

Full paper

Redundancy Control of a Humanoid Head for Foveation and Three-Dimensional Object Tracking: A Virtual Mechanism Approach

Damir Omrčen^{a,*} and Aleš Ude^{a,b}^a Jozef Stefan Institute, Jamova cesta 39, 1000 Ljubljana, Slovenia^b ATR Computational Neuroscience Laboratories, 2-2-2 Hikaridai, Seika-cho, Sokaro-gun, Kyoto 619-0288, Japan

Received 6 November 2009; accepted 15 January 2010

Abstract

This paper presents a novel approach for object tracking with a humanoid robot head. The proposed approach is based on the concept of a virtual mechanism, where the real head is enhanced with a virtual link that connects the eye with a point in three-dimensional space. We tested our implementation on a humanoid head with 7 d.o.f. and two rigidly connected cameras in each eye (wide-angle and telescopic). The experimental results show that the proposed control algorithm can be used to maintain the view of an observed object in the foveal (telescopic) image using information from the peripheral view. Unlike other methods proposed in the literature, our approach indicates how to exploit the redundancy of the robot head. The proposed technique is systematic and can be easily implemented on different types of active humanoid heads. The results show good tracking performance regardless of the distance between the object and the head. Moreover, the uncertainties in the kinematic model of the head do not affect the performance of the system. © Koninklijke Brill NV, Leiden and The Robotics Society of Japan, 2010

Keywords

Active stereo vision, foveated vision, humanoid head, virtual mechanism, redundancy resolution

1. Introduction

Human eye movements and gaze direction have a high communicative value [1, 2]. For example, gaze direction is a good indicator of the locus of visual attention. Knowing a person's locus of attention reveals what that person currently considers behaviorally relevant, which is in turn a powerful clue to their intent. The ability to detect another creature looking at you is critical for many species. Many vertebrates

* To whom correspondence should be addressed. E-mail: damir.omrcen@ijs.si

have been observed to change their behavior based on whether or not eyes are gazing at them [3–5]. In humans, eye contact serves a variety of social functions, from indicating interest to displaying aggression [6]. While infants initially lack many social conventions (understanding pointing gestures may not occur until the end of the first year), recognition of eye contact is present from as early as the first month [7, 8]. The dynamic aspects of eye movement, such as staring *versus* glancing, also convey information. Eye movements are particularly potent during social interactions, such as conversational turn-taking, where making and breaking eye contact plays an important role in regulating the exchange. Such cues can also be used to interpret robot behavior. For example, when an anthropomorphic robot moves its eyes and neck to fixate on an object, an observer can conclude that the robot has become interested in that object.

There are different varieties of eye movements. Saccades, smooth pursuit and vergence movements cooperate in the execution of visual tasks [9]. Saccades rapidly reorient the eye to project a different part of the visual scene onto the fovea. If the eyes fixate on a moving object, they can follow it with a continuous tracking movement called smooth pursuit. This type of eye movement cannot be evoked voluntarily, but only occurs in the presence of a moving object [1]. Saccadic and smooth pursuit movements are used in order to compensate for lateral motion of the target. Vergence movements, instead, are mainly based on disparity cues to compensate for motion in depth.

Since the eyes are located on the head, they need to compensate for any head movements that occur during fixation. The vestibulo-ocular reflex uses inertial feedback from the vestibular system to keep the orientation of the eyes stable as the eyes move. This is a very fast response, but is prone to the accumulation of error over time. The optokinetic response is a slower compensation mechanism that uses a measure of the visual slip of the image across the retina to correct for drift. These two mechanisms work together to give humans stable gaze as the head moves [1, 10].

Control circuits that realize three of the most basic oculomotor behaviors and their integration, i.e., the vestibulo-ocular reflex and optokinetic response for gaze stabilization, smooth pursuit for tracking moving objects, and saccades for overt visual attention, have been proposed in Ref. [11].

Many visual tasks require both high resolution and a wide field of view. High acuity is needed for recognition tasks and for controlling precise visually guided motor movements. A wide field of view is needed for search tasks, for tracking multiple objects, etc. A common solution found in biological systems is to have retinas with variable distribution of image elements, thus supporting both types of tasks. This is seen in animals with foveate vision, where the density of photoreceptors is highest at the center of the retina and falls off rapidly towards the periphery. Designers of a number of humanoid robots (Cog [2, 12], Kismet [1], Armar III [13], DB [14], etc.) attempted to mimic the foveated structure of the human eye by using two rigidly connected cameras, one with a wide-angle lens and the other with a

telescopic lens, in each eye. In this paper we discuss the control of such humanoid visual systems and the associated three-dimensional (3-D) vision processing.

Metta *et al.* use log-polar images for recognition and tracking of objects [10, 15]. To control the gaze direction they use learning methods, which define movements of the eye joints. However, since the gaze direction is also affected by the head movement, the system is redundant. To solve that, they use an independent controller to move the neck. To compensate for the neck movement eyes are counter-rotated based on a reactive control using an inertial sensor in the head [10].

Bernardino *et al.* [16] proposed a kinematic and dynamic controller, which is very simple since it decouples the kinematic relations of a robotic head. By decomposition in separate movements they achieved simplification of the sensorimotor process. The simplification results in common movement of both eyes pan, which is in most cases appropriate, human-like movement of the eyes. However, this restricts the diversity of all possible movements. In Ref. [17] a tracking system with log-polar cameras was proposed. The system is based on the estimation of redundant 2-D motion parameters.

Breazeal *et al.* [1] implemented object tracking on a Kismet humanoid robot. Kismet's visuomotor control is modeled after the human oculomotor system. Here, narrow-angle cameras are in the robot eyes, while wide-angle cameras are fixed with respect to the head. Wide-angle cameras define position set-points for the eye motors. This transformation in general requires distance to the object, which is very noisy. Here, the crucial factor to achieve good transformation is also the distance (the transformation) between wide- and narrow-angle cameras. A similar approach was proposed in Ref. [18], where they used a well calibrated head to assure good tracking of an object in space using wide- and narrow-angle cameras. Here, wide-angle cameras define the 3-D position of the object to be tracked.

Ude *et al.* [19] proposed a simpler, decomposed controller, realized as a network of PD controllers. The PD controllers are based on simplified mappings between visual coordinates and joint angles, rather than on a full kinematic model. They use two cameras per eye (wide- and narrow-angle). For that reason they additionally implemented a transformation that assures that the object, which needs to be tracked, is kept in the middle of the narrow-angle cameras, even though it is tracked based on the information provided by wide-angle cameras. Since this transformation depends on the depth, the authors proved that it assures good results if an object is not too close to the cameras. To improve the accuracy of fixation in foveal views, depth information would need to be utilized. Despite a relatively simple controller, the resulting head movements look very natural.

The initial goal of our system is to obtain high-resolution images of an object using a humanoid robot head. As the peripheral cameras have a very wide field of view, we cannot extract detailed object features from these images. Just like humans must fixate on an object to discriminate fine detail, our foveal cameras must be pointed in the direction of a given object in order to provide sufficient resolution.

To accurately track an object, we need to solve both perceptual and control problems:

- The perceptual problem deals with the estimation of the location of a target object. We estimate the 3-D position of the object based on kinematics of the robot head and wide-angle camera image information.
- The control problem addresses the control of the head. It assures that an object is kept in the center of the narrow- or wide-angle cameras using the estimated target motion. To solve the control problem, we propose the concept of a virtual mechanism. The concept of a virtual mechanism is the main contribution of the paper and originates from this paper. The virtual mechanism is an auxiliary mechanism that points from both robot eyes to a 3-D point in space. It enables us to properly define a task (the gaze direction) and solve the control problem more systematically. The idea of introducing additional (virtual) degrees of freedom as such has been applied also in other contexts, e.g., in Ref. [20], where the authors used additional degrees of freedom to describe a manipulation task.

A robotic head is an example of a mechanism that has some degrees of freedom in one common kinematic chain (e.g., the neck), while some of degrees of freedom are specific to each branch of a system (e.g., each eye pan). This kind of systems can be treated as a branching mechanism [21]. The most general way to control a branching mechanism is to treat both branches equally so that the movement of the common degrees of freedom is defined by the tasks of both subbranches equally, while the motion of a particular branch only depends on a particular task. In other cases, one of the branches can be dominant and can define the motion entirely, while the motion of the other branch adapts to this motion. In our paper both branches have the same priority, which is also the case in humans. Additionally, the symmetry of the head indicates the same priority of the subbranches.

A humanoid head is redundant with respect to the task of fixating on an object. A redundant manipulator has more degrees of freedom than is required to solve the task. A redundant manipulator is more dexterous than a nonredundant manipulator [22] and has the ability to move in the joint space without affecting the motion in the task space. Therefore, a redundant manipulator can execute a given task (called the primary task) together with an additional, less important subtask (called the secondary task). For example, a redundant manipulator can track a trajectory while avoiding obstacles [23] or singularities, optimizing joint torques [24], or optimizing various performance criteria (e.g., manipulability) [25]. The approach proposed in this paper will not only solve the redundancy by applying additional constraints as proposed in Refs [10, 15, 16] but it will also exploit the robot's redundancy in such a way that the robot can perform additional tasks that are typical for humans.

There are many reasons for exploiting the redundancy of the robot head. For example, when a robot neck is close to the joint limit the robot can use other joints (e.g., eyes) to accomplish the task (see joint limit avoidance in Ref. [26]). We can optimize manipulability [25] and consequently achieve shorter reaction time when

an object suddenly moves. Similarly, we can optimize joint torques [24] and achieve lower energy consumption (or low fatigue in case of a human). Or we can apply different voluntary head motions such as nodding and still assure a stable gaze on the object.

In the paper we often refer to a human body and suggest human-like behavior. However, imitating human motion with a humanoid robot is not always the most optimal solution because the mechanical structure and the joint actuators of a mechanical robot are not exactly the same as on the human body. To support interaction and communication with people, it is often preferable that a humanoid robot performs human-like motion. In this paper we show that our approach is suitable for the generation of such movements. On the other hand, our method is general and supports also the generation of head movements that are not human-like if this is what the task requires.

The paper is organized as follows. First, Section 2.1 indicates how to solve the perceptual problem, i.e., how to acquire the 3-D position of a point in space using stereo vision on an active humanoid head. Next, Section 2.2 addresses the control problem, i.e., how to move the head and eyes in order to keep the object in the center of narrow-angle images, even though it is perceived by the wide-angle cameras. Here, we introduce the virtual mechanism, which simplifies and systematizes the description of the tracking task. In Section 3 we present experimental results that show how to apply the proposed approach to implement object fixation using 3-D vision, proving that the system is accurate enough to realize foveated vision. Finally, we demonstrate that the system can solve secondary tasks by exploiting the redundancy of the mechanism while fixating on the object.

2. Methods

To evaluate our approach we carried out several experiments with a humanoid head (Fig. 1), which is similar to the head used on the *Armar III* humanoid robot [13]. The head has 7 mechanical d.o.f. and two eyes. The eyes have a common tilt and can pan independently. The visual system is mounted on a 4-d.o.f. neck, which is realized as a pitch–roll–yaw–pitch mechanism. Each eye is equipped with two digital color cameras (wide- and narrow-angle) to allow visuomotor behaviors such as tracking and saccadic motions towards salient regions, as well as more complex tasks such as hand–eye coordination. The head features human-like characteristics in motion and response, i.e., the neck and the eyes have a human-like speed and range of motion.

In this work we address the control problem of how to track an object in 3-D space with the narrow-angle cameras. A 3-D point in the robot viewfield is acquired by stereo vision using wide-angle cameras. The control problem is solved using the 3-D position of an object that needs to be tracked. Therefore, the perception has to provide 3-D information, which is not a simple task on an active vision system.

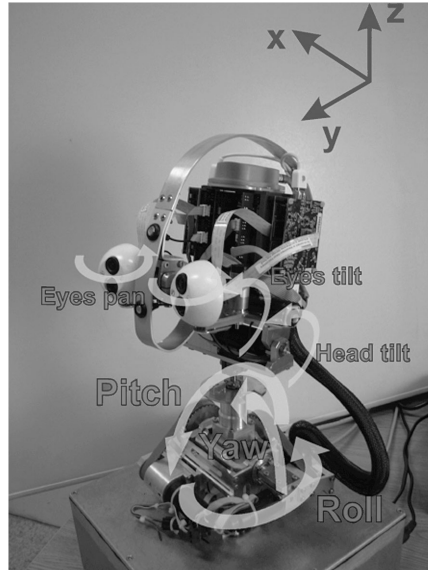


Figure 1. Humanoid robot head.

2.1. Estimating 3-D Position of an Object (the Perceptual Problem)

The perceptual problem can be solved using two images from the wide-angle stereo cameras. A good intrinsic model of the camera optics and of the head kinematics (extrinsic camera model) is crucial for this purpose. This section only indicates how to solve this demanding problem.

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. It is well known how to calibrate a static stereo camera at a specific configuration [27]. Here, we explain how to estimate the transformation between the camera and motor coordinate systems, and describe how to realize 3-D vision when the eyes and the rest of the body move.

2.1.1. Acquiring Eye–Camera Transformation

It is very difficult to mount the cameras on the head so that the internal camera coordinate system is 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 (Fig. 2). We denote this transformation by $\mathbf{T}_{ec}^{l \text{ or } r}$. To estimate it, a calibration object is placed at a fixed location in front of the robot and the robot moves its eyes to a number of known orientations. The poses of the calibration object are estimated at all these configurations using the method described in Ref. [27]. Let \mathbf{T}_{co}^j and \mathbf{T}_{er}^j , $j = 0, \dots, n$, 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). Here, we drop the index l or r for the sake of simplicity. \mathbf{T}_{er}^j can

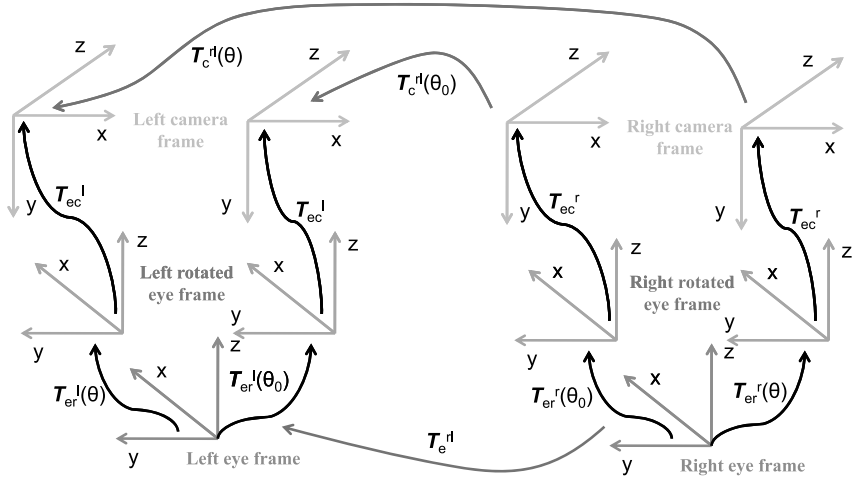


Figure 2. Coordinate systems that need to be accounted for to realize 3-D vision on an active humanoid robot head.

be easily computed using the joint angles obtained by robot joint sensors and the kinematics of the eye’s motor system. On our head, we use the eye pan and tilt degree of freedom to generate these eye configurations. The presented approach is, however, more general and does not make this specific assumption.

There exists the following relationship for each j , $j = 1, \dots, n$:

$$\mathbf{T}_{co}^0(\mathbf{T}_{co}^j)^{-1} = \mathbf{T}_{ec}^{-1}(\mathbf{T}_{er}^0)^{-1}\mathbf{T}_{er}^j\mathbf{T}_{ec}. \tag{1}$$

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

$$\mathbf{A}_j\mathbf{X} = \mathbf{X}\mathbf{B}_j, \tag{2}$$

where \mathbf{A}_j , \mathbf{B}_j , $\mathbf{X} \in \text{SE}(3)$. $\text{SE}(3)$ is the special Euclidean group of rigid-body transformations. This equation often arises in problems associated with sensor-robot calibration and can be solved for $\mathbf{X} = \mathbf{T}_{ec}$ analytically for the left or right eye [28, 29].

2.1.2. Active Stereo Vision

The transformation $\mathbf{T}_c^{rl}(\theta)$ between the left and right eye is needed to calculate the 3-D position of an observed point. However, this transformation is not constant on an active system, but depends on the eye joint angles θ (Fig. 2). It therefore needs to be estimated as the eyes move. This can be accomplished by utilizing the results of the static camera calibration process, the eye–camera transformation of Section 2.1.1 and by making use of the known eye kinematics. Let $\mathbf{T}_c^{rl}(\theta_0)$ be the transformation from the right to left camera at joint configuration θ_0 , which is estimated by the static stereo camera calibration, and let $\mathbf{T}_{er}^l(\theta_0)$ and $\mathbf{T}_{er}^r(\theta_0)$, respectively, be the configuration of the left and right eye at this configuration. As before let \mathbf{T}_{ec}^l and \mathbf{T}_{ec}^r be the transformations between the left and right eye and

camera, respectively. They can be estimated by the calibration process of Section 2.1.1 and do not depend on the eyes' joint angles. By $\mathbf{T}_{er}^l(\theta)$ and $\mathbf{T}_{er}^r(\theta)$ we denote the current eye postures at joint angles θ . $\mathbf{T}_{er}^l(\theta_0)$, $\mathbf{T}_{er}^r(\theta_0)$, $\mathbf{T}_{ec}^l(\theta)$ and $\mathbf{T}_{ec}^r(\theta)$ can be computed using the known eye kinematics and proprioception.

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

$$\mathbf{T}_e^r = \mathbf{T}_{er}^r(\theta_0)\mathbf{T}_{ec}^r\mathbf{T}_c^r(\theta_0)(\mathbf{T}_{ec}^l)^{-1}\mathbf{T}_{er}^l(\theta_0)^{-1}. \quad (3)$$

\mathbf{T}_e^r 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. 2):

$$\mathbf{y}_r = \mathbf{T}_c^r(\theta)\mathbf{y}_l = (\mathbf{T}_{ec}^r)^{-1}(\mathbf{T}_{er}^r(\theta))^{-1}\mathbf{T}_e^r\mathbf{T}_{er}^l(\theta)\mathbf{T}_{ec}^l\mathbf{y}_l. \quad (4)$$

The above transformation allows us to calculate the 3-D point coordinates \mathbf{y}_l in the rotated left camera coordinate system using standard stereo triangulation. Finally, the following transformation can be applied to compute the position in the robot body coordinates

$$\mathbf{y}_b = \mathbf{T}_{be}^l(\theta)\mathbf{T}_{er}^l(\theta)\mathbf{T}_{ec}^l\mathbf{y}_l, \quad (5)$$

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

We note that the accuracy of 3-D estimation crucially depends on the accuracy of the estimation of joint angles θ and on the accuracy of the head kinematic model. In the following we shall show that pursuit of objects is possible even when these parameters are not known with high accuracy. For more details regarding active camera calibration, see Ref. [30].

2.2. The Control Problem

The task of the robot head is to keep an object in the center of both narrow-angle camera images. The head has to assure proper gaze direction of both eyes (cameras). Therefore, the task has 4 d.o.f., since the gaze direction of each eye is defined by two parameters (by two angles). Gaze direction is a function of a 3-D point in space as well as the position of the eyes (Fig. 3). When the head is moved, the position of the eyes changes and that affects the gaze direction. Thus, the task is a function of a point in space as well as a function of the head configuration. To solve this problem in a more systematic way, we propose the use of a virtual mechanism approach.

2.2.1. Virtual Mechanism Approach

The main purpose of the virtual mechanism is the systematization of the task definition. That allows us to use the well-known control approaches to control the robot head.

Let us explain the virtual mechanism on a simple illustrative planar case. The task of the robot head is to keep the object in the center of the camera image.

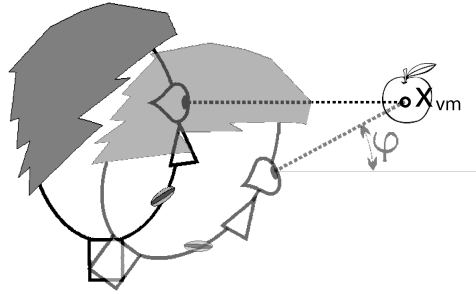


Figure 3. Gaze direction φ changes during head movement. The end of the virtual mechanism touches the object.

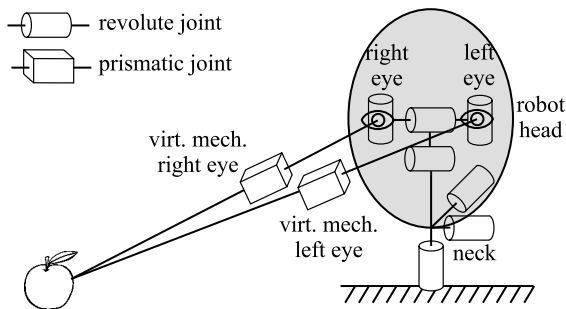


Figure 4. Schematics of a humanoid head enhanced with virtual mechanisms.

Figure 3 shows an example where the eyes are turned toward the object. When the head moves, the eyes have to change their orientation (φ) in order to keep the gaze on the object. Thus, the robot task can be defined as the angle of the eye φ , and is a function of the object position and also camera (eye) position. The position of the camera depends on the head configuration, which means that the head configuration is involved in the task definition:

$$\mathbf{task}_1 \text{ d.o.f.} = \varphi = \mathbf{f}(\text{object pos.}, \text{head conf.}).$$

The above specification of the problem is not the most common way to describe a task in robotics — in general, a task is not a function of the robot configuration. Therefore, it is very complex to implement well-known control approaches to control the robot head. To solve this problem in a more systematic way, we propose the use of a virtual mechanism. The purpose of the virtual mechanism approach is to define the task in such a way that it is not configuration dependent.

Let us expand our humanoid head mechanism with an additional virtual link (mechanism) in the eye. This virtual link can be treated as an additional prismatic joint and is fixed to the eye (see schematics in Fig. 4). When the eye moves the virtual link also moves. By adding the virtual link we add an additional degree of freedom to the system. The extra degree of freedom is the length of the virtual mechanism.

The task can now be formulated differently. Instead of controlling the view angle, we can control the position of the end of the virtual link x_{vm} (see end of virtual link x_{vm} in Fig. 3). By adding the virtual prismatic link to the eye, we require that the end of the eye's virtual extension touches the object that is to be tracked. Contrary to the previous case, the task can now be defined as a simple positioning problem (inverse kinematics) and is not a function of the head position:

$$\mathbf{task}_2 \text{ d.o.f.s} = \mathbf{x}_{vm} = \mathbf{f}(\text{object pos.}).$$

To achieve the proper gaze direction the head still needs to move in the same way as before, regardless of the added virtual mechanism. The only difference is the new task formalization, i.e., the task is defined more naturally and many of the standard robotics techniques are applicable.

In the original task definition we used 1 d.o.f. (φ) to describe the task. By introducing the virtual mechanism we add 1 additional d.o.f. to the system (the length of the virtual mechanism); however, we also add 1 d.o.f. to the description of the task (2 d.o.f. for the \mathbf{x}_{vm} position in the plane). Thus, the degree of redundancy remains the same.

In the spatial 3-D case the circumstances are similar. In this case the task of the robot head is to control the orientation of both cameras and point them directly toward the object. Here, in the original task description the task has 4 d.o.f. (two camera angles per eye) and the robot has 7 d.o.f. After the introduction of two virtual mechanisms (one per eye) the number of degrees of freedom of the robot is increased by 2; however, the degree of the task is also increased by 2. The task of positioning requires 3 d.o.f. per each eye. The introduction of the virtual mechanism increases the number of degrees of freedom of the mechanism as well as of the task.

Let us summarize the most essential characteristics of the virtual mechanism once again. In both task descriptions the goal is to keep the object in the center of the camera images. In both descriptions the angles of the eyes have to assure proper gaze direction. However, in the original description the task is defined as the viewing angles (which are configuration dependent). In contrast, in the virtual mechanism description the task is defined as the position of the object (which is not configuration dependent). Now, instead of specifying the desired pointing direction (angles), we can consider the problem as a classic inverse kinematics task. Nevertheless, regardless of the task description the final head configuration (or camera angles) is the same in both descriptions (the head always has to be directed toward the object). The benefit of the virtual mechanism is the systematization of the task description and this brings us the essential simplification in the controller design. Consequently, we can apply optimal robotics methodologies to improve the tracking results and can exploit the redundancy of the mechanism.

2.2.2. Controller

By having an additional virtual mechanism in each eye, the kinematics of the head is given in the following form:

$$\mathbf{x}_{vm} = \mathbf{f}(\mathbf{q}_{\text{head}}, l_{\text{virt.m.}}), \quad (6)$$

where \mathbf{x}_{vm} denotes the position of the end of the virtual link, while \mathbf{q}_{head} and $l_{virt.m.}$ denote the head joint angles and the lengths of the virtual mechanisms. To simplify the notation we treat the length of the virtual mechanisms as an additional joint variable, such that $\mathbf{q} = [\mathbf{q}_{head}, l_{virt.m.}]$. The relationship between joint and task velocities is given by the robot Jacobian \mathbf{J} :

$$\dot{\mathbf{x}}_{vm} = \mathbf{J}\dot{\mathbf{q}}. \quad (7)$$

As already stated, the head has more degrees of freedom than needed to accomplish the given task. To achieve a good tracking performance while exploiting the redundancy, the following velocity controller can be applied:

$$\dot{\mathbf{q}}_c = \mathbf{J}^\# \dot{\mathbf{x}}_{vmc} + \mathbf{N}\dot{\mathbf{q}}_n, \quad (8)$$

where $\dot{\mathbf{q}}_c$ denotes the vector of joint velocities, $\mathbf{J}^\#$ is the weighted generalized inverse of the Jacobian matrix \mathbf{J} , $\dot{\mathbf{x}}_{vmc}$ is the desired velocity in the task space, \mathbf{N} is the projection onto the null space of \mathbf{J} , and $\dot{\mathbf{q}}_n$ is the desired joint velocity in the null space. The product $\mathbf{J}^\# \dot{\mathbf{x}}_{vmc}$ represents the joint velocities due to the task space motion and $\mathbf{N}\dot{\mathbf{q}}_n$ represents the joint velocities of the null space motion. $\mathbf{J}^\#$ and \mathbf{N} can be calculated as:

$$\mathbf{J}^\# = \mathbf{W}^{-1} \mathbf{J}^T (\mathbf{J} \mathbf{W}^{-1} \mathbf{J}^T)^{-1}, \quad \mathbf{N} = \mathbf{I} - \mathbf{J}^\# \mathbf{J}, \quad (9)$$

where \mathbf{W} is the weighting matrix. By using such a definition of the generalized inverse the weighted joint velocity is minimized. To control the position of the virtual link, the following $\dot{\mathbf{x}}_{vmc}$ controller is proposed:

$$\dot{\mathbf{x}}_{vmc} = \dot{\mathbf{r}} + K_p \mathbf{e},$$

where \mathbf{e} , $\mathbf{e} = \mathbf{r} - \mathbf{x}_{vm}$, is the task space tracking error and \mathbf{r} is the desired task space position, i.e., the position of the object that has to be tracked.

In the case of the robot head, the controller is not so simple as shown above. The topology of the robot head is a tree-like structure with one main branch and two subbranches (Fig. 4). The main branch refers to the joints and links that are common to both parts of the system. Those are the neck joints and the eye tilt. On top of that, there are joints that are specific to each subbranch of the system. Those are the eye pans and the lengths of the virtual mechanisms.

When controlling a branching mechanism we have two options. The first option is to treat one of the subbranches as the dominant one. In this case the task of the dominant branch defines the motion of it and all the common joints, while the less important branch only adapts to that motion. The second and more general option is to treat all (both) subbranches equally, so that the movement of the common degrees of freedom is defined by the tasks of both subbranches equally. Similarly, the movement of a particular branch only depends on the particular subtask [21].

It is typical for humans that they fixate on an object with both eyes and none of them is dominant. Both eyes as well as the rest of the head work together in order to achieve a comfortable gaze on the object. Additionally, the symmetry of the head

indicates the same priority of the subbranches. Therefore, the robot system should be treated as a branching kinematic tree with the same priority of the subbranches.

2.2.3. *Branching Mechanism (Kinematic Tree)*

The proposed approach for controlling the head and eyes is very similar to a classic branching robot with a common base whose task is to touch the same point in space with both arms. An efficient and very systematic approach to control a branching mechanism was proposed in Ref. [21].

We express the relationship between the joint and task velocities for the left $(\cdot)_L$ and right $(\cdot)_R$ eye as shown in (7):

$$\dot{\mathbf{x}}_{vmL} = \mathbf{J}_L \dot{\mathbf{q}}_{all}, \tag{10}$$

and:

$$\dot{\mathbf{x}}_{vmR} = \mathbf{J}_R \dot{\mathbf{q}}_{all}, \tag{11}$$

where $\dot{\mathbf{q}}_{all}$ is the vector of joint velocities of all head joints together with virtual joints as shown here:

$$\dot{\mathbf{q}}_{all} = \left[\begin{array}{c} \dot{q}_{roll} \\ \dot{q}_{pitch} \\ \dot{q}_{yaw} \\ \dot{q}_{neck\ tilt} \\ \dot{q}_{eyes\ tilt} \\ \hline \dot{q}_{left\ eye\ pan} \\ \dot{q}_{left\ eye\ virt.\ m.\ dist.} \\ \hline \dot{q}_{right\ eye\ pan} \\ \dot{q}_{right\ eye\ virt.\ m.\ dist.} \end{array} \right] \left. \begin{array}{l} \} \\ \} \\ \} \\ \} \\ \} \\ \} \\ \} \\ \} \\ \} \\ \} \end{array} \right\} \begin{array}{l} \text{common joints} \\ \text{left branch joints} \\ \text{right branch joints.} \end{array}$$

The Jacobian matrices \mathbf{J}_L and \mathbf{J}_R for the left and right branches have zero elements for the joints that do not contribute to the motion of the specific branch:

$$\mathbf{J}_L = \begin{bmatrix} j_{11} & j_{12} & j_{13} & j_{14} & j_{15} & j_{16} & j_{17} & 0 & 0 \\ j_{21} & j_{22} & j_{23} & j_{24} & j_{25} & j_{26} & j_{27} & 0 & 0 \\ j_{31} & j_{32} & j_{33} & j_{34} & j_{35} & j_{36} & j_{37} & 0 & 0 \end{bmatrix},$$

and:

$$\mathbf{J}_R = \begin{bmatrix} j_{11} & j_{12} & j_{13} & j_{14} & j_{15} & 0 & 0 & j_{18} & j_{19} \\ j_{21} & j_{22} & j_{23} & j_{24} & j_{25} & 0 & 0 & j_{28} & j_{29} \\ j_{31} & j_{32} & j_{33} & j_{34} & j_{35} & 0 & 0 & j_{38} & j_{39} \end{bmatrix}.$$

The first five columns indicate that the motion of the first five joints contributes to both left and right branches, while the last four columns indicate contribution only to the particular branch.

The task of the head is to control the position of the virtual mechanisms of both left and right branches simultaneously. To achieve this let us join the relationship of the left and right branches of (10) and (11) in the matrix form:

$$\begin{bmatrix} \dot{\mathbf{x}}_{\text{vmL}} \\ \dot{\mathbf{x}}_{\text{vmR}} \end{bmatrix} = \begin{bmatrix} \mathbf{J}_L \\ \mathbf{J}_R \end{bmatrix} \dot{\mathbf{q}}_{\text{all}}. \quad (12)$$

The joined Jacobian matrix $\begin{bmatrix} \mathbf{J}_L \\ \mathbf{J}_R \end{bmatrix}$ is not a square matrix. The size of the matrix is 6×9 , which indicates that the degree of redundancy is 3. By following the suggestions pointed out in Section 2.2.2, the following controller is proposed:

$$\dot{\mathbf{q}}_{\text{all}_c} = \begin{bmatrix} \mathbf{J}_L \\ \mathbf{J}_R \end{bmatrix}^\# \begin{bmatrix} \dot{\mathbf{x}}_{\text{vmL}_c} \\ \dot{\mathbf{x}}_{\text{vmR}_c} \end{bmatrix} + \mathbf{N}_{LR} \dot{\mathbf{q}}_{\text{all}_n},$$

where the desired motion in the task space is defined for each branch individually:

$$\begin{bmatrix} \dot{\mathbf{x}}_{\text{vmL}_c} \\ \dot{\mathbf{x}}_{\text{vmR}_c} \end{bmatrix} = \begin{bmatrix} \dot{\mathbf{r}}_L + K_p \mathbf{e}_L \\ \dot{\mathbf{r}}_R + K_p \mathbf{e}_R \end{bmatrix},$$

where $\mathbf{e}_L = \mathbf{r}_L - \mathbf{x}_{\text{vmL}}$ and $\mathbf{e}_R = \mathbf{r}_R - \mathbf{x}_{\text{vmR}}$. \mathbf{r}_L and \mathbf{r}_R are the reference position of the left and the right branch, respectively. The task is to keep the gaze of both eyes on the object, i.e., the end-points of both branches have to be at the position of the tracked object. The 3-D position of the tracked object is estimated by the stereo cameras as explained in Section 2.1. The target position for both branches is, thus, the 3-D position of the object:

$$\mathbf{r}_R = \mathbf{r}_L = \begin{bmatrix} X_{\text{obj}} \\ Y_{\text{obj}} \\ Z_{\text{obj}} \end{bmatrix}.$$

2.2.4. Accuracy of the Kinematic Model

To accurately define the 3-D object position it is crucial to have a good intrinsic model of the cameras as well as of the head kinematics (extrinsic camera model). The accuracy of the head kinematics can be quite low on a lightweight humanoid robot head and that raises doubt about the success of the proposed algorithm. In the following we show that the kinematic model accuracy does not play an important role in the object tracking accuracy; however, it remains important for the accuracy of 3-D position estimation.

The position of the object is estimated using the kinematic model of the head. The same kinematic model is used to define positions of the ends of the virtual links (\mathbf{x}_{vmL} , \mathbf{x}_{vmR}). If the kinematics of the head is inaccurate, the estimated 3-D object position as well as the positions of the virtual links are inaccurate. Fortunately, these two errors cancel themselves out in the controller. Despite the low accuracy of the kinematic model, the accuracy of the object tracking is still very high, as shown in Section 3.1.

The accuracy of the kinematics also affects the depth information. Estimation of depth is quite noisy when using stereo vision; however, these disturbances produce very small angle changes in the eyes and do not affect the motion of the head significantly.

2.3. Exploiting Redundancy

A redundant manipulator can solve a less important secondary task without affecting the motion of a more important primary task. It is therefore beneficial to exploit the redundancy instead of only solving it. The robot has to be able to move the head while keeping eye contact with an object. Since the proposed virtual mechanism approach is very systematic we can use a great number of control approaches and optimization strategies proposed in the literature [22].

To control the system we have used the controller given in (8). Here, $\dot{\mathbf{q}}_n$ is the desired joint velocity in the null space that does not affect the task space motion. The task is defined as a positioning task of the ‘end-effectors’ of the virtual mechanisms. If we pre-multiply (8) with the robot Jacobian \mathbf{J} we obtain the following equation:

$$\mathbf{J}\dot{\mathbf{q}}_c = \mathbf{J}\mathbf{J}^\# \dot{\mathbf{x}}_{vmc} + \mathbf{J}\mathbf{N}\dot{\mathbf{q}}_n.$$

Using (9) we can see that $\mathbf{J}\mathbf{J}^\# = \mathbf{I}$ and $\mathbf{J}\mathbf{N} = \mathbf{0}$. It yields:

$$\dot{\mathbf{x}}_{vmc} = \mathbf{J}\dot{\mathbf{q}}_c = \dot{\mathbf{x}}_{vmc} + \mathbf{0}\dot{\mathbf{q}}_n.$$

It is obvious that the null space velocity $\dot{\mathbf{q}}_n$ does not affect the primary task $\dot{\mathbf{x}}_{vmc}$ since it is multiplied by 0.

3. Results

The proposed approach has been verified on the real humanoid head described in Section 2 and shown in Fig. 1. We defined the kinematic model of the head including the virtual mechanism using Denavit–Hartenberg notation. The model of the kinematic structure of the head is quite accurate because the head has been manufactured using CAD technology. The initialization process, however, defines the joint angle offsets and is much less accurate, which makes the complete kinematic model inaccurate.

Each eye of the robot has two cameras where the wide-angle camera is placed above the narrow-angle one. The vertical distance between them is 18 mm and the resolution of all cameras is 640×480 . The intrinsic camera parameters and transformations between cameras were calculated on a set of chess board snapshots. The robot head was controlled with MATLAB/Simulink *via* UDP connection. Sample time of the control loop was set to 100 Hz, while the frame rate of the visual processing was 60 Hz. For estimating the object position in camera images we used color segmentation software.

The weighting matrix \mathbf{W} used in (9) is a diagonal matrix with elements [2 0.5 0.5 0.1 0.1 0.05 0.00001 0.05 0.00001]. Here, higher values in the matrix result in less

motion of the corresponding joint, due to the fact that the weighted joint velocity is minimized. In our case that means that the neck joints move less while the lighter and faster eye joints move more. The null space term in the robot controller has been defined in such a way that all the joints are attracted to their initial position as follows: $\dot{\mathbf{q}}_n = -K_n(\mathbf{q} - \mathbf{q}_0)$. By using the suggested weighting matrix and the null space velocity we achieve very natural and human-like motion.

We performed a series of experiments in which a human demonstrator holds a colored object in his hand and moves it in front of the humanoid head. We compared the use of different kinematic models (accurate and inaccurate) in order to show that the accuracy does not play a major role in the tracking performance. In the following we demonstrate the tracking of the narrow-angle cameras based on the wide-angle images. These tests have been performed at different distances from the object to demonstrate robustness against changes in distance. The section is concluded with an experiment that involves exploiting the redundancy of the head.

3.1. Comparing Different Kinematic Models

In general, obtaining the 3-D position of an object by active stereo vision is not very accurate. Therefore, a more robust controller and better results are expected using algorithms that perform tracking directly in 2-D images [19] compared to tracking in 3-D space.

That is true to some extent. However, the approach presented in this paper is to a great extent insensitive to the uncertainties in the kinematic model. The reason is that the same kinematic model is used to obtain the 3-D position from two 2-D images and to control the gaze direction. Both estimation processes produce systematic error due to the uncertainties in the kinematic model. It turns out, however, that these two errors cancel each other out and the gaze direction is still accurate.

In the experiments a human demonstrator moved the object as shown in Table 1. The movements between the four positions, which were approximately 30 cm apart, took approximately 1 s. The distance from the robot eyes and the object was approximately 50 cm. In the experiments shown here both the 3-D position estimation and the control (tracking) were realized using wide-angle cameras.

3.1.1. Experiments with the Accurate Kinematic Model

The first experiment was performed using the most accurate model. The kinematic model parameters were taken from the CAD model of the head and the cameras

Table 1.

Time (s)	Object position
0–10	in front of the head
10–20	on the right side of the head
20–30	on the left side of the head
30–40	in front of the head

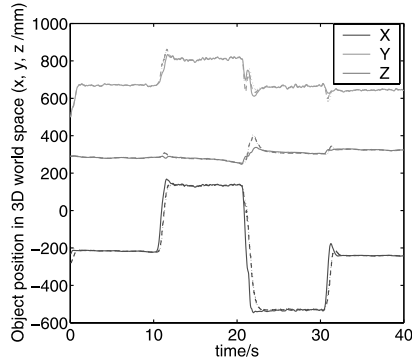


Figure 5. Position of the object in 3-D space: accurate kinematic model.

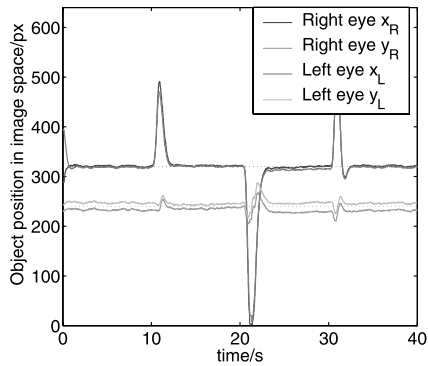


Figure 6. Position of the object in left and right images — accurate kinematic model.

were carefully calibrated. The demonstrator moved an object in front of the head as instructed in Table 1. The solid lines of Fig. 5 show the position of the object in 3-D world space. The positions were estimated using two wide-angle camera images using (5).

Figure 5 shows the position of the object as well as the positions of the left and right virtual link (dotted and dashed line). The reference and virtual link positions are nearly the same, which indicates good gaze tracking performance in 3-D space.

The main task is to keep the object in the center of the images, which means that the more relevant measure of the tracking performance is in the image space rather than in 3-D space. Figure 6 shows the position of the object in the left and right image. Since the camera resolution is 640×480 , the center of the image is at (320, 240) pixels. Figure 6 shows that the tracking in both images is very good.

3.1.2. Experiments with an Inaccurate Kinematic Model

The kinematic model used in the previous experiments was quite accurate. However, it is very difficult to have a good kinematic model using an active, badly calibrated humanoid head. In order to demonstrate robustness against the inaccuracies in the kinematic model, we added an error to the head kinematics. The structure

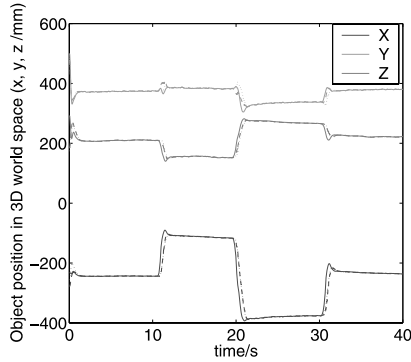


Figure 7. Position of the object in 3-D space — inaccurate kinematic model.

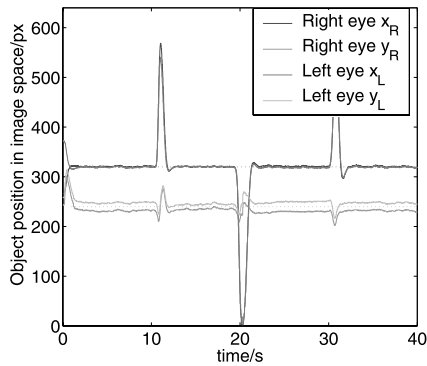


Figure 8. Position of the object in left and right images — inaccurate kinematic model.

of the kinematic model remained the same, while the joint offsets were modified. The following, arbitrary selected angular offsets were added to all head joints: $[10^\circ, 20^\circ, -10^\circ, 30^\circ, -20^\circ, 5^\circ, -5^\circ]$. These joint offsets are quite large. It is expected that the kinematic model of the head can always be better than the one used here.

Figure 7 shows the estimated movement of the object in 3-D space. Although the movement was similar to the movement in the first experiment, the acquired 3-D positions differ significantly. The reason for that is the modified kinematics. Due to the modified kinematics the camera-extrinsic parameters are not estimated correctly and consequently the acquired positions are very inaccurate. However, since the same kinematics are used in the control part, the 3-D tracking still performed very well, as shown with dashed and dotted lines in Fig. 7.

Despite the huge uncertainties in the kinematic model, the tracking in the image space still performs very well as shown in Fig. 8. During tracking the head movement looked a bit less natural because the head does not move directly towards the object; however, it achieves the goal in a reasonable short time.

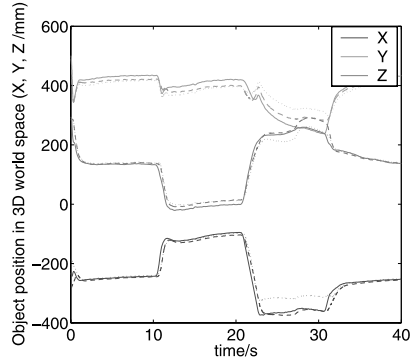


Figure 9. Position of the object in 3-D space — different kinematic models.

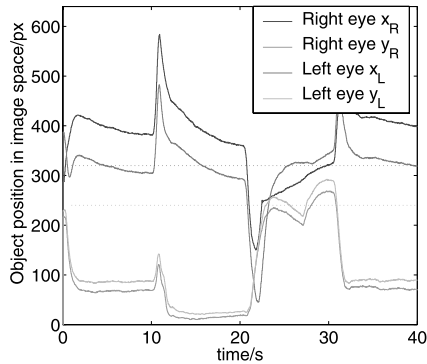


Figure 10. Position of the object in left and right images — different kinematic models.

3.1.3. Experiments with Different Kinematic Models in the Perceptual and Control Part

In the experiments described before the kinematic model was the same in both parts of the system. In this section different kinematic models were used in both parts. In the control part we used the accurate kinematic model while in the perceptual part the modified model was applied. Thus, the kinematic models in the perceptual and control parts differ.

Figures 9 and 10 show the tracking in the 3-D and image space. As we can see the tracking in image space is very bad. This is due to the fact that different models were used in both parts of the system. In this case the perceptual error does not cancel out the control error.

3.2. Object Tracking with Narrow-Angle Cameras

Many visual tasks require both high resolution and a wide field of view. High acuity is needed for recognition tasks and a wide field of view is needed for search tasks. As explained in Section 1, our system models the foveated structure of biological vision systems by having two cameras in each eye, i.e., a narrow-angle foveal cam-



Figure 11. Snapshots of wide- and narrow-angle cameras. (a) Wide-angle camera image. (b) Narrow-angle camera image.

era and a wide-angle camera for peripheral vision. Snapshots from both cameras are shown in Fig. 11.

The object position was acquired using information from wide-angle cameras, while the object was tracked by the narrow-angle cameras. Since both cameras are rigidly connected, the transformation from the wide-angle to narrow-angle coordinate frame is constant, and was acquired in the cameras calibration process.

Ude *et al.* [19] derived a relationship between the 2-D position of a point in the wide- and narrow-angle camera image with respect to the distance of the object from the eye. The difference between 2-D point positions in both images changes drastically when the object is close to the cameras. However, for larger distances the relationship between the 2-D point positions in both cameras can be assumed to be constant with satisfactory accuracy.

There is no problem with unaligned optical axes of the cameras using the approach proposed in the paper. In the perceptual phase the 3-D position of an object is estimated based on the wide-angle camera images and the kinematics of the wide-angle cameras. In the control phase, where the object tracking has to be performed, the kinematics of the narrow-angle cameras has to be used, since they are used for tracking.

To demonstrate the tracking ability of the narrow-angle cameras, the demonstrator was instructed to move an object on a rectangular trajectory in a plane as shown in Table 2. Figure 12 shows the movement of the object in 3-D space, where the object moved at the distance of 30 cm from the eyes.

To demonstrate the robustness against distance of the object from the eyes, we performed two different experiments at two different distances, i.e., 30 and 400 cm from the cameras.

3.2.1. Object is Close to the Cameras

First a demonstrator moved an object on a rectangular trajectory 30 cm away from the cameras. Using the proposed approach, the head was able to fixate on the object

Table 2.

Time (s)	Object position
0–10	in front of the head
10–20	on the right side of the head
20–30	on the right and up side of the head
30–40	on the left and up side of the head
40–50	on the left side of the head
50–60	in front of the head

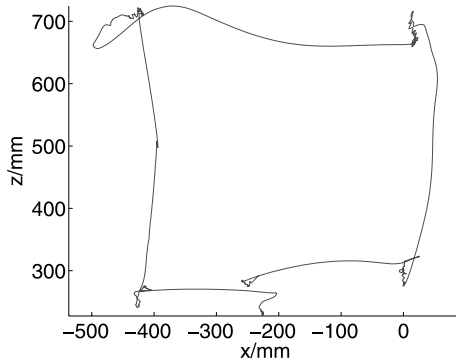


Figure 12. Movement of the object in front of the head.

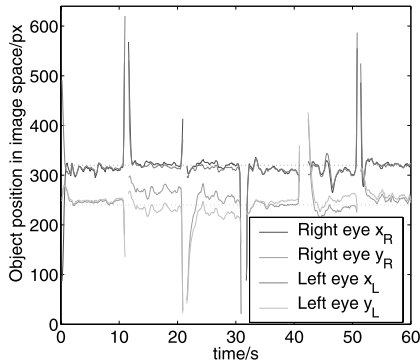


Figure 13. Object position in narrow-angle images — object distance is 30 cm.

in the narrow-angle images (Fig. 13). However, as expected, the object is not in the center of wide-angle images as shown in Fig. 14.

If the kinematic model of the wide-angle cameras was used in the control part, then the object would be in the center of wide-angle images instead of in the center of the narrow-angle images, as can be seen in Figs 15 and 16. It is only possible to achieve an object in the center of both images at one point in space, i.e., at the point where the optical axes of the narrow- and wide-angle cameras intersect. This

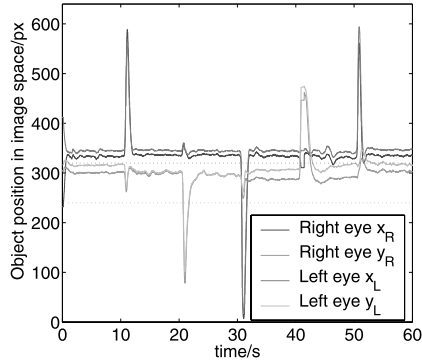


Figure 14. Object position in wide-angle images — object distance is 30 cm.

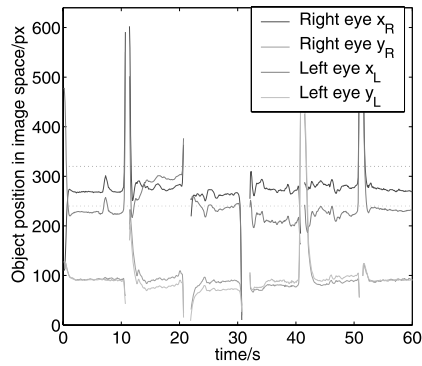


Figure 15. Object position in narrow-angle images using wide-angle camera kinematics — object distance is 30 cm.

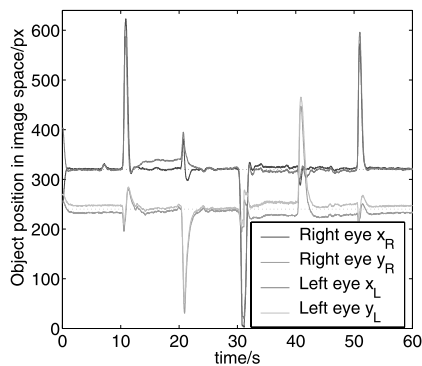


Figure 16. Object position in wide-angle images using wide-angle camera kinematics — object distance is 30 cm.

happens at infinity for systems where the optical axes of narrow- and wide-angle cameras are parallel.

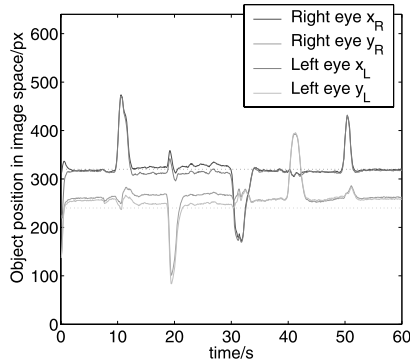


Figure 17. Object position in narrow-angle images — object distance is 400 cm.

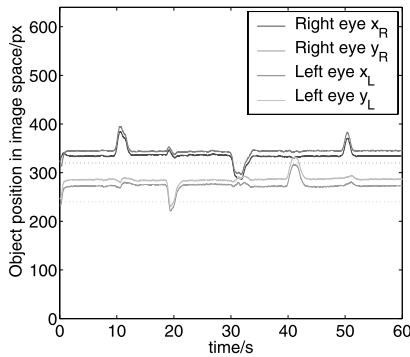


Figure 18. Object position in wide-angle images — object distance is 400 cm.

3.2.2. *Object is Far from the Cameras*

The errors in depth are much larger when an object is far from the cameras. We therefore carried out another experiment in which a human demonstrator moved an object 400 cm away from the cameras. The achieved foveation accuracy, i.e., 2-D position of the object in narrow-angle images, is shown in Fig. 17. The corresponding position in wide-angle images is shown in Fig. 18. This demonstrates that the proposed approach assures that the object is always in the center of the narrow-angle images even when the distance of the object from the camera is changing.

A crucial factor in assuring a good foveation performance is the accuracy of the transformation between the narrow- and wide-angle cameras. As shown in Section 3.1, the accuracy of the rest of the kinematics is not critical because the same kinematic model is used both in the perceptual and in the control part. Here, however, two different models are used: the narrow- and the wide-angle camera models. A good estimation of the static transformation between the two coordinate frames is therefore crucial for good tracking. Since both cameras are rigidly connected, this static transformation can be acquired with a high accuracy.

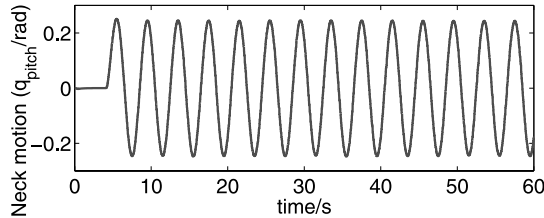


Figure 19. Actual motion in the head neck pitch joint in the case of no redundancy exploitation.

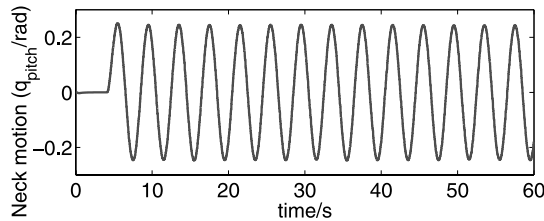


Figure 20. Actual motion in the head neck pitch joint in the case of redundancy exploitation.

3.3. Experiment on Exploiting Redundancy

In order to demonstrate how to exploit the redundancy with the proposed control algorithm, we present a simple example where the head moves forward and backward (nodding) while keeping eye contact with an object. In contrast to the simplified nodding movement, the movement of the head could be a function of a more complicated task. For example, in our other work we used a Nintendo Wii remote (Wiimote) as an input device [31] for the desired neck movement.

In this example the null space motion is defined in such a way that the neck pitch tracks a simple sinusoidal motion:

$$\dot{q}_{n_{pitch}} = K_n(A * \sin(\omega t) - q_{pitch_{actual}}), \quad (13)$$

while the velocities for the other joints are equal to zero: $\dot{q}_{n_i} = 0, \forall i \neq pitch$.

We compared two different controllers. In the first case the neck motion was not compensated and was therefore disturbing the accuracy of object fixation, while in the second case the redundancy was exploited and the neck motion was compensated by the motion of other joints as presented in the paper. Figures 19 and 20 show the motion in the neck pitch for both cases. Figures 21 and 22 show the position of the object in both cameras during the sinusoidal neck movement. When exploiting redundancy the tracking results are significantly better. The neck pitch motion is compensated by the other head joints by exploiting redundancy of the head.

4. Conclusions

We presented an approach for controlling the gaze direction of a humanoid head, which is equipped with two cameras per eye. The proposed approach uses wide-angle cameras to acquire the 3-D position of an object in space. This information

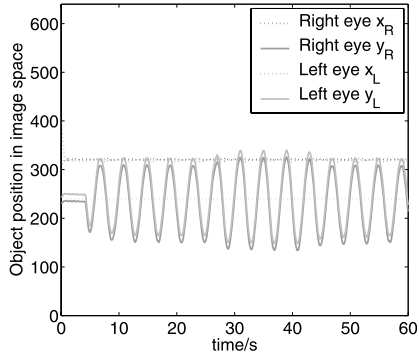


Figure 21. Object position in wide-angle images. Motion is considered as a disturbance.

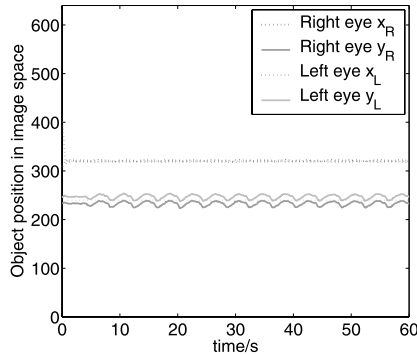


Figure 22. Object position in wide-angle images. The redundancy of the head is exploited and the object is kept in the image center despite the neck motion.

is later used in order to bring and keep an object in the center of the narrow-angle images regardless of the distance of the object from the eyes. To achieve that we introduced a virtual mechanism, which is the main contribution of the paper.

The introduction of the virtual mechanism simplifies the description of the task. This brings us essential simplification in the controller design, and results in better tracking performance and ability to exploit redundancy. Experiments on a real robot head are very promising. Experimental results confirm that the tracking performance is very good, regardless of the distance of the object from the eyes. Additionally, the accuracy of the kinematic model does not play an important role in the accuracy of the tracking. However, it is still crucial for the estimation of the 3-D object position in space.

There are many advantages of the proposed approach over the others described in the literature:

- Compared to the decomposed controllers that control each joint individually without considering the complete head kinematics (e.g., Ref. [19]), the pro-

posed controller results in a more optimal head motion and brings the object to the image center in a more optimal path.

- Getting an object to the center of narrow-angle images even if it is tracked in wide-angle images is simplified and does not depend on the precise placement of the cameras or on the distance of the object from the cameras.
- The proposed approach indicates how to exploit the redundancy of the head instead of only solving it. As mentioned in Section 1, exploiting redundancy has many advantages. Without exploiting redundancy higher tracking error appears when an additional head motion is introduced.
- When using controllers that rely on image-based visual servoing it is more complex to achieve the placement of the desired object in the center of the narrow-angle images, since the object is sometimes visible only in wide-angle images.
- Since the proposed approach is very systematic, it can be easily implemented on different types of active stereo vision mechanisms.

Acknowledgments

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. Breazeal, A. Edsinger, P. Fitzpatrick and B. Scassellati, Active vision for sociable robots, *IEEE Trans. Syst. Man Cybernet. A* **31**, 443–453 (2001).
2. B. Scassellati, Eye finding via face detection for a foveated, active vision system, in: *Proc. 15th AAAI/IAAI Conf.*, Madison, WI, pp. 969–976 (1998).
3. D. J. Povinelli and T. M. Preuss, Theory of mind: evolutionary history of a cognitive specialization, *Trends Neurosci.* **18**, 418–424 (1995).
4. G. M. Burghardt, Cognitive ethology and critical anthropomorphism: a snake with two heads and hognose snakes that play dead, in: *Cognitive Ethology: The Minds of Other Animals: Essays in Honor of Donald R. Griffin*, C. A. Ristau (Ed.). Lawrence Erlbaum, Hillsdale, NJ, pp. 53–90 (1991).
5. C. A. Ristau, Before mindreading: attention, purposes and deception in birds, in: *Natural Theories of Mind: Evolution, Development and Simulation of Everyday Mindreading*, A. Whiten (Ed.), pp. 209–222. Basil Blackwell, Oxford (1991).
6. T. Nummenmaa, The language of the face, *Studies in Education, Psychology and Social Research* **9**, University of Jyväskylä (1964).
7. U. Frith, *Autism: Explaining the Enigma*, 2nd edn. Blackwell, Oxford (2003).
8. S. Thayer, Children's detection of on-face and off-face gazes, *Dev. Psychol.* **13**, 673–674 (1977).
9. R. Carpenter, *Movements of the Eyes*. Pion, London (1988).

10. F. Panerai, G. Metta and G. Sandini, Visuo-inertial stabilization in space-variant binocular systems, *Robotics Autonomous Syst.* **30**, 195–214 (2000).
11. T. Shibata, S. Vijayakumar, J. Conradt and S. Schaal, Biomimetic oculomotor control, *Adapt. Behav.* **9**, 189–207 (2001).
12. J. Yamato, Tracking moving object by stereo vision head with vergence for humanoid robot, Master's Thesis, Department of Electronic Engineering and Computer Science, MIT (1998).
13. 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 Int. Conf. on Humanoid Robots*, Genoa, pp. 169–175 (2006).
14. C. G. Atkeson, J. G. 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 Intell. Syst.* **15**, 46–56 (2000).
15. G. Metta, A. Gasteratos and G. Sandini, Learning to track colored objects with log-polar vision, *Mechatronics* **14**, 989–1006 (2004).
16. A. Bernardino and J. Santos-Victor, Binocular visual tracking: integration of perception and control, *IEEE Trans. Robotics Automat.* **15**, 1080–1094 (1999).
17. A. Bernardino, J. Santos-Victor and G. Sandini, Foveated active tracking with redundant 2D motion parameters, *Robotics Autonomous Syst.* **39**, 205–221 (2002).
18. C.-Y. Tang, Y.-P. Hung and Z. Chen, Automatic detection and tracking of human heads using an active stereo vision system, *Lect. Notes Comp. Sci.* **1351**, 632–639 (1998).
19. A. Ude, C. Gaskett and G. Cheng, Foveated vision systems with two cameras per eye, in: *Proc. IEEE Int. Conf. Robotics and Automation*, Orlando, FL, pp. 2378–2384 (2006).
20. B. Nemeč and L. Žlajpah, Automatic trajectory generation using redundancy resolution scheme based on virtual mechanism, in: *Contemporary Robotics, Challenges and Solutions*, A. D. Rodić (Ed.), Chapter 1. IN-TECH, Vienna, Austria (2009).
21. K.-S. Chang and O. Khatib, Operational space dynamics: Efficient algorithms for modeling and control of branching mechanisms, in: *Proc. Int. Conf. on Robotics and Automation*, San Francisco, CA, pp. 850–856 (2000).
22. D. N. Nenchev, Redundancy resolution through local optimization: a review, *J. Robotic Syst.* **6**, 769–798 (1989).
23. L. Žlajpah and B. Nemeč, Force strategies for on-line obstacle avoidance for redundant manipulators, *Robotica* **21**, 633–644 (2003).
24. J. M. Hollerbach, Redundancy resolution of manipulators through torque optimization, *J. Robotics Automat.* **3**, 308–316 (1987).
25. T. Yoshikawa, Basic optimization methods of redundant manipulators, *Lab. Robotics Automat.* **8**, 49–60 (1996).
26. O. Khatib, Real-time obstacle avoidance for manipulators and mobile robots, *Int. J. Robotics Res.* **5**, 90–98 (1986).
27. Z. Zhang, A flexible new technique for camera calibration, *IEEE Trans. Pattern Anal. Mach. Intell.* **22**, 1330–1334 (2000).
28. 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.* **5**, 16–27 (1989).
29. F. C. Park and B. J. Martin, Robot sensor calibration: solving $\mathbf{AX} = \mathbf{XB}$ on the Euclidean group, *IEEE Trans. Robotics Automat.* **10**, 717–721 (1994).
30. A. Ude and E. Oztop, Active 3-D vision on a humanoid head, in: *Proc. 14th Int. Conf. on Advanced Robotics ICAR*, Munich, pp. 1–6 (2009).

31. A. Gams, A. J. Ijspeert, S. Schaal and J. Lenarčič, On-line learning and modulation of periodic movements with nonlinear dynamical systems, *Autonomous Robots* **27**, 3–23 (2009).

About the Authors



Damir Omrčen received his PhD in Robotics from the University of Ljubljana, Slovenia, in 2005. He is employed as a Research Assistant at the Department of Automation, Biocybernetics and Robotics at the Jozef Stefan Institute in Ljubljana. His fields of interest include vision and robot control where he combines classical model-based approaches and more advanced approaches based on exploration and learning.



Aleš Ude studied Applied Mathematics at the University of Ljubljana, Slovenia, and received his Doctoral degree from the Faculty of Informatics, University of Karlsruhe, Germany. He was awarded the STA Fellowship for Postdoctoral studies in ERATO Kawato Dynamic Brain Project, Japan. He has been a Visiting Researcher at ATR Computational Neuroscience Laboratories, Kyoto, Japan, for a number of years and is still associated with this group. Currently, he is a Senior Researcher at the Department of Automatics, Biocybernetics, and Robotics, Jozef Stefan Institute, Ljubljana, Slovenia. His research focuses on imitation and action learning, perception of human activity, humanoid robot vision, and humanoid cognition.