A Self-Trainable Depth Perception Method from Eye Pursuit and Motion Parallax

Tanapol Prucksakorn[☆a], Sungmoon Jeong[☆a,b], Nak Young Chong[a]

[a] *School of Information Science, Japan Advanced Institute of Science and Technology, 1-1 Asahidai, Nomi, Ishikawa 923-1292, Japan.*
[b] *Bio-medical Research Institute, Kyungpook National University Hospital, 130 Dongdeok-ro, Jung-gu, Daegu 41944, Korea*

Abstract

When humans move in a lateral direction (frontal plane), they intuitively understand the motion parallax phenomenon while jointly developing sensory neurons and pursuit eye movements with the help of their life-long learning experiences. At that time, various ranges of motion parallax effects are used to extract meaningful pieces of information such as relative depth of variously positioned objects and the spatial separation between the robot and the fixating object (absolute distance). By mimicking the visual learning in mammals to realize an autonomous robot system, a visual learning framework [1] was proposed to concurrently develop both visual sensory coding and pursuit eye movement with an addition of depth perception. Within the proposed framework, an artificial neural network was used to learn the relationship between the eye movements and the absolute distance. Nonetheless, the limitation of the proposed framework is that the predefined single lateral body movement can not fully evoke the motion parallax effect for depth perception. Here, we extend the presented visual learning framework to accurately and autonomously represent the various ranges of absolute distance by using pursuit eye movements from multiple lateral body movements. We show that the proposed model, which is implemented in a HOAP3 humanoid robot simulator, can successfully enhance the smooth

[☆] These authors contribute equally.
Email addresses: tanapol.pr@jaist.ac.jp (Tanapol Prucksakorn[☆]), jeongsm00@gmail.com (Sungmoon Jeong[☆]), nakyoung@jaist.ac.jp (Nak Young Chong)

pursuit eye movement control with the self-calibrating ability and the distance estimation comparing to the single lateral movement based approach.

Keywords: Active depth perception, Developmental Vision, Motion parallax, Eye pursuit, Sensory-motor Coordination

1. Introduction

For living organisms such as humans and mammals, when they were born they do not instantly understand how to use the information they perceived. They continuously learn and improve their perception while interacting with the environments around them during their lifetime. This is described as developmental learning. The essence of building a biologically plausible cognitive robot is based on the developmental learning of perceptual and behavioral abilities in humans and developed organisms. Recently, there are many studies on computer vision field that are related to human cognitive systems, inspired by the facts that humans can autonomously develop and recover their perceptual and behavioral abilities to survive in various environments. These abilities are not only useful for extracting visual information for guiding actions, but they are also for perceiving the environments. Synthetic approaches based on explanations and designs could be proposed to overcome the shallow knowledge [2]. However, it is still a very challenging task to implement the cognitive developmental system in an autonomous learning manner. In order to realize the cognitive developmental robot, the system should equip two important learning principles which are (1) autonomous development through their artificial life and (2) unified-learning of action and perception. By establishing a tight connection between action and perception, the visual information can be used to improve the robot's behavior, while the resulted actions effectively reinforce the perceptual learning.

The same idea also applies to active depth perception which is a process of producing different kinds of eye and body movement to utilize active visual depth cues. Moreover, it is required that several cognitive developments such as

visual representation (sensory coding), eye movement control (action strategy), and depth representation (high-level sensory perception) are simultaneously performed during their lifetime (life-long learning). However, the underlying ideas of the active depth perception are still unclear. In [3, 4], they utilized the efficient coding theory together with a reinforcement learning algorithm to tightly couple action and perception for a robot to generate vergence based eye movements on the encoded information. The active efficient coding framework for the autonomous self-calibration of active perception was proposed in [5, 6, 7]. It originates in the theories of efficient sensory coding in Neuroscience. The main concept is that it exploits the statistical properties of the sensory signals to encode the sensory signals efficiently. Following the works, there are studies proposed frameworks that can generate smooth pursuit eye movements to track a moving object [8, 9] based on the same concept. Recently in [10], they also took a similar approach by using Gabor filters for binocular disparity coding and Hebbian learning for the eye movement control. In these mentioned studies, the behavior does not simply learn by itself, but it also learns with the help of the perception part, and vice versa. In other words, action and perception are not only connected, but they encourage each other to enhance themselves to extract more meaningful information creating an action-perception cycle. However, the works do not consider self-induced body movements and depth perception such as motion parallax.

Motion parallax is a phenomenon that can be observed in daily life. It provides useful information that helps the observer to visually understand the surrounding environment. When the observer moves in a lateral direction (frontal plane), various ranges of motion parallax effect occur by maintaining its visual fixation on a visual stimulus. Motion parallax effect provides two different kinds of depth perception which are the distance from the observer to the fixating object (egocentric distance), and the distance from the fixating object to other objects (allocentric distance). Usually, allocentric distance is extracted from the motion parallax phenomenon such as in [11], they discuss how it is possible to generalize the relationship between the eye movements and the allocentric

distance. However, that is not the only strong point of utilizing the motion parallax effect. In [12], they showed that it is possible for humans to extract the egocentric distance information. Also, the retinal motion induced by the motion parallax effect can be utilized to observe the apparent depth (egocentric distance) appears on the sagittal plane [13].

In [1], they proposed a developmental learning framework for active depth perception that utilizes the motion parallax phenomenon. The study successfully created a model which could estimate the egocentric distance with a learning scheme. However, the limitation of the model is that the lateral body movement is limited to a single movement. The issue is that large lateral body movement produces large parallax (angle between the two different lines of sight) which raises the difficulty of fixating the visual stimulus. In addition, the proposed model could only obtain the distance estimation after the training is done which is inappropriate in the developmental learning scheme.

Therefore, in this paper, we extend the concept of the related works to generate reliable eye movements for various ranges of motion parallax. We show that the eye movement information could be successfully used to represent the egocentric distance information by coupling the action and perception.

2. Related Works

Currently, there are two major approaches to implement a human-like visually guided cognitive system with visual depth perception ability. One of the approaches is to individually develop perceptual and behavioral abilities. Perceptual ability guides the behavior to solve a given task in a straightforward way to link perception and action together.

Remarkably, there are many studies that proposed image processing and machine learning techniques to implement depth perception for solving a given task [14, 15, 16, 17]. In [18], they utilized multiple frames captured with a single camera to predict distances. Prediction algorithm was designed and used as a distance estimator under the assumption that the camera motions are known.

[19] proposed a biologically plausible visual attention system to selectively localize a salient area. [20] used an information theoretic approach to minimize an uncertainty. In [21], they utilized a monocular vision-based obstacle avoidance system by coupling a reinforcement learner together with a linear regression method. Some studies used a visual servo approach [22, 23, 24] to establish the coordination of action and perception by utilizing the kinematic link between the visual information and the camera velocity. However, with the aforementioned works, it is quite challenging to create such a system that can develop and adapt itself to the different environments by developing both of perceptual and behavioral abilities at the same time. The main reason is that manual calibrations and prior knowledge are required to finely tune the system during their artificial life.

Nonetheless, developmental systems can learn and adapt to various environments, thus it could be the second approach to achieve the biologically inspired cognitive robot. In [25], they proposed a way to implement developmental learning of eye-head gaze control in human infants in a humanoid robot. They used a constraint-based field-mapping approach for the learning of gaze control. In [26], a convolutional network was used to train vergence eye movements, but it required supervised signal to minimize the cost function. In [27], they successfully demonstrated a framework that generates multiple eye movements which are smooth pursuit and vergence eye movements to track an object. The model encourages the relation between action and perception which are learned by themselves without any supervision.

In this paper, we propose a solution to the issues based on the previous studies [1, 3, 4, 8, 9, 27]. We analyze and examine how a visual system can understand various ranges of motion parallax effects through acquiring visual sensory representations and eye movements control with multiple self-induced lateral body movements. This study considers three important mechanisms to obtain the smooth pursuit eye movement and the egocentric depth estimation: (1) visual sensory representation for low level visual signal processing by using sparse coding technique, (2) eye movement generation that maximizes the re-

dundancy between input visual signals by reinforcement learning algorithm, (3) multiple lateral body movements for generating various motion parallax effects.

The extended framework can also be described as a low-level visual cue in the primary visual cortex (V1) [28], as it only focuses on maximizing the sensory encoding efficiency (sparse coding) of the available visual stimuli. Since allocentric depth requires a higher understanding of the concept of the object such as border ownership which is represented by some of the V2 and V4 neurons in the visual cortex [29], this research focuses on observing the egocentric distance.

To the best of our knowledge, no study has yet attempted to propose a motion parallax based active depth perception framework for the cognitive developmental robot under the efficient coding theory with multiple lateral movements. This approach does not only enable the robot to autonomously learn sensory representation and eye movement controls, but it is also the first step toward creating active depth perception during self-induced body movements.

3. Methods

Motion parallax is one of the visual depth cues to perceive the depth based on monocular vision system. This phenomenon is generated when an observer moves in a lateral direction, i.e., left or right direction (frontal plane), while fixating a visual stimulus. Therefore, by letting a robot move laterally and capture the successive images, it can observe a motion parallax phenomenon under different conditions, such as positions and translation speeds. Then, two different kinds of depth can be extracted from the motion parallax effect. The first one is the spatial interval that separates the robot and the visual stimulus (egocentric distance). Second, the spatial interval between the fixating visual stimulus and other visual stimuli (allocentric distance). In this research, we assume that the robot can perfectly control its lateral body movements without uncertainty. The developments of the related cognitive functions, such as visual representation and eye movements control, will only be focused on to understand the motion parallax and the distance information (egocentric).

3.1. Model Architectures

The goal of this framework is to generate eye movements that can fixate on certain visual stimuli at various lateral positions. Fig. 1 shows the architecture of the framework. When the robot generates a self-induced lateral motion, the robot rotates the camera to fixate the visual stimulus (smooth pursuit eye movement). The number of eye movements is used to represent the distance information by mapping between the eye pursuit and distance. To achieve that, there are three required main cognitive functions: (1) visual representation based on sparse coding, (2) eye movements control based on reinforcement learning, (3) artificial neural network to represent the distance information supervised by a human. First, we utilize multiple sparse coding schemes as a sensory coding model coupled with a reinforcement learner to achieve the efficient coding for the visual inputs. The sensory coding model learns to represent the input images, while the reinforcement learner tries to generate actions that increase the efficiency of the coding model. Multiple lateral movements help the framework to understand various ranges of motion parallax. Finally, the generated number of eye movements are used as inputs for the artificial neural network during the human-robot interaction to represent the distance information.

3.2. Single & Multiple Lateral Positions

In this research, we consider two different learning strategies which are based on a single lateral body movement and multiple lateral body movements. The key difference between the two strategies is visualized in Fig. 2. By having more than one lateral body movement, the robot can go through the gradual steps of learning difficulties. As shown in Fig. 2(a), if the lateral body movement is large, then the learning difficulty is significantly larger which is inappropriate at the beginning of the learning stage. While, the multiple lateral body movements strategy, Fig. 2(b), provides levels of learning difficulty. Therefore, the multiple lateral movements strategy is helpful for the reinforcement learner to grasp the control of small movements first.

3.2.1. Position Setup

At the beginning of the training, an image I_0 is captured from the camera at the original position as a reference. Then, the robot moves laterally from its original position l_0 to the lateral position l_1 which is from the lateral position list $L = \{l_1, l_2, \ldots, l_p, \ldots, l_r\}$. p is the index of lateral positions on the list L as shown in Fig. 3. Then, the framework proceeds to the first iteration.

3.2.2. Motion Parallax

After the robot moves laterally for $l = l_1$ as shown in Fig. 4, the motion parallax phenomenon is induced. In this research, the parallax angle is the angle between the two different lines of sight which is shown as q in the figure.

To collect the information of the visual stimulus for generating smooth pursuit eye movements, an image $I_{l_k}(t)$ is captured from the left eye camera at the lateral position k-th. The two captured images $I(t) = \begin{bmatrix} I_0 & I_{l_k}(t) \end{bmatrix}$ are then input to the sensory encoder to generate one eye movement from the reinforcement learner. This process of capturing $I_{l_k}(t)$ and generating eye movement are repeated for t_h iterations (one trial). Theoretically, the framework should produce a total number of eye movements that is similar to q.

After one trial, the robot moves to the next lateral position (l_2) in L. It repeats the process until it reaches the final position l_r. When the robot reaches the final position in L, the robot simply moves back to the lateral position l_0 preparing for the next visual stimulus.

3.3. Sensory Coding Model

Sensory systems should encode sensory information in an efficient manner by exploiting redundancies in their inputs [5, 6, 7]. We use sparse coding to learn efficient representations of the sensory inputs. The key idea of this efficient encoding is that the reinforcement learner receives the reward signal depending on how well the sensory model can represent the input.

Figure 5 visualizes the process and the flow of the input images for the eye movement generation. The input images are first converted to gray-scale. There

are two cropping windows representing fine and coarse scale images. They are used to crop the input images $I(t)$ by 150x150 pixels and 250x250 pixels from the center, respectively. The cropped images are then sub-sampled according to the cropping windows. Coarse images are sub-sampled by a factor of $P_s^C = 8$. Fine images are sub-sampled by a factor of $P_s^F = 2$. We use a Gaussian pyramid algorithm for the sub-sampling.

The two scales of the images represent the foveal system in human eyes. The fine scale image represents a foveal region in eyes which can pick more detail from the center of vision. While the coarse scale represents the parafoveal area which has lower detail. Square patches where the lengths of each side are $P_l = 10$ pixels, i.e., 10 by 10 pixels patches, are then extracted from the sub-sampled cropped images, whose locations are generated by 1 pixel and 4 pixels shifts horizontally and vertically for coarse scale and fine scale, respectively. The patches are reshaped to be one-dimensional vectors which have zero mean and unit norm, $\gamma_i^j(t)$. i is the index of the patch, which $j \in \{C, F\}$. C is for coarse scale, and F stands for fine scale.

The sub-sampled images let the framework handle image disparities that are larger than the patch width. Note that the fine scale helps in fine-tuning the eye movements. Discussions and comparisons between using one scale and multiple scales have been done in [4]. They discussed how gaining the access to multi-scale images could improve the learning of the framework. On the other hand, having only one scale might prevent the system from learning appropriately.

For the coarse scale and the fine scale, the two one-dimensional vectors are then combined into a single vector $\gamma^j(t)$. The first 100 elements of the vectors are from the first image I_0 and the remaining are from the second image $I_{l_k}(t)$. The resulted vectors ($\gamma^C(t)$ and $\gamma^F(t)$) consist of $K = 200$ elements.

Later, the patches are encoded by a sparse coding algorithm in a linear fashion. Each patch is represented by a linear combination of basis functions picked up from a dictionary $\phi^j(t) = \{\phi_n^j(t)\}_{n=1}^N$[30]. We use $N = 288$ basis functions. Two dictionaries are randomly initialized and normalized. One is for coarse scale and the another is for fine scale.

We use the matching pursuit algorithm defined in [31] to estimate and find the sparse representation of the input vector by the weighted sum as follows:

$$\gamma_i^j(t) \approx \hat{\gamma}_i^j(t) = \sum_{n=1}^{N} b_{i,n}^j(t)\phi_n^j(t) \ . \tag{1}$$

The matching pursuit algorithm suits to the concept of sparse coding, because it can estimate $\gamma_i^j(t)$ by using a limited number of coefficients. In this research, the maximum number of non-zero scalar coefficients $b_{i,n}(t)$ is set to be 10 elements to ensure the sparseness of the efficient coding. For later use in reinforcement learner part, pooled activity, $\theta_n^j(t)$, which represent the activity of each neuron cell is calculated from the coefficients from the matching pursuit algorithm as follows:

$$\theta^j(t) = \begin{bmatrix} \theta_1^j(t) \\ \theta_2^j(t) \\ \vdots \\ \theta_N^j(t) \end{bmatrix} \ . \tag{2}$$

Where, each element of the vector $\theta^j(t)$ is described as:

$$\theta_n^j(t) = \frac{1}{P_N} \sum_{i=1}^{P_N} b_{i,n}^j(t)^2, \tag{3}$$

where P_N is the number of patches extracted from one input image. A reconstruction error is introduced as a unified cost function that links the sensory coding model and the reinforcement learner. It measures the estimation error of vector $\gamma(t)$. The reconstruction error is defined as:

$$e^j(t) = \frac{1}{P_N} \sum_{i=1}^{P_N} \frac{\|\gamma_i^j(t) - \sum_{n=1}^{N} b_{i,n}^j(t)\phi_n^j(t)\|^2}{\|\gamma_i^j(t)^2\|}. \tag{4}$$

A gradient descent method is used to update the dictionaries with the reconstruction error as the cost function. After each update, the dictionaries are normalized.

3.4. Reinforcement Learning

The state representation of the reinforcement learner can be described by a combination of coarse scale and fine scale pooled activity, $\theta_n(t)$ as follows:

$$\theta(t) = \begin{bmatrix} \theta^C(t) \\ \theta^F(t) \end{bmatrix} . \tag{5}$$

The reward is a negative of the summation of reconstruction error from both scales which is described as:

$$R(t) = -(e^C(t) + e^F(t)) . \tag{6}$$

An actor-critic algorithm number 3 proposed in [32] is employed for the leaner agent. For action selection, we use Gibbs distribution (softmax) for probabilistically choosing an action as follows:

$$\pi(\theta(t), a_t) = \frac{\exp(z_a)}{\sum_{a' \in A} \exp(z_{a'})}, \tag{7}$$

where $\exp\colon x \mapsto \mathrm{e}^x$ is the exponential function. For each action, the activation value z_a is given by:

$$z_a = \sum_{n=1}^{N} w_a(t)\theta_n(t) , \tag{8}$$

where $w_a(t)$ is a weight vector from the state $f(t)$ to action a. The action is pan angle of the cameras in degrees. Possible actions a are contained in a set of actions A. In this research we use $A = \{-0.2^\circ, -0.1^\circ, -0.05^\circ, 0^\circ, 0.05^\circ, 0.1^\circ, 0.2^\circ\}$. Thus, the policy maps $\theta(t)$ to $a \in A$.

3.5. Egocentric Distance Representation

To extract the distance information, one may calculate it directly with the knowledge it has, such as traveled distances and eye movements. Since in this research, we focus on building the framework that can adapt to various configurations and environments, it is impractical to specifically calculate the distance information which usually requires the exact system's parameters.

To let the system learn the relationship between the distance and the eye movements, the robot must know (1) lateral distance and (2) number of eye

movements. Since the robot moves according to the lateral position list L, the speed of the lateral translation is constant. So, the knowledge of the lateral distance can be excluded from the learning. The eye movements (q in Fig.3) are memorized and accumulated in the vector $s = \begin{bmatrix} q_1 & q_2 & \dots & q_p & \dots & q_r \end{bmatrix}^\top$ at the end of each trial of each lateral position.

We suppose that the robot can remember c of its previous eye movement vectors s. The previous eye movement vectors are concatenated to create a queue-memory matrix

$$S = \begin{bmatrix} s_1 & s_2 & \cdots & s_c \end{bmatrix}, \tag{9}$$

where s_1 is the latest eye movement vector collected. s_c is the oldest eye movement vector that the robot can remember. When a new eye movement information s is available, s_c in the matrix S is discarded (dequeued). The indexes of the remaining vectors are then shifted by one, i.e., s_k is assigned to be s_{k+1}. The new vector is then assigned (queued) as s_1

Here, we use a feed-forward neural network with a hidden layer as the egocentric distance learner to interpret the eye movements to the distance information. We use the Levenberg-Marquardt method [33] for training the neural network. The input of the neural network is S (batch training). The sigmoid activation function is used in the hidden layer which has 30 neurons. The output layer uses the linear activation function. The target is ground truth distances provided by the supervisor. The supervised signals (ground truth absolute distances) are provided for letting the robot understand the metric system. The neural network starts to train after the robot has filled the memory matrix S, i.e., s_c exists. The training occurs every iteration when a new s is available.

3.5.1. Normalization of Generated Eye Movements for Neural Networks

Sensitive information should be carefully treated before inputting to a neural network since it could prevent the neural network from learning appropriately. In this research, information from large lateral body movements is considered as sensitive, because generating eye movement at those positions are more difficult.

We consider a disparity score to distinguish lateral body movements that gives sensitive eye movement information. The disparity score measures the spatial separation between the two viewpoints, which are the original position and relocated positions in the lateral direction, in pixels. The score is simply calculated by using geometry. As an example, Table 1 shows the disparity score for simulations that will be done in the next section.

A lateral movement that has a disparity score larger than or approximately equal to the effective patch size is considered sensitive since it means that there is no overlap between the two input images initially. If the images were not overlapped initially, the framework would have a difficulty finding the redundancy in the inputs. The effective patch size is calculated as follows:

$$P_e = P_l \cdot \sqrt{P_s^j}. \tag{10}$$

If the lateral position l_p is considered as the sensitive lateral position, then the lateral positions from l_p to l_r are sensitive. To reduce the negative effect of the sensitive eye movements, weighting is applied to the neural network input S as follows:

$$S = \begin{bmatrix} w_1 q_{1,1} & w_1 q_{1,2} & & w_1 q_{1,c} \\ w_2 q_{2,1} & w_1 q_{2,2} & & w_1 q_{2,c} \\ \vdots & \vdots & & \\ w_p q_{p,1} & w_p q_{p,2} & \cdots & w_1 q_{p,c} \\ \vdots & \vdots & & \\ w_r q_{r,1} & w_r q_{r,2} & & w_1 q_{r,c} \end{bmatrix}. \tag{11}$$

The weight w_k is defined as:

$$w_k = \begin{cases} \dfrac{1}{(1 + l_r - l_k)} \cdot \dfrac{1}{y}, & \text{if } k \geq p \\ 1, & \text{otherwise} \end{cases}, \tag{12}$$

where y is a hyper-parameter.

3.5.2. Input and Output of the Neural Network

Each column of the matrix S is the input of the neural network, as shown in Fig. 6. There are 30 neurons in the hidden layer which are defined as $h_1, h_2, h_3, \cdots, h_{30}$. The activation function of the hidden layer is sigmoid activation function defined as follows:

$$h_n = \sigma(s \cdot v_n^{(1)} + s_b), \tag{13}$$

$$\sigma(x) = \frac{1}{1 + \exp(-x)} \tag{14}$$

where s is any neural network's input vector. $v_n^{(1)}$ is a weight vector that contains weights connecting every input in s to the hidden layer's neuron h_n. s_b is a bias in the input layer. Finally, the depth information is calculated by the weighted sum of activations in the hidden layer including the bias h_b as follows:

$$d = h \cdot v^{(2)} + h_b, \tag{15}$$

where $h = \begin{bmatrix} h_1 & h_2 & h_3 & \cdots & h_{30} \end{bmatrix}^\top$. The weight vector $v^{(2)}$ contains weights that connect every neurons in the hidden layer to the output d. Note that the activation of the output layer is linear activation ($f(x) = x$).

After S is updated, i.e., a new s is available, the neural network is trained by using the Levenberg-Marquardt method.

4. Simulations & Results

4.1. Experimental Setup

We use V-REP, a robot simulator, as a 3-dimension environment visualization for testing the framework, while the model is implemented and developed in MATLAB. The simulated environment comprises a HOAP3 robot, an object with an interchangeable texture, and a still background image as shown in the top picture in Fig. 1. The lateral movement of the robot is simplified to be pick-and-place.

We test the depth cue to estimate the distance between the robot and the object. The separation distance between the robot and the object is ranged from 3 meters to 10 meters with 0.1 meters interval, i.e., $3.0, 3.1, 3.2, \cdots, 10.0$ meters. The number of iterations in one trial, t_h, is 30 iterations. We prepared a set of 100 different images for the framework to learn various visual stimuli.

4.2. Joint Development of Active Depth Perception

In this section, we test and analyze the performance of the framework. Eye movement generation and reconstruction errors are observed to verify the progress of learning. $c = 300$ sets of eye movements are used as inputs for the neural network. To track the performance of the eye movement generation, eye movements at 30 iteration are recorded and compared with the expected eye movements in form of mean absolute errors.The mean absolute error (MAE) is computed to evaluate the training. MAE is defined as follows:

$$MAE(t) = \frac{1}{1000} \sum_{k=0}^{999} |\alpha(t + 29 + 30k) - \alpha^*(t + 29 + 30k)|, \tag{16}$$

where $\alpha(t)$ and $\alpha^*(t)$ are the pan and the target pan at t.

4.2.1. Single Lateral Position and Multiple Lateral Positions

For comparison, we first do the single lateral position experiments. Each experiment contains 3 simulations which use different lateral positions which are $L = 5, 7, 10, 13, 15$ and 20 cm.

We picked these lateral positions based on the disparity score which is an approximated distance between the two input images in pixels, as seen in Table 1 below. The disparity score is calculated by using geometry at 3 meters distance because at the closest distance we can observe the maximum disparity for each lateral position.

Because the coarse scale is sub-sampled from the original image by a factor of 8, the maximum horizontal disparity that causes no overlap between two 10x10 pixels patches, i.e., the effective patch size, would be $P_e = 10 \cdot \sqrt{8} \approx 28$. At 15 cm lateral position, two patches are barely overlapped at the start of each

Table 1: Disparity score of the two input images at the beginning of each trial at 3 meters distance

Lateral Position (cm)	Disparity (pixel) at 3 m
5	9
7	12
10	18
13	23
15	27
20	35

trial. While 20 cm lateral position completely separates the patches. These two lateral positions are good examples to show the effect of having large lateral movement.

After we have confirmed the training of the single lateral position simulation, we test the multiple lateral movements case. We perform two experiments. Each experiment contains 3 simulations which use different sets of lateral positions which are $L = \{5, 6, 7, \ldots, 10\}$ (cm) and $L = \{5, 6, 7, \ldots, 20\}$ (cm).

4.2.2. Simulation Results

The training results from the multiple lateral positions simulations are shown in Fig. 8, for 5-10 cm, and Fig. 9, for 5-20 cm, before the vertical dashed lines in each figure. The blue dashed lines represent the variance of each trial from the 3 simulations. The solid line is the average of the MAE from 3 simulations. Table 2 shows a comparison between the single lateral body movement and the multiple lateral movements simulations.

As shown in Table 2, all of the simulations except for the 15 cm and 20 cm single lateral position show a similar performance for the last 100 trials eye movement MAE. In the beginning period of the training, there are a lot of combinations of textures and distances of the object to be learned, so the rises and the declines of the MAE can be expected as shown in Fig. 9. When the lateral body movement is too large, it makes the separation between the

Table 2: Performance of the single lateral position (Sing.) and multiple lateral positions (Mult.)

Lateral Position	Average (Variance) of last 100 trials		
	Sing.	Mult. 5-10 cm	Mult. 5-20 cm
5	0.18 (0.10)	0.18 (0.08)	0.25 (0.13)
7	0.16 (0.06)	0.16 (0.06)	0.18 (0.06)
10	0.16 (0.02)	0.17 (0.06)	0.18 (0.07)
13	0.18 (0.02)	-	0.17 (0.03)
15	0.30 (0.08)	-	0.17 (0.03)
20	0.91 (1.15)	-	0.19 (0.04)

two input images I_1 and $I_2(t)$ initially large. Large separation hinders the framework's ability to utilize the redundancy between the two images effectively. This leads to unstable eye movement generation as shown in Fig 7. However, it can still maintain the eye movement with some level of precision.

5. Robustness Test

We test the robustness of the framework by applying two kinds of perturbation to the system. First, we rotate the left camera by 15 degrees. Second, we add blur to the camera by applying a Gaussian filter with the standard deviation of 2 to the captured images.

The disturbances are applied after the training that is shown in sub-section 4.2.2. We then continue the training of the framework with the disturbances. As shown in Fig. 7 – Fig. 9 and Table 3, noticeable increases in the MAE are observed right after the gray dashed lines in the figures. The dashed lines represent the time when the disturbances are applied.

For the small single lateral positions, the framework can recover from the disturbances with similar performances before the interferences. For the lateral positions from 10 cm, we can see that they do not have MAE similar to the performances before the perturbations. However, they can still recover and

Table 3: Performance of the single lateral position (Sing.) and multiple lateral positions (Mult.) after perturbations applied.

Lateral Position	Average (Variance) of last 100 trials		
	Sing.	Mult. 5-10 cm	Mult. 5-20 cm
5	0.12 (0.01)	0.13 (0.01)	0.16 (0.02)
7	0.14 (0.01)	0.15 (0.01)	0.16 (0.03)
10	0.21 (0.04)	0.18 (0.02)	0.18 (0.01)
13	0.36 (0.12)	-	0.21 (0.04)
15	0.54 (0.21)	-	0.25 (0.04)
20	1.09 (1.03)	-	0.39 (0.13)

maintain the MAE. Interestingly, for the 20 cm lateral position simulation, it manages to maintain the MAE as shown in Fig. 7.

Noticeably, the lateral positions from 5 to 10 cm can fully recover to similar or even better performances from the disturbances. Some perform better after the disturbances because they simply have more time to learn. Also, disturbances encourage the framework to explore and learn more. While for the lateral positions from 13 cm to 20 cm, the framework can not fully recover from the disturbances. Because the effects of the high disparity scores and the disturbances obstruct the framework to learn to generate eye movement effectively. However, they can still maintain the MAE.

6. Distance Estimation

Large lateral movements are useful for estimating far distances such as 5 m to 10 m. However, the eye movement information is not quite useful for estimating the close distances such as 3 m since the information is sensitive. However, based on the proposed weighting, we show that the proposed model can successfully represent not only for the close distance but also the far distance. Moreover, the proposed model can recover from the perturbations.

We investigate the distance estimation performance and robustness test as

Table 4: Average distance estimation error for each range of distances.

Simulation	3 to 4.9 (m) distances	5 to 6.9 (m) distances	7 to 10 (m) distances	Total Average
5-10 cm without weighting	3.97%	4.09%	4.74%	4.33%
5-20 cm without weighting	6.39%	2.24%	3.06%	3.77%
5-20 cm with weighting	**3.66%**	**1.89%**	**2.55%**	**2.69%**

shown in Fig 10, Fig. 11, respectively. They show the distance estimation performances of the single lateral position simulations and the multiple lateral positions simulations. The dashed lines represent the minimum and the maximum values of the distance estimation errors for all of the single lateral position simulations. The blue solid line is the average of the distance estimation errors of the single lateral position simulations. The other solid lines are the distance estimation error for the multiple lateral positions simulations.

Fig. 10 shows that the multiple lateral positions strategies provide the better overall performances comparing to the average and the minimum of the single lateral position. With the weighting (red line), it shows a significant improvement in the 3-5 m distance range and a little improvement in the 9-10 m distance range with respect to the multiple lateral positions without weighting (magenta line). We can see that the proposed learning strategy with weighting improves the distance estimation performance.

For Fig. 11, we can see that the performances after the disturbances are quite similar to the performances before the disturbances for all of the simulations. Table 4 and Table 5 show the distance estimation error in each range of distances. We can see that with the lateral position 5-20 cm with weighting performs better than the other two strategies in every case. In addition, the performances of every lateral movement strategy are robust to the perturbations. This means that the proposed learning scheme is robust to the changing of the system's parameters.

Table 5: Average distance estimation error after perturbations for each range of distances.

Simulation	3 to 4.9 (m) distances	5 to 6.9 (m) distances	7 to 10 (m) distances	Total Average
5-10 cm without weighting	3.95%	3.52%	5.50%	4.48%
5-20 cm without weighting	4.30%	2.76%	3.03%	3.31%
5-20 cm with weighting	**2.29%**	**0.98%**	**0.82%**	**1.30%**

7. Conclusion and Discussion

In this research, we propose a novel visual learning framework to actively perceive the various ranges of distances from motion parallax by integrating the learning of sensory representation and the eye pursuit during self-induced multiple lateral body movements. An artificial neural network is used to represent the egocentric distance by autonomously understanding the relationship between the number of eye movements and the distance information under a human supervision instead of a certain equation. The generated eye movements are effectively used to represent the distance information. The proposed framework has a better accuracy to perceive the distance than a single lateral body movement. Moreover, the proposed model can seamlessly recover from the perturbations such as image blur and rotation.

To fully implement an autonomous learning system, two important future works are expected: (1) an appropriate body movement will be autonomously selected by robot-self. It should let the robot be able to decide when to stop moving laterally for generating enough information to perceive the distance according to the lower bound of motion required to distinguish distances [34] (2) An additional learning unit may be included to autonomously generate the meaning of depth and distance information by using embodied intelligences instead of the human supervision. This means that the autonomous system could be fully developed by itself without the external supervised signals such as distance information.

Acknowledgement

This research is supported by the project Autonomous Learning of Active Depth Perception: from Neural Models to Humanoid Robots from Japan Agency for Medical Research and Development, AMED.

References

[1] T. Prucksakorn, S. Jeong, J. Triesch, H. Lee, N. Y. Chong, Self-calibrating active depth perception via motion parallax, in: Development and Learning and Epigenetic Robotics (ICDL-Epirob), 2016 Joint IEEE International Conferences on, IEEE, 2016.

[2] M. Asada, K. F. MacDorman, H. Ishiguro, Y. Kuniyoshi, Cognitive developmental robotics as a new paradigm for the design of humanoid robots, Robotics and Autonomous Systems 37 (2) (2001) 185–193.

[3] Y. Zhao, C. Rothkopf, J. Triesch, B. Shi, A unified model of the joint development of disparity selectivity and vergence control, in: Development and Learning and Epigenetic Robotics (ICDL), 2012 IEEE International Conference on, 2012, pp. 1–6. doi:10.1109/DevLrn.2012.6400876.

[4] L. Lonini, Y. Zhao, P. Chandrashekhariah, B. Shi, J. Triesch, Autonomous learning of active multi-scale binocular vision, in: Development and Learning and Epigenetic Robotics (ICDL), 2013 IEEE Third Joint International Conference on, 2013, pp. 1–6. doi:10.1109/DevLrn.2013.6652541.

[5] F. Attneave, Some informational aspects of visual perception, Psychol. Rev (1954) 183–193.

[6] H. B. Barlow, Possible principles underlying the transformation of sensory messages, Cambridge, MA: MIT Press, 1961.

[7] D. J. Field, What is the goal of sensory coding?, Neural Comput. 6 (4) (1994) 559–601. doi:10.1162/neco.1994.6.4.559.

[8] C. Teulière, S. Forestier, L. Lonini, C. Zhang, Y. Zhao, B. Shi, J. Triesch, Self-calibrating smooth pursuit through active efficient coding, Robotics and Autonomous Systems 71 (2015) 3–12.

[9] C. Zhang, Y. Zhao, J. Triesch, B. E. Shi, Intrinsically motivated learning of visual motion perception and smooth pursuit, in: Robotics and Automation (ICRA), 2014 IEEE International Conference on, IEEE, 2014, pp. 1902–1908.

[10] A. Gibaldi, A. Canessa, F. Solari, S. P. Sabatini, Autonomous learning of disparity–vergence behavior through distributed coding and population reward: Basic mechanisms and real-world conditioning on a robot stereo head, Robotics and Autonomous Systems 71 (2015) 23–34.

[11] K. Stroyan, M. Nawrot, Visual depth from motion parallax and eye pursuit, Journal of mathematical biology 64 (7) (2012) 1157–1188.

[12] S. H. Ferris, Motion parallax and absolute distance., Journal of experimental psychology 95 (2) (1972) 258.

[13] M. E. Ono, J. Rivest, H. Ono, Depth perception as a function of motion parallax and absolute-distance information., Journal of Experimental Psychology: Human Perception and Performance 12 (3) (1986) 331.

[14] S. Sengupta, E. Greveson, A. Shahrokni, P. H. Torr, Urban 3d semantic modelling using stereo vision, in: Robotics and Automation (ICRA), 2013 IEEE International Conference on, IEEE, 2013, pp. 580–585.

[15] X. Kuang, M. Gibson, B. E. Shi, M. Rucci, Active vision during coordinated head/eye movements in a humanoid robot, IEEE Transactions on Robotics 28 (6) (2012) 1423–1430.

[16] M. Antonelli, A. P. Del Pobil, M. Rucci, Depth estimation during fixational head movements in a humanoid robot, in: International Conference on Computer Vision Systems, Springer, 2013, pp. 264–273.

[17] A. Saxena, S. H. Chung, A. Y. Ng, Learning depth from single monocular images, in: Advances in neural information processing systems, 2006, pp. 1161–1168.

[18] H. Zhuang, R. Sudhakar, J.-y. Shieh, Depth estimation from a sequence of monocular images with known camera motion, Robotics and autonomous systems 13 (2) (1994) 87–95.

[19] S. Jeong, S.-W. Ban, M. Lee, Stereo saliency map considering affective factors and selective motion analysis in a dynamic environment, Neural networks 21 (10) (2008) 1420–1430.

[20] H. Lee, S. Jeong, N. Y. Chong, Unsupervised learning approach to attention-path planning for large-scale environment classification, in: Intelligent Robots and Systems (IROS 2014), 2014 IEEE/RSJ International Conference on, IEEE, 2014, pp. 1447–1452.

[21] J. Michels, A. Saxena, A. Y. Ng, High speed obstacle avoidance using monocular vision and reinforcement learning, in: Proceedings of the 22nd international conference on Machine learning, ACM, 2005, pp. 593–600.

[22] F. Chaumette, S. Hutchinson, Visual servo control. i. basic approaches, IEEE Robotics & Automation Magazine 13 (4) (2006) 82–90.

[23] C.-Y. Tsai, C.-C. Wong, C.-J. Yu, C.-C. Liu, T.-Y. Liu, A hybrid switched reactive-based visual servo control of 5-dof robot manipulators for pick-and-place tasks, IEEE Systems Journal 9 (1) (2015) 119–130.

[24] Y. Wang, H. Lang, C. W. de Silva, A hybrid visual servo controller for robust grasping by wheeled mobile robots, IEEE/ASME transactions on Mechatronics 15 (5) (2010) 757–769.

[25] J. Law, P. Shaw, M. Lee, A biologically constrained architecture for developmental learning of eye–head gaze control on a humanoid robot, Autonomous Robots 35 (1) (2013) 77–92.

[26] N. Chumerin, A. Gibaldi, S. P. Sabatini, M. M. Van Hulle, Learning eye vergence control from a distributed disparity representation, International journal of neural systems 20 (04) (2010) 267–278.

[27] T. Vikram, C. Teuliere, C. Zhang, B. Shi, J. Triesch, Autonomous learning of smooth pursuit and vergence through active efficient coding, in: Development and Learning and Epigenetic Robotics (ICDL-Epirob), 2014 Joint IEEE International Conferences on, IEEE, 2014, pp. 448–453.

[28] W. E. Vinje, J. L. Gallant, Sparse coding and decorrelation in primary visual cortex during natural vision, Science 287 (5456) (2000) 1273–1276.

[29] H. Zhou, H. S. Friedman, R. Von Der Heydt, Coding of border ownership in monkey visual cortex, Journal of Neuroscience 20 (17) (2000) 6594–6611.

[30] B. A. Olshausen, D. J. Field, Sparse coding with an overcomplete basis set: A strategy employed by v1?, Vision Research 37 (23) (1997) 3311 – 3325.

[31] S. G. Mallat, Z. Zhang, Matching pursuits with time-frequency dictionaries, Signal Processing, IEEE Transactions on 41 (12) (1993) 3397–3415.

[32] S. Bhatnagar, R. S. Sutton, M. Ghavamzadeh, M. Lee, Natural actor–critic algorithms, Automatica 45 (11) (2009) 2471–2482.

[33] J. J. Moré, The levenberg-marquardt algorithm: implementation and theory, in: Numerical analysis, Springer, 1978, pp. 105–116.

[34] J. Holmin, M. Nawrot, Motion parallax thresholds for unambiguous depth perception, Vision research 115 (2015) 40–47.

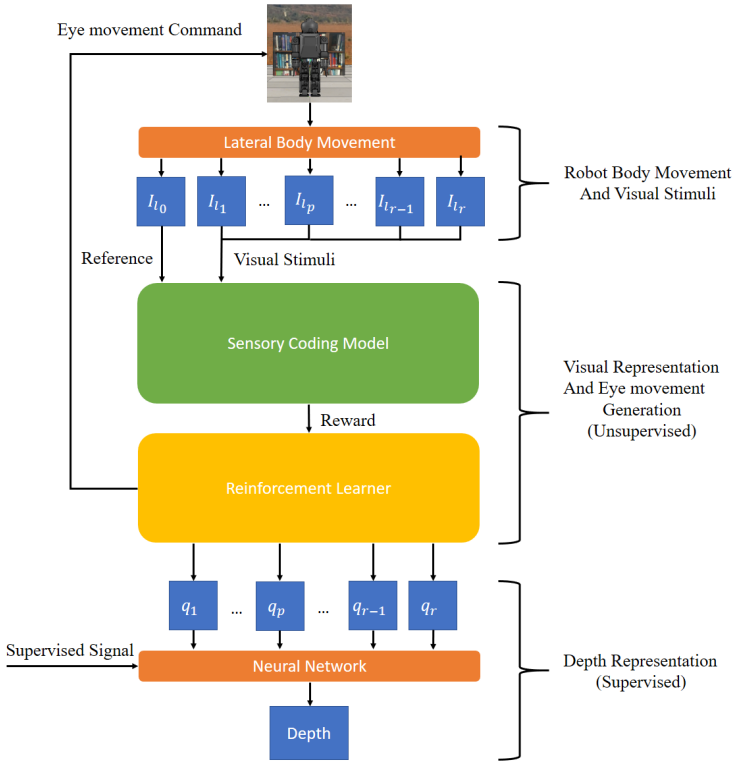
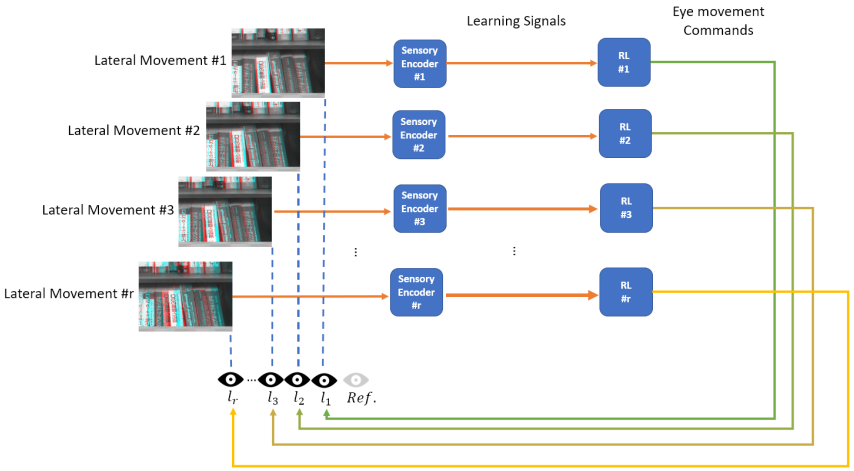
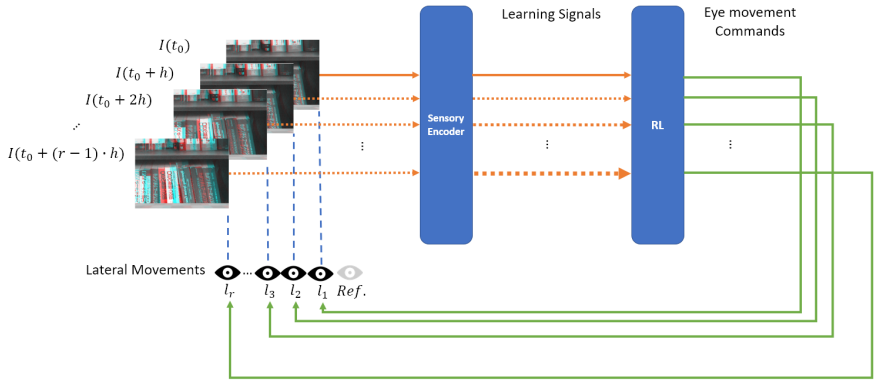


Figure 1: Model architecture. The robot captures a reference image and then moves to the lateral position l_k from L. To perform the motion parallax, the successive images $I(t)$ are inputted into the sensory encoders with multiple image scales. Then, an output reward signal generated from the sensory encoders is sent to the reinforcement learner to generate an appropriate eye movement to hold the fixation during the body movement. Finally, a pan command is sent to the robot which then generates the smooth pursuit eye movement to maximize the redundancy between the input images. The memorized eye movements (q_1, q_2, ..., q_r) are used as an input for the neural network to represent the distance information which is given by human-robot interaction.



(a) Single lateral distance



(b) Multiple lateral distance

Figure 2: (a) shows a learning scheme when using only single lateral movement. It has only one difficulty of learning signal. While, (b) shows the flow of performing the same task but with multiple lateral body movements. It can provide multiple scales of difficulty of the learning signal to the reinforcement learner.

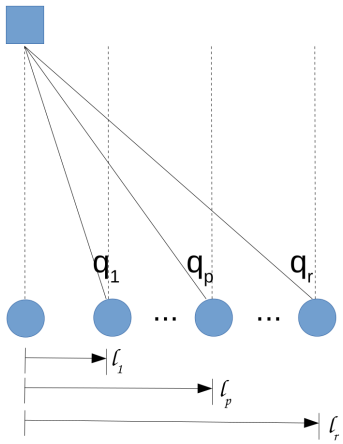


Figure 3: The lateral body movement of the robot and the total eye movements at each position. The robot moves laterally for a certain distance from L. Then it tries to generate eye movements $q_1, q_2, \cdots, q_p, \cdots, q_r$ to fixate the visual stimulus at the center of the gaze.

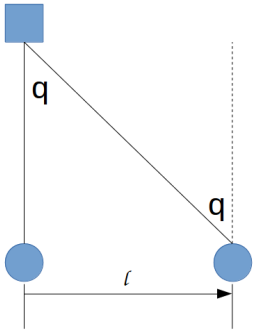


Figure 4: The parallax angle q which is identical to the total eye movement required to fixate the stimulus at a certain lateral distance l.

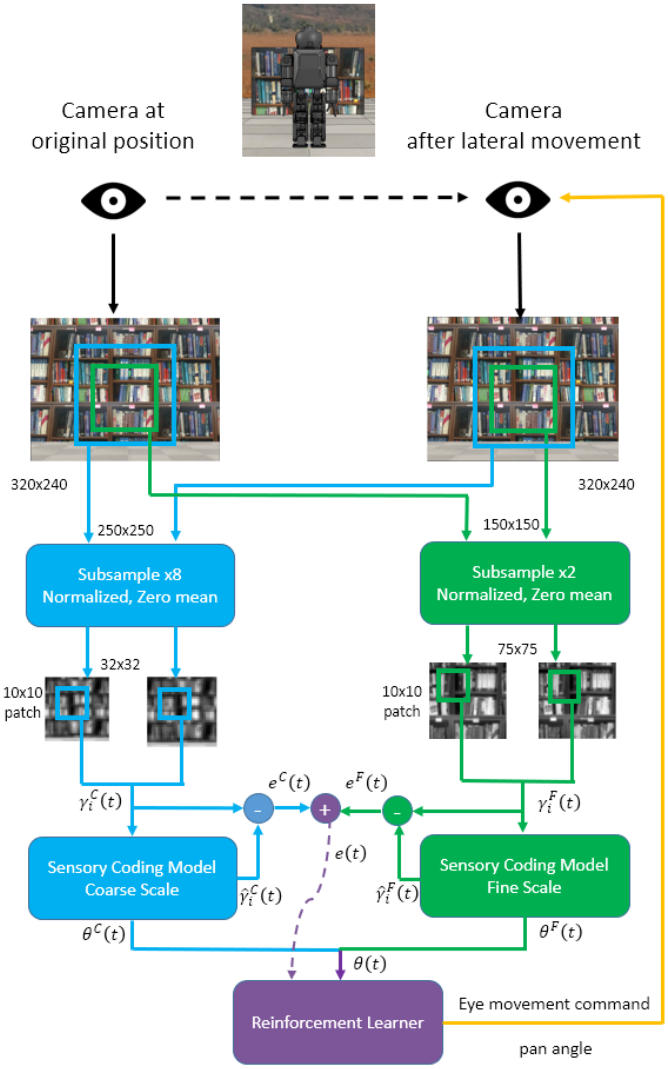


Figure 5: Sensory coding model and eye movement generation. Images captured from the camera are cropped and processed before inputting into the sensory encoders. The cropped images are then sub-sampled and normalized with respect to the two scales which are coarse and fine. The sensory encoders then encode the patches extracted from the processed images. The reinforcement learner takes the pooled activities $\theta^j(t)$ and the errors from the sensory encoders to learn and generate the eye movement (smooth pursuit)

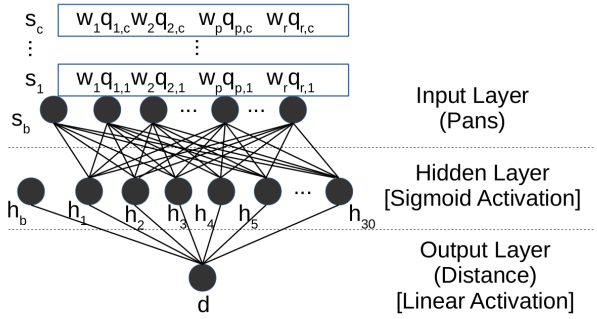


Figure 6: The 3 layers feed forward neural network for estimating the egocentric distance. The feature inputs are the eye movements from each lateral position in L. Sigmoid activation function is used in the hidden layer, while the output layer uses linear activation function. The output layer has only one node which is the absolute distance.

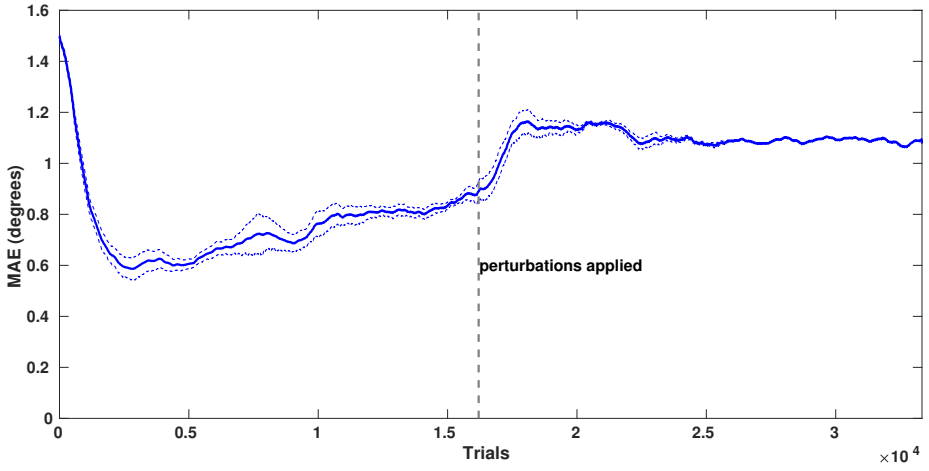


Figure 7: Eye movement MAE of single lateral position at 20 cm after the disturbances

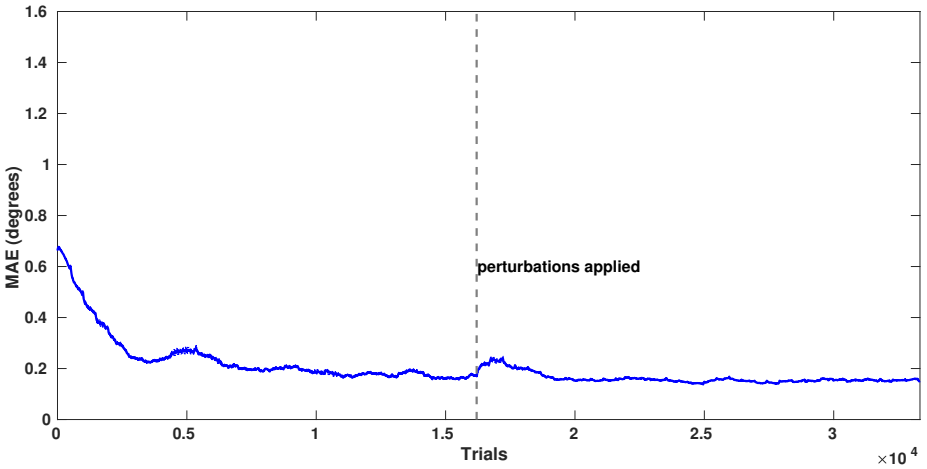


Figure 8: Eye movement MAE of multiple lateral positions 5-10 cm after the disturbances

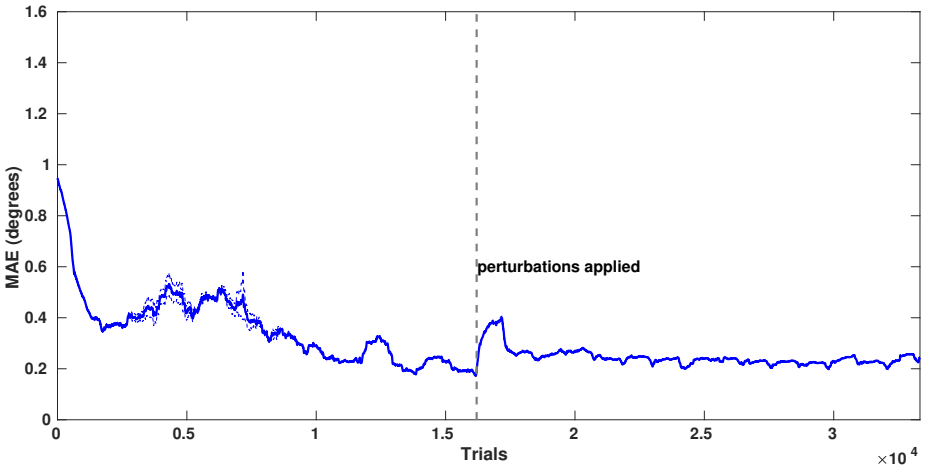


Figure 9: Eye movement MAE of multiple lateral positions 5-20 cm after the disturbances

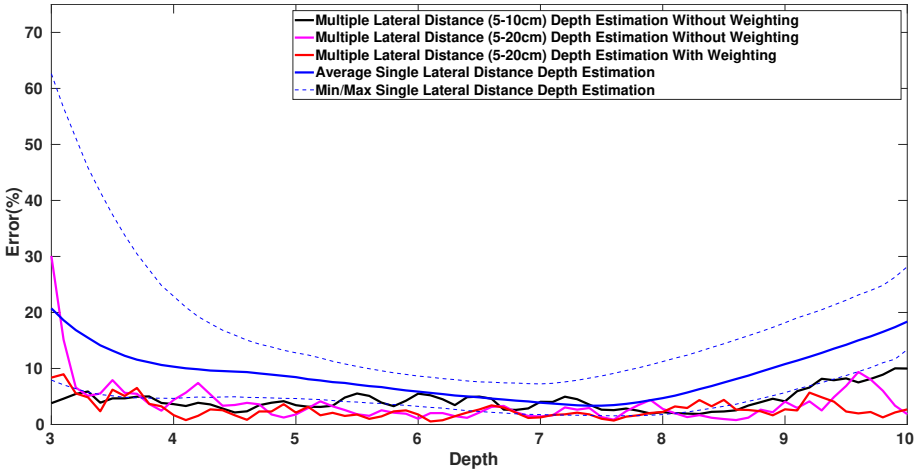


Figure 10: Distance estimation error

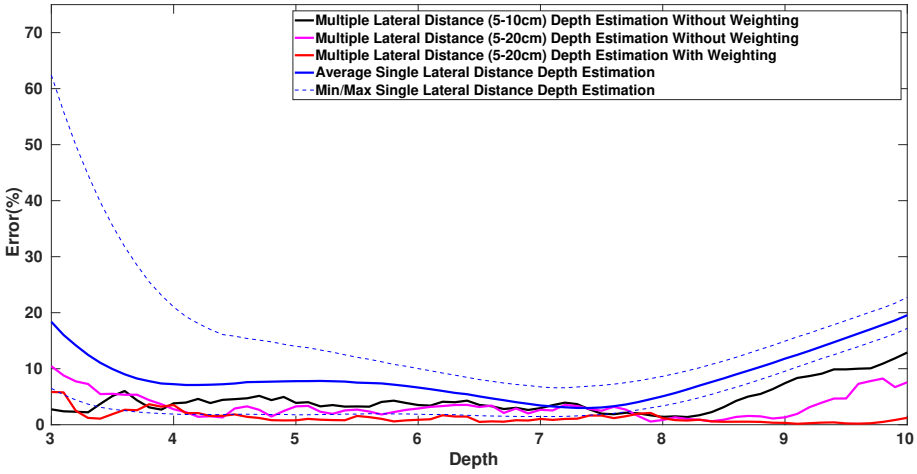


Figure 11: Distance estimation error at each distance after the disturbances