Visual Servoing for Humanoid Grasping and Manipulation Tasks

Nikolaus Vahrenkamp, Steven Wieland, Pedram Azad, David Gonzalez, Tamim Asfour and Rüdiger Dillmann
Institute for Computer Science and Engineering
University of Karlsruhe,
Haid-und-Neu-Strasse 7, 76131 Karlsruhe, Germany
Email: {vahrenkamp,wieland,azad,gonzalez,asfour,dillmann}@ira.uka.de

***Abstract*—Using visual feedback to control the movement of the end-effector is a common approach for robust execution of robot movements in real-world scenarios. Over the years several visual servoing algorithms have been developed and implemented for various types of robot hardware. In this paper, we present a hybrid approach which combines visual estimations with kinematically determined orientations to control the movement of a humanoid arm. The approach has been evaluated with the humanoid robot ARMAR III using the stereo system of the active head for perception as well as the torso and arms equipped with five finger hands for actuation. We show how a robust visual perception is used to control complex robots without any hand-eye calibration. Furthermore, the robustness of the system is improved by estimating the hand position in case of failed visual hand tracking due to lightning artifacts or occlusions. The proposed control scheme is based on the fusion of the sensor channels for visual perception, force measurement and motor encoder data. The combination of these different data sources results in a reactive, visually guided control that allows the robot ARMAR-III to execute grasping tasks in a real-world scenario.**

I. Introduction

Object manipulation with humanoid robotic systems is an essentially different problem compared to solving the same task with an industrial robotic manipulator. The main difference lies in the accuracy of the hand-eye calibration. With industrial robotic systems, this problem can be solved easily, since the inverse kinematics of an industrial robotic arm is very precise, and often a static stereo camera system is used.

With a humanoid robotic system with light-weight arms and often using wire-driven mechanics, the repeatability is significantly lower. Furthermore, and even more critical, is the problem of the usually imprecise inverse kinematics and therefore the hand-eye calibration. The reason for this is that the kinematics of the robot is formulated on the basis of CAD models of the robot. However, in practice, small translative and rotative deviations in each joint occur during manufacturing. These lead to a large accumulated error of the calibration between the camera system and the end effectors, i.e. the robot's hands. One has to keep in mind that for a 7 DoF robot arm and 3 DoF neck with even fixed eyes, already the kinematic chain between the hand and the eye consists of 10 DoF. In practice, the final absolute error of the inverse kinematics with respect to the camera system might be in the order of $> \pm$ 5 cm.

One approach to tackle this problem is to learn the hand-eye calibration on the basis of 3D object tracking where the tracking object is attached to the hand, which is observed by the robot's camera system. This special calibration process can be avoided when using visual servoing for positioning the end effector. Within this context visual servoing means that both the robot's hand and the target object are tracked by the camera system, and a specific control law defined on the measured 3D pose difference incrementally moves the hand towards the object.

For such an approach, it is usually not sufficient to compute the object pose once and then execute the grasp. The reason is that the robot's hip and head often must be involved in the servoing procedure, in order to extend the working space of the robot. Both head and hip movements change the relative pose of the object with respect to the camera coordinate system. Since, as explained, the kinematics of humanoid robotic systems are often not accurate enough to update the pose, the pose update must be computed *within* the closed-loop visual servoing controller.

In this paper we present a hybrid visual servoing approach which is applied for grasping and manipulation tasks on a humanoid robot. We show how the execution of grasping operations can be visually controlled without any hand-eye calibration. Furthermore we present several realizations of grasping and manipulation tasks to show the applicability and the robustness of the proposed algorithms.

II. Visual Servoing

Visual servoing approaches can be used to control the movements of a manipulator over a visual sensor system, which usually consists of one or more cameras. The tool center point (TCP) and/or features of a target object are tracked by the vision system and the algorithms in the control loop try to bring the TCP to a desired position and orientation [1].

Hill and Park [2] introduced the term visual servoing in 1979 to describe a visual-closed control loop. Since their work, a lot of setups for the visual perception like *fixed camera*, *multiple cameras* or *camera in hand*-approaches have been developed. Furthermore, three main control strategies have been developed over the last years which we want to introduce briefly (a more detailed introduction to visual servoing can be found in the survey from Chaumette and Hutchinson given in [3] and [4]).

A. Image-Based Visual Servoing (IBVS)

The position of the TCP in the camera image(s) is acquired by image features, and through the movements of these features a local Image-Jacobian (*Feature-Sensitivity Matrix, Interaction Matrix, B Matrix*) can be derived ([5], [6], [7],

[8]). The Jacobian depends on the camera parameters and the feature positions. By computing the pseudo-inverse of this Jacobian, the desired 2D movements of the features in the image plane can be used to calculate the movements of the joints that bring the TCP to the desired position. These approaches work without any stereo computations and thus with a single camera. Problems can arise, if the features leave the field of view and some situations like the *Chaumette Conundrum* described in [9] cannot be solved.

B. Position-Based Visual Servoing (PBVS)

Position-Based Visual Servoing approaches determine the Cartesian position and/or orientation of the TCP with respect to the target pose ([10], [11], [12], [13]). Different configurations (camera-in-hand, fixed camera, active head, mono or multi camera system) lead to varying formulations of the control scheme. The main aspect of PBVS is that a Cartesian error, determined using visual perception, is controlled to zero. Unlike the IBVS, position-based visual servo control scheme needs a module that is able to control the TCP in workspace. Furthermore, the computer vision algorithms have to be more sophisticated, in order to compute the Cartesian position and orientation of the TCP and the target object. A control scheme based on PBVS has the advantage, that the TCP position is controlled in Cartesian workspace and thus constraints coming from collision checking or path planning components can be easily included.

C. 2.5D Visual Servoing

Hybrid visual servoing approaches (2.5D Visual Servoing), first presented in [14], can be used to decouple the translational and the rotational control loop. Since the position and orientation of the TCP is controlled by two independent control loops, there are some advantages compared to IBVS and PBVS. By selecting adequate visual features defined in part in 2D (IBVS), and in part in 3D (PBVS), it is possible to generate a control scheme that always converges and avoids singularities.

III. Perception

The vision component of our system is able to recognize, localize and track a set of daily objects encountered in a kitchen environment. In our experiments we are using single-colored plastic dishes, textured everyday kitchen boxes, door handles and dish washer basket, for which we are able to track the poses with a processing rate of 15–30 Hz. Thus we are able to track moving objects and adapt our control loop to differing target positions.

A. Manipulation Objects

In [15], we have presented our approach to recognition and 6D pose estimation of textured and single-colored objects, which will be summarized briefly in the following. Textured objects are recognized and localized in 2D on the basis of local point features. As features we use a combination of the Harris corner detector [16] and the SIFT descriptor [17], with an extension allowing for scale-invariance without performing an explicit scale space analysis, as described in [18]. Recognition and 2D localization is performed on the basis of 2D-2D feature point correspondences, using a Hough transform and an iterative estimation of an affine transformation, which is refined to a full homography in the last iteration. In contrast to conventional approaches, the 6D pose is not computed on the basis of 2D-3D point correspondences, but on the basis of triangulated subpixel-accurate stereo correspondences within the estimated 2D area of the object, yielding 3D-3D point correspondences with a training view. For boxes with a planar surface, a refined pose is computed on the basis of the optimal 3D plane fit through the 3D data.

Single-colored objects cannot be recognized and localized on the basis of local features, since their only characteristic feature is their shape. Our approach for such objects combines model-based view generation for an appearance-based approach with stereo-based position estimation. Orientation information is retrieved from the matched views; an accurate 6D pose is calculated by a pose correction procedure, as presented in [18]. Exemplary results of the two integrated systems are given in Fig. 1.

Fig. 1. Exemplary results with the integrated object recognition and pose estimation systems. The computed object pose has been applied to the 3D object model and the wireframe model has been overlaid.

B. Environment Objects

In this approach, environmental elements, i.e. doors and handles were robustly recognized by means of gaussian classification using tailored characteristics feature spaces for each element type.

The components of feature vectors representing doors were obtained from the eigenvectors of the covariance matrix from extracted color-segmented regions (*blobs*) from stereo images, as well as ratios involving eigenvalues $E = \sigma_0/\sigma_1$ and the angles $A = \theta_1/\theta_2$ between the blob's diagonal axes. (see Fig. 2). Subsequently, the left-right cross match using

Fig. 2. Recognized environmental elements and pose estimation for grasping skills. The upper left image shows the recognized kitchen doors and handles. The zoomed detail on upper right image shows the approach robustness against difficult illuminations conditions. The center left and right images show the cross matching of recognized blobs using the epipolar lines and incidence criteria. The bottom left image shows the results of the dishwasher basket handle recognition phase while visual servoing is being executed, whereas the bottom right image shows the results obtained at the first target search.

size, position, orientation, distance to the epipolar line and standard disparity constraints allows powerful rejection of blob outliers.

The handle's feature vectors use previously computed blobs descriptors to extract additional characterizing components. Only those blobs which simultaneously present an elongation ratio E above a threshold τ and corner features on both ends of the main axis proceed to the following stage. Next, both the image and its Harris response function [16] are sampled along the main axis.

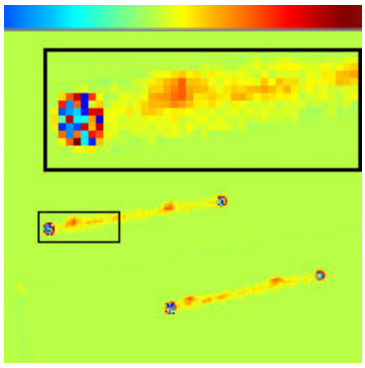

Fig. 3. The Harris response function.

This unidimensional signal contains information about the chromatic and sharpness (edges-corners) nature of the blob which is taken into account as four components, the mean and standard deviation of both color and harris domain (see Fig. 3). Once the handle has been recognized, a linear regression of edge pixels on both sides is computed in order to improve the subpixel precision of the perceived position (see Fig. 2).

Furthermore, a dish washer handle recognition and position estimation has been developed in order to illustrate and expand the aptness and suitability of our approach. The recognition begins by segmenting the image base on color as follows. For every pixel $P_i = I(x, y) \in R^3$ within a normalized RGB space a distance $D_\alpha(P_i, P_t) = 1 - \frac{P_i \cdot \widehat{P_t}}{\|P_i\|}$ to our target color P_t is computed, this implies an angular relation to the desired color, i.e. a cone in color space whose aperture angle $\alpha = \arccos(1 - D_\alpha(P_i, P_t))$ parameterizing color similarity with less sensibility to intensity. In addition, based on their pixel-connectivity, all pixels inside this chromatic cone having a minimal color-euclidean distance to $\|P_i - P_t\|$ are used to form the required blobs for the next stage. In the subsequent phase blobs characteristics are use to arrange feature vectors F as follows:

$$F := [W, S, \delta, \overline{x}, \overline{y}, \sigma_0/\sigma_1]^t, \quad (1)$$

where the amount of pixels W in the blob and the area S of the eigen-axes bounding box are used to define the third component as a density descriptor $\delta = \frac{W}{S}$ of the blob which turns to be a very powerful clue to reject false blobs because the grid where the handle is attached depicts a blob with big area but lower density. The remaining components describe the center of the blob in image coordinates and the elongation criterion base on the ratio of the eigen values. Finally, those blobs whose feature vector were not rejected by a gaussian classifier pass to the next phase where the blobs external contour is used to compute a hough transformation, here the resulting voting cell are normalized according the size of the contour. In case the blob has at least few cells above the threshold C_t then the lines will be found on the lower side of the blob which later are use to compute the end points of the handle, see Fig. 2. The three dimensional position of the handle is calculated using similar restrictions (left-right cross check, distance to epipolar side, etc) as before.

C. Hand Tracking

To estimate the position of the TCP, we use an artificial marker, since the robust tracking of human-like 5-finger hand is a challenging problem which is not scope of this paper. To mark the hand, a red sphere is used, which is mounted on the wrist of the humanoid arm. This position avoids self-covering and allows a good visibility in most configurations of the arm. With this artificial marker we are able to use different versions of the hand, just by changing some parameters, e.g. the offsets from the marker to the center of the hand or to the fingertips.

This marker can only be used to retrieve a position relative to the wrist since there is no possibility of getting an orientation of a uniform sphere. To retrieve an approximated orientation of the hand, the direct kinematic of the robot arm is used. The orientational component of the hand pose will not be exact, since the arm is just roughly calibrated, but as our experiments show, these errors are admissible for grasping and manipulation tasks. The positioning errors, in contrast, have much more influence on successful execution of manipulation tasks which is shown in section VII. The tracking of the marker is done in real-time (30 Hz). Therefore, the positioning of the hand always operates on the correct data.

D. Force Measurement

To improve the robustness and to generalize the approach, the perception is extended by force and torque values that are measured at the wrist with a six-dimensional force/torque sensor. This additional sensor channel improves the robustness, since the force feedback provides the servoing loop with information about contact events and allows the system to react on critical situations like collisions.

IV. Execution on ARMAR-III

A. Software Framework

All experiments are implemented on the humanoid robot Armar-III ([19], [20]). The software architecture of the robot provides mechanisms, which allow integrating different perception, control and planning modules as well as an access to the robot sensors and actors. In this architecture *skills* are implemented capabilities of the robot, which can be regarded as atomic (e.g. *MovePlatform*, *SearchForObject*, *HandOver*, *CloseDoor*, *LookTo*). Tasks are combinations of several skills for a specific purpose (e.g. the task *BringObject* parametrizes and executes the skills *SearchForObject*, *MovePlatform*, *GraspObject* and *HandOver*). A brief overview of the processing framework can be seen in Fig. 4.

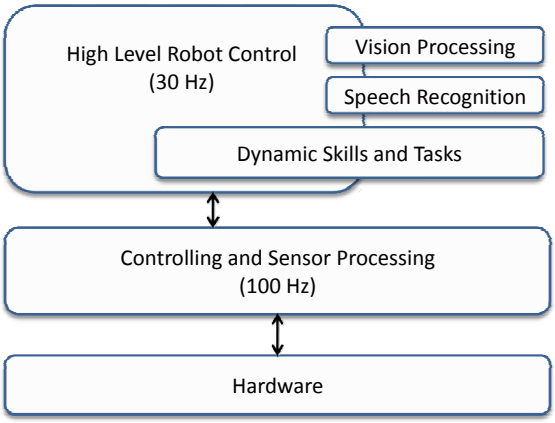

Fig. 4. The Control Framework of ARMAR-III.

The high-level robot control, which runs with 30 Hz, provides an interface where skills and tasks can be loaded and parametrized on runtime. For the experiments, the two skills *GraspObject* and *GraspHandle* where included in the skill library. The vision and speech components run asynchronous with 10-30 Hz depending on the requested operations.

B. Arms and Hip

To execute grasping tasks on the humanoid robot ARMAR III, the seven degrees of freedom (DoF) of one arm are used (all grasping tasks are implemented for the left and the right arm). In order to increase the area of operation, the hip yaw joint is included for all arm movements. These eight joints of the robot can be addressed via a velocity controller which allows smooth movements of the robot.

C. Hands

The modular mechatronics concept of ARMAR-III allows mounting of different versions of grippers or hands. We performed our experiments with two different types of the humanoid 5-finger hand. Initially, we used the hand described in [21] (see Fig. 8). During our experiments an improved version of the hand was build, which we used for the experiments described in section VII B and C (see Fig. 10 and 11). The hand is controlled by predefined shapes, which is sufficient for the grasping tasks described in this paper. We use five different preshapes: *open*, *close all*, *close fingertips*, *close fingers* and *close thumb*.

D. Head

To track the marker located on the wrist and to receive the position of the target object, the active head of the robot with seven DoF is used. To enable stereo localization the three DoF of the eyes are fixed and the inverse kinematics for the neck is used to focus 3D coordinates in workspace. The Cartesian coordinates of the marker and the target are computed in the camera coordinate system and then transformed to world coordinates.

V. Control

A. Reference Frames

The reference frames used for visual servoing can be seen in Fig. 5. The position and orientation (pose) of the manipulation object leads to a target pose which is used in the control loop. The wrist pose is a combination of the visually received wrist position and the kinematically computed orientation of the hand. By adding a hand-specific offset value, the current hand pose is calculated.

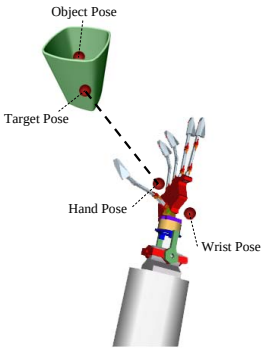

Fig. 5. The different poses.

B. Control Scheme

The control scheme for a grasping task is depicted in Fig. 6. The input values for the control algorithm consists of the current robot configuration $\boldsymbol{q}$, obtained from joint sensor reading, and a difference vector $\boldsymbol{\Delta x} = \boldsymbol{x}_{target} - \boldsymbol{x}_{hand}$, describing the difference between hand and target position. The current joint configuration $\boldsymbol{q}$ is used to build up the Jacobi matrix $J(\boldsymbol{q})$ and its pseudoinverse $J^{+}(\boldsymbol{q})$. The desired joint velocities $\dot{\boldsymbol{\Theta}}$ are computed as described in Eq. 2, where α is a gain factor controlling the speed of the servoing.

$$\dot{\boldsymbol{\Theta}} = \alpha J^{+}(\boldsymbol{q})\boldsymbol{\Delta x} \tag{2}$$

To ensure that the marker of the hand is always visible, the head is adjusted using inverse kinematics methods to hold the hand in the field of view of the stereo camera system. The perception system always searches for the target object, and updates the absolute position if the object is found. Once the

hand is in the vicinity of the target, both objects (target object and marker) can be recognized and the servoing will become more robust, since the positioning values used for servoing are determined with the same sensor channel and the same algorithms.

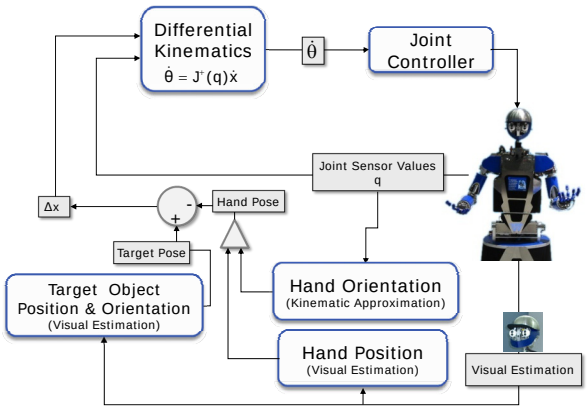

Fig. 6. The Control Loop.

VI. IMPROVING ROBUSTNESS

A. Visual Limitations

The visual servoing strongly relies on the information that is computed by the visual perception component. This component may fail due to different reasons (e.g. reflections, blurred images or sub-optimal parameters, features are outside one or both camera images). To ensure a robust execution, the visual servoing control loop stores the last acquired poses of the hand and the target. The target position will remain constant until the perception component informs the servoing loop about a newly perceived object position. The hand tracking is more fragile since the hand is moving and the pose does change continuously. To handle situations in which the hand tracking failed, the difference $\boldsymbol{\delta}_{HandPose}$ between the kinematically calculated and the visual determined hand positions are stored in each control loop (see Eq. 3).

$$\boldsymbol{\delta}^{t}_{HandPose} = \boldsymbol{x}^{t}_{vision} - \boldsymbol{x}^{t}_{kinematic} \tag{3}$$

In the case of failed visual perception these values are used to compute an approximated pose $\boldsymbol{x}^{t+1}_{hand}$ of the hand, which is used in the control loop (see Eq. 4).

$$\boldsymbol{x}^{t+1}_{hand} = \boldsymbol{x}^{t+1}_{kinematic} + \boldsymbol{\delta}^{t}_{HandPose} \tag{4}$$

Since the offset between the kinematically and the visually determined hand position is only valid in the vicinity of the current hand pose and the approximation gets worse when moving too far, the speed is reduced and if the hand is not recognized for a specific amount of time, the servoing loop is aborted to avoid unpredictable or even dangerous situations. An example for the calculated offsets between kinematically and visually determined hand positions are shown in Fig. 7.

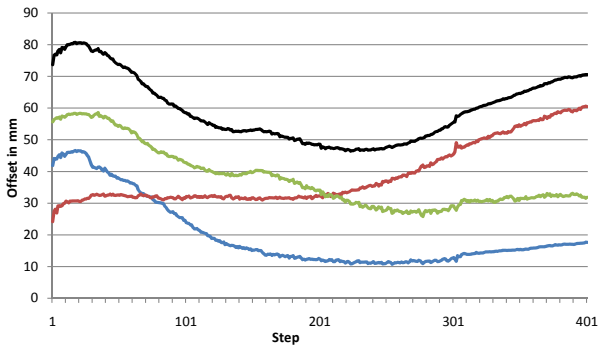

Fig. 7. Difference between the kinematically and the visually determined hand positions. The absolute offsets are shown in blue (x), red (y) and green (z), the total offset is plotted in black.

B. Reacting on Forces

As mentioned before, the 6D force/torque sensor values are used to adapt the control strategy to unknown or undiscovered obstacles. If the measured force exceeds a contact threshold, we have an indication for a contact/collision with an obstacle. A collision should never occur in an ideal environment, but in a real-world scenario with inaccuracies, the system should be able to react on these situations. An effective way of reacting on undiscovered obstacles or inaccurate measurements, can be achieved by combining the visual servoing control and a zero force control loop, which has a higher priority. The zero force control loop tries to bring all acting forces to zero, which means that obstacles (e.g. top of a table) will suspend the visual controlled servoing loop, when they are hit with the hand.

VII. EXPERIMENTS

In this section we show how the proposed control scheme is working on the robot ARMAR-III. Different grasping tasks have been implemented to enable the robot to operate in an every day kitchen scenario. No external sensors are needed and the robot operates completely autonomous.

A. Grasping a cup

In this experiment, ARMAR-III is standing in front of a uniform colored cup which is known and recognizable by the system. In the first step, the robot searches a cup and if one is found, the Cartesian position is transformed to world coordinates and the target pose is calculated. The servoing loop starts to grasp the cup with the right arm. If the Cartesian distance between hand and target pose falls below a threshold (5 mm), the hand is closed and the grasped cup is raised. When all operations are finished, the skill reports a successful state to the framework and further operations can be triggered. Fig. 8 shows the execution of the *VisualGrasp*-skill.

The Cartesian servoing distances between the hand and the target are shown in Fig. 9. The distances are determined every servoing step (33 ms) and transformed to the coordinate

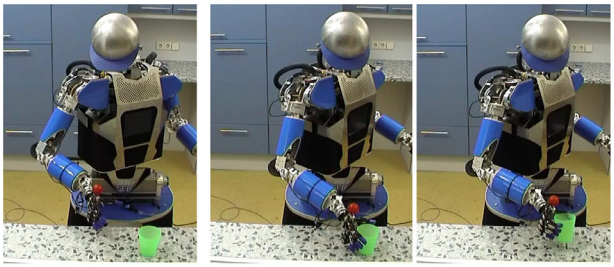

Fig. 8. Armar-III grasping a cup.

system of the hand. The figure shows the distances in x (blue), y (red), z (green) coordinates and the total distance in black. As expected, the values decrease over the time and the servoing is stopped when the total distance falls below a threshold (5 mm). The target position is determined in advance, since at the start of the servoing only the marker is in the field of view. At step 180, the target comes into the field of view again and the position is recognized and updated by the vision thread. Because of the kinematic chain between camera and hand (consisting of ten DoF), the calculated target position slightly differs from the initial position, although the real position did not change. Here, the advantages of visual servoing approaches are shown clearly. The pose of the hand is controlled relatively to a target object and since the target and the hand are tracked with the same algorithms, their Cartesian relation is determined with high accuracy. The differential kinematic algorithms are fault-tolerant as long as the errors are locally bounded and the Jacobian remains regular.

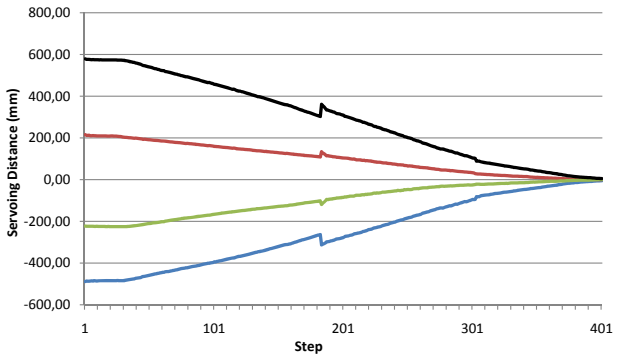

Fig. 9. The Cartesian distances (hand to target), shown in blue (x), red (y), green (z). The total distance is shown in black.

B. Grasping a box

With the proposed control scheme, ARMAR-III is also able to grasp rectangular boxes of varying sizes. The dimensions of the boxes are known in advance, so that the target poses can be derived from these values (see Fig. 5). The control scheme has to be modified slightly by adding a constraint for an aligned reaching of the hand. The alignment is achieved by adjusting the orientation of the hand to the orientation of the object. Furthermore, the flexibility was increased by adding an intermediate reaching point next to the object. The hand is moved to this preposition before it is moved towards the final target position. The pictures in Fig. 10 are captured with the stereo cameras of ARMAR-III while the robot is executing the grasping task.

Fig. 10. ARMAR-III is grasping a mashed potatoes box, captured by the robot's camera system.

C. Grasping a handle

The proposed visual servoing scheme can be used to enable ARMAR-III grasping handles of kitchen doors as well. In this case the Cartesian position of the handle is used to determine a target position and a pre-position which is below and in front of the handle. Instead of using a *Hand Pose* in the center of the TCP (see Fig. 5), the *GraspHandle*-skill uses a different offset which defines the distance from the marker to the fingertips. Hence the position of the fingertips is controlled via the visual servoing control loop. The grasping of the handle is performed in four steps:

- The TCP is moved to a pre-position, in front of the handle.
- The TCP is moved to the final grasp position.
- If the grasp position is reached, or if the force sensor values indicate a contact, the hand configuration is set to the preshape *close fingertips* and an impedance controller is started which aligns the hand to the handle.
- Finally the thumb is closed to finish the grasp.

In Fig. 11 some intermediate stages of the execution of the *GraspHandle*-skill are shown.

VIII. CONCLUSIONS AND FUTURE WORK

A. Conclusions

We showed how it is possible to enable the humanoid robot ARMAR-III to execute grasping actions in a robust manner. The presented hybrid visual servoing tasks can be used to avoid hand-eye calibration routines and to improve

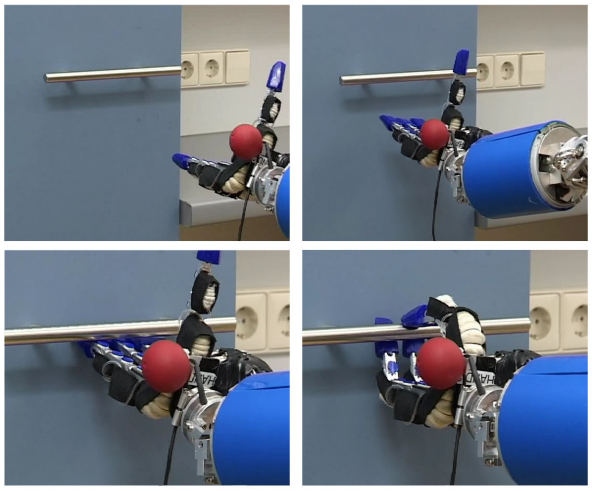

Fig. 11. Grasping a fridge handle.

the manipulation skills of the robot. The robustness of the approach was improved by two enhancements:

- Observing the offset between kinematically determined and visually approximated marker positions, to improve the hand tracking.
- The visual guided movements have been enhanced by a reactive component by including the data coming from the 6D force/torque sensor into the visual servoing control structure. Therefore, safety of the robot, environment and humans is increased.

Furthermore, the presented framework allows to support different manipulation targets and hand types by parameterizing the skill structure. The approach was successfully implemented for different grasping tasks which enable the robot to execute more complex activities.

B. Future Work

The ability of grasping varying objects in a robust manner is the basis for more extensive scenarios. The grasping skills can be used for higher level tasks which set the parameters on-the-fly and thus allow the implementation of more complex tasks. Future work will address the post-grasping behavior like the check if a grasping action was successful. These checks can be used to adopt parameters on-the-fly or to report that the parametrization should be optimized (e.g. the light conditions changed). Furthermore, we want to investigate more complex grasping actions like two arm manipulations or more sophisticated grasping types.

IX. ACKNOWLEDGMENTS

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

REFERENCES

[1] S. Hutchinson, G. Hager, and P. Corke, "A tutorial on visual servo control," *IEEE Trans. on Robotics and Automation*, vol. 12, no. 5, pp. 651–670, Oct. 1996.

[2] J. Hill and W. Park, "Real time control of a robot with a mobile camera," in *The 9th International Symposium on Industrial Robots*, 1979, pp. 233–246.

[3] F.Chaumette and S.Hutchinson, "Visual servo control, part I: Basic approaches," *IEEE Robotics and Automation Magazine*, vol. 13, no. 4, pp. 82–90, dec 2006.

[4] F. Chaumette and S. Hutchinson, "Visual servo control, part II: Advanced approaches," *IEEE Robotics and Automation Magazine*, vol. 14, no. 1, pp. 109–118, March 2007.

[5] L. Weiss, A. C. Sanderson, and C. P. Neuman, "Dynamic sensor-based control of robots with visual feedback," *IEEE Journal on Robotics and Automation*, vol. RA-3, no. 5, October 1987.

[6] K. Hosoda and M. Asada, "Versatile visual servoing without knowledge of true jacobian," in *Intelligent Robots and Systems '94. 'Advanced Robotic Systems and the Real World', IROS '94.*, vol. 1, September 1994, pp. 186–193.

[7] F. Chaumette, "Visual servoing using image features defined upon geometrical primitives," in *33rd IEEE Conf. on Decision and Control*, vol. 4, Orlando, Florida, December 1994, pp. 3782–3787.

[8] K. HoWon, C. JaeSeung, and K. InSo, "A novel image-based control-law for the visual servoing system under large pose error," in *Intelligent Robots and Systems, 2000. (IROS 2000).*, vol. 1, 2000, pp. 263–268.

[9] F. Chaumette, "Potential problems of stability and convergence in image-based and position-based visual servoing," in *The Confluence of Vision and Control*, D. Kriegman, G. . Hager, and A. Morse, Eds. LNCIS Series, No 237, Springer-Verlag, 1998, pp. 66–78.

[10] G. b. W.J. Wilson, C.C. Williams Hulls, "Relative end-effector control using cartesian position based visual servoing." *IEEE Transactions on Robotics and Automation.*, vol. 12, pp. 684–696, October 1996.

[11] G. Bell and W. Wilson, "Coordinated controller design for position based robot visual servoing in cartesian coordinates," in *IEEE International Conference on Robotics and Automation*, vol. 2, 1996, pp. 1650–1655.

[12] P. Martinet and J. Gallice, "Position based visual servoing using a nonlinear approach," in *IEEE/RSJ Int. Conf. On Intelligent Robots and Systems*, 1999, pp. 531–536.

[13] B. Thuilot, P. Martinet, L. Cordesses, and J. Gallice, "Position based visual servoing: Keeping the object in the field of vision," in *ICRA*, 2002, pp. 1624–1629.

[14] E. Malis, F. Chaumette, and S. Boudet, "2 1/2 d visual servoing," *IEEE Transaction on Robotics and Automation*, vol. 15, no. 2, pp. 234–246, April 1999.

[15] P. Azad, T. Asfour, and R. Dillmann, "Stereo-based 6D Object Localization for Grasping with Humanoid Robot Systems," in *International Conference on Intelligent Robots and Systems (IROS)*, San Diego, USA, 2007.

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

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

[18] P. Azad, "submitted thesis: Visual Perception for Manipulation and Imitation in Humanoid Robots," Ph.D. dissertation, University of Karlsruhe, Karlsruhe Germany, 2008.

[19] T. Asfour, K. Regenstein, P. Azad, J. Schröder, A. Bierbaum, N. Vahrenkamp, and R. Dillmann, "Armar-III: An integrated humanoid platform for sensory-motor control." in *IEEE-RAS International Conference on Humanoid Robots (Humanoids)*, December 2006.

[20] T. Asfour, P. Azad, N. Vahrenkamp, K. Regenstein, A. Bierbaum, K. Welke, J. Schröder, and R. Dillmann, "Toward humanoid manipulation in human-centred environments," *Robot. Auton. Syst.*, vol. 56, no. 1, pp. 54–65, 2008.

[21] A. Kargov, T. Asfour, C. Pylatiuk, R. Oberle, H. Klosek, S. Schulz, K. Regenstein, G. Bretthauer, and R. Dillmann, "Development of an anthropomorphic hand for a mobile assistive robot," *Rehabilitation Robotics, 2005. ICORR 2005. 9th International Conference on*, pp. 182–186, 28 June-1 July 2005.