# Saccade Adaptation on a 2 DoF Camera Head

## Tobias Rodemann, Frank Joublin, Edgar Körner

### 2004

**Preprint:**

This is an accepted article published in Third Workshop on SelfOrganization of AdaptiVE Behavior (SOAVE 2004) Ilmenau. The final authenticated version is available online at: https://doi.org/[DOI not available]

*Presented at SOAVE 2004 Workshop, 28.–30. September 2004 in Ilmenau*
*Published in Fortschritts-Berichte VDI, Reihe 10, Nr. 743, pages 94–103*

# Saccade Adaptation on a 2 DOF Camera Head

Tobias Rodemann Frank Joublin
and Edgar Körner
Honda Research Institute Europe
Carl-Legien-Strasse 30, D-63073 Offenbach (Main),
Germany
{Tobias.Rodemann, Frank.Joublin, Edgar.Koerner}@honda-ri.de

**Abstract**

*The ability to saccade to a given point, i.e. to move the fovea on a specific, salient object in the world, is essential for any biological vision system that operates in a realistic environment and also for technical systems with a limited viewing range like cameras. The saccade movement requires a mapping from the retinal position of the target point to motor commands that change the viewing direction of the recording element (camera or eye), i.e. changing the pan and tilt angle of the camera. The precise mapping between retinal position and camera angles is non-linear and may change over time due to mechanical wear-down, changes in environmental conditions (e.g. temperature), or when a new camera system is installed. The continuous operation in a real-world environment therefore requires a* permanent, online adaptation *of the mapping function. In this article we detail a model for an online-adaptation of saccades. The basic idea of the approach is to learn a mapping between points in a retinal coordinate system and corresponding points in a gaze-centered, spherical motor coordinate system. Learning is based on the comparison of visual inputs on the retina before and after a saccade and the relation to a motor command.*

## 1 Introduction

Saccades are used to bring the fovea of the eye on a new, relevant target. The position of the target stimulus is normally defined in a different coordinate system than that of head, eye, or camera motors. Most target stimuli are selected visually, they are naturally referenced by their retinal coordinates. The easiest representation for accessing eye-, camera- and head-motors is a rotational motor-coordinate system. The task for learning saccade control is therefore to go from a retinal representation of the interesting stimuli to a motor-coordinate system. This mapping can be very complex and also be subject to variations due to changes in eye-head sensor or motor parameters (e.g. different lens, motor damage). Therefore it is necessary to have an adaptation mechanism to learn the mapping and automatically correct errors in the saccade generation mechanism shortly after they appear. To allow an operation of the system over longer periods of time an *on-line* adaptation is necessary. This is especially important in the context of a humanoid robot system like Honda's humanoid robot Asimo.

## 1.1 Biological Background

Current biological data [1, 5] suggests that there are two different centers for saccade control: the Superior Colliculus (*SC*) and the Frontal Eye Field (*FEF*). The *SC* is responsible for fast, reflexive saccades to salient targets. It consists of several layers. Collicular neurons in the upper layers are mostly fed by visual input, while in the deeper layers also auditory and somato-sensory inputs arrive. In the deepest layers neurons send outputs to the motor centers of the brain stem. The *FEF* is a cortical structure and seems to be the relay center for voluntary saccades, e.g. for scene exploration. It interacts with and often inhibits the *SC*. It is also active for saccades to targets in visual memory. An important structure for the proposed saccade adaption mechanism is the cerebellum that seems to be needed to adapt motor commands including gaze shifts [8, 6]. Our proposed algorithm covers the interaction between *SC, FEF* and cerebellum for adapting the saccade control. It seems that saccade targets are chosen, based on visual or other sensory input, in either the SC or the frontal eye fields. These structures can then initiate and control the movement of eye, head and torso. Learning and adaptation of saccade control involves the cerebellum.

## 1.2 Embedding into the Brain-like Active Sensing System

The work is embedded into a larger endeavor striving for an integrated system for active sensing and pattern recognition (publication in preparation). The set-up for the whole system consists of a neck and camera system that provides images (see Fig. 1), some preprocessing modules including saliency point computation, an object identification system, modules for gaze target selection, modules for the memory trace and inhibition of return, and the modules for online-adaptation of gaze control. The neck can change its pan and tilt angles but otherwise the system is considered to be fixed. In the current system only one camera (left eye) will be used for saccades while the other camera operates in parallel without an extra control for vergence. A subtask for the system is the fovealization of objects. The target's retinal position is selected by the gaze selection algorithm described below and a number of targets are supposed to be inspected in a scene. The correspondence between retinal position and required camera movement can be calculated analytically if the exact camera geometry is known and the camera has been calibrated. However, camera calibration is a difficult and time consuming process and not easily performed on-line because it normally negatively affects the normal operation of the system. Exchanging parts of the camera system (e.g. the lens) will require a quick recalibration and/or modification of the explicit mapping function, while operation in a real-world scenario will surely lead to slow but continuous changes. We therefore believe that only a permanent, online adaptation of the saccade control mechanism (based on errors of the gaze control system), similar to the one observed in the primate oculo-motor system, is a satisfactory solution. In fact, for biological systems it is the only viable solution.

## 1.3 Gaze selection mechanism

Selection of targets is based on the output of a saliency computation module. This saliency map is transformed into motor coordinates so that selected positions can trivially be saccaded to (the position in the map directly corresponds to a vector of motor angles). Generally the most salient target is selected. However, different factors can also contribute. Certain locations can be blocked (e.g. to limit the camera range to mechanically possible values) and others can be specifically attended to (attention mechanism). Furthermore additional, memory based stimuli can be used. The dynamics of the gaze selection mechanism also lead to interactions between several potential targets. To avoid selecting the optimal stimulus again a previously selected stimulus is inhibited for some time [2]. The characteristics of the gaze selection mechanism have an influence on the choice of the saccade

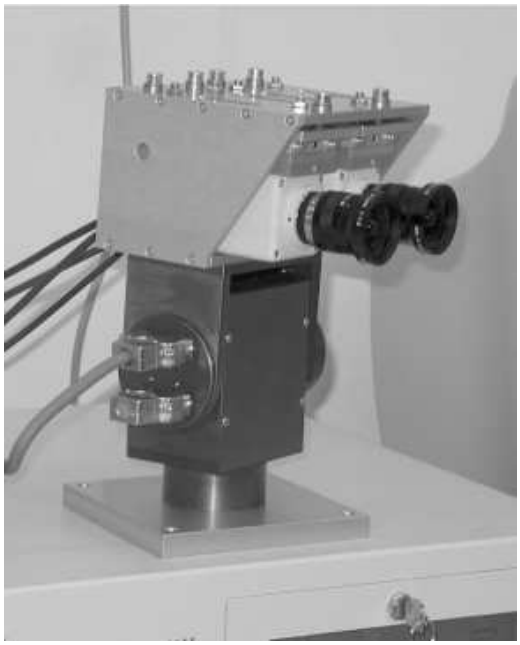

*Figure 1. The stereo cameras on a pan-tilt head element.*

adaptation model that can be used. The gaze selection scheme described here requires a mapping from retinal to motor coordinates which should be learned. It also decides about visual targets (i.e. saccade targets are not chosen by the saccade learning algorithm). An important aspect is that due to the inhibition the chosen targets before and after the saccade will not be same. For more details on the gaze selection algorithm see [4].

For the specific algorithm that is the focus of this paper it is not important how the saccade target is selected. The mechanism described hereafter will function even with a random selection of targets.

# 2 Methods

We want to learn the mapping from retinal coordinates to motor commands on-line during the normal operation of the system. The mapping is represented by an array $W(x, y)$ of vectors from retinal positions $(x, y)$ to motor commands ($\phi, \theta$). Learning is based on the comparison of images before and after the saccade. By finding corresponding positions in the images and aligning them with the executed motor command we can learn the relation between changes in the visual input and changes in the system's viewing direction. We now describe the different aspects of the algorithm in more detail.

## 2.1 Representing the Mapping

The mapping between image coordinates $(x, y)$ and motor coordinates $(\phi, \theta)$ is described by the referencing matrix $W(x, y)$. Each entry of $W$ contains a two dimensional vector that represents the corresponding position in motor space. After learning, $W(x, y)$ is the command that moves position $(x, y)$ into the fovea. The size of $W$ is given by the number of pixels in the camera image, while the size of the motor space is limited by the precision of the camera motors. Each retinal position is associated with just one motor position. In this article we generally assume that the size of the retinal space is as big or bigger than the size of motor space. For the case of a larger target space, i.e. more motor positions than image pixels, single pixels have to be mapped to a region in motor space. On

the other hand for a larger image space several pixels are mapped to a single motor position. In this case an appropriate preprocessing should be performed (e.g. smoothing the image before mapping it to motor space). It is possible to map the full visual image to motor space and decide to which position to actually saccade to in this representation or, alternatively, to select a single point in image coordinates and transform this one into the corresponding motor commands. For our case it was easier to perform the target selection in motor coordinates, because in this coordinate system information about previously targeted objects can be more easily integrated. Therefore, we first map the whole visual input (saliency map) to motor space and then decided on a saccade target.

## 2.2 Image Correspondences

We have to align gaze movements, represented by motor commands $\vec{m}$, with changes in the position of objects on the retina. There are two methods which can be used: The first one consists of comparing the position of the saccade target before and after the saccade. This algorithm is effective in a very simple environment with only a few salient stimuli. The second approach, the one used in our algorithm, looks for the position of the patch in the pre-saccadic image that ends up in the fovea after the saccade. This position is then obviously moved by the used motor-command onto the fovea. This algorithm requires a feature-rich visual environment to clearly identity the correct patch in the pre-saccadic image. Because this approach directly links image positions with the correct motor command for fovealization no approximative calculation is required. Furthermore we want our system to operate in a normal, visually-rich environment. Therefore we prefer this algorithm over the first option (tracking a single salient target). To get an impression of the visual environment we used, take a look at Fig. 2.

*Figure 2. Two example snap-shots of the visual environment used for testing our adaptation mechanism. The target of the vision system is fovealizing objects on the table.*

We compute a function $C(x, y)$ (called correspondence) that is a measure of the similarity between a defined foveal window after the saccade and an image patch around position $(x, y)$ in the pre-saccadic image. For the correspondence functions between the two patches ($\vec{r}^f$ is the foveal patch in the post-saccadic image and $\vec{r}(x, y)$ a patch from the pre-saccadic image around position $(x, y)$) we use the following equation:

$$C(x,y) = \frac{\vec{r}^f \cdot \vec{r}(x,y)}{||\vec{r}^f|| \cdot ||\vec{r}(x,y)||} \quad . \tag{1}$$

This is a simple template matching operation with a normalization for brightness. The point of maximum correspondence $(x_{max}, y_{max})$ is now taken as the position in the image patch that was moved by the motor command $\vec{m}$ to the fovea. We thus have to learn the association between $\vec{m}$ and $(x_{max}, y_{max})$.

## 2.3 Learning Algorithm

We can now update the connection matrix $W(x, y)$ in the following way. For a given motor-command $\vec{m} = (i_m, j_m)$ and the best correspondence value found at position $(x_{max}, y_{max})$ we use the following learning rule:

$$\Delta W(x_{max}, y_{max}) = -\alpha \cdot \kappa \cdot (W(x_{max}, y_{max}) - \vec{m}) \quad , \tag{2}$$

with $\alpha$ as the learning step size (a fixed parameter) and $\kappa$ as the confidence value (see section 2.4). In this equation only one reference vector is adapted. Performance can be improved by adapting not only one reference vector but all vectors in the vicinity of the best matching one. As an example we take a Gaussian neighborhood function, wherein we reduce the amount of adaptation with increasing distance from the best matching vector:

$$\Delta W(x, y) = -\alpha \cdot \exp\left(-\frac{(x - x_{max})^2 + (y - y_{max})^2}{\sigma^2}\right) \cdot \kappa \cdot (W(x, y) - \vec{m}) \quad . \tag{3}$$

In this equation $\sigma$ is the width of the adaptation region (a system parameter). How to dynamically set $\sigma$ is explained in section 2.5.

## 2.4 Confidence Measure

Under real world conditions, wrong or multiple correspondences between image patches are very likely. Learning with wrong inputs will disrupt the mapping which is highly undesirable as the system is supposed to operate normally during learning. To avoid this problem we compute a confidence measure $\kappa$ (see eqn. 3). We start from our calculation of the correspondence matrix and take the maximum value $c_{max}$. The higher $c_{max}$ the higher the confidence in the correspondence search. In a first step we compute the confidence value $\kappa'$ by the following equation:

$$\kappa' = \frac{1}{1 + \exp(-c_s * (c_{max} - c_t))} \quad . \tag{4}$$

This is a sigmoid with threshold value $c_t$ and a slope $c_s$. This function is used to adapt the confidence value to the statistics of correspondence values. We compute $c_t$ and $c_s$ through a common parameter ($\tau$) by $c_s = 10.0/(1.0 - \tau)$ and $c_t = (1 + \tau)/2.0$. The parameter $\tau$ represents a lower bound on the accepted correspondence level. The purpose of this processing step is to get strongly peaked correspondence map. To reduce the confidence in case of multiple good matches (a common problem for homogeneous patches) we perform a normalization of the confidence value by the number of entries in the correspondence map with value above a threshold of $T = R * c_{max}$, where $R$ is the percentage of the maximum correspondence value to count as a competing match. All entries with correspondence values above $T$ count as other potential matching candidates and increase the normalization factor. In our implementation we also weight these additional good matches by their distance to the optimum. Other good matches close to the best one are weighted less than distant ones. The characteristic range is defined by $\sigma_C = r_t \cdot$ *size of retina/image*. The parameter $r_t$ gives the tolerance radius. It is basically the width of one peak in the correspondence map. Every entry in the correspondence map with a value above $T$ will increase the normalization factor by the following value $N(x, y)$:

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

With this the final confidence value can be computed to:

$$\kappa = \kappa' \cdot \frac{1}{1 + \sum_{x,y} N(x,y)} \quad . \tag{6}$$

We used the following parameter values in our simulation: $\tau = 0.5, R = 0.9, r_t = 0.1$. The choice of the parameters has to be adapted to the visual environment, especially the statistics of the correspondence values. The exact choice of $\tau$ and $r_t$ is not very critical, however, $R$ has to be rather precisely adapted to the statistics of the data and the way the correspondence is calculated.

## 2.5 Adapting the Learning

We need to set the learning parameters, more specifically the step-size of adaptation $\alpha$ (how much do we move the reference vector to its target position) and the width of the population that is adapted ($\sigma$). It turned out that the step size can be kept constant. A value of 0.8 led to good results. To adapt the population width $\sigma$ we compute the saccade error — the difference between the planned and the actual position of the target after the saccade. We use the mean saccade error $\bar{E}$ (averaged over for example the last 10 saccades) as a measure to modify the adaptation width:

$$\sigma = \sigma^{max} \cdot \frac{1}{(1 + \exp(-s * (\bar{E}/E_{max} - t)))} \quad . \tag{7}$$

In this equation $\sigma^{max}$ is the maximum adaptation width (a parameter that has to be set in relation to the size of the source map, e.g. 30% of the image size), $E_{max}$ the maximum possible saccade error (a normalization constant), $s$ the slope of the sigmoid, and $t$ the threshold of the sigmoid. These parameters determine how much adaptation is performed depending on the amount of saccadic error. This mechanism ensures that for a well-adapted mapping only smaller changes are performed but that as soon as bigger saccade errors appear, e.g. after changing the lens, the adaptation rate will increase and the mapping will quickly adapt to the new situation. In our simulation we used $\sigma_{max} = 30\%$ of the size of the retina, $s = 20$ and $t = 0.2$.

## 2.6 Linearization of the Map

The reduced model has many similarities with a standard self-organizing map (*SOM*) of the Kohonen type (see [3, 7]). We use a two-dimensional lattice of nodes (in retinal space) with attached reference vectors into another two dimensional space. The main difference to the standard *SOM* model is how these reference vectors are adapted. In the standard *SOM* nodes with reference vectors close to an input vector are adapted. In our model the choice of nodes to be adapted is determined by the point of best correspondence in the retinal map. The reference vectors of these nodes are then moved toward the motor command vector. A further important extension is the result of a, normally minor, problem in SOMs: It is well known that standard SOMs do not fully fill the space of input vectors (or in our case the motor space), but rather 'shrink' away from the borders. For us that means that motor commands at the boundaries of the allowed motor range are not targeted by the mapping. In the implemented system we transform a retinal saliency map to motor space and then select a target there. Because of the border shrinkage effect no motor commands from the border of motor space will be generated. Therefore the effective training space for the SOM will shrink—creating a new border. The mapping will now again shrink away from the new borders until all retinal positions are mapped to the center of motor space. The basic learning algorithm locally adapts reference vectors toward the executed motor command. Thereby we assume a flat local mapping, as we move all local reference vectors toward

the same motor-command. As an extension and solution to the map shrinking problem we propose to use a local *linear* mapping, that is we assume that reference vectors around the optimum point are moved to a position $m_{LLM}$ that is a linear function of the motor command and the distance to the best matching vector:

$$\vec{m}_i^{LLM} = \vec{m}_x + (x - x_{max}) \cdot s_x \cdot F_x \tag{8}$$
$$\vec{m}_j^{LLM} = \vec{m}_y + (y - y_{max}) \cdot s_y \cdot F_y \tag{9}$$

and we get the new equation for the adaptation step:

$$\Delta W(x,y)_i = -\alpha \cdot g_\sigma(x,y) \cdot \kappa \cdot (W(x,y)_i^t - \vec{m}_i^{LLM}) \tag{10}$$
$$\Delta W(x,y)_j = -\alpha \cdot g_\sigma(x,y) \cdot \kappa \cdot (W(x,y)_j^t - \vec{m}_j^{LLM}) \tag{11}$$

with $g_\sigma(x,y)$ as the Gaussian neighborhood function as used in equation 3. The linear model is represented by two variables. The first one is the sign factor $s_x$ (and $s_y$) which is computed by multiplying the signs of the motor vector and the retinal position vector $P_x$: $s_x = sign(m_x * P_x)$. By this we are capable of adapting even to inversions in the relation between retinal coordinates and motor commands[1]. The second parameter of the linear model is the slope $F_x$ which is calculated from the dimensions of the image and the motor map: $F_x$ = width of motor map / width of image map. This factor is fixed and ensures that the map is extended to the full size. The same relation also applies to $F_y$ but using the height of retina and motor map. Simulations have shown that the local linearization extension improves learning speed, increases the smoothness of the map and prevents map shrinkage. For strongly non-linear mappings the local linearization helps to pre-structure the map. With a reduction in the adaptation width the effect of the linearization decreases so that also non-linear functions can be learned. For an example on how the local linearization works for a simple test function see Fig. 3.

# 3 Results

We tested our algorithm in a real world scenario with the described vision system. Initially the mapping was set to random and the system was designed to saccade to a number of salient objects presented in a table scenario. The system was able to learn the mapping between image and motor coordinates, which is not trivial due the geometry of our camera head. After 100 time steps (= saccades) we introduced a prism effect that mirrored the image on the horizontal axis (upside-down). Again the system quickly adapted (without any user interference). The system performed the saccading and object recognition task in parallel to the learning, which was never disabled. Figures 4 and 5 show an example run of our algorithm. The final accuracy of the system after a longer adaptation period (ca. 1000 saccades) is in the order of 5% of the maximum saccade error. This sometimes requires a second corrective saccade to fovealize a target, as it is also observed for the human eye, where the error after the first saccade can reach up to 25% of the saccade length. We note that precision is inherently limited because we can't get the exact 3D position of the target (no distance cue) which would be necessary for a correct mapping. Furthermore we used a very small motor map with a limited resolution. For our purposes the achieved precision of the saccades was satisfying.

[1]For this model to work we have to shift retinal and motor coordinates so that the center position for both is at (0,0) and we therefore get both positive and negative values.

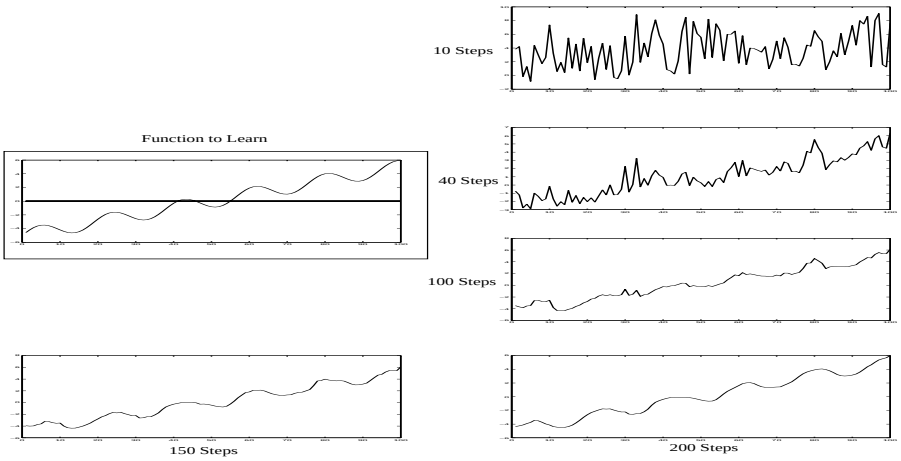


*Figure 3. The performance of the learning algorithm with local linearization on a simple function which consists of a linear term and a superimposed sine term. The function to be learned is depicted in upper left plot. The algorithm starts from random initialization and first learns the linear term (panels for 10, 40, 100 steps) and then later the sinusoidal, superimposed function (bottom row).*

# 4 Summary, Conclusion and Outlook

We have presented a biologically inspired saccade adaptation system for a pan-tilt camera head system based on the analysis of visual input before and after the saccade. Learning can operate during the normal operation of the system (visual scene exploration), and can continuously adapt the system. This makes any (re-)calibration of the system unnecessary even when modifying or exchanging neck, camera, or any software preprocessing stages. The adaptation is very flexible and provides the required accuracy for our purposes. The adaptation system is very fast (about 100 ms per saccade on conventional hardware ) and can therefore be used for real-time robotics applications. It does not depend on specific aspects of camera system, saliency computation or gaze target selection. A similar learning mechanism could also be used for other aspects of the gaze control system like the Vestibular-Ocular reflex or binocular viewing (vergence control).

A necessary extension of the algorithm is to include depth information to improve the quality of the mapping. This would require e.g. a stereo camera-based distance measurement of the target location. Another important aspect is an improvement in the correspondence matching algorithm. The current version is rather slow and still faces the problem of mismatches, while the number of parameters especially for the confidence calculation is too large. Finally we want to go beyond the linearization approach for dealing with the map shrinking problem, because it requires the specification of motor and image map sizes.

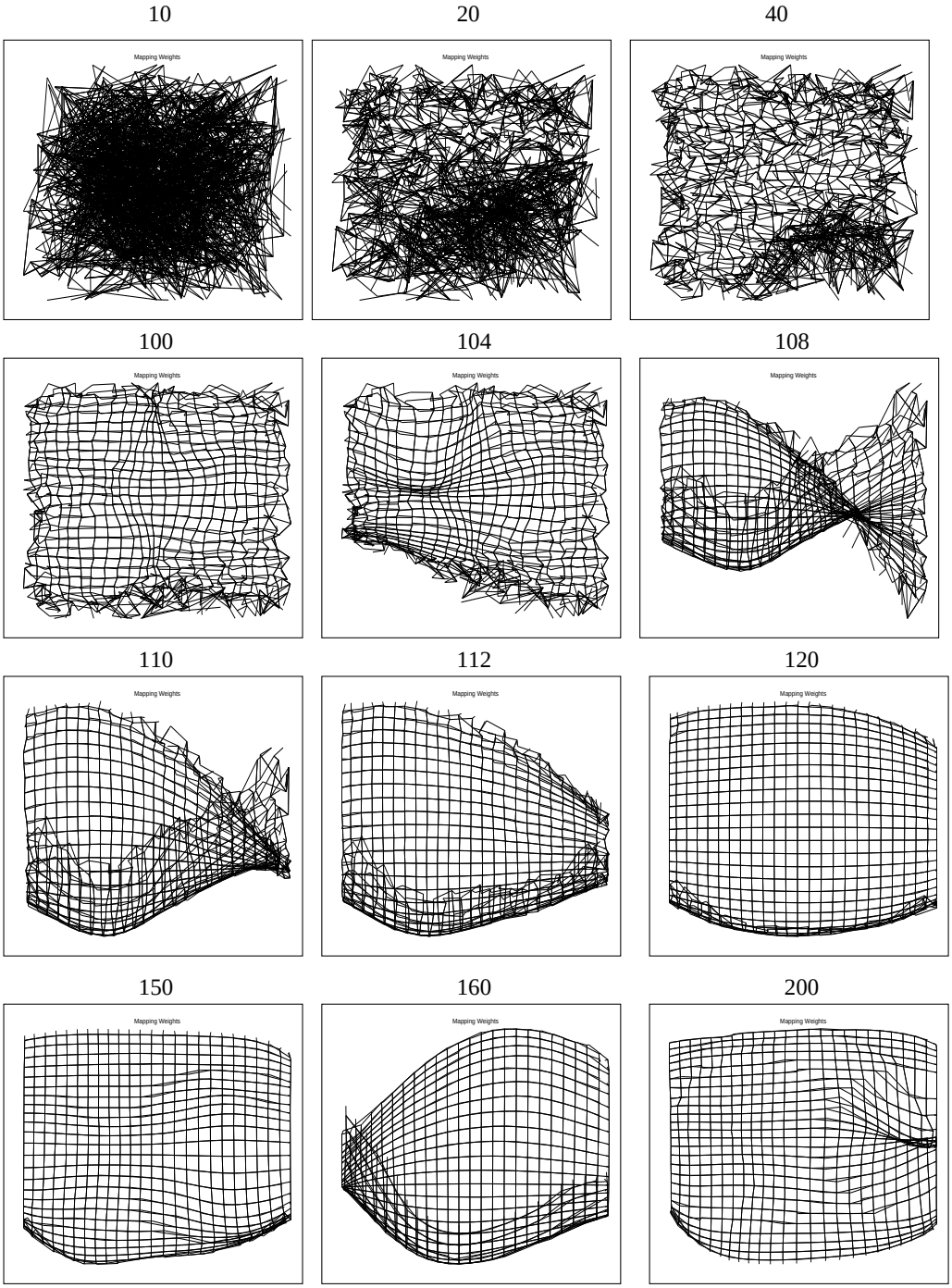


*Figure 4. Mapping vectors for an example run of the saccade learning algorithm. Grid-points denote positions in motor space, while edges represent neighborhood relations in retinal space. Mapping vectors are initialized randomly and adapt to form a regular grid within a few 10 steps (top row). At step 100 an up-down inverting prism is added and the mapping quickly adapts to the new situation by flipping around (rows 2 and 3). The final row depicts the development when the prism is removed at step 150 and the mapping flips again.*

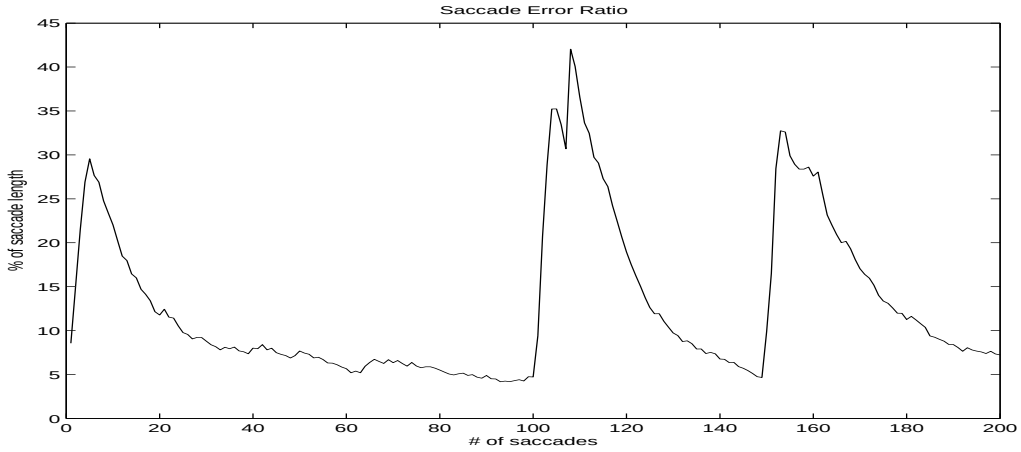


*Figure 5. Development of the saccade error for the simulation run depicted in Fig. 4. One can easily see that the system quickly adapts from both a random mapping and an inversion of the visual input within a few 10 time-steps. Note that the saccade error is smoothed over many time steps, because it shows a high variety between consecutive saccades. This is also the reason for the low initial saccade error.*

# References


[1] Charles J. Bruce and Harriet R. Friedman. *Encyclopedia of the Human Brain*, volume 2, chapter Eye Movements, pages 269–297. 2002.

[2] Laurent Itti and Christof Koch. A saliency-based search mechanism for overt and covert shifts of visual attention. *Vision Research*, 40:1489–1506, 2000.

[3] T. Kohonen. *Self-Organization and Associative Memory*. Springer-Verlag, Berlin, 3rd edition, 1989.

[4] Inna Mikhailova. Saliency-based gaze direction control for active vision. Diplomarbeit, Technische Universität Darmstadt, Juni 2003.

[5] Douglas P. Munoz and Stefan Everling. Look away: The anti-saccade task and the voluntary control of eye movement. *Nature Reviews Neuroscience*, 5:218–228, February 2004.

[6] Denis Péllison, Laurent Goffart, and Alain Guilaume. *Progress in Brain Research*, chapter Control of saccadic eye movements and combined eye/head gaze shifts by the medio-posterior cerebellum. Elsevier Science, 2003.

[7] H. Ritter, T. Martinetz, and K. Schulten. *Neuronale Netze.* Addison–Wesley, 1990.

[8] Xiaoxing Wang, Jesse Jin, and Marwan Jabri. Neural network models for the gaze shift system in the superior colliculus and cerebellum. *Neural Networks*, 15:811–832, 2002.