Active 3-D Vision on a Humanoid Head

Aleš Ude [1,3], Erhan Oztop [2,3]

[1]*Jožef Stefan Institute, Dept. of Automatics, Biocybernetics, and Robotics, Slovenia* ales.ude@ijs.si

[2]*Japan Science and Technology Agency, ICORP, Computational Brain project, Japan* erhan@atr.jp

[3]*ATR Computational Neuroscience Laboratories, Dept. of Humanoid Robotics and Computational Neuroscience, Japan*

***Abstract*— In this paper we describe and experimentally evaluate how to model, control, and use the capabilities of a humanoid visual system with foveated vision. We present a computational process that can be utilized to identify and update the parameters of the robot's eyes under motion, which enables the use of 3-D vision on an active humanoid head. We also derive the formulas expressing the geometry of our foveated vision setup. Based on these results we can actively control the eye gaze towards the potential regions of interest and analyze these areas using foveation and 3-D vision processing. Experimental results showing the accuracy of the system are provided. The system has been demonstrated to be sufficiently accurate to realize grasping using active 3-D vision.**

I. Introduction

Foveated vision refers to the property of a human retina, on which the resolution gradually decreases away from the fovea. The main benefit of this arrangement is the relatively low average image resolution over the complete field of view while maintaining high resolution at the center of view. In this way human vision balances the trade-offs between the necessary computational resources and the accuracy of computation. It is clear that foveated vision only makes sense on an active system where it is possible to direct the gaze towards areas of interest that need to be processed with higher precision. By replicating the foveated structure of the human eye and the human oculomotor system, humanoid robot vision becomes significantly more complex than standard active vision.

There are various ways to realize foveated vision on a humanoid robot. The approach we followed (see Fig. 1) is to use two cameras in each eye equipped with lenses with different focal lengths [1], [2], [3], [4]. In this way we can *simultaneously* acquire wide and narrow field-of-view images. While the physical resolution of all cameras is the same, wide-angle (or peripheral) cameras provide images of larger regions at a lower resolution, whereas narrow-angle (or foveal) cameras provide images of smaller areas but at a higher resolution. For many tasks it is important that the robot can use information from peripheral views to get the area of interest into foveal views because it is difficult to move the eyes quickly and accurately enough to keep the area of interest over the center of foveal views at all times. For example, when tracking objects can easily be lost from foveal views, thus we cannot control the head using foveal cameras only [5]. On the other hand, the object is much less likely to disappear from peripheral views due to the wider field of view of these cameras.

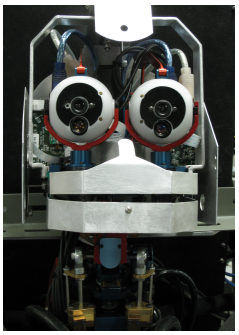

Fig. 1. The head of humanoid robot CB-i [6]. Foveation is realized using two cameras in each eye. Each eye has two independent degrees of freedom: pan (right-left) and tilt (up-down) rotation. The foveal cameras are placed vertically below the peripheral cameras.

The ability to use 3-D vision is very important for manipulation tasks in which the robot performs high-precision actions such as grasping. However, using 3-D vision on an active system is much more difficult than on a static system because various transformations need to be updated on-line to enable 3-D computations. On the other hand, we do not want to give up the ability to use foveated vision. Hence we need to understand how to model the kinematic relationships between different cameras and how to update them. In this paper we put the foundations for the integrated use of 3-D vision and foveation.

II. Calibration of a Humanoid Visual System

As described in the introduction, a humanoid vision system consists of two eyes, each with one or more independent degrees of freedom. Since the relative arrangement of the cameras mounted in different eyes changes as the eyes move, 3-D vision is possible only if both the optics and the motor system of the eyes are properly modeled. Here we first briefly describe how to calibrate the cameras at a specific, static configuration[1]. This is needed to explain the real contributions of the paper. We continue by providing the methodology to compute the transformation between the camera and motor coordinate systems and finally describe how to realize 3-D vision when the eyes move.

A. Static Stereo Camera Calibration

For the purpose of stereo calibration, we model the cameras by a standard pinhole camera model. We denote

[1]The eyes of our robot have independent pan and tilt degrees of freedom. We chose the arrangement in which the cameras are aligned ($\mathrm{pan} = \mathrm{tilt} = 0$) as a special configuration used to calibrate the optical system.

a 3-D point by $\mathbf{y} = \begin{bmatrix} x & y & z \end{bmatrix}^T$ and a 2-D point by $\mathbf{u} = \begin{bmatrix} u & v \end{bmatrix}^T$. Let $\tilde{\mathbf{y}} = \begin{bmatrix} x & y & z & 1 \end{bmatrix}^T$ and $\tilde{\mathbf{u}} = \begin{bmatrix} u & v & 1 \end{bmatrix}^T$ be the homogeneous coordinates of $\mathbf{y}$ and $\mathbf{u}$, respectively. The relationship between a 3-D point $\mathbf{y}$ and its projection $\mathbf{u}$ is then given by [7]

$$s\tilde{\mathbf{u}} = \mathbf{A} \begin{bmatrix} \mathbf{R} & \mathbf{t} \end{bmatrix} \tilde{\mathbf{y}}, \tag{1}$$

where s is an arbitrary scale factor, $\mathbf{R}$ and $\mathbf{t}$ are the extrinsic parameters denoting the rotation and translation that relate the world coordinate system to the camera coordinate system and $\mathbf{A}$ is the intrinsic matrix

$$\mathbf{A} = \begin{bmatrix} \alpha & \gamma & u_0 \\ 0 & \beta & v_0 \\ 0 & 0 & 1 \end{bmatrix}. \tag{2}$$

Here α and β are the scale factors, γ is the parameter describing the skewness of the two image axes, and (u_0, v_0) is the principal point. The internal camera parameters $\mathbf{A}$ can be estimated by a calibration process described in [7]. For $\mathbf{R} = \mathbf{I}$ and $\mathbf{t} = 0$, the projection is given in the internal image coordinate system.

The effects of lens distortion are not considered in the above camera model. Such an assumption is justified for foveal cameras, which are equipped with lenses with relatively long focal lengths that normally do not exhibit noticeable distortion effects. This is especially true because the distortion function is usually dominated by radial components [8], [7]. Conversely, to achieve wide field of view, peripheral cameras need to have lenses with shorter focal lengths. Cameras with such lenses often produce significantly distorted images. However, the distortion can be corrected in a preprocessing step using a suitable distortion correction procedure, e. g. the one described in [7]. Equation (1) is valid for the distortion-corrected pixels.

For stereo vision, we need to estimate the transformation matrix between the two cameras, e. g. from right to left camera. In this case the calibration of a stereo camera system involves the estimation of the internal parameters of left and right camera $\mathbf{A}_\mathrm{l}$ and $\mathbf{A}_\mathrm{r}$, respectively, and the estimation of the transformation matrix from the right to the left camera frame $\mathbf{T}_\mathrm{c}^\mathrm{rl}$. This is a classic problem and we omit the details here. Given the corresponding points $\mathbf{u}_\mathrm{l}$, $\mathbf{u}_\mathrm{r}$, the 3-D position $\mathbf{y}_\mathrm{l}$ in the left camera coordinate system can be calculated by solving

$$\tilde{\mathbf{u}}_\mathrm{l} = \mathbf{A}_\mathrm{l}\mathbf{y}_\mathrm{l}, \tag{3}$$

$$\tilde{\mathbf{u}}_\mathrm{r} = \mathbf{A}_\mathrm{r}\mathbf{y}_\mathrm{r} = \mathbf{A}_\mathrm{r}\mathbf{T}_\mathrm{c}^\mathrm{rl}\mathbf{y}_\mathrm{l}. \tag{4}$$

Note that distortion should be corrected before applying the above formulas and we therefore do not need to consider it in our analysis.

B. Acquiring Eye-Camera Transformation

Now we turn our attention towards the parameters that need to be estimated on an active humanoid vision system. It is very difficult to mount the cameras on the head so that the internal camera coordinate system would be aligned with the eye rotation axes precisely. Hence to calculate how the cameras move, we need to estimate the (unknown) transformation from the eye coordinate system to the camera coordinate system (see Fig. 2). We denote this transformation by T_ec. To learn it, a calibration object is placed at a fixed location in front of the robot, which moves the eyes to a number of orientations. The poses of the calibration object are estimated at all these configurations using the method described in [7]. Let $\mathbf{T}_\mathrm{co}^j$ and $\mathbf{T}_\mathrm{er}^j$, $j = 0, \ldots, n$, respectively be the poses of the calibration object in the camera coordinate system and the poses of the eyes in the fixed eye coordinate system (in which the eye rotations are defined). $\mathbf{T}_\mathrm{er}^j$ can be easily computed using the joint angles obtained by robot joint sensors and the kinematics of the eye's motor system. On our robot, each eye has an independent pan and tilt degree of freedom with orthogonal and intersecting rotational axes, and the resulting transformation matrices $\mathbf{T}_\mathrm{er}^j$ are pure rotations. The presented approach is, however, more general and does not make this specific assumption.

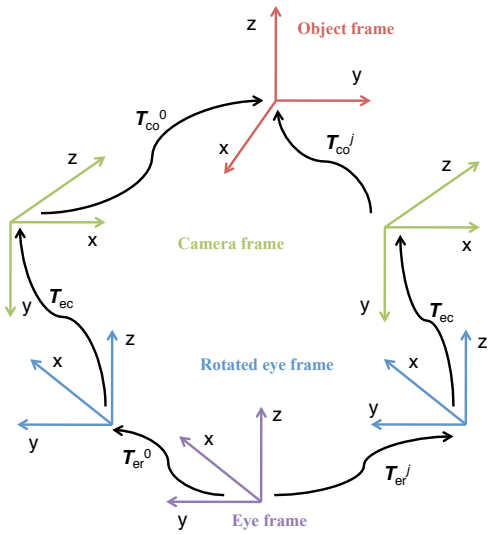

Fig. 2. Coordinate systems that need to be considered for the calculation of eye-camera transformation

Based on Fig. 2, we have the following relationship for each j, $j = 1, \ldots, n$,

$$\mathbf{T}_\mathrm{co}^{0\ -1}\,\mathbf{T}_\mathrm{co}^j = \mathbf{T}_\mathrm{ec}\mathbf{T}_\mathrm{er}^0\,\mathbf{T}_\mathrm{er}^{j\ -1}\,\mathbf{T}_\mathrm{ec}^{-1}. \tag{5}$$

Lets denote $\mathbf{A}_j = \mathbf{T}_\mathrm{co}^{0\ -1}\,\mathbf{T}_\mathrm{co}^j$, $\mathbf{B}_j = \mathbf{T}_\mathrm{er}^0\,\mathbf{T}_\mathrm{er}^{j\ -1}$, and $\mathbf{X} = \mathbf{T}_\mathrm{ec}$. Then we can rewrite the above equations as

$$\mathbf{A}_j\mathbf{X} = \mathbf{X}\mathbf{B}_j, \tag{6}$$

where $\mathbf{A}_j,\ \mathbf{B}_j,\ \mathbf{X} \in \mathrm{SE}(3)$. $\mathrm{SE}(3)$ is the special Euclidean group of rigid body transformations. As noticed by the authors of [9], [10], this equation often arises in problems associated with sensor-robot calibration.

The equation system (6) can be solved analytically by considering the properties of the *logarithmic map* on SE(3). See [11] for an in-depth discussion of the special Euclidean group and the logarithmic map. Writing rigid body transformations as

$$\mathbf{X} = \begin{bmatrix} \mathbf{R}_\mathbf{X} & \mathbf{t}_\mathbf{X} \\ 0 & 1 \end{bmatrix},$$

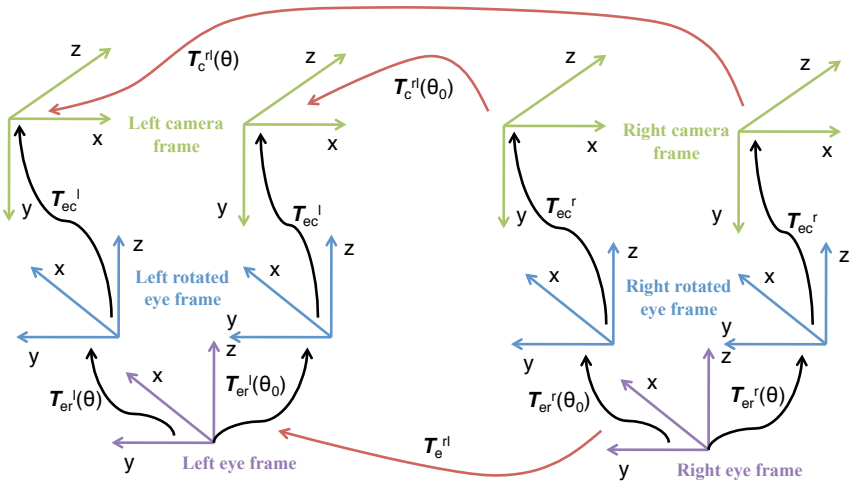

Fig. 3. Coordinate systems that need to be accounted for to realize 3-D vision on an active humanoid robot head

logarithmic map can be applied to transform rotation matrices into skew symmetric matrices $[\mathbf{a}_j] = \log\left(\mathbf{R}_{\mathbf{A}_j}\right)$ and $[\mathbf{b}_j] = \log\left(\mathbf{R}_{\mathbf{B}_j}\right)$ [11], where

$$[\mathbf{x}] = \begin{bmatrix} 0 & -x_3 & x_2 \\ x_3 & 0 & -x_1 \\ -x_2 & x_1 & 0. \end{bmatrix}.$$

Based on the fact that $[\mathbf{b}_j] = \log\left(\mathbf{R}_{\mathbf{B}_j}\right) = \log\left(\mathbf{R}_{\mathbf{X}}\mathbf{R}_{\mathbf{A}_j}\mathbf{R}_{\mathbf{X}}^T\right) = \mathbf{R}_{\mathbf{X}}\log\left(\mathbf{R}_{\mathbf{A}_j}\right)\mathbf{R}_{\mathbf{X}}^T = [\mathbf{R}_{\mathbf{X}}\mathbf{a}_j]$ and using the results from [12], a least-squares solution for (6) was provided in [10], which we repeat here for the sake of completeness. First equation system (6) is rewritten as

$$\mathbf{R}_{\mathbf{X}}\mathbf{a}_j = \mathbf{b}_j \quad (7)$$

$$\left(\mathbf{I} - \mathbf{R}_{\mathbf{A}_j}\right)\mathbf{t}_{\mathbf{X}} = \mathbf{t}_{\mathbf{A}_j} - \mathbf{R}_{\mathbf{X}}\mathbf{t}_{\mathbf{B}_j}. \quad (8)$$

The least squares solution of the equation system (7) on SO(3) (group of all rotation matrices) is then given by [12]

$$\mathbf{R}_{\mathbf{X}} = \left(\mathbf{M}^T\mathbf{M}\right)^{-1/2}\mathbf{M}^T,\ \mathbf{M} = \sum_{j=1}^{N}\mathbf{b}_j\mathbf{a}_j^T. \quad (9)$$

Once we know $\mathbf{R}_{\mathbf{X}}$, Eq. (8) becomes a classical least-squares problem that can be solved for $\mathbf{t}_{\mathbf{X}}$ using standard methods.

Thus by solving Eq. (6) we can compute the transformation between the eye and camera coordinate systems $\mathbf{T}_{\mathrm{ec}}^{\mathrm{l}}$ and $\mathbf{T}_{\mathrm{ec}}^{\mathrm{r}}$ for both left and right camera. Note that this equation can be solved uniquely if and only if $\log(\mathbf{A}_i)\times\log(\mathbf{A}_j) \neq 0$ and $\log(\mathbf{B}_i)\times\log(\mathbf{B}_j) \neq 0$ for at least one pair of i, j. This condition is fulfilled on systems with independent pan and tilt, which are the most useful systems for foveated vision. Note also that it is important to identify the eye kinematics together with the rest of the robot's kinematics, otherwise it is not possible to transform the positions in the eye coordinates into the positions in the robot's body coordinates using Eq. (12). We use standard approaches for the identification of the robot kinematics for this purpose [6].

C. Active Stereo Vision

The transformation $\mathbf{T}_{\mathrm{c}}^{\mathrm{rl}}$ of Eq. (4) is not constant on an active system and therefore needs to be estimated as the eyes move. This can be accomplished by utilizing the results of the static camera calibration process of Sec. II-A, eye-camera transformation of Sec. II-B, and using known eye kinematics. The robot's eye coordinate systems are shown in Fig. 3. Here $\mathbf{T}_{\mathrm{c}}^{\mathrm{rl}}(\theta_0)$ is the transformation from right to left camera at joint configuration θ_0, which is estimated by the static stereo camera calibration. $\mathbf{T}_{\mathrm{ec}}^{\mathrm{l}}$, $\mathbf{T}_{\mathrm{ec}}^{\mathrm{r}}$ are the transformations between the left and right eye and camera, respectively. They are estimated by the calibration process of Sec. II-B and do not depend on the eyes' joint angles. $\mathbf{T}_{\mathrm{er}}^{\mathrm{l}}(\theta)$, $\mathbf{T}_{\mathrm{er}}^{\mathrm{r}}(\theta)$ are the current eye postures at joint angles θ. $\mathbf{T}_{\mathrm{er}}^{\mathrm{l}}(\theta_0)$, $\mathbf{T}_{\mathrm{er}}^{\mathrm{r}}(\theta_0)$, $\mathbf{T}_{\mathrm{er}}^{\mathrm{l}}(\theta)$, $\mathbf{T}_{\mathrm{er}}^{\mathrm{r}}(\theta)$ can be computed using the known eye kinematics and proprioception.

To compute transformation $\mathbf{T}_{\mathrm{c}}^{\mathrm{rl}}(\theta)$, which changes as the eyes move, we first calculate the transformation $\mathbf{T}_{\mathrm{e}}^{\mathrm{rl}}$ between the fixed eye coordinate systems (with respect to the robot head)

$$\mathbf{T}_{\mathrm{e}}^{\mathrm{rl}} = \mathbf{T}_{\mathrm{er}}^{\mathrm{r}}(\theta_0)\mathbf{T}_{\mathrm{ec}}^{\mathrm{r}}\mathbf{T}_{\mathrm{c}}^{\mathrm{rl}}(\theta_0)\ {\mathbf{T}_{\mathrm{ec}}^{\mathrm{l}}}^{-1}\mathbf{T}_{\mathrm{ec}}^{\mathrm{l}}(\theta_0)^{-1}. \quad (10)$$

$\mathbf{T}_{\mathrm{e}}^{\mathrm{rl}}$ is constant and consequently there are only constant terms in the above equation. To transform the coordinates of a 3-D point from the left to the right camera frame, we can use the following formula (see also Fig. 3)

$$\mathbf{y}_{\mathrm{r}} = \mathbf{T}_{\mathrm{c}}^{\mathrm{rl}}(\theta)\mathbf{y}_{\mathrm{l}} = {\mathbf{T}_{\mathrm{ec}}^{\mathrm{r}}}^{-1}\mathbf{T}_{\mathrm{er}}^{\mathrm{r}}(\theta)^{-1}\mathbf{T}_{\mathrm{e}}^{\mathrm{rl}}\mathbf{T}_{\mathrm{er}}^{\mathrm{l}}(\theta)\mathbf{T}_{\mathrm{ec}}^{\mathrm{l}}\mathbf{y}_{\mathrm{l}}. \quad (11)$$

The above transformation allows us to calculate the 3-D point coordinates $\mathbf{y}_l$ in the rotated left camera coordinate system by solving the equation system (3), (4). Finally, the following transformation can be applied to compute the position in the robot body coordinates

$$\mathbf{y}_{\mathrm{b}} = \mathbf{T}_{\mathrm{be}}^{\mathrm{l}}(\theta)\mathbf{T}_{\mathrm{er}}^{\mathrm{l}}(\theta)\mathbf{T}_{\mathrm{ec}}^{\mathrm{l}}\mathbf{y}_{\mathrm{l}}, \quad (12)$$

where $\mathbf{T}_{\mathrm{be}}^{\mathrm{l}}(\theta)$ is the position and orientation of the left eye in the body coordinate system before the eye rotation.

III. Implementing Foveation Using 3-D Vision

In this section we derive the mathematical formulas that enable the robot to move the eyes so that the area of interest is placed over the center of the fovea based on information from peripheral images. As mentioned in the introduction,

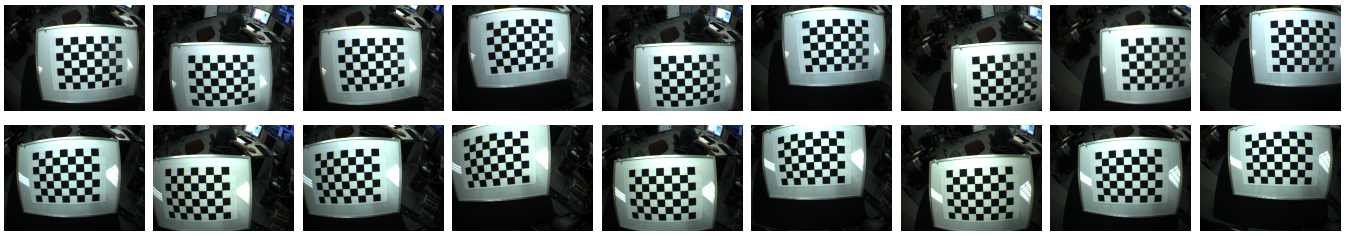

Fig. 4. A typical calibration sequence (upper row: left eye images, lower row: right eye images). The eyes are moved to acquire the snapshots of the calibration object placed at the same location, but from different eye orientations.

it is important that the robot can use information from peripheral views to control the eyes because it is difficult to move the cameras quickly and accurately enough to keep the area of interest over the center of foveal views at all times.

We denote by $\mathbf{A}_\mathrm{f}$ and $\mathbf{A}_\mathrm{p}$ the projection matrices of the peripheral and foveal cameras. Since the mathematics is the same for both left and right eyes, we can drop the corresponding eye indices. Without loss of generality we can assume that the origin of the image coordinate system coincides with the camera coordinate system up to the translation along the optical axis. This can be achieved by subtracting the coordinates of the principal point (u_0, v_0) from the pixel coordinates. In this case we have

$$\mathbf{A}_\mathrm{f} = \begin{bmatrix} \alpha_\mathrm{f} & \gamma_\mathrm{f} & 0 \\ 0 & \beta_\mathrm{f} & 0 \\ 0 & 0 & 1 \end{bmatrix} \text{ and } \mathbf{A}_\mathrm{p} = \begin{bmatrix} \alpha_\mathrm{p} & \gamma_\mathrm{p} & 0 \\ 0 & \beta_\mathrm{p} & 0 \\ 0 & 0 & 1 \end{bmatrix}.$$

Let $\mathbf{t}_\mathrm{fp}$ be the position of the origin of the peripheral camera coordinate system expressed in the foveal camera coordinate system and let $\mathbf{R}_\mathrm{fp}$ be the rotation matrix that rotates the basis vectors of the foveal camera coordinate system into the basis vectors of the peripheral camera coordinate system. Since the foveal and peripheral camera are rigidly attached to each other, $\mathbf{t}_\mathrm{fp}$ and $\mathbf{R}_\mathrm{fp}$ are both constant and can be estimated using standard calibration techniques. Let $\mathbf{y}_\mathrm{f}$ and $\mathbf{y}_\mathrm{p}$ be the position of a 3-D point[2] expressed in the foveal and peripheral camera system, respectively. We then have

$$\mathbf{y}_\mathrm{p} = \mathbf{R}_\mathrm{fp}^T(\mathbf{y}_\mathrm{f} - \mathbf{t}_\mathrm{fp}). \tag{13}$$

The projections of a 3-D point $\mathbf{y}_\mathrm{f} = (x, y, z)$ onto the planes of both cameras are given by

$$u_\mathrm{f} = \frac{\alpha_\mathrm{f} x + \gamma_\mathrm{f} y}{z}, \tag{14}$$

$$v_\mathrm{f} = \frac{\beta_\mathrm{f} y}{z}, \tag{15}$$

and

$$u_\mathrm{p} = \frac{\alpha_\mathrm{p}\mathbf{r}_1 \cdot (\mathbf{y}_\mathrm{f} - \mathbf{t}_\mathrm{fp}) + \gamma_\mathrm{p}\mathbf{r}_2 \cdot (\mathbf{y}_\mathrm{f} - \mathbf{t}_\mathrm{fp})}{\mathbf{r}_3 \cdot (\mathbf{y}_\mathrm{f} - \mathbf{t}_\mathrm{fp})}, \tag{16}$$

$$v_\mathrm{p} = \frac{\beta_\mathrm{p}\mathbf{r}_2 \cdot (\mathbf{y}_\mathrm{f} - \mathbf{t}_\mathrm{fp})}{\mathbf{r}_3 \cdot (\mathbf{y}_\mathrm{f} - \mathbf{t}_\mathrm{fp})}, \tag{17}$$

where $\mathbf{r}_1$, $\mathbf{r}_2$, and $\mathbf{r}_3$ are the rows of the rotation matrix $\mathbf{R}_\mathrm{fp}^T = \begin{bmatrix} \mathbf{r}_1^T & \mathbf{r}_2^T & \mathbf{r}_3^T \end{bmatrix}^T$. $\mathbf{y}_\mathrm{f}$ projects onto the principal point in the foveal camera if $u_\mathrm{f} = v_\mathrm{f} = 0$. Assuming that the point is in front of the camera, hence $z > 0$, we obtain from Eq. (14) and (15) that $x = y = 0$, which means that the point must lie on the optical axis of the foveal camera. Inserting this into Eq. (16) and (17), we obtain the following expression for the ideal position $(\hat{u}_\mathrm{p}, \hat{v}_\mathrm{p})$ in the peripheral camera image that results in the projection onto the principal point in the foveal camera image

$$\hat{u}_\mathrm{p} = \frac{\alpha_\mathrm{p}\mathbf{r}_1 \cdot \mathbf{t}_\mathrm{fp} + \gamma_\mathrm{p}\mathbf{r}_2 \cdot \mathbf{t}_\mathrm{fp} - (\alpha_\mathrm{p} r_{13} + \gamma_\mathrm{p} r_{23})z}{\mathbf{r}_3 \cdot \mathbf{t}_\mathrm{fp} - r_{33} z}, \tag{18}$$

$$\hat{v}_\mathrm{p} = \frac{\beta_\mathrm{p}\mathbf{r}_2 \cdot \mathbf{t}_\mathrm{fp} - \beta_\mathrm{p} r_{23} z}{\mathbf{r}_3 \cdot \mathbf{t}_\mathrm{fp} - r_{33} z}, \tag{19}$$

where $\begin{bmatrix} r_{13} & r_{23} & r_{33} \end{bmatrix}^T$ is the third column of $\mathbf{R}_\mathrm{fp}^T$. Note that the ideal position in the periphery is independent from the intrinsic parameters of the foveal camera. It depends, however, on the distance z of the point of interest from the cameras. Hence to use this formula, we need to be able to calculate depth information, which can be accomplished based on the approaches of the previous section.

By applying the above equations, we can turn the eye gaze towards the object and keep the object in the center of foveal cameras based on information from peripheral views and using image-based closed-loop control [13].

IV. Experimental Results

It is well known that the error in depth is proportional to the disparity error and increases with the squared distance of the object from the camera [14]

$$\epsilon_z = \frac{z^2}{bf}\epsilon_d, \tag{20}$$

where ϵ_z is the depth error, z is the depth, b is the baseline, f is the focal length of the camera in pixels, and ϵ_d is the disparity error in pixels. The error in depth is always larger than the errors in directions along the camera axes. In static systems with accurately calibrated camera parameters, the disparity error is essentially the same as the matching error. However, in the case of an active humanoid head, there is another source of error, namely errors caused by the inaccurate re-calibration of the system when the eyes move, i.e. the errors in $\mathbf{T}_\mathrm{c}^\mathrm{rl}(\theta)$ (see Fig. 5). This error can be caused by inaccurate sensor readings, time delays between image capture and reading of joint angles, vibrations, etc. For the pixel at the image center, we have the following expression

$$\epsilon_d = f \tan(\alpha). \tag{21}$$

[2]In practice $\mathbf{y}$ would be the center of the area of interest

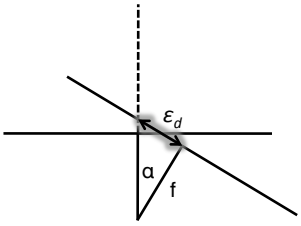

Fig. 5. Disparity error because of the inaccurate orientation. Here α is the error in the orientation between the two image planes, f the focal length in pixel units, and ϵ_d the disparity error for the pixel at the center of the image plane caused by the error in orientation.

The above equation shows that the disparity error caused by the errors in the re-calibration process increases with the focal length of the camera. Hence the conventional wisdom that depth measurements are more accurate when using narrower lenses with longer focal lengths is not necessarily true on active systems because the disparity errors can increase with focal length.

In our experiments we first calibrated the cameras at a particular configuration, which we chose to be the one with the straightforward gaze (eye angles equal to zero). In the second phase, transformation $\mathbf{T}_{\mathrm{e}}^{\mathrm{rl}}$ of Eq. (10) was estimated. We then moved the eyes along the following trajectories

$$\theta^1_{\mathrm{pan}} = 0.25\cos(t),\ \theta^1_{\mathrm{tilt}} = 0.25\cos(t/1.5), \quad (22)$$

$$\theta^2_{\mathrm{pan}} = 0.25\cos(t/2),\ \theta^2_{\mathrm{tilt}} = 0.25\cos(t/2.5). \quad (23)$$

In the first test both eyes remained parallel and moved according to Eq. (22), whereas in the second test the first eye (left eye) moved along the trajectory given by Eq. (22) and the second eye (right eye) along the trajectory given by Eq. (23). The observed object was fixed in all experiments and did not move in space. The task was to estimate the object position in a fixed head coordinate system. The x axis of this coordinate system was roughly aligned with the optical axis of the camera at zero configuration, thus the estimated x coordinates correspond to the depth measurement. See the submitted video to inspect the resulting image motion caused by the eye movement.

Fig. 6 and 7 show that the system is fairly accurate from a distance of about 0.5 meter. Although the eye movement travelled the course of 0.5 radians for each eye degree of freedom, the system was able to estimate a rather constant object position. As one could expect, the system was more accurate for parallel eye movements, which are also much more natural than divergent eye movements, which humans normally do not perform. The largest errors can be found in the depth estimates, but this is expected for stereo vision.

Our results confirm that the error increases significantly with the distance of the object from the eyes (see Fig. 8 and 9). Nevertheless, the results remain accurate except for the depth estimation. The standard deviation in the depth error for the case of parallel eye movements was 6.6 cm, which is still sufficient to roughly reach for the object. However, this result shows that some corrective movements need to be performed when the robot attempts to grasp an object. Since at this point of the grasping action the robot is close to the object and since the cameras do not need to move any more, we can expect smaller errors in this situation and we can conclude that our system is accurate enough to implement grasping behaviors.

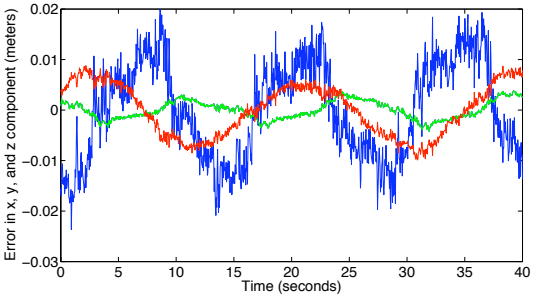

Fig. 6. Fluctuation of the estimated positions in x (blue), y (green), and z (red) coordinates for the parallel eye movement. The observed point was fixed and the estimated mean position was $(0.426, -0.021, -0.081)$ m. The standard deviation was $(0.010, 0.002, 0.005)$ m. Focal length of the lens was 2.8 mm.

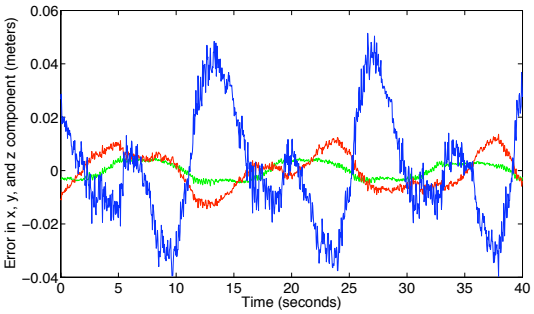

Fig. 7. Fluctuation of the estimated positions in x (blue), y (green), and z (red) coordinates for the divergent eye movement. The observed point was fixed and the estimated mean position was $(0.421, -0.072, -0.088)$ m. The standard deviation was $(0.020, 0.003, 0.007)$ m. Focal length of the lens was 2.8 mm.

In Fig. 10 and 11 we show the results when using lenses with a longer focal length. Considering a slightly longer depth distance, these results are comparable to the results of Fig. 6 and 7. Because of the not completely accurate re-calibration of the system, we cannot expect to significantly increase the accuracy when using lenses with longer focal lengths. We note however, that the resolution of objects in such images is still higher than in the images acquired by cameras with wider field-of-view, which can be important for vision processing.

The analysis of Eq. (18) and (19) shows that the ideal image position for foveation in the peripheral images converges to a fixed value as depth (z in this case) tends to infinity. Hence the errors in depth at large distances do not significantly influence the estimates of the image position for foveation. We may thus conclude that the implemented system is sufficiently precise to implement foveation based on Eq. (18) and (19) and image based feedback control.

V. Conclusions

In this paper we developed mathematical machinery necessary to implement 3-D vision and foveation on an active humanoid head. The system was fully implemented on a humanoid robot CB-i. Our experiments have shown that it is possible to acquire all the necessary parameters using a

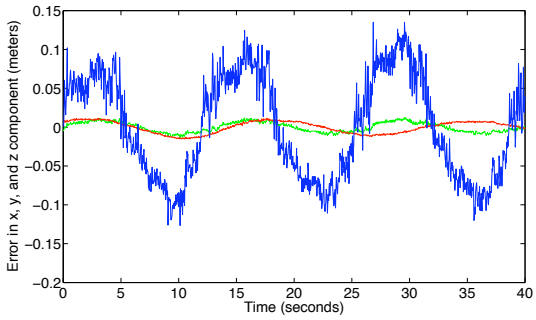

Fig. 8. Fluctuation of the estimated positions in x (blue), y (green), and z (red) coordinates for the parallel eye movement. The observed point was fixed and the estimated mean position was $(1.106, 0.017, 0.003)$ m. The standard deviation was $(0.066, 0.006, 0.008)$ m. Focal length of the lens was 2.8 mm.

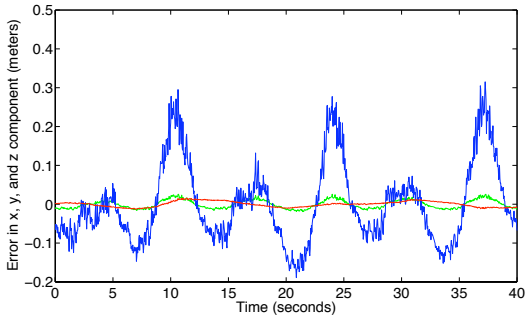

Fig. 9. Fluctuation of the estimated positions in x (blue), y (green), and z (red) coordinates for the divergent eye movement. The observed point was fixed and the estimated mean position was $(1.109, 0.017, 0.001)$ m. The standard deviation was $(0.102, 0.011, 0.007)$ m. Focal length of the lens was 2.8 mm.

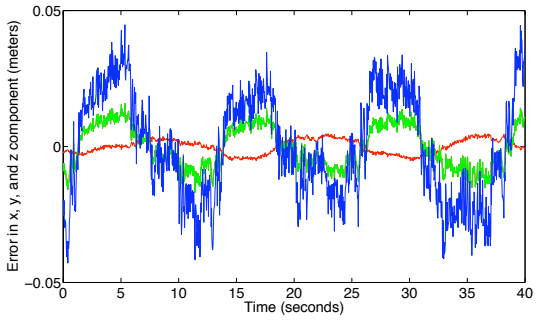

Fig. 10. Fluctuation of the estimated positions in x (blue), y (green), and z (red) coordinates for the parallel eye movement. The observed point was fixed and the estimated mean position was $(0.642, 0.173, -0.035)$ m. The standard deviation was $(0.019, 0.007, 0.003)$ m. Focal length of the lens was 4.0 mm.

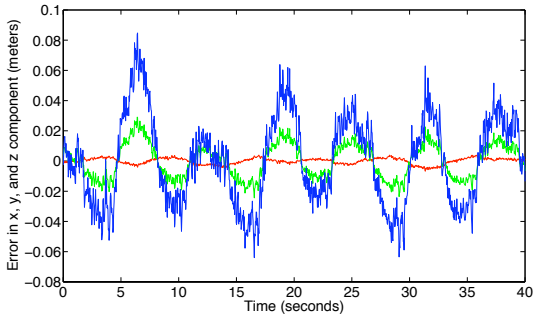

Fig. 11. Fluctuation of the estimated positions in x (blue), y (green), and z (red) coordinates for the divergent eye movement. The observed point was fixed and the estimated mean position was $(0.643, 0.173, -0.035)$ m. The standard deviation was $(0.029, 0.011, 0.002)$ m. Focal length of the lens was 4.0 mm.

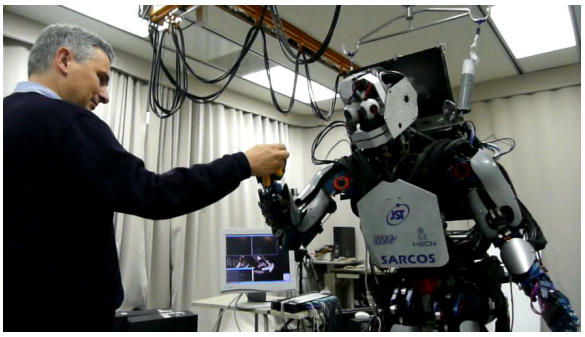

Fig. 12. Grasping an object held by a person. The eyes were continuously centered on the object and the 3-D object position was updated during grasping.

planar calibration pattern such as the one in Fig. 4. This is important because such patterns are much easier to construct and process than classic 3-D calibration objects. The system have been shown to be accurate enough to realize grasping behaviors using active 3-D vision (see Fig. 12).

Acknowledgment: The work described in this paper was partially conducted within the EU Cognitive Systems project PACO-PLUS (FP6-2004-IST-4-027657) funded by the European Commission.

REFERENCES

[1] C. G. Atkeson, J. Hale, F. Pollick, M. Riley, S. Kotosaka, S. Schaal, T. Shibata, G. Tevatia, A. Ude, S. Vijayakumar, and M. Kawato, "Using humanoid robots to study human behavior," *IEEE Intelligent Systems*, vol. 15, no. 4, pp. 46–56, July/August 2000.

[2] H. Kozima and H. Yano, "A robot that learns to communicate with human caregivers," in *Proc. Int. Workshop on Epigenetic Robotics*, Lund, Sweden, 2001.

[3] B. Scassellati, "A binocular, foveated active vision system," MIT A.I. Memo No. 1628, Cambridge, Mass., 1999.

[4] 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 *Proc. IEEE-RAS/RSJ Int. Conf. Humanoid Robots*, Genoa, Italy, 2006.

[5] A. Ude, C. G. Atkeson, and G. Cheng, "Combining peripheral and foveal humanoid vision to detect, pursue, recognize and act," in *Proc. 2003 IEEE/RSJ Int. Conf. Intelligent Robots and Systems*, Las Vegas, Nevada, 2003, pp. 2173–2178.

[6] G. Cheng, S.-H. Hyon, J. Morimoto, A. Ude, J. G. Hale, G. Colvin, W. Scroggin, and S. C. Jacobsen, "CB: a humanoid research platform for exploring neuroscience," *Advanced Robotics.*, vol. 21, no. 10, pp. 1097–1114, 2007.

[7] Z. Zhang, "A flexible new technique for camera calibration," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 22, no. 11, pp. 1330–1334, 2000.

[8] R. Y. Tsai, "A versatile camera calibration technique for high-accuracy 3d machine vision metrology using off-the-shelf tv cameras and lenses," *IEEE J. Robotics Automat.*, vol. 3, no. 4, pp. 323–344, 1987.

[9] Y. C. Shiu and S. Ahmad, "Calibration of wrist-mounted robotic sensors by solving homogeneous transform equations of the form $\mathbf{AX} = \mathbf{XB}$," *IEEE Trans. Robotics Automat.*, vol. 5, no. 1, pp. 16–27, 1989.

[10] F. C. Park and B. J. Martin, "Robot sensor calibration: Solving $\mathbf{AX} = \mathbf{XB}$ on the Euclidean group," *IEEE Trans. Robotics Automat.*, vol. 10, no. 5, pp. 717–721, 1994.

[11] R. M. Murray, Z. Li, and S. S. Sastry, *A Mathematical Introduction to Robotic Manipulation*. Boca Raton, New York: CRC Press, 1994.

[12] A. Nadas, "Least squares and maximum likelihood estimation of rigid motion," IBM Research Report RC9645, Yorktown Heights, 1978.

[13] S. Hutchinson, G. D. Hager, and P. I. Corke, "A tutorial on visual servo control," *IEEE Trans. Robotics Automat.*, 12, pp. 651–670, 1996.

[14] D. Gallup, J.-M. Frahm, P. Mordohai, and M. Pollefeys, "Variable baseline/resolution stereo," in *Proc. IEEE Comp. Soc. Conf. Computer Vision and Pattern Recognition*, Anchorage, Alaska, June 2008.