INSTITUT FÜR INFORMATIK

UND PRAKTISCHE MATHEMATIK

Manipulator and Head Servoing for Tool Handling and Object Inspection

Josef Pauli

Bericht Nr. 9813
November 1998

CHRISTIAN-ALBRECHTS-UNIVERSITÄT

KIEL

Institut für Informatik und Praktische Mathematik der
Christian-Albrechts-Universität zu Kiel
Olshausenstr. 40
D – 24098 Kiel

Manipulator and Head Servoing for Tool Handling and Object Inspection

Josef Pauli

Bericht Nr. 9813
November 1998

e-mail: jpa@ks.informatik.uni-kiel.de

Abstract

We define *image-based robot servoing* as a continual process of perception-action cycles for the task of tool handling or object inspection. Image analysis techniques and control rules are presented as the basic components of a behaviour-based robot system. Our robot hardware consists of a bisight head on a movable platform with several degrees-of-freedom, an articulation manipulator on a stationary platform with a parallel jaw gripper including a hand-mounted single camera, and finally a rotary table. The approaching, assembling, and continual handling of the gripper tool is illustrated. For the purpose of object inspection the head-camera system or the manipulator (carrying the object) are controlled to reach a desired size, resolution and orientation of the depicted object. Manipulator and head servoing is also used for self-calibration, i.e., determining the optical axes, the fields of visibility and the location of the head in the manipulator coordinate system. Finally, the significant role of offline visual demonstration is exemplified for specifying visual goal situations in robot servoing.

1 Introduction

This work gives a review of the wide application spectrum of *image-based robot servoing (IBRS)* using a multi-component robot system in a realistic scene. We are convinced that the usefulness is far from being sufficiently realized which is due to several reasons. First, the various *degrees-of-freedom (DOF)* of a robot head, e.g., pan, tilt, vergence, focus, focal length, and aperture of the head-cameras, must be controlled in cooperation in order to exploit their complementary strengths [1]. Our work contributes in several aspects to this problem. Second, nearly all contributions to robotic visual servoing describe systems consisting of just one robot, e.g., exclusively a robot manipulator or a robot head. Instead of that, we present applications of image-based robot servoing for a *multi-component robot system* consisting exemplary of a movable robot head, a stationary manipulator, and a rotary table. Third, for solving robot tasks in realism, perhaps a priori models of objects and their arrangement are hardly available and consequently model-free exploring robots are required. The book edited by [2] gives the state of the art of exploratory vision and includes a chapter on robots that explore. Image-based robot servoing must play a significant role especially in model-free exploration of scenes. Our work proposes *visual demonstration* as a means for supporting visual exploration.

In the following we describe shortly important contributions of IBRS relevant for our work. The book edited by [3] gives an overview of various approaches of automatic control of mechanical systems using visual sensory feedback. To mention just the introductory work of [4] there two approaches of visual servoing are proposed, the *position-based* and the *feature-based*. In position-based control features are extracted from the image and used in conjunction with a geometric model of the target to determine the pose of the target with respect to the camera. In image-based servoing the last step is omitted, and servoing

is done on the basis of image features directly. In our applications geometric object models are hardly available and accordingly the visual feedback controller must be feature-based. A further classification criterion is whether a current robot state (e.g., position and orientation of a gripper) is used as additional feedback information for successive control. The *dynamic look-and-move approaches* use it, but the *servo approaches* only rely on visual feedback. In our system we can request the manipulator or head state during the movement and can also alter this movement dynamically. Furthermore the images can be taken and analysed parallel with the control. Therefore our control scheme is a *feature-based dynamic look-and-move approach.*

This approach is also used by [5] who describe a system that positions a robot manipulator using visual information from two stationary cameras. The end-effector and the visual features defining the goal position are simultaneously tracked using a PI controller. We adopt the idea of using Jacobians for describing the $3D$–$2D$ relation but taking projection matrices of a poorly calibrated head-camera–manipulatior relation into account instead of explicit camera parameters.

Similary the system of [6] tracks a moving object with a single camera mounted on a manipulator. A visual feedback controller is used which is based on an inverse Jacobian matrix for transforming changes from image coordinates to robot joint angles. The work is interesting to us because the role of a *teach-by-showing method* is mentioned. Offline the user teaches the robot desired motion commands and generates reference vision-feature data. In the online playback mode the system executes the motion commands and controls the robot until the extracted feature data correspond to the reference data.

The authors [7] present an algorithm for robotic camera servoing around a static target object with the purpose of reaching a certain relation to the object. This is done by moving the camera (mounted on a manipulator) such that the image projections of certain feature points of the object reach some desired image positions. In our work a similar problem occurs in controlling a manipulator to carry an object towards the head-camera such that a desired size, resolution and orientation of the depicted object is reached.

The system of [8] reconstructs the $3D$ structure of geometric primitives like cylinders from controlled motion of a single camera. The intention is to obtain a high accuracy by focusing at the object and generating optimal camera motions. An optimal camera movement for reconstructing the cyclinder would be a cycle around it. This camera trajectory is acquired via visual servoing around a cylinder by keeping the object depiction in vertical orientation in the image center. The work is related to our approach of using IBRS for determining the optical axis and the field of visibility of a head-camera.

This first chapter mentioned important contributions related to our work. In the second chapter IBRS is discussed in simple general terms. The third chapter presents an approach of self-calibration of the head-camera–manipulator relation and uses *image-based manipulator servoing (IBMS)* to determine the optical axis and field of visibility of the head-camera. In the fourth chapter image-based manipulator servoing is applied to tool handling (first principal goal). The fifth chapter combines image-based manipulator servoing with *image-based head servoing (IBHS)* for object inspection (second principal goal). A summary in chapter six concludes the work.

2 Definition of image-based robot servoing

Image-based robot servoing is the gradual actuator movement of a robot system continually controlled with visual sensory feedback.
This definition can best be understood in its wide range by first introducing an exemplary camera-based robot system (see Figure 1).

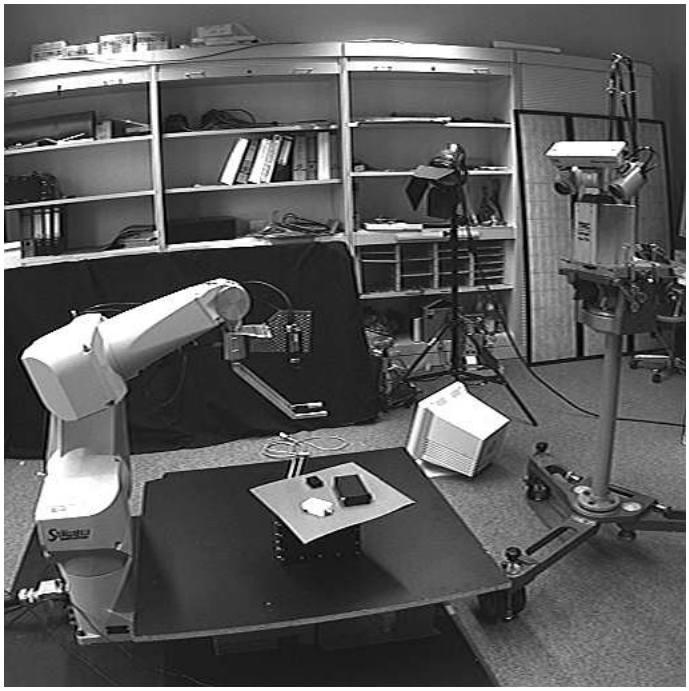

FIG. 1: Exemplary architecture of a camera-based robot system.

A stationary manipulator is shown with six rotational joints for positioning its hand and one linear joint for opening/closing parallel jaw fingers. Furthermore a single camera is fastened at the manipulator hand with the viewing direction straight through the fingers. Beside the manipulator a movable platform for a stereo camera system is shown to observe the scene at variable viewing points. The camera system belongs to a robot head with pan, tilt, vergence DOF, and zooming/focusing facilities. In between the manipulator and the robot head a rotary table is located which can turn objects if desired. By using the inverse manipulator kinematics a goal position (in $3D$ coordinates X, Y, Z) and goal orientation (in Euler angles yaw, pitch, roll) of the manipulator hand can be transformed into six joint angles [9]. The working space of the hand, i.e., arbitrary orientation in a certain space, is a cube of about $400mm$ sidelength. The movement of the platform for the stereo camera is specified in a local attached $2D$ coordinate system. The pan and tilt DOF of the robot head are from -90 to $+90$ degrees each. The vergence DOF for each camera is from -45 to $+45$ degrees. The focal length of the camera can vary between 11 and $69mm$. The turning angle of the rotary table is specified between 0 and 360 degrees. Generally, a robot system to be controlled can be characterized by a *fixed state vector* S_f which is inherent constant in the system, and by a *variable state vector* $S_v(t)$ which can

be changed through a *vector of control signals* $C(t)$ at time t. For example, the fixed state vector of a robot manipulator contains the *Denavit-Hartenberg parameters length, twist, offset* for each link which are constant for rotating joints [9]. On the basis of the variable state vector $S_v(t)$ and control vector $C(t)$ the transition function f determines the next state vector $S_v(t+1)$:

$$S_v(t+1) := f(C(t), S_v(t)) \tag{1}$$

For example, if the vectors $C(t)$ and $S_v(t)$ are of equal dimension with the components corresponding pairwise, and the function f is the vector addition, then $C(t)$ serves as an increment vector for $S_v(t)$. The vector $S_v(t)$ could be the 6-dimensional state of position and orientation of the robot hand, and $C(t)^T := (\triangle X, \triangle Y, \triangle Z, 0, 0, 0)$, then after the movement the state vector $S_v(t+1)$ describes a new position of the hand preserving the orientation. Both state and control vector are specified in the manipulator coordinate system.

In each state of the robot system the cameras take images from the scene. This is symbolized by a function g which produces a *current measurement vector* $Q(t)$ at time t (in coordinate systems of the cameras).

$$Q(t) := g(S_v(t), S_f) \tag{2}$$

Given the current measurement vector $Q(t)$, the current state vector $S_v(t)$, and a *desired measurement vector* Q^*, the controller generates a *control vector* $C(t)$.

$$C(t) := h(Q^*, Q(t), S_v(t)) \tag{3}$$

The *control rule* h describes the relation between changes in different coordinate systems, e.g., $Q(t)$ in the head-camera and $C(t)$ in the manipulator coordinate system. The control vector $C(t)$ is used to update the state vector into $S_v(t+1)$, and then a new measurement vector $Q(t+1)$ is acquired that should be more closer to Q^* than $Q(t)$. In the case that the desired situation is already reached after the first actuator movement, the *one-step controller* can be thought of as an exact *inverse model* of the robot system. Unfortunately, in realistic control environments only approximations for the inverse model are available. In consequence of that, it is necessary to run through cycles of gradual actuator movement and continual visual feedback to successive reach the desired situation. Frequently, the control rule h is a *linear approximation* of the unknown inverse model, i.e., the parameters $Q^*, Q(t), S_v(t)$ are linear combined to produce $C(t)$. Some articles in [3] also describe nonlinear, fuzzy logic, and neural network control schemes.

IBRS is organized into an offline-phase and an online-phase. Offline we specify the approximate *head-camera–manipulator relation* of coordinate systems and define the control rule thereof. Online the control rule is applied during which the system recognizes a current situation and compares it with a certain goal situation. In case of deviation an actuator is moving to bring the new situation closer to the goal situation. This cycle is repeated until a certain threshold criterion is reached.

3 Calibration of head-camera–manipulator relation

The relation between the coordinate systems of the head-camera and the manipulator is acquired roughly by taking the agility of the manipulator into account and tracking

systematic gripper movements. This is the basis for nearly all applications of image-based manipulator servoing presented in this work. For example, in this chapter IBMS will be applied to determine the optical axis and the field of visibility of a head-camera. These informations are extraordinary important in the active vision paradigm (see chapter five). Finally, we present a strategy for locating the head-camera system in the manipulator coordinate system using once again IBMS.

3.1 Approximate head-camera–manipulator relation

The approach computes perspective projection matrices describing the head-camera–manipulator mapping. In general this estimated relation is poor because the camera platform is movable. The head-camera system is put up in a position and orientation that the common field of visibility of the two cameras contains a large enough subspace of $3D$ working space of the manipulator. A certain reference point of the gripper is defined as the tool center point (i.e., the gripper tip) for which the $3D$ coordinates in the manipulator coordinate system are known. From this gripper tip the $2D$ coordinates must be determined in the stereo images.

The gripper systematic moves in the working space, stops on equidistant places, and from the gripper tip the $3D$ coordinates and the twice $2D$ coordinates are recorded. Based on the resulting samples the head-camera–manipulator mapping can be approximated directly without putting a calibration object in between. The number of samples for this mapping is variable due to steerable distances between the stopping places. Furthermore calibration points both on the surface and within the working space are considered. The only serious problem is to extract the gripper tip from the stereo images as accurate as possible.

First, by correlation matching which is based on the sum of squared distances the gripper tip is located roughly (see Figure 2). As the manipulator systematic moves we can predict the location of gripper tip in the following image and thus restrict the search area. Second, to verify the place of maximum correlation and locate the position of the reference point exactly we additional extract geometric attributes of the gripper appearance. The gripper reference point is defined in the image as the intersection point between the middle straight line and the end straight line of the parallel jaw gripper (see Figure 3). Hough transformation [10] can be used for extracting straight lines of the finger contours restricted on the gripper tip region. Taking the polar form for representing lines the Hough image can be defined such that the horizontal axis is for the radial distance and the vertical axis is for the orientation of a line. According to this agreement an image line is Hough transformed such that a peak occurs in the Hough image and its position just specifies the line parameters in the grey level image. For example the two pairs of long lines for the two fingers occur in the Hough image as four peaks which are nearly horizontal due to similar line orientations. Therefore according to the specific pattern of four peaks the long finger lines are extracted and from those the middle straight line. Furthermore the Hough image can be used for constructing the end straight line.

FIG. 2: Gripper, gripper tip region, correlation image.

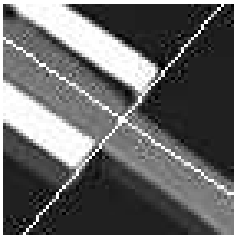

FIG. 3: Middle straight and end straight line.

For each camera, $k \in \{1,2\}$, a perspective projection matrix is computed by using corresponding $3D$ points and $2D$ points (respective for each of the two head-cameras).

$$M_k := \begin{pmatrix} M_k^1 \\ M_k^2 \\ M_k^3 \end{pmatrix} \quad ; \quad with \quad \begin{array}{l} M_k^1 := (m_k^{11}, m_k^{12}, m_k^{13}, m_k^{14}) \\ M_k^2 := (m_k^{21}, m_k^{22}, m_k^{23}, m_k^{24}) \\ M_k^3 := (m_k^{31}, m_k^{32}, m_k^{33}, m_k^{34}) \end{array} \tag{4}$$

The scalar parameters m_k^{ij} are determined with linear methods according to ([11], pp. 55-58). They represent a combination of extrinsic and intrinsic camera parameters which we leave implicit. The usage of the projection matrix is specified within the following context. Given a point in homogeneous manipulator coordinates $P := (X, Y, Z, 1)^T$ the position in homogeneous image coordinates $p_k := (x_k, y_k, 1)^T$ can be obtained by solving

$$p_k := \frac{1}{\delta_k} \cdot M_k \cdot P \quad ; \quad with \quad \delta_k := M_k^3 \cdot P \tag{5}$$

The equations (4) and (5) are easily derived by taking the perspective projection of a pinhole camera into account. Next we describe how a certain change in manipulator coordinates affects a change in image coordinates. The Jacobian J_k for the mapping in equation (5) is used.

$$J_k(P) := \begin{pmatrix} \frac{\partial x_k}{\partial X}(P) & \frac{\partial x_k}{\partial Y}(P) & \frac{\partial x_k}{\partial Z}(P) \\ \frac{\partial y_k}{\partial X}(P) & \frac{\partial y_k}{\partial Y}(P) & \frac{\partial y_k}{\partial Z}(P) \end{pmatrix} = \begin{pmatrix} \frac{m_k^{11} \cdot M_k^3 \cdot P - m_k^{31} \cdot M_k^1 \cdot P}{(M_k^3 \cdot P) \cdot (M_k^3 \cdot P)} & \cdots \\ \vdots & \ddots \end{pmatrix} \tag{6}$$

3.2 Manipulator servoing for determining the optical axis

IBMS can be applied for determining the optical axis of a head-camera. During the procedure the robot head is motionless and the manipulator gripper will be servoed to two distinct points located on the optical axis. It is assumed that all points located on this axis are projected to the image center approximately. Accordingly, we must servo the gripper such that the two-dimensional projection of the gripper tip approaches the image center. In the goal situation the $3D$ position of the gripper tip (which is the known tool center point in $(\vec{X}, \vec{Y}, \vec{Z})$ manipulator coordinate system) is taken as a point on the optical

axis. For simplifying the servoing task two planes are specified which are parallel to the $(\vec{Y}, \vec{Z})$ plane with constant offsets X^1 and X^2 on the $\vec{X}$-axis and the movement of the gripper is restricted just on these planes (see Figure 4). Generally, in IBRS the deviation between a current situation and a goal situation is specified in image coordinates. To transform a desired change from image coordinates back to manipulator coordinates the inverse or pseudo inverse of the Jacobian of the projection matrices is computed. In this application the Jacobian $J_k, k \in \{1, 2\}$, in equation (6) for the mapping in equation (5) can be restricted to the second and third columns because the coordinates on the $\vec{X}$-axis are fixed. Accordingly, the inverse of the quadratic Jacobian matrix is computed, $J^{\dagger}(P) := J_k(P)^{-1}$.

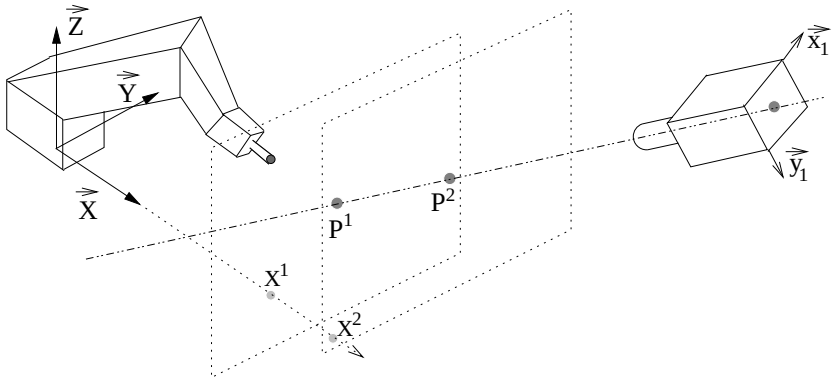

FIG. 4: Determining the optical axis of a head-camera.

The current measurement vector $Q(t)$ is defined as the $2D$ image location of the gripper tip and the desired measurement vector Q^* as the image center point. The variable state vector $S_v(t)$ consists of the two variable coordinates of the tool center point in the selected plane (X^1, Y, Z) or (X^2, Y, Z). Then the control scheme is as follows

$$C(t) := \begin{cases} s \cdot J^{\dagger}(S_v(t)) \cdot (Q^* - Q(t)) & : \quad |Q^* - Q(t)| > thresh \\ 0 & : \quad else \end{cases} \tag{7}$$

with the servoing factor s to control the velocity of approaching the optical axis. The gripper position is changed by a non-null vector $C(t)$ if desired and current positions in the image deviate more than a threshold $thresh$. Actually equation (7) defines a proportional control law (P-controller), meaning that the change is proportional to the deviation between the desired and the current position.[1] First the gripper tip is servoed to the intersection point P^1 of the unknown optical axis with the plane (X^1, Y, Z), and second to the intersection point P^2 with plane (X^2, Y, Z). The two resulting positions of the tool center point specify the axis which is represented in the manipulator system. Figure 5 shows for manipulator servoing on one plane the succession of extracted gripper positions in the image with the final point at the image center (servoing factor $s := 0.3$).

[1]Alternatively the P-controller can be combined with an integral and a derivative control law to construct a PID-controller. However the P-controller is good enough for this simple control task.

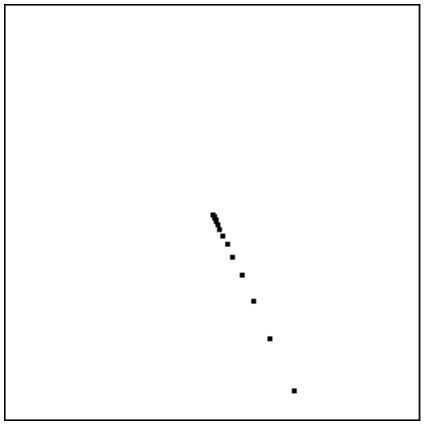

FIG. 5: Course of detected gripper.

3.3 Determining the field of visibility and sharpness

IBMS is a means for constructing the field of visibility and sharpness of a head-camera which can be approximated as a truncated pyramid with top and bottom rectangles normal to the optical axis (see Figure 6). The top rectangle is small and near to the camera, the bottom rectangle is larger and at a greater distance from the camera.

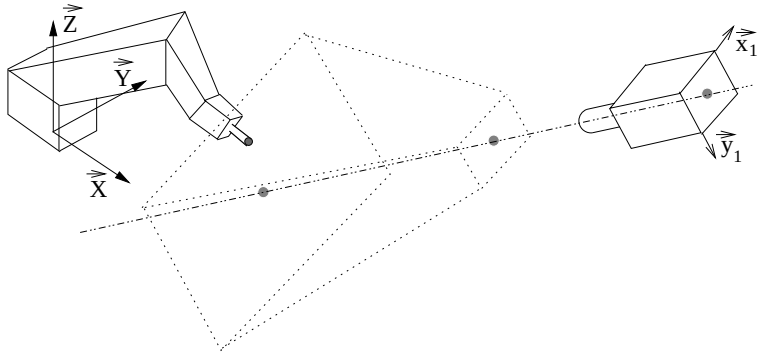

FIG. 6: Pyramid field of visibility and sharpness.

For determining the range of sharp focus the gripper tip is servoed along the optical axis and the sharpness of the depicted gripper is evaluated. As the gripper tip is located in the image center we extract a small rectangular patch surrounding the center and compute the sharpness in it. For example, a measure of sharpness is obtained by computing the magnitudes of grey level gradients and taking the mean of 10 percent of maximum responses. Figure 7 shows these measurements for a head-camera with focal length $69mm$. The gripper is starting at a distance of $1030mm$ to the camera and approaches to $610mm$ with stopping places every $30mm$ (this gives 15 measurements). We specify a threshold value Q^* for the measurements $Q(t)$ for defining the acceptable level of sharpness. In Figure 7 four measurements surpass the threshold, numbers $9, 10, 11, 12$, which means

that the depth of sharpness is about $90mm$, reaching from $700mm$ to $790mm$ distances from the camera. The control procedure consists of two stages, first reaching the sharp field, and second moving through it.

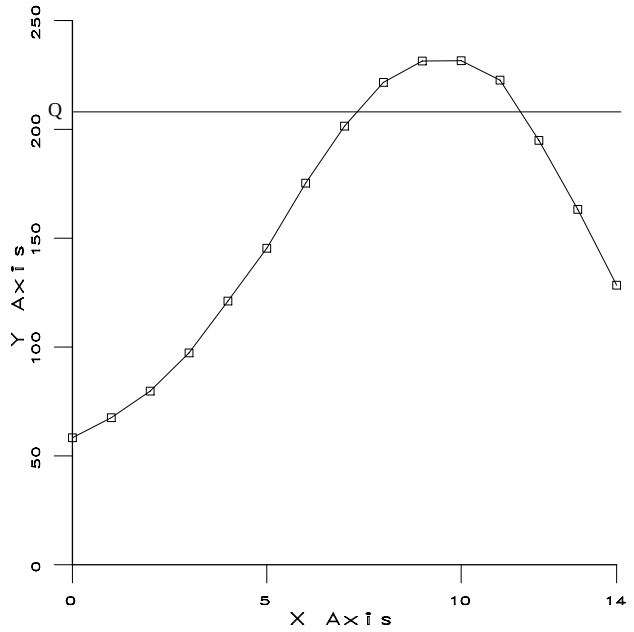

FIG. 7: Sharpness measurements.

The variable state vector $S_v(t)$ is just a scalar defining the position of the gripper tip on the optical axis and the control vector $C(t)$ is constant scalar (e.g., $r := 30mm$).

$$C(t) := \begin{cases} r & : \quad (Q^* - Q(t)) > 0 \\ 0 & : \quad else \end{cases} \quad ; \quad C(t) := \begin{cases} r & : \quad (Q^* - Q(t)) < 0 \\ 0 & : \quad else \end{cases} \tag{8}$$

The width and height of visibility must be determined at the top and bottom point of sharpness which are incident to the top and bottom rectangle of the truncated pyramid. Once again the agility of the manipulator comes into play to determine the rectangle corners. First the gripper is servoed on the top plane and second on the bottom plane. Sequentially the gripper must reach those four $3D$ positions for which the gripper tip is projected onto one of the image corners. The control schema is equal to the one for determining the optical axis with redefined measurement vectors and control vectors. Repeating the procedure for both planes we obtain the eight corners of the truncated pyramid. For example, using quadratic images from the our head-camera (focal length $69mm$) the sidelength of the top rectangle is $80mm$ and of the bottom rectangle $90mm$.

3.4 Locating the robot head

The perspective projection matrices of the head-camera–manipulator relation are computed roughly and therefore a localization of the robot head is inaccurate if using the

matrices directly. Fortunately, we can construct the optical axes of the head-cameras exactly using IBMS and determine from those the head position in the manipulator coordinate system. The tilt rotation axis and the two vergence rotation axes intersect at the focal points of the two cameras (see Figure 8). Two arbitrary angles ϕ^1 and ϕ^2 of the tilt DOF are used, and for each the optical axes of the two head-cameras are determined. This gives two pairs of intersecting straight lines, one intersection point P_1^H is equal to the focal point of the left camera and the other point P_2^H is the one of the right camera (see Figure 9). The middle point P^H between both specifies the head position.

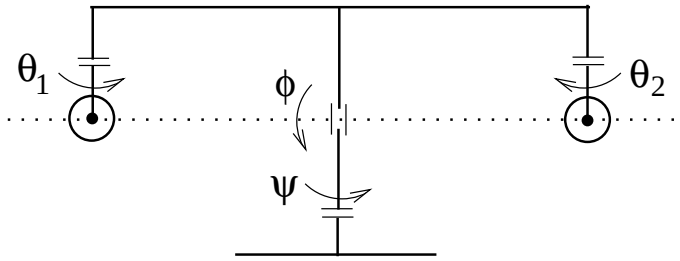

FIG. 8: Degrees of freedom of the robot head.

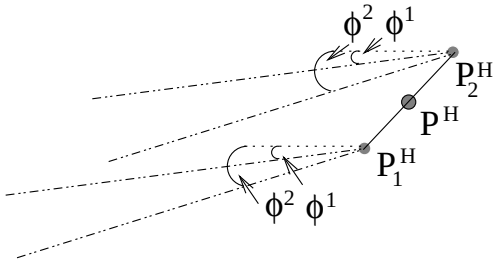

FIG. 9: Optical axes under changing tilt.

4 Manipulator servoing for tool handling

A principal goal of IBRS is manipulating objects. The manipulator carries a tool for changing the pose or shape of an object. Tool handling is composed of four successive stages. First the tool approaches the object and second is fine-controlled until it takes on a certain spatial relation to the object. Third the tool works, i.e., it must be fine-controlled continually and through careful movement the object is manipulated. Fourth the tool will be decoupled from the object.

4.1 Approaching the tool to a object

The head-cameras are used for taking stereo images from the manipulator working space continually. In the images the object position and the current tool position are detected and according to the control rule an increment vector for moving the tool nearer to the object is computed (similar to [5]).

Extracting image positions of gripper and object

In each of the stereo images both the gripper tip and the target object must be located. Gripper detection has already been tackled in chapter 3. The detection of an object also requires a specific operator which is tuned to specific object attributes. In [12] a neural network of radial basis functions is trained with the grey level images of various object views and thus the appearance manifold is represented. Furthermore this network is extended with an output layer in which the weights can be trained such that the whole network computes a value of reliability for the object to be recognized. Alternatively the approach for object detection in [13] uses geometric features which are invariant under perspective projection. Based on the Hough transformation a set of invariants are extracted and combined for the purpose of detecting an object of approximate parallelepiped shape. However these approaches are time-consuming and therefore should be applied only prior to the control cycle. For stationary or slow moving objects this is acceptable because during the servoing cycle an efficient procedure for change detection can be used to verify or determine the new object position.

Determining the control rule

Corresponding object positions in the stereo images must be related to positions in the manipulator coordinate system (i.e., changes of positions). The Jacobians $J_1(P)$ and $J_2(P)$ of equation (6) for the two head-cameras are simply combined in a (4×3) matrix. To transform a desired change from stereo image coordinates into manipulator coordinates the pseudo inverse $J^{\dagger}(P)$ is computed.

$$J(P) := \begin{pmatrix} J_1(P) \\ J_2(P) \end{pmatrix} \quad ; \quad J^{\dagger}(P) := (J^T(P) \cdot J(P))^{-1} \cdot J^T(P) \tag{9}$$

The variable state vector $S_v(t)$ is defined by the $3D$ coordinate vector $P(t)$ of the gripper tip. The desired measurement vector is a $4D$ vector of the $2D$ positions of the object in the stereo images, the current measurement vector represents the stereo 2D positions of the gripper tip.

$$Q^* := \begin{pmatrix} p_1^* \\ p_2^* \end{pmatrix}; \quad Q(t) := \begin{pmatrix} p_1(t) \\ p_2(t) \end{pmatrix} \tag{10}$$

With these definitions the control scheme in equation (7) can be applied. The manipulator gripper approaches the object, and if the object is moving then the gripper will follow it.

Experiments

The usefulness of servoing is exemplified for inaccurate head-camera–manipulator relations. The manipulator working space is a cube of sidelength $400mm$. The spatial dis-

tance between head-camera and manipulator is about $1500mm$, the head-camera focal length is taken as $12mm$. The self-calibration procedure has been applied for three different densities of calibration points (i.e., stopping places of the gripper). Distances of $100mm$, $200mm$, or $400mm$ yield 125, 27, or 8 calibration points respectively from which three projection matrices are computed. In all experiments the gripper starts at a corner and must be servoed to the center of working space. For a servoing factor $s := 0.5$ it turns out that at most 10 cycle iterations are necessary until convergence. After convergence we make mesurements of the deviation from the $3D$ center point. First, the servoing procedure is applied under the use of the three mentioned projection matrices. The result is that the final deviation from the goal position is at most $5mm$ with no direct correlation to the density of calibration points (i.e., the accuracy of the projection matrices). According to that it is sufficient to use just eight corners of the working space for head-camera–manipulator self-calibration. Second, the servoing procedure is applied after changing certain geometric parameters of the robot head respectively. Changing the head position in a circle of radius $100mm$, or changing pan or tilt DOF within angle interval of 10 degrees yield deviations from goal position of at most $25mm$. The errors occur mainly due to the restricted image resolution of 256×256 pixels. According to that the head-camera–manipulator relation need not be re-calibrated in case of the mentioned changes of the pre-calibrated arrangement.

4.2 Assembling the tool to an object

The gripper has approached a motionless object such that a safe distance is kept both over and in fornt of the object (first image in Figure 10). Now the manipulator will be carefully servoed to an optimal grasping situation based on the manipulator mounted camera (fourth image in Figure 10). Using an objective with large focal length the situations are depicted with high resolution which is a precondition to reach a high accuracy during assembling. To simplify recognition of changing situations first a rotation and then a translation takes place.

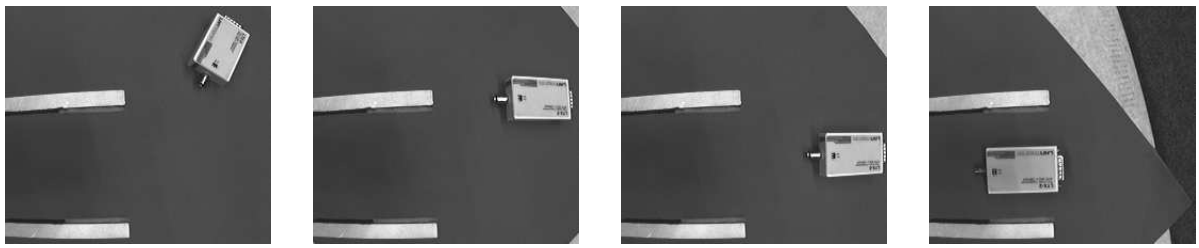

FIG. 10: Assembling the gripper to an object.

Rotational movement of the gripper

The purpose of gripper rotation is to make the finger orientation equal to the principal orientation of the object (second image in Figure 10). For simplicity it is assumed to work on a horizontal plane and thus deal only with one rotating degree of freedom (i.e.,

the roll parameter $R(t)$). Both fingers and the object are detected in the grey level image (see procedures in chapters 3.1 and 4.1). As a result we assume a binarized image of greylevel edges which originate from fingers and object boundary. We perserve the edges orientations and construct a histogram thereof. Figures 11 and 12 show these histograms prior and after the cycles of rotational gripper movement (for first and second image in Figure 10). The position of the first peak in Figure 11 specifies the principal orientation r_o of the object and the second larger peak the gripper orientation r_g in the image. During the servoing cycle the gripper orientation changes but due to the mounted camera a change of the object orientation appears. Accordingly, the first histogram peak must move to the right until it unifies into the second peak (Figure 12).

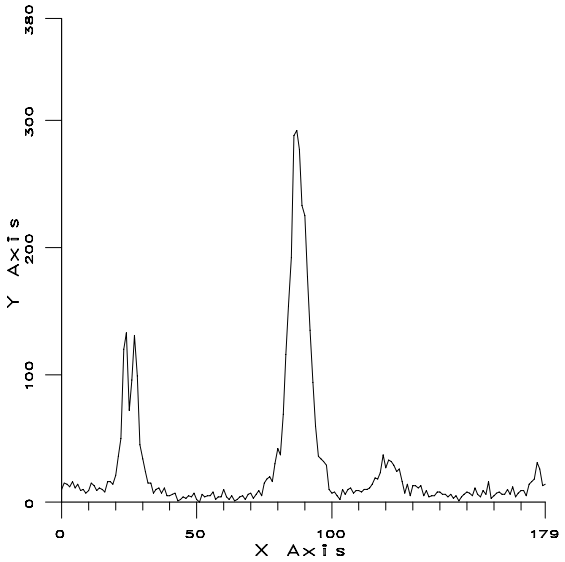

FIG. 11: Edge orientation histogram prior to rotation.

For the control scheme we define the variable state vector $S_v(t) := R(t)$, the current measurement vector $Q(t) := r_o(t)$, the desired measurement vector $Q^* := r_g$.

$$C(t) := \begin{cases} r \cdot signum(Q^* - Q(t)) & : \quad |Q^* - Q(t)| > thresh \\ 0 & : \quad else \end{cases} \tag{11}$$

For the case that desired and current orientation deviate more than a threshold $thresh$ the orientation changes by a small value r.

Translational movement of the gripper

The gripper will be servoed such that the gripper reference point is collinear with the principal axis of the object and then is servoed along the direction of this axis until a certain grasping situation is reached. A constant increment vector r is prefered (similar to the case of rotation) for better surveying the movement. Reasonable values for r

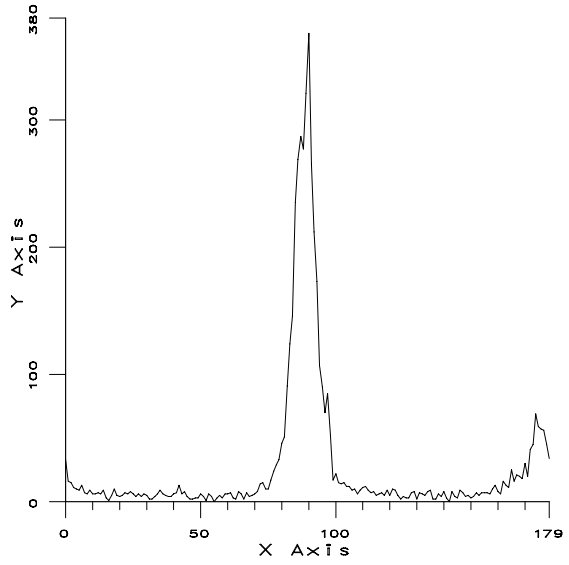

FIG. 12: Edge orientation histogram after rotation.

are obtained by offline experimentation. For defining grasping situations we can take the gripper reference point and the object center point into account, e.g., computing the city block distance between both. If this distance falls below a certain threshold $thresh$ then the desired grasping situation is reached, else the gripper translates in small increments. An alternative approach for evaluating the grasping stability is demonstrated in [12] which avoids the use of geometric features. A neural network learns to evaluate the stability of grasping situations on the basis of training examples. These example situations are represented as patches of filter responses in which a band pass filter is tuned to specifically respond on certain relationships between grasping fingers and object. These filter responses $E(t)$ implicit represent a measurement of distance of the gripper from the most stable position. For example, when the gripper moves step by step to the most stable grasping pose and then moves off the network learns a parabolic curve with the maximum at the most stable situation. A precondition for using the approach is that gripper and object must be in a small neighborhood so that the filter can catch the relation. Instead of computing for the vector of filter responses a value of grasping stability it is possible to associate an appropriate increment vector for moving the gripper. In that case the control rule h is implemented as neural network which is applied to a filter response vector $E(t)$.

4.3 Continual handling of a tool

Frequently for object manipulation it is required to move the tool along a certain trajectory and furthermore keep a certain orientation relative to the object. For example, we assume that a gripper finger must be servoed at a certain distance over an object surface and must be kept normal to the surface. For example, taking the application area of dismantling computer monitors a plausible strategy is to detach the front part of a monitor

case using a laser beam cutter. The trajectory of the cutter is approximately a rectangle which surrounds the front part and during this course the beam orientation should be kept orthogonal to the lines of the rectangle. Figure 13 shows stereo images of a monitor (focal length $12mm$) and in more detail the finger–monitor relation (focal length $69mm$). For this advanced application of IBMS the control problem is much more complicated. First, the goal situation actually is an ordered sequence of intermediate goal situations which must be reached step by step. Second, the measurement vector describung a situation must be partitioned into two subvectors the first one consisting of attributes which should be kept constant and the second one consisting of attributes which must change systematically to reach the next goal situation. Third, for specifying criteria under which the goal situations are reached it is advantageous to visually demonstrate these situations in an offline stage.

FIG. 13: Stereo images of a monitor, and detailled finger–monitor relation.

Supporting manipulator servoing by visual demonstration

The control cycles for approaching and assembling a tool to a target object are running as long as the deviation between current situation and goal situation is larger than a certain threshold *thresh*. However the value for this parameter must be specified in terms of pixels which is inconvenient for system users. Unfortunately in complicated applications even a *vector of threshold values* must be specified. To simplify this kind of user interaction it makes sense to manually arrange certain goal situations prior to the servoing cycles and take images. These images are analysed with the purpose of automatically extracting the goal situations and furthermore determining relevant thresholds which describe acceptable deviations. E.g., for servoing the finger we must specify in terms of pixels the permissible tolerance for the orthogonality to the surface and for the distance from the surface. Actually, this tolerances are a priori known in the euclidean $3D$ space but must be determined in the images. Figure 14 shows in the first and second image exemplary the tolerance concerning orthogonality and distance and in the third and fourth image non-acceptable deviations. For determining the acceptable variances in both parameters a simple image subtraction and a detailled analysis of the subtraction area is useful.
A further even more important aspect of visual demonstration is to acquire operators for situation recognition prior to the servoing cycles. The goal situations (including typ-

FIG. 14: Acceptable and non-acceptable finger–monitor relations.

ical permissible variations) are manually arranged and by taking images for each goal situation a manifold of situation apperances is constructed. From that, operators for situation recognition can be learned, e.g., using RBF networks in [12]. The great strength of this approach is that we don't have to provide geometric models for the recognition task. Instead, operators for recognition are learned on the basis of real examples from the appearances of situations.

Behaviour-based strategy for continual object handling

The complex task of continual object handling can best be organized by several behaviours each one performing perception–action cycles with the purpose of retaining or striving for a certain subgoal. To obtain an overall behaviour as desired the basic behaviours must cooperate appropriately, e.g., working in parallel or exclusive, or one suppressing or animating the other [14]. For the task of detaching the front part of a monitor case one basic behaviour is responsible for servoing the tool over the monitor (*go–over behaviour*) and another one for keeping the tool in a certain relation to a part of the surface (*keep–relation behaviour*). We assumed an approximate rectangular trajectory over the front part of the monitor. The go–over behaviour strives for moving along an exact rectangle but will be modified slightly by the keep–relation behaviour.
For the go–over behaviour four intermediate subgoals are defined which are the four corners of the monitor front. The head-cameras are used for taking stereo images each of which containing the whole monitor front and the gripper finger. In both images we extract the four (virtual) corner points of the monitor, e.g., using one of the recognition approaches discussed above. By combining the corresponding $2D$ coordinates between the stereo images we obtain four $4D$ vectors which represent the intermediate goal positions in the stereo images, i.e., we must pass successively four desired measurement vectors $Q^*(1), Q^*(2), Q^*(3), Q^*(4)$. The variable state vector $S_v(t)$ is defined as the $3D$ coordinate vector $P(t)$ of the finger tip, and the current measurement vector $Q(t)$ represents its position in the stereo images. The pseudo inverse $J^\dagger(S_v(t))$ of the Jacobian is taken from equation (9). The control rule for approaching the desired measurement vectors $Q^*(i), i \in \{1,2,3,4\}$, is as follows.

$$C(t) := \begin{cases} s \cdot \frac{J^\dagger(S_v(t)) \cdot (Q^* - Q(t))}{|J^\dagger(S_v(t)) \cdot (Q^* - Q(t))|} & : \quad |Q^* - Q(t)| > thresh \\ 0 & : \quad else \end{cases} \tag{12}$$

In the application phase parameter i is running from 1 to 4, i.e., as soon as $Q^*(i)$ is passed taking threshold $thresh$ into account then the behaviour is striving for $Q^*(i+1)$. Due

to the normalization involved in the control rule an increment vector of constant length is computed which makes sense because in the application a movement with constant velocity is favourable.
The keep–relation behaviour is responsible for keeping the finger in an orthogonal orientation near to the current part (determined by the go-over behaviour) of the monitor surface. For taking images from the situations at a high resolution (see Figure 14) the manipulator camera is used. Similar to the assembling task (chapter 4.2) a rotational and/or a translational movement takes place if the current situation is non-acceptable.
For rotational servoing simply histograms of edge orientations are used to distinguish between acceptable and non-acceptable angles between finger and surface. Coming back to the role of visual demonstration it is necessary to acquire three classes of histograms prior to the servoing cycles. One class consisting of acceptable relations and two other classes representing non-acceptable relations with the distinction of clockwise or counter-clockwise deviation from orthogonality. Based on that a certain angle between finger and surface will be classified during the servoing cycles using its edge orientation histogram.[2] For example an RBF neural network [15] can be used in which a collection of hidden nodes represents the three manifolds of histograms and an output node computes an evidence value indicating the relevant class, e.g., value near to 0 for acceptable relations and values near to 1 or -1 for non-acceptable clockwise or counter-clockwise deviation. As usual the hidden nodes are created on the basis of the *c-means clustering algorithm* and the link weights to the output node are determined by the *pseudo inverse technique*. The control rule for the rotation task is similar to equation (11) with the distinction that a measure of distance between current and desired measurement vectors (i.e., edge orientation histograms) is computed by the RBF network.
For translating the finger to reach and then keep a certain distance to the monitor a strategy similar to the translational movement of the gripper can be applied (see chapter 4.2). The cooperation between the go–over behaviour and the keep–relation behaviour is according to the principle of alternation. The go–over behaviour follows step by step the corners of the monitor und computes in each iteration of its control cycles a small increment towards the next monitor corner. Then the control cycles of the keep–relation behaviour starts to bring the tool into the desired relation to the monitor. Next, again an iteration of the go–over control cycle comes into play, and so on.

5 Manipulator and/or head servoing for object inspection

A further primary goal of image-based robot servoing is to acquire information about certain objects (e.g., object inspection). The optical axes and the fields of visibility of the head-cameras are important for this purpose. For example, the manipulator can carry an object into the field of visibility of a head-camera, then move the object along the optical

[2]The strength of applying the learning process to the raw histogram data is that the network can generalize from a large amount of data. However, if data compression would be done prior to learning (e.g., computing symbolic values from the histograms) then quantization or generalization errors are unavoidable.

axis towards the camera to increase image resolution and finally rotate the object to view the object from various orientations. Alternatively, the degrees-of-freedom of the robot head can be controlled to move the field of visibility to the object place.

5.1 Role of the optical axis for object inspection

The optical axis is a useful guideline for manipulator or head servoing in order to extract object information from adequate images. Taking this optical axis constraint into account various techniques become simplified or even applicable at all.

Reasonable size, resolution, and orientation of an object

We assume that an object of interest is located at a point on the optical axis. For depicting the object with reasonable size and resolution the focal length of the head-camera can be servoed. An appropriate object orientation is reached with the rotary table. Figure 15 shows an object taken under large (left) and small (middle) focal length, and under degenerate orientation (right).

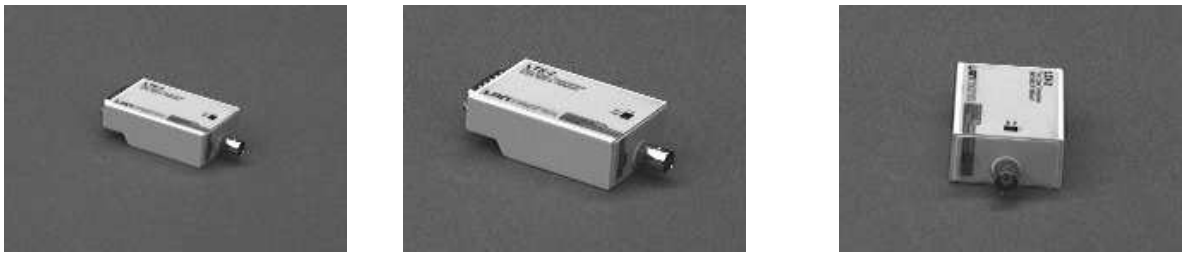

FIG. 15: Transceiver box, taken under large and small focal length, and under degenerate orientation.

The change of the depicted object size can be evaluated by image subtraction, active contour construction, optical flow computation, etc. For example, an active contour approach [16] is simply started by putting an initial contour at the image center and then expanding it step by step until the background image area of the object is reached which is assumed to be homogeneous. Based on this representation it is easy evaluated whether the object silhouette is of a desired size or locally touches the image border and thus meets an optimality criterion concerning depicted object size.
The change of object resolution in the image can be evaluated by frequency analysis, Hough transformation, steerable filters, etc. For example, using Hough transformation we extract boundary lines and evaluate distances between approximate parallel lines. A measure of resolution is based on the pattern of peaks within a horizontal stripe in the Hough image. Figure 16 shows for the images in Figure 15 the Hough image respectively. For the case of low (high) resolution the horizontal distances between the peaks are small (large). Having the object depicted at the image center the straight boundary lines of a polyhedral object can be approximated as straight image lines due to minimal perspective distortions.

FIG. 16: Hough transformation of binarized images in Figure 15.

For the purpose of object recognition we are interested in taking images under a general object orientation, e.g., three visible faces of the transceiver box in Figure 15 (left and middle). However the degenerate object view in Figure 15 (right) only shows two faces. Taking the peak pattern of the Hough transformation into account we can differentiate between general and degenerate views (see Figure 16, middle and right). According to that the object can be rotated appropriately while preserving its position on the optical axis.

Depth and shape reconstruction

For reconstructing depth or shape of an object once again the optical axis is important. By taking two images under different focal length (see Figure 15, left and middle) we can use simple constraints for solving the serious matching problem. Corresponding features (e.g., lines) between the images are detected under the reasonable assumption that they must expand with the image center as the focus of expansion. Alternatively, a depth-from-focus strategy of shape reconstruction can be applied [17] in which the manipulator carries the object along the optical axis. The image is partitioned regulary and for each patch a measure of sharpness is computed (e.g., see chapter 3.3). During the manipulator movement we take images at certain positions and evaluate for each patch the sharpness. This gives for each patch a series of sharpness values and from that we look for the maximum and associate the relevant position of the manipulator gripper to it. Accordingly, for each patch individual positions of the manipulator gripper are determined which indicate the depth and shape of the object. Usually the aspect of image point motion must be taken into account [18] which is simplified under the assumption that the focus of expansion is located at the image center.

5.2 View control of the head-camera system

Suppose the head-camera system is used to visually control the actions of the manipulator, e.g., approaching the gripper to an object. As a precondition the task-specific working space of the manipulator must be contained in the common field of visibility of the two head-cameras.

Visibility of a certain working space

For simplifying the task we construct a sphere which minimally surrounds the volume of

the working space. Its position is known in the manipulator coordinate system. On the other hand also the head position is represented in the coordinate system of the manipulator (see chapters 3.1, 3.2 and 3.4). Furthermore a relationship is obtained between the quadrupel of pan, tilt, and two vergence values (given in the robot head) and the orientation of the optical axes (given in the manipulator coordinate system). Finally, for a certain value of focal length the field of visibility, i.e., the size and position of the truncated pyramid, is determined for each head-camera (see chapter 3.3). If the sphere fits into the visibility pyramid of a head-camera then the focal length should be taken as appropriate. Else it must be decreased systematically until the fitting condition is fulfilled. With all these data the robot head can be steered directly such that both optical axes intersect at the center of the sphere and both vergence angles are equal. This arrangement of the robot head is considered as optimal for observing the manipulator actions.

Object inspection with view control

In contrast to the previous case the head-cameras can be used simultaneously but dissimilar concerning the focal length. If we use a constant small focal length for the left and a constant large focal length for the right head camera we take a wide range of the environment with the left and a small with the right one. For example, in Figure 15 the left image shows a (large) object as a whole and the right image shows an object part in detail. As opposed to the above arrangement these two images could be taken by the head-cameras simultaneously. Accordingly the field of visibility of the right camera must be contained in the field of visibility of the left camera. For this arrangement the pan and tilt DOF are servoed systematically such that the left camera always depicts the object as a whole and the right camera successive inspects certain parts of the object in detail. For an automatic control the visible part of the scene taken by the right camera must be known in the image taken by the left camera. Then a strategy similar to continual object handling in chapter 4.3 can be applied to inspect the front of a monitor case. The robot head will be servoed such that the right camera inspects successive the border of the monitor.

6 Summary

The usefulness of image-based robot servoing was demonstrated for a multi-component robot system consisting of a movable robot head, a stationary manipulator, and a rotary table. The various degrees-of-freedom can be controlled in cooperation to overcome their individual restrictions and exploit their complementary strengths. The most serious problem is image-based situation recognition as a precondition to determine appropriate control signals (desirable in video frame rate). We are convinced that visual demonstration is a step towards solution. Prior to the servoing cycles certain goal situations are manually arranged and from the images thereof a set of appropriate operators for image analysis must be learned automatically. As these operators are grounded in actual situations the application during the servoing cycles should be successful and efficient.

Acknowledgment: Many thanks to Prof. Sommer for the valuable discussions. The contributions of S. Kunze, F. Lempelius, M. Päschke, and A. Schmidt are greatfully appreciated.

References

[1] N. Ahuja and A. Abbott. Active stereo - integrating disparity, vergence, focus, aperture, and calibration for surface estimation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 15:1007–1029, 1993.

[2] M. Landy, L. Maloney, and M. Pavel, editors. *Exploratory Vision – The Active Eye.* Springer Verlag, Berlin, 1995.

[3] K. Hashimoto, editor. *Visual Servoing.* World Scientific Publishing, Singapore, 1993.

[4] P. Corke. Visual control of robot manipulators – a review. In K. Hashimoto, editor, *Visual Servoing*, pages 1–31. World Scientific Publishing, Singapore, 1993.

[5] G. Hager, W. Chang, and A. Morse. Robot hand-eye coordination based on stereo vision. *IEEE Control Systems*, pages 30–39, February 1995.

[6] J. Feddema, C. Lee, and O. Mitchell. Model-based visual feedback control for a hand-eye coordinated robotic system. *Computer*, pages 21–31, August 1992.

[7] P. Papanikolopoulos and P. Khosla. Robotic visual servoing around a static target - an example of controlled active vision. In *Proceedings of the American Control Conference*, pages 1489–1494, 1992.

[8] F. Chaumette, S. Boukir, P. Bouthemy, and D. Juvin. Structure from controlled motion. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 18:492–504, 1996.

[9] J. Craig. *Introduction to Robotics.* Addison-Wesley Publishing Company, Massachusetts, 1989.

[10] V. Leavers. Survey: Which Hough transform ? *Computer Vision, Graphics, and Image Processing – Image Understanding*, 58:250–264, 1993.

[11] O. Faugeras. *Three-Dimensional Computer Vision.* The MIT Press, Cambridge, Massachusetts, 1993.

[12] J. Pauli. Learning to recognize and grasp objects. *Autonomous Robots*, 5:407–420, 1998.

[13] J. Pauli. Learning to recognize and grasp objects. *Machine Learning*, 31:239–258, 1998.

[14] R. Brooks. A robust layered control system for a mobile robot. *IEEE Journal of Robotics and Automation*, 2:14–23, 1986.

[15] D. Hush and B. Horne. Progress in supervised neural networks. *IEEE Signal Processing Magazine*, 10:8–39, January 1993.

[16] D. Williams and M. Shah. A fast algorithm for active contours and curvature estimation. *Computer Vision, Graphics, and Image Processing – Image Understanding*, 55:14–26, 1992.

[17] T. Darell and K. Wohn. Depth from focus using a pyramid architecture. *Pattern Recognition Letters*, 11:787–796, 1990.

[18] S. Olsen. Image point motion when zooming and focusing. In *The 10th Scandinavian Conference on Image Analysis*. Finnland, 1997.