledge from different sources. Models can be acquired either from large databases of stored models, in interaction with a human teacher where the human teacher performs the pushes instead of the robot, or fully autonomously. Such an approach is essential to prevent on the one hand excessively long learning times and on the other hand to enable acquisition of new knowledge as need arises. We believe that our integrated approach makes an important step towards truly autonomous robots.

VI. Acknowledgments

Research leading to these results was supported in part by the EU Seventh Framework Programme under grant agreement no. 270273, Xperience, "Brain Machine Interface Development", SBRPS, MEXT, and Grant-in-Aid for Scientific Research on Innovative Areas: Prediction and Decision Making 23120004. A. Ude would like to thank NICT for its support within the JAPAN TRUST International Research Cooperation Program.

References

[1] 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.

[2] 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*, Prague, Czech Republic, 2004.

[3] J. Feldman. What is a visual object? *Trends in Cognitive Sciences*, 7(6):252–256, 2003.

[4] M. A. Fischler and R. C. Bolles. Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography. In *Communications of the ACM*, volume 24, 1981.

[5] P. Fitzpatrick. First contact: an active vision approach to segmentation. In *Proc. 2003 IEEE/RSJ Int. Conf. Intelligent Robots and Systems*, pages 2161–2166, Las Vegas, Nevada, 2003.

[6] P. Forssen. Maximally stable colour regions for recognition and matching. In *IEEE Conf. Computer Vision and Pattern Recognition*, 2007.

[7] J. Gibson. *The Ecological Approach to Visual Perception*. Houghton Mifflin, Boston, MA, 1979.

[8] C. Harris and M. Stephens. A combined corner and edge detector. In *Alvey Vision Conference*, page 147151, 1988.

[9] A. J. Ijspeert, J. Nakanishi, and S. Schaal. Movement imitation with nonlinear dynamical systems in humanoid robots. In *Proc. IEEE Int. Conf. Robotics and Automation*, pages 1398–1403, Washington, DC, 2002.

[10] J. Kenney, T. Buckley, and O. Brock. Interactive segmentation for manipulation in unstructured environments. In *Proc. IEEE Int. Conf. Robotics and Automation*, pages 1377–1382, Kobe, Japan, 2009.

[11] D. Kraft, N. Pugeault, E. Baseski, M. Popovic, D. Kragic, S. Kalkan, F. Wörgötter, and N. Krüger. Birth of the object: Detection of objectness and extraction of object shape through object-action complexes. *Int. J. Humanoid Robot.*, 5(2):247–265, 2008.

[12] M. Krainin, P. Henry, X. Ren, and D. Fox. Manipulator and object tracking for in-hand 3D object modeling. *Int. J. Robotics Res.*, 2011 (online first).

[13] S. Lazebnik, C. Schmid, and J. Ponce. Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*, pages 2169–2178, New York, NY, 2006.

[14] W. H. Li and L. Kleeman. Segmentation and modeling of visually symmetric objects by robot actions. *Int. J. Robotics Res.*, 30(9):1124–1142, 2011.

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

[16] 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.

[17] 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.

[18] 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.

[19] C. E. Rasmussen and C. Williams. *Gaussian Processes for Machine Learning*. MIT Press, Cambridge, MA, 2006.

[20] S. Schaal and D. Sternad. Programmable pattern generators. In *Proc. Int. Conf. on Computational Intelligence in Neuroscience*, Research Triangle Park, NC, 1998.

[21] E. Stergaršek-Kuzmič and A. Ude. Object segmentation and learning through feature grouping and manipulation. In *Proc. IEEE-RAS Int. Conf. on Humanoid Robots*, pages 371–378, Nashville, TN, 2010.

[22] A. Ude, D. Omrčen, and G. Cheng. Making object learning and recognition an active process. *Int. J. Humanoid Robot.*, 5(2):247–265, 2008.

[23] A. Ude and E. Oztop. Active 3-D vision on a humanoid head. In *Proc. 14th Int. Conf. Advanced Robotics*, Munich, Germany, 2009.

[24] K. van de Sande, T. Gevers, and C. Snoek. Evaluating color descriptors for object and scene recognition. In *IEEE Trans. Pattern Analysis and Machine Intelligence*, pages 1582–1596, 2010.

[25] J. Zhang, M. Marszalek, S. Lazebnik, and C. Schmid. Local features and kernels for classification of texture and object categories: A comprehensive study. *Int. J. Comput. Vision*, 73(2):123–138, 2007.