Online Adaptation of Gaze Fixation for a Stereo-Vergence System with Foveated Vision

Cem Karaoguz, Mark Dunn, Tobias Rodemann, Christian Goerick

2009

Preprint:

This is an accepted article published in International Conference on Advanced Robotics (ICAR). The final authenticated version is available online at: https://doi.org/[DOI not available]

Accepted for ICAR 2009, June 22 – 26, 2009 in Munich, Germany

Online Adaptation of Gaze Fixation for a Stereo-Vergence System with Foveated Vision

Cem Karaoguz, Mark Dunn, Tobias Rodemann and Christian Goerick

***Abstract*— In active vision systems, which direct their gaze to different visual targets in their environment, targets are represented in image coordinates and commands by which the gaze direction is changed are represented in motor coordinates. This requires knowledge of the mapping between two coordinate frames. In this work we present a robust mechanism that learns such a mapping. The mechanism can be applied to any active vision system performing arbitrary gaze shifts and runs online without interfering with any process in the vision system which it is integrated into. We show the feasibility of our approach by simulation and implementation of a stereo vision system with vergence and foveation.**

I. INTRODUCTION

Humans employ binocular vision with foveae and different oculomotor movements that allow the vision system to bring and keep different visual stimuli on the foveae in order to achieve a high resolution view. This allows obtaining the greatest possible amount of information from the fixated stimuli. Two examples of such movements are saccades and vergence. Saccades are rapid movements of the eyes that change the gaze direction to bring the foveae on a new target while vergence is a disconjugate movement of the eyes that brings and keeps the fovea of both eyes on the same visual target along the gaze direction.

Today there is an increasing interest in research of stereo vision systems with vergence and foveation in robotics ([1], [3], [7], [11], [12]). Several methods exist to achieve foveation. We use custom made fish-eye lenses in order to combine high resolution visual data extraction with a large field of view (see fig. 1(b)). Vergence achieves binocular fusion, which leads to a maximization of the overlap between the visual fields of the two cameras. This may facilitate and improve complex visual processes like target-background segmentation, depth and motion estimation, etc.

In active vision, visual sensors have to be directed to areas of interest. The target stimuli are often referenced in image coordinates while the representation of the camera/head orientation is done in motor coordinates. This requires a mapping between two coordinate systems. Such a mapping can be established by calibration, which is a time consuming process. Moreover, changes to the camera system require re-calibration. These changes can be voluntary (e.g. lens change) or involuntary (e.g. motor damage, decalibration) and the latter one is inevitable for systems running over long time periods. For those reasons, a mechanism that learns the mapping between two coordinate systems and corrects it shortly after errors appear is necessary.

All authors are with the Honda Research Institute Europe GmbH, Carl-Legien Str. 30 D-63073 Offenbach/Main, Germany {cem.karaoguz, mark.dunn, tobias.rodemann, christian.goerick}@honda-ri.de

A saccade adaptation scheme for monocular vision systems was previously presented by Rodemann et al. in [10]. We modified this approach to work with stereo vision systems with vergence and foveation. New problems were introduced by these modifications. Section II details how these problems are solved. We also propose a fixation paradigm that consists of coordinated saccade and fine vergence movements. This strategy ensures that every gaze shift ends up with a visual target fixated on the foveae and allows the algorithm to learn with arbitrary gaze shifting movements.

A. Related Work

In [8], Pagel et al. presented similar work where a mapping between a six dimensional input space $(x_l, y_l, x_r, y_r, t, v)$ and a three dimensional motor space $(\Delta p, \Delta t, \Delta v)$ is learned via a growing neural gas network. In contrast to our work, a multiple-saccade strategy (main saccade + corrective saccades) without foveation is used for target fixations.

Rao et al. also presented a similar application for learning monocular saccadic eye movements where multiscale spatial filters are used to construct an iconic representation of the scene [9]. They also adopted a multiple-saccade strategy, which is dependent on a certain target selection process. Foveation is applied in this work by log-polar image sensors.

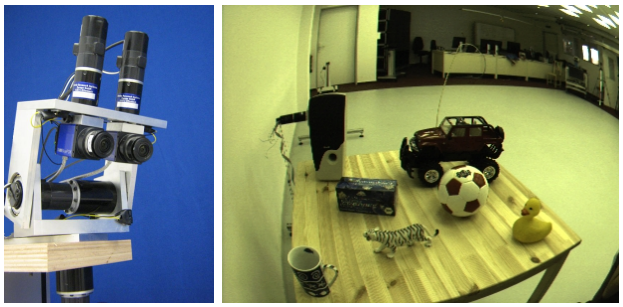
B. Hardware Setup

An experimental stereo vision head (shown in fig. 1(a)) with 4 DoF (2 DoF for head and 1 DoF for each camera) is used as the hardware platform. Our vision system is equipped with Matrix Vision BlueFOX USB Cameras and NIKON custom made fish-eye lenses. The cameras have 7.28×5.04 mm CCD image sensors; the lenses have 5.4 mm focal length in the center, 90° horizontal, 62° vertical and 150° diagonal angle of view. The lens characteristics are plotted in fig. 2(b) and an example image taken with these lenses is shown in fig. 1(b).

The adaptation mechanism is embedded into an active vision system that incorporates several oculomotor movements. Targets are fixated with a combination of saccades and vergence movements. This will be explained in detail in section II.

II. METHODS

We modified the monocular saccade adaptation scheme introduced in [10] for a stereo vision system with vergence and



(a) (b)

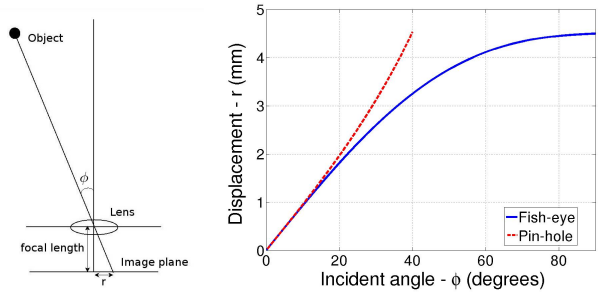
Fig. 1. (a) The experimental stereo camera head (b) Image taken from the stereo camera system equipped with custom made fish-eye lenses showing the visual environment in which online experiments took place

foveation and dealt with problems specific to this layout. The mapping between the image and motor coordinate systems is done with a variant of the Kohonen style Self Organizing Map (SOM) [4]. The nodes of the map represent motor commands, which bring visual targets to the foveae, and connections between these nodes denote the neighborhood relations in image space. The training of the map is done by using the correct association of the motor commands with the image coordinates. Such an association is made by comparing the images taken before and after a fixation as proposed in the previous work. The validity of the association is rated by a confidence measure. The number of nodes that are being adapted is regulated internally in order to speed up the adaptation process.

Extending the mapping scheme used in the previous work with stereo images in the input space and an additional degree of freedom (vergence) in the output space increases the 4 dimensional mapping function to 7 dimensions. We applied simplifications in dimensionality as explained in section II-A in order to reduce computational complexity.

The approach for making associations between executed motor commands and resulting changes in the images after a fixation proposed in [10] requires modifications for our framework due to distortion of the images caused by foveation. We extended this approach for foveated images as explained in section II-B.

Saccades are usually directed towards specific targets that are determined by the analysis of visual information. However, it is possible that saccades do not accurately land on desired targets due to systematic (e.g. the mapping is not yet precisely learned) or external reasons (e.g. the target has changed its position in the world during saccade). We propose a gaze fixation mechanism that allows the algorithm to learn with arbitrary gaze shifting movements and ensures that adaptation under improper fixation conditions is avoided. This is accomplished by using coordinated movements of saccade and vergence in two steps: First a coarse fixation on the target is achieved by a saccade (using pan, tilt and vergence movements), then the precision of the fixation on the target is improved by a fine vergence adjustment controlling the vergence angle. A saccade is considered to be completed only after proper fixation has been achieved. Eventually the adaptation algorithm is provided with data only from proper fixation conditions.



(a) (b)

Fig. 2. (a) Pinhole camera geometry. Projection of a light ray from an object onto the image plane (denoted as r) and incident angle (denoted as ϕ) are shown. (b) Displacement of projection on image plane (r) as a function of incident angle (ϕ) for a pinhole and a fish-eye lens.

A. Mapping Representation and Dimensionality

The mapping is a function $W(x_l, y_l, x_r, y_r)$ which associates image positions to motor commands. The stereo camera images define a four dimensional coordinate frame where x_l and y_l are the pixel positions in the left image, x_r and y_r are the pixel positions in the right image. Reducing the dimensionality of the mapping would eventually increase computational performance and decrease adaptation time. A reduction of the input size can be achieved by replacing one of the horizontal components (x_l or x_r) by disparity ($d = x_r - x_l$). This substitution allows us to select an appropriate disparity range depending on the application instead of using the whole horizontal dimension of the image. Coupled vertical movement of cameras also allows omitting one of the vertical components (y_l or y_r), although fish-eye lenses highly distort the images and introduce vertical disparities (i.e. $y_l \neq y_r$). Using a Matlab simulation[1] we found out that this does not have a crucial effect on our system. However, the mapping can be extended with vertical disparity as another dimension if lenses causing greater distortion are used. After those simplifications, the input space of the mapping is represented as $W(x, y, d)$ where $x = x_l$, $y = y_l$ and $d = x_r - x_l$.

Rotation of the cameras also causes a rotation of the images around the optical axes and this rotation depends on the current tilt and vergence angles [8]. This dependence is also excluded from our representation of input space for further reduction of dimensionality. Despite these simplifications our algorithm is able to perform with the required accuracy (see section III).

The output space is $(\Delta\theta_p, \Delta\theta_t, \Delta\theta_v)$ which denote the

[1]Vertical disparities are computed for the projection of a scene point at 30 cm on horizontal and vertical axes with respect to the stereo camera system in a depth range from 30 cm to 300 cm. The maximum vertical disparity is found as 3.472 pixels (4.6% of the total image height).

relative motor commands for pan, tilt and vergence[2] respectively. Therefore, current motor positions have to be known to execute saccades. The size of the output space is limited by the precision of the head and camera motors.

B. Image Correspondences

A method is necessary to make associations between motor commands $\overrightarrow{m}$ executed for a fixation and the image position $\overrightarrow{r}$ that has been brought to the foveae as the result of the executed motor commands. We adopted the same correspondence calculation method that has been previously introduced in [10]. This method is based on searching for a patch from the foveal region of the post-fixation image (referred to as I^t) in the pre-fixation image (referred to as I^{t-1}) in order to find out which part of the image has been moved to the fovea by the given motor command $\overrightarrow{m}$. We use Normalized Cross Correlation (NCC) as a similarity measure between the patches. A correspondence map $C(x, y)$ is computed by

$$C(x,y) = \Delta(\overrightarrow{I^t}(x_0,y_0), \overrightarrow{I^{t-1}}(x,y)) \cdot f_1(x,y) \cdot f_2(x,y) \quad (1)$$

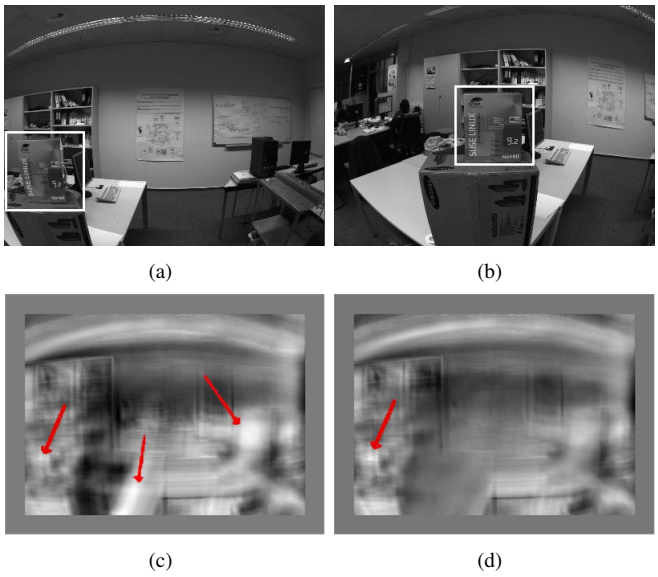
where Δ indicates the NCC operation, $\overrightarrow{I^t}(x_0, y_0)$ is the foveal patch from I^t, and $\overrightarrow{I^{t-1}}(x, y)$ is the patch from I^{t-1} around the point (x, y). Normally NCC distinguishes the corresponding positions quite well when it is used with non-foveated images. However, in our case due to the spatial distortion caused by the fish-eye camera lenses, using NCC alone was not successful to reveal the corresponding positions especially when features that are searched for reside in the periphery of the pre-fixation image where distortion is high. Mean and variance of the patches however, are not spatially very much affected by the distortion. Therefore, we introduced factors $f_1(x, y)$ and $f_2(x, y)$ in the following way:

$$f_1(x,y) = |mean(I^t(x_0,y_0)) - mean(I^{t-1}(x,y))|^{-1} \quad (2)$$

$$f_2(x,y) = |var(I^t(x_0,y_0)) - var(I^{t-1}(x,y))|^{-1} \quad (3)$$

The correspondence maps before and after the factors have been applied are shown in fig. 3(c) and 3(d). Theoretically, the maximum correspondence $c_{max} = C(x_{max}, y_{max})$ is the position where the foveal region has been before the cameras were moved. However, real world conditions may cause ambiguities so that more than one peak occurs in the correspondence map. A confidence measure of whether the candidate peak corresponds to the real pre-movement position of the foveal patch is explained in section II-C.

Correspondence calculations for both left and right images are done and maximum correspondence positions are obtained as $c_{max,i} = C(x_{max,i}, y_{max,i})$ where the subscript i is l or r for left and right images. The correspondence position is defined as $\overrightarrow{r} = [x^*, y^*, d^*]$ where $(x^* = x_{max,l})$, $(y^* = y_{max,l})$ and $(d^* = x_{max,r} - x_{max,l})$.



(a) (b) (c) (d)

Fig. 3. Correspondence calculation in foveated images. Subfigure (a) shows an image taken before fixation and (b) shows an image taken after fixation. The foveal patch of the post-fixation image and its location in the pre-fixation image are also marked. The plain correspondence map of above images without factors f_1 and f_2 is shown in (c) where multiple peaks (marked with arrows) indicate potential matches. The correct position of the patch is acquired after factors f_1 and f_2 are applied to the correspondence calculation, as shown in (d).

C. Confidence Measure

Under real world conditions it is very likely to have a visual environment with relatively big homogeneous structures and elements lacking details and texture like walls, carpets, etc. Under these conditions it is possible to have wrong or multiple correspondences as the result of the correspondence calculation. Fig. 4 illustrates such a kind of situation. This causes wrong associations between the inputs and outputs of the mapping. To avoid this problem the computation of a confidence value is introduced as done in [10]. The confidence value is composed of two factors. The first factor c_1 is a confidence from the correspondence value c_{max}. Since a high value of c_{max} means high similarity, the confidence factor c_1 is computed as

$$c_{1,i} = \frac{1}{1 + e^{-c_s \cdot (c_{max,i} - c_t)}} \quad (4)$$

where the subscript i is l or r for left and right images. This is a sigmoid with a slope of c_s and a threshold of c_t. c_s and c_t can be calculated through a common parameter τ by $c_s = \frac{10}{1-\tau}$, $c_t = \frac{1+\tau}{2}$. The parameter τ represents the minimum accepted correspondence value.

The second factor c_2 is the normalization factor. If multiple candidate matches are found by the correspondence calculation the confidence should be reduced. A threshold operation

[2]The vergence angle is obtained from the triangular geometry of the the left and right camera angles as $\theta_v = |\theta_l| + |\theta_r|$. The control of the vergence angle is done by the symmetrical control of the left and right camera angles ($\theta_l = -\theta_r$)

is done on the correspondence matrix with a threshold value of $\delta = R \cdot c_{max}$ where R is the percentage of the maximum correspondence value to count as a competing match. If the candidate peaks are close to c_{max} it is more likely for c_{max} to be the correct match, so they are not so critical. If they are distant from c_{max} the probability of the candidate peaks being potential matches increases. This weighting operation is done by computing a normalization map:

$$N(x,y) = T(C(x,y) - \delta) \cdot \left(\left(\frac{x - x_{max}}{\sigma_c} \right)^2 + \left(\frac{y - y_{max}}{\sigma_c} \right)^2 \right) \quad (5)$$

where $T(f)$ is a threshold function which gives 0 for $f < 0$ and f otherwise. The characteristic range is defined by $\sigma_c = r_t \cdot s_{img}$ where r_t defines the percentage of the tolerance radius and s_{img} is the size of the image. From the normalization map the normalization factor is calculated as:

$$c_{2,i} = \frac{1}{1 + \sum_{x,y} N_i(x,y)} \quad (6)$$

where the subscript i is l or r for left and right images.

The final confidence value is obtained by the multiplication of the two confidence factors of the left and right images:

$$c = c_{1,l} \cdot c_{2,l} \cdot c_{1,r} \cdot c_{2,r} \quad (7)$$

The confidence factors explained here are used to check two independent situations that solely cause adaptation with incorrect data. By multiplying the confidence factors, it is ensured that either of the situations can cause a considerable decrease in the confidence alone.

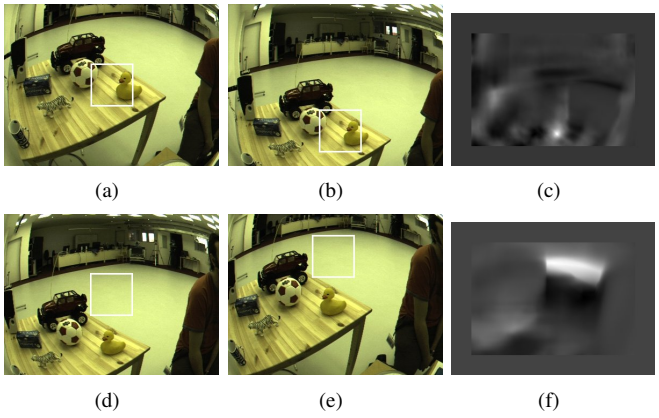


Fig. 4. Correspondence calculations for two different cases. The upper row shows the first case and the lower row shows the second case. Figures (a), (b) and (c) with their below counterparts correspond to the post-fixation image, pre-fixation image and correspondence map respectively for the two cases. In the first case, the foveal patch that is taken from the post-fixation image (a) has rich visual detail. Therefore, it could be found easily in the pre-fixation image (b) with a correspondence calculation, as can be seen by the correspondence map (c). However, in the second case, the foveal patch that has been taken from the post-fixation image (d) has little structure, making it difficult to find it in the pre-fixation image (e). The search for the patch has caused ambiguities in the correspondence map (f). Confidence values that are calculated as explained in section II-C are 0.93 for the first case and 0.39 for the second case.

D. Robustness of Adaptation

An adaptation step is done by updating the node of a correspondence position $\overrightarrow{r}$ following a rule explained in section II-E. As introduced in [10] the learning process can be sped up by updating a population of nodes in the neighborhood of the correspondence position instead of just one node. The adaptation strength and the number of the neighboring nodes can be determined by a Gaussian neighborhood function $G(x,y,d)$ with a peak at position (x^*, y^*, d^*) and standard deviation of σ_x, σ_y and σ_d in each dimension. The standard deviation parameters define the population width and are computed dynamically from the error in the input space (retinal error) so that a wide range of adaptation can be reached for large errors and more local updates are made for small errors in input space. This error (referred to as retinal error) is computed as:

$$E = \sqrt{\left(\frac{x^* - x_d}{E_{max,x}} \right)^2 + \left(\frac{y^* - y_d}{E_{max,y}} \right)^2 + \left(\frac{d^* - d_d}{E_{max,d}} \right)^2} \quad (8)$$

where x_d, y_d and d_d are the retinal positions that were associated with the given motor command $\overrightarrow{m}$ in the mapping before the adaptation takes place. The normalization factors $E_{max,x}$, $E_{max,y}$ and $E_{max,d}$ are the components of the maximum possible retinal error. In this work 50% of the size of the corresponding dimension is selected for these values. The population widths for the input space are calculated from the mean error in the input space $\overline{E}$:

$$\sigma = \sigma_{max} \cdot \frac{1}{1 + e^{-s \cdot (\overline{E} - t)}} \quad (9)$$

where σ_{max} is the maximum population width. Following (9) adaptation widths σ_x, σ_y and σ_d are computed separately with respective maximum population widths $\sigma_{x,max}$, $\sigma_{y,max}$ and $\sigma_{d,max}$ that are determined in relation to the size of the input space. We have selected 20% of the size of the corresponding dimension for these values. s and t are the slope and the threshold values for the sigmoid function respectively. $\overline{E}$ denotes the sum of errors in the input space averaged over a time window (e.g. averaged over the last ten errors calculated).

E. Adaptation Algorithm

The adaptation is done according to the basic delta rule:

$$W^{t+1}(x,y,d) = W^t(x,y,d) + \alpha \cdot \Delta W(x,y,d) \quad (10)$$

where α is the adaptation step size (a fixed parameter). The change in the mapping is:

$$\Delta W(x,y,d) = -c \cdot G(x,y,d) \cdot (W^t(x,y,d) - \overrightarrow{m}) \quad (11)$$

where c is the calculated confidence value and $G(x,y,d)$ is the Gaussian neighborhood function defining the adaptation region.

III. RESULTS

Results are presented as learning curves depicting errors in input and output spaces over adaptation iterations (i.e. saccades). All errors are averaged over the last 10 iterations. Calculation and scaling of the retinal error is explained in section II-D. The update vector in (11) is used to derive the errors in the output space as $|\Delta W(x^*, y^*, d^*)|$. Corresponding elements of the vector show the pan, tilt and vergence errors. The errors are plotted against the number of fixations. One fixation takes approximately 2 sec with our setup. All parameters and their selected values in our experiments are listed in table I. Parameters marked with S mostly depend on the system (hardware and software). Parameters marked with A depend on the application and require fine tuning. The unmarked parameters already produce good results with most of the applications and can be used without tuning. In all experiments the mapping has been initialized randomly.

TABLE I
LIST OF PARAMETERS

Parameter	Symbol	Value
Parameters for Learning		
Input Size	s_{in}	100x75x21 (S)
Output Size	s_{out}	50x50x50 (S)
Adaptation Rate	α	0.6
Max. Adaptation Sigma	σ_{max}	0.2 (A)
Adaptation Slope	s	5
Adaptation Threshold	t	0.2
Windowing Parameter	w	10
Parameters for Correspondence Calculation		
Patch Size	s_{pat}	20x20 (S)
Min. Correlation	τ	0.2 (A)
Min. Ratio to Compete	δ	0.95
Tolerance Radius	r_t	0.1

A. Learning with Simulated Data

In order to verify the adaptation algorithm, we first constructed a model of our stereo vision system (including kinematics and fish-eye lens distortions) using the Epipolar Geometry Toolbox (EGT) designed for Matlab (see [6] for more information). Randomly selected fixation commands in the motor space and their corresponding positions in image space are retrieved from the model and applied to our adaptation algorithm. A satisfactory performance was reached in a short time (fig. 5).

We investigated the robustness of the system with several experiments (e.g. inverting images with a prism, swapping left and right images, simulating motor defects). Our algorithm was able to adapt to such changes in every case. In this paper, we present one of these experiments, which has been motivated by the ontogenetic development of the eye distance in humans. The distance between the cameras in the model is increased by 10 mm every 1000 iterations. In order to show the impact and the adaptation, the change is applied not gradually as in human ontogeny, but instantaneously. Our system was able to adapt to the change quickly (fig. 6).

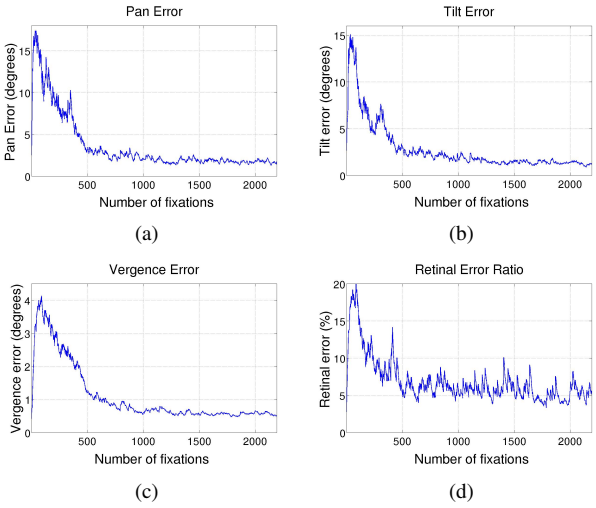


Fig. 5. Learning curves for simulated data using EGT. After 2000 iterations the following error values were reached: pan error = 1.3°, tilt error = 1.07°, vergence error = 0.5°, retinal error = 3.68% (≈ 4.6 pixels).

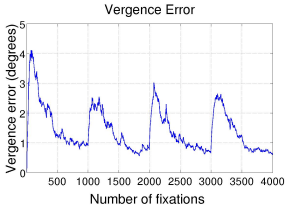


Fig. 6. Experiment with simulated data for adaptation to development of eye baseline. The impact of the change in the baseline after every 1000 iterations can be observed as peaks in the vergence error plot. Re-adaptation is achieved in ≈500 fixations after every change.

B. Online System Implementation

The adaptation mechanism is implemented in an active vision system as explained in section I-B. In the experiments targets for fixation have been selected randomly in motor space instead of saliency computation in image space. A snapshot from the visual environment used for online experiments is shown in fig. 1(b). The mapping has been learned from random initialization in a short time (fig. 7).

Adaptation experiments were also done with the online system implementation. In the first scenario, a prism effect that flips the image upside-down was applied after 1000 iterations. As a second scenario an artificial pan angle defect is applied to the system by adding a 10° of offset to the pan motor angle that is received from the motor encoder after 1000 iterations. Again our mechanism was able to adapt to the changes quickly (fig. 8 and fig. 9).

IV. SUMMARY AND OUTLOOK

In this work, we presented a capable and dependable approach for vision systems that eliminates the necessity of a (re)calibration process by learning and adapting a mapping between image and motor coordinates. We extended the learning strategy for monocular vision systems that is explained in [10] to be used with a stereo-vision

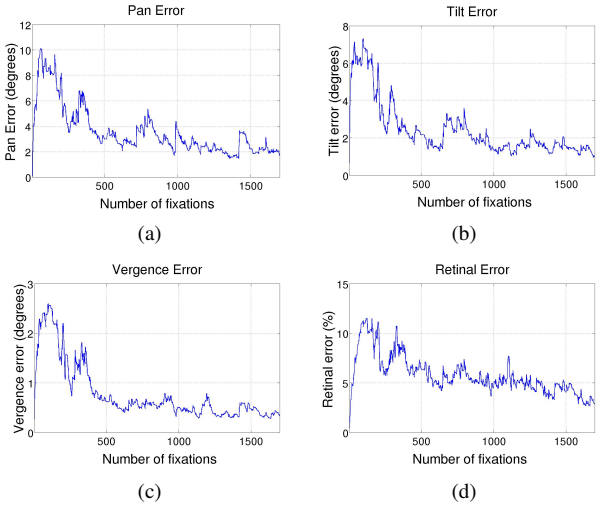


Fig. 7. Learning curves of online experiment. After 1700 iterations the following error values were reached: pan error = 1.7°, tilt error = 1.00°, vergence error = 0.34°, retinal error = 2.6% (≈ 3.25 pixels).

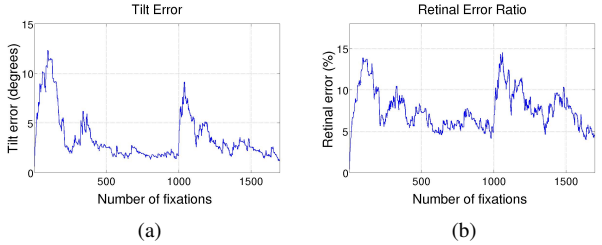


Fig. 8. Online experiment for adaptation to prism effect along vertical axis. 700 iterations after the prism effect is applied the following error values were reached: retinal error = 3.9% (≈ 4.8 pixels), tilt error = 1.20° (pan and vergence commands are not dramatically affected). The disturbance from the prism effect can be seen in tilt and retinal errors after 1000 iterations.

system with vergence and foveation. In this framework, we introduced enhancements to correspondence calculation for dealing with distortion caused by foveated lenses. We also proposed a fixation strategy in which gaze shifts are done as a combination of saccade and fine vergence movements. This approach enables us to avoid adaptation under improper fixation conditions.

We have shown that our mechanism learns the mapping in a short time. A reasonably high accuracy (≈3.25 pixels of retinal error) can be achieved in only 2000 iterations. This is quicker than both of the results that are presented by Pagel et al. (≈10000 iterations) in [8] and Rao et al. (≈5000 iterations) in [9]. The previous work learns the mapping in ≈100 iterations [10]. Our algorithm takes longer than that due to the higher complexity of the mapping.

Our approach is capable of providing online adaptation to vision systems without interrupting any processes by being independent of the target selection process. Therefore, it can be integrated with any active vision system performing arbitrary gaze-shifts using the gaze fixation strategy that we propose. In the future we plan to embed our system into an architecture of a large scale incremental behavior

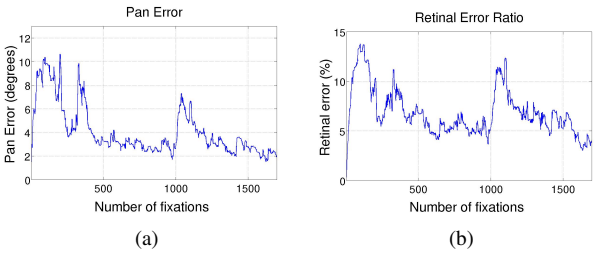


Fig. 9. Online experiment for adaptation to pan defect. 700 iterations after the pan defect is applied, the errors have been reduced to an acceptable level: retinal error = 3.06% (≈ 3.8 pixels), pan error = 1.50° (tilt and vergence commands are not affected). The disturbance from the pan defect can be seen in pan and retinal errors after 1000 iterations.

control system, called ALIS. A previous instance of this system running on the humanoid robot ASIMO is described in [2]. Further issues can be considered for discussion and improvement. For example, our correspondence calculation often exhibits satisfactory results with foveated images. However, scale invariant object matching techniques (such as [5]) may improve correspondence results thus, a quicker learning process may be obtained.

V. ACKNOWLEDGMENTS

The authors would like to thank Frank Joublin for fruitful discussions and Achim Bendig for his support with the hardware.

REFERENCES

[1] A. Bernardino and J. Santos-Victor. Binocular visual tracking: Integration of perception and control. *IEEE Transactions on Robotics and Automation*, 15(6), December 1999.

[2] B. Bolder, H. Brandl, M. Heracles, H. Janssen, I. Mikhailova, J. Schmuedderich, and C. Goerick. Expectation-driven autonomous learning and interaction system. In *IEEE-RAS International Conference on Humanoid Robots*, 2008.

[3] A. Dankers and A. Zelinsky. Cedar: A real-world vision system: Mechanism, control and visual processing. *Machine Vision and Applications*, 16(1):47–58, December 2004.

[4] T. Kohonen. *Self-organization and associative memory: 3rd edition*. Springer-Verlag New York, Inc., New York, NY, USA, 1989.

[5] D. G. Lowe. Object recognition from local scale-invariant features. *Proc. of the International Conference on Computer Vision*, 1999.

[6] G. L. Mariottini and D. Prattichizzo. EGT: a toolbox for multiple view geometry and visual servoing. *IEEE Robotics and Automation Magazine*, 3(12), December 2005.

[7] G. Metta. An attentional system for a humanoid robot exploiting space variant vision. *IEEE-RAS International Conference on Humanoid Robots*, pages 359–366, 2001.

[8] M. Pagel, E. Mael, and C. Von Der Malsburg. Self calibration of the fixation movement of a stereo camera head. *Autonomous Robots*, 5(3-4):355–367, July 1998.

[9] R. P. N. Rao and D. H. Ballard. Learning saccadic eye movements using multiscale spatial filters. In *Advances in Neural Information Processing Systems 7*, pages 893–900. MIT Press, 1995.

[10] T. Rodemann, F. Joublin, and C. Goerick. Continuous and robust saccade adaptation in a real-world environment. *KI-Künstliche Intelligenz*, 03:23–26, 2006.

[11] T. Shibata, S. Vijayakumar, J. Conradt, and S. Schaal. Humanoid oculomotor control based on concepts of computational neuroscience. *Humanoids 2001, Second IEEE-RAS Intl. Conf. on Humanoid Robots, Waseda Univ., Japan*, pages 278–285, 2001.

[12] A. X. J. Zhang and A. L. P. Tay. Vergence control of 2 DOF pan-tilt binocular cameras using a log-polar representation of the visual cortex. *Neural Networks*, pages 4277– 4283, July 2006.