Proceedings of the 2007 IEEE/RSJ International
Conference on Intelligent Robots and Systems
San Diego, CA, USA, Oct 29 - Nov 2, 2007

Stereo-based 6D Object Localization for Grasping with Humanoid Robot Systems

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— **Robust vision-based grasping is still a hard problem for humanoid robot systems. When being restricted to using the camera system built-in into the robot's head for object localization, the scenarios get often very simplified in order to allow the robot to grasp autonomously. Within the computer vision community, many object recognition and localization systems exist, but in general, they are not tailored to the application on a humanoid robot. In particular, accurate 6D object localization in the camera coordinate system with respect to a 3D rigid model is crucial for a general framework for grasping. While many approaches try to avoid the use of stereo calibration, we will present a system that makes explicit use of the stereo camera system in order to achieve maximum depth accuracy. Our system can deal with textured objects as well as objects that can be segmented globally and are defined by their shape. Thus, it covers the cases of objects with complex texture and complex shape. Our work is directly linked to a grasping framework being implemented on the humanoid robot ARMAR and serves as its perception module for various grasping and manipulation experiments in a kitchen scenario.**

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 some 3D rigid model in the world coordinate system with sufficient accuracy.

When taking a look at commonly applied image-based vision systems for robot grasping, one finds that in many cases a very simplified scenario is assumed: objects of simple geometries and a simplified hand. Only recently, research on grasping and manipulation of objects with arbitrary geometries with an anthropomorphic five-fingered hand has become feasible and therefore of interest. However, currently, vision systems that can fulfill the requirements for research in this area are rare.

We will present an object recognition and localization system for two different classes of objects: textured objects and objects that can be segmented globally, e.g. by color, and are defined by their shape. The focus is in both cases on accurate 6D localization with respect to a rigid object model using stereo vision. The application for which this system is designed for is vision-based grasping with humanoid robot systems, for which we have presented our framework in [1]. Furthermore, determining the object pose with the same sensor system and the same stereo algorithms as for the pose estimation of the robot hand allows the implementation of visual servoing techniques without any additional calibration such as hand-eye-calibration.

The paper is organized as follows: In Section II, the requirements for a component of a vision system in the context of autonomous 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 the Sections IV and V. For both subsystems, the focus is on full 6D localization in terms of rigid object models. Experimental results with the proposed system performed with the humanoid robot ARMAR in a kitchen environment are presented in Section VI. The results are discussed in Section VII.

Fig. 1. The humanoid robot ARMAR in a kitchen environment.

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-based 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.

1-4244-0912-8/07/$25.00 ©2007 IEEE.

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 the 3D camera coordinate system and in terms of a 3D representation: It is not sufficient to fit the object model to the image, but it is crucial that the calculated pose is sufficiently accurate in the 3D camera coordinate system. In particular, the assumption that depth can be recovered from scaling with sufficient accuracy is questionable in practice.
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.

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

Most vision systems in the context of grasping and manipulation assume simple object shapes. Furthermore, the benefits of stereo-vision are rarely used together with state-of-the-art feature-based object recognition and localization systems. In this section, we want to show the limits of state-of-the-art systems in the context of vision-based grasping with humanoid robot systems.

A. Model-based Methods

Model-based object tracking algorithms are based on relatively simple CAD wire models of objects. Using such models, the starting and end points of lines can be projected 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 Fig. 2.

Fig. 2. Illustration of a 3D model of a can. Left: wire model. Right: rendered model.

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 is explained in [2]. This is due to the fact that even with offscreen rendering with hardware acceleration, approximately 100 projections per second is the maximum speed that can be achieved for a 3D model as illustrated in Fig. 2. Furthermore, such a projection does not allow establishing correspondences between model points and image points, which would be necessary for any kind of optimization procedure as commonly used in model-based tracking.

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. 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 briefly introduce methods using local features and show their 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 [3], Shi-Tomasi features [4], SIFT features [5], and Maximally Stable Extremal Regions [6]. All object recognition and localization systems based on such features depend on the successful extraction of a sufficient number of features for each object.

It has been shown that powerful object recognition systems can be built on the base of local features ([7], [6], [8]). However, in general, localization is performed on the base of a single camera image and feature correspondences only. Therefore, depth information is determined on the base of scaling. However, a higher accuracy can achieved by using a calibrated stereo system, for which we will show our approach in Section V.

IV. Recognition and Localization based on Shape

Our approach for shape-based object recognition and localization is inspired by the global appearance-based object recognition system proposed in [9], 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, Principal Component Analysis (PCA) is applied for reducing dimensionality. However, the system proposed in [9] from 1996 is far away from being applicable for a humanoid robot in a realistic scenario:

- Different views are produced using a rotation plate. Thus, objects are not localized in 6D but in 1D, which is not suitable for grasping applications.
- Recognition is performed with the same setup as for learning i.e. a humanoid robot would not be allowed to move, since this would cause a change of the viewpoint.

In the following, we will give an outline of the system presented in [2], in which appearance-based methods and stereo vision are combined. A 3D model of the object is used for generating multiple views.

A. Segmentation

For the proposed shape-based approach, the objects have to be segmented. In the presented examples, this is done by performing color segmentation in HSV color space for colored dishes. In order to use stereo vision, segmentation is performed for the left and the right image. The properties of the resulting blobs are represented by 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 efficiently and effectively.

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. The resizing can be done with or without keeping the aspect ratio of the region. As illustrated in Fig. 3, 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:

$$(w_0, h_0) := \begin{cases} (k, \lfloor \frac{kh}{w} + 0.5 \rfloor) & : \quad w \geq h \\ (\lfloor \frac{kw}{h} + 0.5 \rfloor, k) & : \quad \text{otherwise} \end{cases} \tag{1}$$

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). In the second step, the gradient image is calculated for the normalized window, which leads to a more robust matching procedure, as shown in [2]. Finally, in order to achieve invariance to constant multiplicative illumination changes, the signal energy of each gradient image I is normalized, so that $\sum_n = I^2(n) = 1$ (see [9], [2]). By normalizing the intensity of the *gradient* image, variations in the embodiment of the edges can be handled.

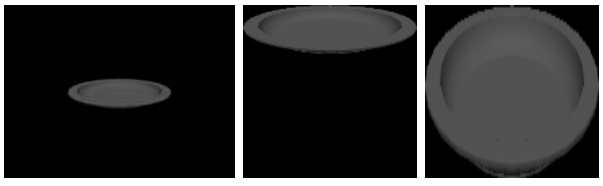

Fig. 3. Illustration of size normalization. Left: original view. Middle: normalization with keeping aspect ratio. Right: normalization without keeping aspect ratio.

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 it is not possible to solve the problem in this six dimensional space directly within adequate time. Therefore, we solve the problem by calculating the position and the orientation independently in first place. A first estimate of the position is calculated by triangulating the centroids of the color blobs. A first estimate of the orientation is retrieved from the database for the matched view. Since the position influences the view and the view influences the position of the centroids, corrective calculations are performed afterwards. Details are given in [2].

Fig. 4. Effect of corrective calculations for three objects lying on a flat table. Left: before correction. Right: after correction.

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

A suitable hardware setup for the acquisition of the view set for an object would consist of an accurate robot manipulator and a stereo camera system. However, the hardware effort is quite high, and the calibration of the kinematic chain between the head and the manipulator has to be known for the generation of accurate data. Therefore, we have used a 3D model of the object to generate the views.

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. To achieve real-time performance, we use PCA to reduce dimensionality from $64 \times 64 = 4096$ to 100. 3D models of rather simple shapes can be generated manually. For more complicated objects we use the interactive object modeling center presented in [10].

V. Recognition and Localization based on Texture

Our system for the recognition and localization of textured objects builds on top of the approach proposed in [7]. After

comparing the different features that we have tested, a summary of the framework for recognition and 2D localization using 2D feature correspondences will be given. Then, we will introduce our approach for 6D localization of planar objects with respect to a rigid object model using stereo vision. In Section VII, we will briefly discuss an extension for partly cylindrical objects such as bottles.

A. Feature Calculation

Various texture-based 2D point features have been proposed in the past. One has to distinguish between the calculation of feature points and the calculation of the feature descriptor. A feature point itself is determined by the 2D coordinates (u, v). Since different views of the same image patch around a feature point vary, the image patches can not be correlated directly. The task of the feature descriptor is to achieve a sufficient degree of invariance with respect to the potentially differing views. In general, such descriptors are computed on the base of a *local* planar assumption.

We have tested three different features respectively descriptors: Shi-Tomasi features and representing a patch by a view set, the Maximally Stable Extremal Regions (MSER) in combination with the Local Affine Frames (LAF) as presented in [6], and the SIFT features [7]. The first approach has been motivated by the work in [8] and our system for the recognition of multiple objects has been presented in [11]. However, despite the use of shared features using k-means clustering, this method does not scale well with an increasing number of objects. This is due the fact that one patch is represented by 100-200 views to achieve invariance. In addition to robustness considerations, the total amount of patches leads to long computation times for the Principal Component Analysis (PCA) compression and k-means clustering. Learning 20 objects with a view set consisting of 20000 patches per object in average takes more than 20 hours on a 3 GHz CPU.

The MSER features are a powerful method for segmenting homogenous regions of arbitrary gray values. In particular, regions with sharp borders such as letters and symbols lead to robust MSER features. Since such regions can be of any size, scale invariance is supported naturally. Invariance to affine transformations is achieved by computing the LAF descriptor. However, although theoretically the LAF are fully scale invariant, in practice the limited resolution often leads to varying embodiments of one MSER at lower scaling steps. In particular, the resolution of 640×480 of ARMAR's cameras turned out to be too low for a robust application of the MSER in a realistic scenario. However, in the future, when more computational power and miniaturized cameras with higher resolution will be availabe, the MSER can serve as valuable features in combination with traditional gradient-based features.

The best results could be achieved with the SIFT features. The SIFT descriptor is fully rotation invariant and invariant to skew and depth to some degree. The feature information used in the following is the position (u, v), the rotation angle φ and a feature vector $\{\boldsymbol{x}_j\}$ consisting of 128 floating point values. These feature vectors are matched using a cross correlation. As the SIFT features are gradient-based, sharp input images with high contrast lead to more features of high quality.

B. Object Recognition

Given a set of n features $\{u_i, v_i, \varphi_i, \{\boldsymbol{x}_j\}_i\}$ with $i \in \{1, \ldots, n\}$ and $j \in \{1, \ldots, 128\}$ that have been calculated for an input image, the first task is to recognize which objects are present in the scene. Simply counting the features does not lead to a robust system since the number of wrong matches increases with the number of objects. Therefore, it is necessary to incorporate the feature positions with respect to each other into the recognition process. The state-of-the-art technique for this purpose is the general Hough transform. We use a two dimensional Hough space with the parameters u, v; the rotative information φ and the scale are used within the voting formula. Given a feature with u, v, φ in the current scene and the matched feature with u', v', φ', the following bins of the Hough space are incremented:

$$\begin{pmatrix} u_k \\ v_k \end{pmatrix} = r \left[\begin{pmatrix} u \\ v \end{pmatrix} - s_k \begin{pmatrix} \cos\Delta\varphi & -\sin\Delta\varphi \\ \sin\Delta\varphi & \cos\Delta\varphi \end{pmatrix} \begin{pmatrix} u' \\ v' \end{pmatrix} \right] \tag{2}$$

where $\Delta\varphi := \varphi - \varphi'$ and $s_k := 0.5 + k \cdot 0.1$ with $k \in \{0, \ldots, 5\}$. The scaling parameters s_k can be adapted depending on the desired scale space. Instead of using a three dimensional hough space, we vote at several scales in a two dimensional hough space, which practically means voting along a straight line. An alternative is to use the scale available from the SIFT descriptor. However, we experienced that this scale information is not as reliable as it is the case for the rotative information. The parameter r is a constant factor denoting the resolution of the Hough space. Currently, we use $r = 0.05$, which means that the Hough space is smaller than the image by a factor of 20 in each direction. The rotation assumes that the v-axis of the image coordinate system is oriented from top to bottom i.e. the coordinate system is left-handed. After the voting procedure, instances of an object in the scene are represented by maxima in the Hough space.

C. 2D Object Localization

After having found an instance of an object, the feature correspondences for this object are filtered by considering only those ones that have voted for this instance. For these correspondences, a homography is computed, which will be explained in the following. A homography is a 2D transformation described by $\boldsymbol{x}' = H\,\boldsymbol{x}$ where H is a 3×3 matrix:

$$H = \begin{pmatrix} h_1 & h_2 & h_3 \\ h_4 & h_5 & h_6 \\ h_7 & h_8 & h_9 \end{pmatrix} .$$

Since this formulation is defined on homogenous coordinates i.e. $\boldsymbol{x} = (u\ v\ 1)^T$, any multiple of H specifies the same transformation. In general, one chooses $h_9 = 1$. An affine transformation is a restriction of a homography with $h_7 = h_8 = 0$. First, an estimate of an affine transformation is

determined by solving the following linear system $A\,\boldsymbol{h} = \boldsymbol{b}$:

$$\begin{pmatrix} u_1 & v_1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & u_1 & v_1 & 1 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ u_n & v_n & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & u_n & v_n & 1 \end{pmatrix} \begin{pmatrix} h_1 \\ h_2 \\ h_3 \\ h_4 \\ h_5 \\ h_6 \end{pmatrix} = \begin{pmatrix} u'_1 \\ v'_1 \\ \vdots \\ u'_n \\ v'_n \end{pmatrix} \quad (3)$$

After having determined the parameters $h_1 \dots h_6$ by, e.g., computing $(A^T A)^{-1} A^T \boldsymbol{b}$ or using a QR-decomposition, the projection error for each correspondence is calculated and compared to the mean error. Correspondences with an error greater than three times the mean error are removed. This procedure is performed in an iterative manner with three iterations. In addition to this technique, we perform a final optimization step, in which we calculate the full homography for the remaining feature correspondences by solving the linear system with the following replaced left side of Equation (3):

$$\begin{pmatrix} u_1 & v_1 & 1 & 0 & 0 & 0 & -u_1\,u'_1 & -v_1\,u'_1 \\ 0 & 0 & 0 & u_1 & v_1 & 1 & -u_1\,v'_1 & -v_1\,v'_1 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ u_n & v_n & 1 & 0 & 0 & 0 & -u_n\,u'_n & -v_n\,u'_n \\ 0 & 0 & 0 & u_n & v_n & 1 & -u_n\,v'_n & -v_n\,v'_n \end{pmatrix} \begin{pmatrix} h_1 \\ \vdots \\ h_8 \end{pmatrix}$$

Using the homography instead of the affine transformation throughout the whole iterative procedure does not lead to a robust system, since the additional two degrees of freedom make the least squares optimization too sensitive to outliers. Only after filtering the outliers by using an affine transformation as described, the computation of the homography is a suitable optimization for the final iteration.

Fig. 5. Difference between 2D localization using an affine transformation only (left) and a homography in the final iteration (right).

If after this process five or more feature correspondences are remaining and the mean error is smaller than two pixels, an instance of the object is declared as recognized. The 2D localization is given by the projection of the four corner points of the front side of the cuboid, which had been marked manually in the training image, as illustrated in Fig. 6.

D. 6D Object Localization

The state-of-the-art technique for 6D localization is to calculate the pose based on the correspondences between 3D model coordinates and image coordinates from one camera image. This is usually done by using the POSIT algorithm [12] or similar methods. The drawback is that the correctness

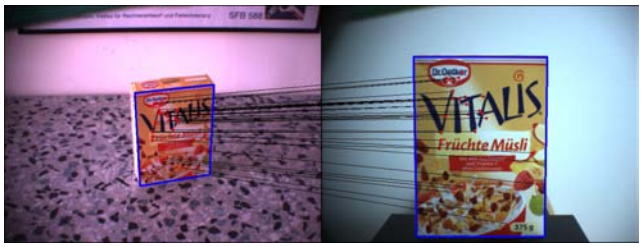

Fig. 6. Correspondences between current view of the scene and training image. Only the valid features after the filtering process are shown. The blue box illustrates the result of 2D localization. Left: input image. Right: training image.

of the calculated pose depends on the accuracy of the 2D correspondences only. In particular, the depth information is very sensitive to small errors in the 2D coordinates of the correspondences. The smaller the area is that the matched features span in relation to the total area of the object, the greater this error becomes. The result of the calculated homography in Figure 7(a) for the right object illustrates this circumstance, where only matches in the upper half of the object could be determined.

However, for a successful grasp, accurate depth information is crucial. Therefore, our strategy is to make explicit use of the calibrated stereo system in order to calculate depth information with maximum accuracy. Our approach for cuboids consists of the following five steps:

1) Determine highly textured points within the calculated 2D contour of the object in the left camera image by calculating Shi-Tomasi features [4].
2) Determine correspondences with subpixel-accuracy in the right camera image for the calculated points by using Zero Mean Cross Correlation (ZNCC) in combination with the epipolar geometry.
3) Calculate a 3D point for each correspondence.
4) Fit a 3D plane into the calculated 3D point cloud.
5) Calculate the intersections of the four 3D lines through the 2D corners in the left camera image with the 3D plane.

The result of this algorithm are the 3D coordinates of the four corners of the object's front surface, given in the world coordinate system. Occlusions can be handled by performing the fitting of the 3D plane with a RANSAC algorithm [13]. To offer the same interface as for the subsystem presented in Section IV, the 6D pose must be determined on the base of the calculated 3D corner points. For this purpose, a simple but yet accurate 3D model of a cuboid for the object is generated manually. The pose of this model with respect to the static pose stored in the file is determined by calculating the optimal transformation between the calculated 3D corner points and the corresponding 3D corner points from the 3D model. This is done by using the method proposed in [14].

VI. Experimental Results

In order to achieve maximum accuracy, we do not undistort or rectify the images but use the distortion and extrinsic camera parameters directly for the stereo calculations. We

have measured the quality of the plane fitting, which indicates the accuracy of the stereo system. With 70-206 3D points for each fitted plane throughout the experiments, the mean error was between 0.7-1.3 mm and the maximum error between 1.8-3.1 mm, with a standard deviation of 0.5-0.9 mm. The next step is to perform an absolute evaluation of both presented systems.

Fig. 7. Recognition and localization result for an exemplary scene. (a) Left input image. (b) 3D visualization of the result.

Figure 7 shows the result of an exemplary scene analysis, which shows that the relative pose of the objects is correct. The absolute depth information depends on the base line and resolution of the stereo system and is therefore scalable. It has to be noted that in the context of grasping with a humanoid robot, the accuracy of the hand-eye calibration is the harder problem. The errors of the 3D measurements of the proposed system are negligible in relation to the hand-eye calibration problem. The benchmark will be the execution of grasping tasks with the humanoid robot ARMAR using the proposed framework in [1].

Currently, the computation time for a complete scene analysis amounts to approximately 1 s on a 3 GHz CPU. The recognition and localization of segmentable objects is performed at frame rate with a database of over 22000 views, as described in [2]. Currently, the computation of the SIFT features takes about 0.4 s and brute-force matching with about 1500 features in the database takes about 0.6 s for one input image. In the future, we will replace the brute-force matching routine by a search using a *k*d-tree structure.

The presented system has been implemented making extensive use of the Integrating Vision Toolkit (IVT), which is available on Sourceforge [15]. The Shi-Tomasi features have been calculated using the OpenCV [16].

VII. Conclusion

We have presented an object recognition and 6D localization system running on the humanoid robot ARMAR with two integrated methods for textured objects as well as objects that can be segmented globally and are defined by their shape. In both subsystems, the 6D pose is calculated by making explicit use of the stereo system to attain maximum depth accuracy. By offering the proposed two solutions with the exact same interface, our system serves as a valuable platform for visual perception in the context of vision-based grasping with humanoid robot systems. It enables research on integrated grasp planning and execution with realistic objects and complex shapes in realistic scenarios.

Currently, we are working on integrating localization of partly cylindrical, textured objects as it is the case for most bottles encountered in a kitchen environment. For this purpose, our presented approach for cuboids can be modified by extending the hough space and fitting a cylinder into the point cloud. Furthermore, we are working on an automatic and accurate calibration procedure for the eye unit of the robot head, which will allow to make use of the stereo calibration also when moving the eyes independently.

Acknowledgment

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

References

[1] 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.

[2] P. Azad, T. Asfour, and R. Dillmann, "Combining Apperance-based and Model-based Methods for Real-Time Object Recognition and 6D Localization," in *International Conference on Intelligent Robots and Systems (IROS)*, Beijing, China, 2006.

[3] C. Harris and M. Stephens, "A Combined Corner and Edge Detector," in *Alvey Vision Conference*, Manchester, UK, 1988, pp. 147–151.

[4] 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.

[5] D. G. Lowe, "Distinctive Image Features from Scale-Invariant Keypoints," *International Journal of Computer Vision (IJCV)*, vol. 60, no. 2, pp. 91–110, 2004.

[6] 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.

[7] D. G. Lowe, "Object Recognition from Local Scale-Invariant Features," in *International Conference on Computer Vision (ICCV)*, Corfu, Greece, 1999, pp. 1150–1517.

[8] 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.

[9] 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.

[10] R. Becher, P. Steinhaus, R. Zöllner, and R. Dillmann, "Design and Implementation of an Interactive Object Modelling System," in *Robotik/ISR*, München, Germany, Mai 2006.

[11] K. Welke, P. Azad, and R. Dillmann, "Fast and Robust Feature-based Recognition of Multiple Objects," in *International Conference on Humanoid Robots (Humanoids)*, Genova, Italy, 2006.

[12] D. DeMenthon, L. Davis, and D. Oberkampf, "Iterative pose estimation using coplanar points," in *International Conference on Computer Vision and Pattern Recognition (CVPR)*, 1993, pp. 626–627.

[13] 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, pp. 381–395, 1981.

[14] B. K. P. Horn, "Closed-form solution of absolute orientation using unit quaternions," *Journal of the Optical Society of America*, vol. 4, no. 4, pp. 629–642, 1987.

[15] P. Azad, "Integrating Vision Toolkit," http://ivt.sourceforge.net.

[16] "OpenCV," http://sourceforge.net/projects/opencvlibrary.