Proceedings of the 2006 IEEE/RSJ
International Conference on Intelligent Robots and Systems
October 9 - 15, 2006, Beijing, China

Combining Appearance-based and Model-based Methods for Real-Time Object Recognition and 6D Localization

Pedram Azad, Tamim Asfour and Ruediger Dillmann
Institute for Computer Science and Engineering
University of Karlsruhe,
Haid-und-Neu-Strasse 7, 76131 Karlsruhe, Germany
Email: azad|asfour|dillmann@ira.uka.de

Abstract— **A general solution for image-based object recognition and localization is still a goal far away. Therefore, the only way to tackle the problem is to apply the suitable approach for each specific problem. The most common techniques can be classified into global appearance-based, model-based, or histogram-based approaches, and approaches based on local features. In this paper, we concentrate on recognition and full 6D localization of solid colored objects of any geometry for real-time application on a humanoid robot system. State-of-the-art model-based methods can only deal with object geometries which can be broken down into 3D lines and planes, and thus can be efficiently projected into the image plane, which is not the case for most objects in a realastic scenario. In contrast, appearance-based methods have the power to be applicable for any object geometry, but are rarely combined with full 6D localization of objects, which is required for any realistic application in the context of grasping with a humanoid robot. We present a system which combines the benefits of global appearance-based and model-based approaches, resulting in a system which can acquire object representations automatically given its 3D model, and can recognize and localize solid-colored objects in 6D in an aribtrary scene in real-time.**

I. Introduction

A vision system suitable for grasping of objects in a realistic scenario sets the highest requirements to a humanoid robot system, more than any other application. Not only have the computations to be performed in real-time and objects have to be recognized in an arbitrary scene, but localization has also to deliver full 6D pose information with respect to a 3D rigid model in the world coordinate system with sufficient accuracy. When taking a look at commonly applied image-based vision systems for robot manipulation, one finds that in all cases a very simplified scenario is assumed: objects of simple geometries and a simplified hand. Only recently, research on manipulation of objects with arbitrary geometries with an anthropomorphic five-fingered hand has become feasible and therefore of interest. However, currently no vision system is known which can fulfill the requirements for research in this area. To simplify the requirements to the vision system, usually objects with simple geometries are used, such as cubes or cuboids. As we will show, the algorithms used can not be extended for application with objects of any geometry; they rely on fitting of simple 3D primitives, such as straight lines and planes, which can be projected very efficiently into the image plane. In the following, we present a vision system, inspired by the idea of performing grasp planning based on visual recognition and localization of 3D object models, as presented in [1]. The goal is to provide a vision system which makes it possible to deal with objects of any shape – an extension which is by far non-trivial, as will be shown.

In Section II, the requirements for a component of a vision system in the context of grasping with a humanoid robot system in a realistic scenario are explained. According to these, the limits of state-of-the-art vision systems are shown in Section III. We present our approach in Section IV, first explaining the class of objects to be recognized and localized, and how segmentation takes place. The focus is on full 6D localization of rigid object models and the combination of appearance-based and model-based methods for efficient acquisition of object representations and real-time recognition and localization, which is explained in the sections IV-C and IV-D. Experimental results with the proposed system performed with the humanoid robot ARMAR in a kitchen environment are presented in Section V. The results will be discussed in Section VI, together with potential future work.

II. Requirements

In general, any component of a vision system for a humanoid robot for application in a realistic scenario has to fulfill a minimum number of requirements. In this section, we briefly discuss these requirements, in particular in the context of vision driven grasping of objects.

1) The component has to deal with a potentially moving robot and robot head: The difficulty caused by this is that the problem of segmenting objects can not be solved by simple background subtraction. The robot has to be able to recognize and localize objects in an arbitrary scene when approaching the scene in an arbitrary way.
2) Recognition of objects has to be invariant to 3D rotation and translation: It must not matter in which rotation and translation the objects are placed in the scene.
3) Objects have to be localized in a 3D world coordinate system, in terms of a 3D representation: It is not suffi-

1-4244-0259-X/06/$20.00 ©2006 IEEE

cient to fit the object model to the image, but it is crucial that the calculated 3D pose is sufficiently accurate in the world coordinate system. In particular, the assumption that depth can be recovered from scaling with sufficient accuracy in practice is questionable.

4) Computations have to be performed in real-time: For realistic application, the analysis of a scene and accurate localization of the objects of interest in this scene should take place at frame rate in the optimal case, and should not take more than one second.

Apart from these requirements, it is desired that object representations can be acquired in a convenient manner. For example, the need of using an accurate industrial robot manipulator for acquisition of new object representations including pose information would make such a system unaffordable for many potential users.

III. The Limits of State-Of-The-Art Systems

Most vision systems in the context of grasping and manipulation of objects assume simple shapes. On the other hand, algorithms for object recognition which have the ability to deal with complex shapes are rarely designed for this purpose. In the following, we show the limits of commonly used model-based and appearance based methods.

A. Model-based Methods

Model-based object tracking algorithms are based on relatively simple CAD wire models of objects, as illustrated in Figure 1. Using such models, the starting and end points of lines can be projected very efficiently into the image plane, allowing real-time tracking of objects with relatively low computational effort. However, the limits of such systems are clearly the shapes they can deal with. Most real-world objects, such as cups, plates and bottles, can not be represented in this manner. The crux becomes clear when taking a look at an object with a complex shape, as it is the case for the can illustrated in Figure 2. The only practical way to represent such an object accurately as a 3D model is to approximate its shape by a relatively high number of polygons. To calculate the projection of such a model into the image plane, practically the same computations a rendering engine would do have to be performed, either in software or with hardware acceleration. But not only the significantly higher computational cost makes common model-based approaches not feasible, also from a conceptual point of view the algorithms can not be extended for complex shapes, as will be explained in the following. Objects which can be represented by straight lines and even planes have the property that each edge of the object is represented by a straight line in the model, which are then used for matching. As soon as an object also exhibits curved surfaces this is not the case anymore: The edges of the polygons do not correspond to potentially visible edges. Already the computation of the contour of such an object is a non-trivial task. If one wants to compute all visible edges there is probably no other feasible solution than projecting the 3D model into the image using an offscreen renderer, and treating the result as an image i.e. computing the edges with a gradient filter. However, even with offscreen rendering with hardware acceleration, approximately 100 projections per second, including edge calculation, is the maximum speed that can be achieved – while still not having geometric representations for the edges. Using this technique for comparing and localizing *one* non-symmetric object model with *one* potential region in the image, given a good enough initial estimation of the position, would already need more than 10000 evaluations for *one* frame – only for searching in the rotational space with $-45^o \leq \alpha \leq 0^o$, $45^o \leq \beta \leq 325^o$, $-45^o \leq \gamma \leq 45^o$, with a resolution of 5^o. Thus, when for example having three object models in a database, recognition and localization for two potential regions in the image would take more than ten minutes using such an approach.

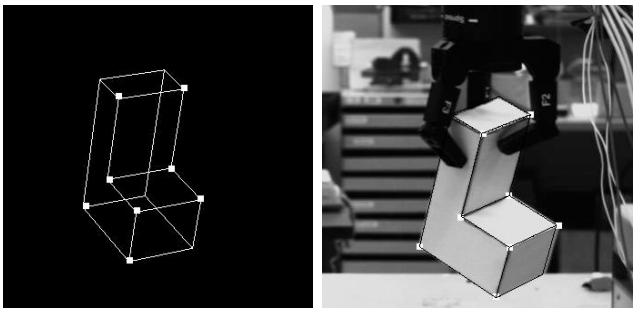

Fig. 1. Illustration of an object modeled by a wire model from [1]

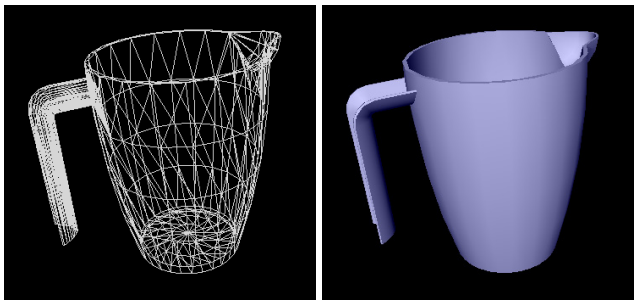

Fig. 2. Illustration of a 3D model of a can

B. Appearance-based Methods

Appearance-based methods span a wide spectrum of algorithms, which can be roughly classified into global and local approaches. While global methods segment a potential region containing an object as a whole, local approaches recognize and localize objects on the base of local features. Global approaches are discussed in Section IV, as our system uses such a method. A further class of methods is based on histograms, which will not be discussed, since they are not suitable for accurate localization in terms of a 3D model. In this section, we want to briefly introduce methods using local features, and show the limits for our intended application.

The use of local features always depends on extracting textural information. Several methods have been proposed for

feature detection, among which are the most popular the Harris corner detector [2], Shi-Tomasi features [3], SIFT features [4], and Maximally Stable Extremal Regions [5]. All object recognition and localization systems based on such features depend on the successful extraction of a sufficient number of features for each object. In Figure 3, the performance of an object tracking system based on Shi-Tomasi features in combination with a 3D object model is illustrated [6]. It has

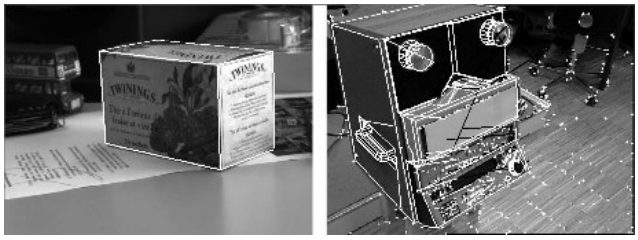

Fig. 3. Illustration of object tracking based on texture features from [6]

been shown that powerful object recognition systems can be built on the base of local features [5], [7]. Thus, it is possible to localize recognized objects based on the same features, as illustrated in Figure 3. However, as already mentioned, this is only possible for objects containing enough features – which is not the case for many objects in our kitchen environment. For example, only rarely dishes contain real texture features, since they are often solid colored with only very few textural information. For such objects, it is more sensible to assume the objects can be segmented, e.g. by color, and solve the problem of recognition and localization with a *global* appearance-based approach, as done in our system, which is explained in detail in Section IV.

IV. Our Approach

Our approach is based on the global appearance-based object recognition system proposed in [8], which is explained briefly in the following. For each object, a set of segmented views is stored, covering the space of possible views of one object. By associating pose information with each view, it is possible to recover the pose through the matched view from the database. For reasons of computational efficiency, PCA [9] is applied for reducing dimensionality. However, the system proposed in [8] from 1996 is far away from being applicable for a humanoid robot in a realistic scenario:

- A black background is assumed.
- Different views are produced using a rotation plate. Thus, objects are not localized in 6D but in 1D.
- Recognition is performed with practically the same setup as for learning.

In the recent past, *global* appearance-based recognition and localization methods have become less popular; the trend goes toward *local* appearance-based methods using texture features. However, there is no practical reason for which global methods are used very rarely. Restricting the range of the material (i.e. texture and color) – not the background – allows relatively easy segmentation of objects. We think that, especially for our intended application, this is a sensible choice for two reasons. First, the goal is the application, therefore it is legitimate to make a simplifying assumption for the segmentation problem, which clearly does not affect the generality of the strategy in any way. Second, segmenting arbitrary objects in an arbitrary scene, which humans are able to do perfectly, is still an unsolved problem in computer vision. Thus, choosing the material so that the objects can be segmented with state-of-the-art methods is a sensible choice to tackle a problem which is unsolved for the general case.

In Section IV-A, we introduce the type of objects our system can deal with and the segmentation method. The region processing pipeline is explained in Section IV-B. In Section IV-C, we present our approach for 6D localization using appearance-based methods and stereo vision. The combination of appearance-based and model-based methods, which leads to a system allowing convenient acquisition and real-time recognition and localization, is presented in Section IV-D.

A. Objects and Segmentation

As already mentioned, we do not restrict the shape of the objects our system can deal with but the material i.e. texture and color. In order to be able to apply state-of-the-art segmentation techniques, we assume a solid colored object. As the focus of this paper clearly is not segmentation, the objects in our test scenario have colors with very high saturation, which allows very robust segmentation in HSV color space.

Color segmentation is a research area of its own. The first choice to make is the color space to operate on; the most commonly used ones are RGB, HSV, and YUV. RGB color space has the drawback that the chrominance is not separated from the luminance components, which is why it is not effective to set bounds for each component. In contrast, especially in HSV color space, setting bounds for the channels H and S is a very effective method for segmenting saturated colors. Independently from the color space, other commonly used methods are either based on histograms or on Gaussian probability density functions using the Mahalanobis distance. In the kitchen environment of our German humanoid robotics program, the lighting conditions are more or less stable, which allows us to use a non-adaptive color model with fixed boundaries for each component in HSV color space. Currently, the optimal parameters are determined manually in an interactive manner. Our experience has shown that once the optimal parameters have been found, the resulting color model is valid throughout the whole kitchen. This is due to the fact that problems usually arise only then when changing daylight produces varying lighting conditions through windows, which is not the case in our test environment. Nevertheless, currently we are working on incorporating an adaptive color model which is capable of dealing with varying lighting conditions. Our experience has shown that the robustness of color segmentation also depends on the camera itself i.e. on the quality of the CCD chip. The segmentation result illustrated in Figure 4 was computed on an input image captured with a *Dragonfly* firewire camera from *Point Grey* [10]. In order to use stereo

Fig. 4. Illustration of the color segmentation result for the colors red and green

vision, color segmentation is performed for the left and the right image. The properties of the color blobs are represented by the color itself, the bounding box, the centroid of the region, and the number of pixels being part of the region. Using this information together with the epipolar geometry, the correspondence problem can be solved very efficiently and effectively. All functionality described above is implemented in the *Integrating Vision Toolkit (IVT)*, developed at our chair at the University of Karlsruhe, and is available on Sourceforge [11].

B. Region Processing Pipeline

Before a segmented region can be used as input for appearance-based calculations it has to be transformed into a normalized representation. For application of the PCA, the region has to be normalized in size. This is done by resizing the region to a squared window of 64×64 pixels. There are two options: resizing with or without keeping the aspect ratio of the region. As illustrated in Figure 5, not keeping the aspect ratio can cause falsifications in the appearance of an object, which lead to false matches. Keeping the aspect ratio can be achieved by using a conventional resize function with bilinear interpolation and transforming the region to a temporary target image with width and height (w_0, h_0), which can be calculated with the following equation:

$$s(w,h,k) := \begin{cases} (k, \lfloor \frac{kh}{w} + 0.5 \rfloor) & : \quad w \geq h \\ (\lfloor \frac{kw}{h} + 0.5 \rfloor, k) & : \quad \text{otherwise} \end{cases} \tag{1}$$

where (w, h) denotes the width and height of the region to be normalized, and k is the side length of the squared destination window. The resulting temporary image of size (w_0, h_0) is then copied into the destination image of size (k, k), which is possible because it is guaranteed that $w_0, h_0 \leq k$. In the second step, the gradient image is calculated for the normalized window. This is done for two reasons. First, symmetries which lead to a very similar projection of an object are less ambiguous in the gradient image, because there the edges gain more significance for correlation. This circumstance is shown in Figure 6, where the half ellipse at the top of the cup in the middle column has more significance. Furthermore, calculating the match on the base of the gradients achieves robustness to varying lighting conditions, which can produce different shading. The second reason is, that some robustness can be achieved to occlusions, because occluded regions do not lead to misclassifications as long as the object is segmented properly and its edges are visible to a sufficient extent, as shown in Figure 10. Finally, in order to achieve invariance to

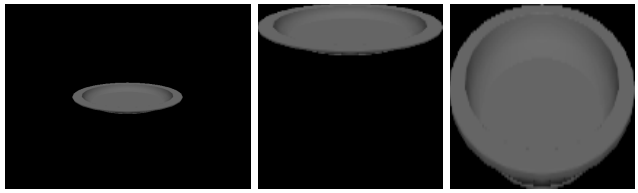

Fig. 5. Illustration of size normalization. Left: original view. Middle: normalization with keeping aspect ratio. Right: normalization without keeping aspect ratio.

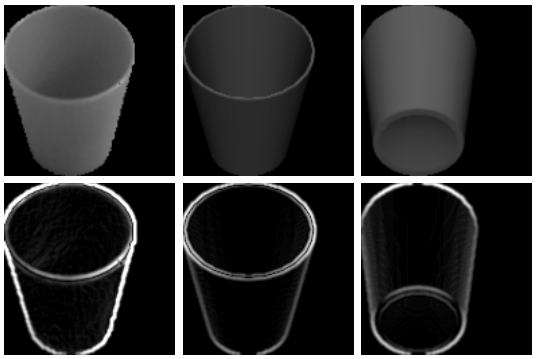

Fig. 6. Illustration of the difference between matching on the grayscale image and on the gradient image. Left: input region. Middle: correct match. Right: wrong similar match.

constant illumination changes i.e. when $\exists c : \forall i \; I_1(i) = cI_2(i)$, the signal energy of each gradient image I is normalized, as proposed in [8], so that:

$$\sum_{n=1}^{k^2} = I^2(n) = 1 \tag{2}$$

By normalizing the intensity of the gradient image, variations in the embodiment of the edges can be handled effectively.

C. Full 6D Localization using Appearance-based Methods

Ideally, for appearance-based 6D localization with respect to a rigid object model, for each object, training views would have to be acquired in the complete six dimensional space i.e. varying orientation *and* position. However, in practice a six dimensional space for the pose is too big for a powerful object recognition and localization system. Therefore, we solve the problem by calculating position and orientation, respectively translation and rotation, independently in the first place. The basis for the calculation of the position is the result of stereo triangulation between the centroids of the matched regions in the left and the right image. However, the result varies with the view of the object, and with its orientation in particular. As illustrated in Figure 7, for each view, a 3D correction vector $\mathbf{c_t}$ can be defined, which is added to the triangulation result during the localization process. Note that this correction

vector is invariant to scaling i.e. to the distance of the object to the camera, since it is a 3D vector and the triangulated point is part of the object. The orientation of an object is

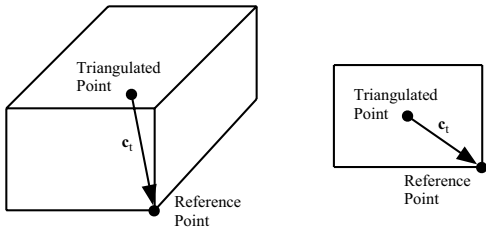

Fig. 7. Illustration of the definition of the position correction vector for two different views of the same object

calculated on the basis of the rotational information which was stored with each view during the acquisition process. However, assuming that translation for each stored view was zero in the x- and y-component causes an error if the object to be localized is not located in the center of the image. Using the previously computed translation, the angles α respectively β in x- respectively y-direction can be calculated, as illustrated in Figure 8 – and on the base of these an orientation correction can be applied. The idea is that the angles α and β cause an error in the pose of the matched view that is equal to the angles themselves. The pose estimation can be formulated by

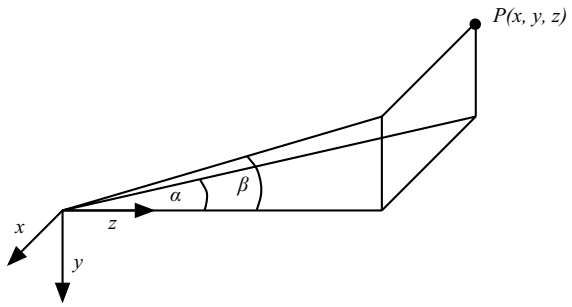

Fig. 8. Illustration of the definition of the angles α and β for orientation correction

the following equations:

$$\mathbf{t} = f(\mathbf{p}_l, \mathbf{p}_r) + \mathbf{c}_t \quad (3)$$

$$\boldsymbol{\theta} = \boldsymbol{\theta}_0 + (\beta\ \alpha\ \alpha)^T \quad (4)$$

where $\mathbf{p}_l$, $\mathbf{p}_r$ denote the centroids of the matching regions in the left and right image, $f : \mathbb{R}^2 \times \mathbb{R}^2 \rightarrow \mathbb{R}^3$ is the transformation performing the triangulation for two matching centroids, $\mathbf{c}_t$ is the position correction vector illustrated in Figure 7, $\boldsymbol{\theta}_0$ is the stored rotation for the recognized view, and $\alpha := \text{atan2}(x, z)$, $\beta := \text{atan2}(y, z)$ are the angles illustrated in Figure 8. Note, that especially the position correction is only an approximation. In order to achieve the highest accuracy possible before a grasp the robot head should center the object of interest in the left image, which is used as input for the appearance-based calculations. The effect of the correction can be seen in Figure 9.

D. Combining Appearance-based and Model-based Methods: Convenient Acquisition and Real-Time Recognition

The approach that has been proposed in the previous sections is purely appearance-based i.e. no model is needed for the acquisition of the views for one object. A suitable hardware setup for the acquisition would consist of an accurate robot manipulator and a stereo head. However, the hardware effort is quite high, and the calibration between the head and the manipulator has to be known for the generation of accurate data. Since our intention is to build a vision system suitable for grasp planning and execution, some kind of 3D model of the objects is needed in any case. For grasp planning and execution, a model in terms of 3D primitives is sufficient, as shown in [1]. If additionally having a rather exact 3D model for each object, it is possible to simulate the views with a 3D engine in software, rather than requiring a hardware setup. Such an approach has several advantages:

- Acquisition of views is more convenient, faster, and more accurate.
- By emulating the stereo setup, the simulation can serve as a valuable tool for the verification of various calculations and optimizations in the optimal case.
- The matched view is automatically given in terms of the 3D model which builds the interface to other applications.

We simulated the stereo setup with *Coin3D* [12], a free implementation of *Open Inventor* for private use and educational purposes. Although focal length is a parameter that can be set in the projective camera model, it is not implemented. Thus, we had to calibrate the simulated projective camera model of Open Inventor. At a resolution of $w \times h = 640 \times 480$ pixels, the measured focal length was $f_x = f_y = 580$ pixels. The measured principal point (c_x, c_y) was the center of the image i.e. $(\frac{w-1}{2}, \frac{h-1}{2})$. No distortion parameters are applied or any other kind of extension of the standard pinhole camera model.

By using an appearance-based approach for a model-based object representation in the core of the system, it is possible to recognize and localize the objects in a given scene in real-time – which is by far impossible with a purely model-based method, as explained in Section III-A. For our experiments, we picked a rotational space of $-45^o \leq \alpha \leq 0^o$, $45^o \leq \beta \leq 325^o$, $-45^o \leq \gamma \leq 45^o$, with a resolution of 5^o, resulting in a search space with $10 \cdot 57 \cdot 19 = 10830$ configurations. For objects which have a rotational symmetry axis, β is set to zero, resulting in $10 \cdot 19 = 190$ configurations. For efficiency considerations, we use PCA to reduce dimensionality from $64 \times 64 = 4096$ to 100. Computation times are explained in Section V.

V. Experimental Results

In this section, we present the experimental results performed with our system. Here, it is not very meaningful to give numbers for the recognition rate and the false positive rate, because the success of recognition depends on the segmentation result – exactly as expected. Segmented objects whose edges are visible to a sufficient extent are recognized and localized

with a rate of 100%. However, a disadvantageous occlusion can lead to a wrong match, which is then filtered by comparing the size of the projected surface of the object to the expected surface, calculated on the base of the determined position and rotation for the recognized object model.

For evaluation of the accuracy, it is suitable to observe position and orientation separately. Sophisticated evaluation of the accuracy of the pose estimation is a very complicated task, since it is hard to determine ground truth information. The accuracy of the position estimation depends on an accurate calibration of the stereo system. With a camera system calibrated with a number of arbitrary views of a chessboard pattern, we could verify that triangulation is accurate with an error of ± 5 mm within the workspace of the robot. We successfully verified the correctness of the orientation by comparing the 3D visualization of the result to the image. Note that the accuracy directly depends on the resolution the views are stored at, which is currently 5^o. The effect of the pose correction presented in Section IV-C is illustrated in Figure 9. However, we are aware that this is only a very qualitative estimation of the accuracy. Since the goal is the task, which is grasping complex shaped objects, we think that it is sensible to make the success of grasping the benchmark for the whole system – although this integrates the errors of all other components such as the estimated kinematics of the head and the arms. The

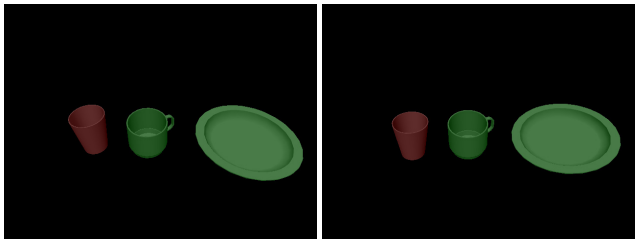

Fig. 9. Effect of the correction formula introduced in Section IV-C in practice. Left: before correction. Right: after correction.

efficiency of the vision system was measured on a 3 GHz CPU. In our tests, the database contained four objects, a cup, a cup with a handle, a measuring cup, and a plate. Since two of these objects have a rotational symmetry axis, the total number of stored views is $2 \cdot 10830 + 2 \cdot 190 = 22040$. With an eigenspace of dimension 100, finding the nearest neighbor with a brute-force approach, comparing the distance in eigenspace for all views, takes approximately 5 ms for one region in the image.

Fig. 10. Recognition and localization result for an exemplary scene. Left: left input image. Right: 3D visualization of the result.

VI. Conclusion

We have presented a vision system for real-time object recognition and 6D localization in terms of a 3D rigid model. The only restriction is the assumption that the objects of interest can be segmented. We could achieve this in our test environment by choosing objects with saturated colors. The focus of our system is to allow accurate localization of objects of any shape with respect to a 3D model, which builds the interface to a grasp planning and grasp execution system. In our experiments, we could verify that our system performs as expected, recognizing the objects in a scene very robustly and reliably in real-time. We are convinced that our system can serve as a valuable basis for research on grasping of complex shaped objects with a humanoid robot with five-fingered hands in a slightly simplified but yet very realistic scenario. Our work in this area is presented in [13]. Future work on the vision system will concentrate on detailed evaluation of the system's accuracy, and on separating the object eigenspace from the universal eigenspace to allow the same performance with a database of many objects, as proposed in [8].

Acknowledgment

The work described in this paper was partially conducted within the EU Cognitive Systems project PACO-PLUS (FP6-2004-IST-4-027657) and funded by the European Commission and the German Humanoid Research project SFB588 funded by the German Research Foundation (DFG: Deutsche Forschungsgemeinschaft).

References

[1] D. Kragic, A. T. Miller, and P. K. Allen, "Real-time tracking meets online grasp planning," in *International Conference on Robotics and Automation (ICRA)*, Seoul, Republic of Korea, 2001, pp. 2460–2465.
[2] C. Harris and M. Stephens, "A combined corner and edge detector," in *Alvey Vision Conference*, Manchester, UK, 1988, pp. 147–151.
[3] J. Shi and C. Tomasi, "Good features to track," in *International Conference on Computer Vision and Pattern Recognition (CVPR)*, Seattle, USA, 1994, pp. 593–600.
[4] D. G. Lowe, "Object recognition from local scale-invariant features," in *International Conference on Computer Vision (ICCV)*, Corfu, Greece, 1999, pp. 1150–1517.
[5] S. Obdrzalek and J. Matas, "Object recognition using local affine frames on distinguished regions," in *British Machine Vision Conference (BMVC)*, vol. 1, Cardiff, UK, 2002, pp. 113–122.
[6] V. Lepetit, L. Vacchetti, D. Thalmann, and P. Fua, "Fully automated and stable registration for augmented reality applications," in *International Symposium on Mixed and Augmented Reality (ISMAR)*, Tokyo, Japan, 2003, pp. 93–102.
[7] E. Murphy-Chutorian and J. Triesch, "Shared features for scalable appearance-based object recognition," in *IEEE Workshop on Applications of Computer Vision*, Breckenridge, USA, 2005.
[8] S. Nayar, S. Nene, and H. Murase, "Real-time 100 object recognition system," in *International Conference on Robotics and Automation (ICRA)*, vol. 3, Minneapolis, USA, 1996, pp. 2321–2325.
[9] G. Dunteman, *Principal Component Analysis*. Sage Publications, 1989.
[10] Point Grey Research, "Dragonfly," http://www.ptgrey.com.
[11] P. Azad, "Integrating Vision Toolkit," http://ivt.sourceforge.net.
[12] Systems in Motion, "Coin3D," http://www.coin3d.org.
[13] A. Morales, T. Asfour, P. Azad, S. Knoop, and R. Dillmann, "Integrated grasp planning and visual object localization for a humanoid robot with five-fingered hands," in *International Conference on Intelligent Robots and Systems (IROS)*, Beijing, China, 2006.