Learning a Visual Forward Model for a Robot Camera Head

Wolfram Schenck and Ralf Möller

Computer Engineering Group, Faculty of Technology, Bielefeld University,
Bielefeld, Germany
wschenck@ti.uni-bielefeld.de

Abstract. Visual forward models predict future visual data from the previous visual sensory state and a motor command. The adaptive acquisition of visual forward models in robotic applications is plagued by the high dimensionality of visual data which is not handled well by most machine learning and neural network algorithms. Moreover, the forward model has to learn which parts of the visual output are really predictable and which are not. In the present study, a learning algorithm is proposed which solves both problems. It relies on predicting the mapping between the visual input and output instead of directly forecasting visual data. The mapping is learnt by matching corresponding regions in the visual input and output while exploring different visual surroundings. Unpredictable regions are detected by the lack of any clear correspondence. The proposed algorithm is applied successfully to a robot camera head with additional distortion of the camera images by a retinal mapping.

1 Visuomotor Prediction

Sensorimotor control is an important research topic in many disciplines, among them cognitive science and robotics. These fields tackle the questions how complex motor skills can be acquired by biological organisms or robots, and how sensory and motor processing are interrelated to each other. So-called “internal models” help to clarify ideas of sensorimotor processing on a functional level [8, 13]. “Inverse models” or controllers generate motor commands based on the current sensory state and the desired one; “forward models” (FWM) predict future sensory states as outcome of motor commands applied in the current sensory state. The present study focuses on the anticipation of visual data by FWMs.

The anticipation of sensory consequences in the nervous system of biological organisms is supposed to be involved in several sensorimotor processes: First, many motor actions rely on feedback control, but sensory feedback is generally too slow. Here, the output of FWMs can replace sensory feedback [9]. Second, FWMs may be used in the planning process for complex motor actions [12]. Third, FWMs are part of a controller learning scheme called “distal supervised learning” [7]. Fourth, FWMs can help to seperate self-induced sensory effects (which are predicted) from externally induced sensory effects (which stand out from the predicted background) [2]. Fifth, it is suggested that perception relies on the anticipation of the consequences of motor actions which could be applied in the current situation. For the anticipation, FWMs are needed [10].

Regarding the fourth function mentioned above, a classical example is the reafference principle suggested by Holst and Mittelstaedt [6]. It explains why (self-induced) eye movements do not evoke the impression that the world around us is moving. As long as the predicted movement of the retinal image (caused by the eye movement) coincides with the actual movement, the effect of this movement is canceled out in the visual perception.

In fields like robotics or artificial life, studies using FWMs for motor control focus mainly on navigation or obstacle avoidance tasks with mobile robots. The sensory input to the FWMs are rather low-dimensional data from distance sensors or laser range finders (e.g.: [12, 14]), optical flow fields [3], or preprocessed visual data with only a few remaining dimensions [5].

We are especially interested in the learning of FWMs in the visual domain, and its application to robot models. In our understanding, visual FWMs predict representations of entire visual scenes. In the nervous system, this could be the relatively unprocessed representation in the primary visual cortex or more complex representations generated in higher visual areas. Regarding robot models, the high-dimensional sensory input and output space of visual FWMs poses a tough challenge to any machine learning or neural network algorithm. Moreover, there might be unpredictable regions in the FWM output (because parts of the visual surrounding only become visible after execution of the motor command). In the present study, we suggest a learning algorithm which solves both problems in the context of robot "eye" movements. In doing so, our main goal is to demonstrate a new efficient learning algorithm for image prediction.

2 Visual Forward Model for Camera Movements

In our robot model, we attempt to predict the visual consequences of eye movements. In the model, the eye is replaced by a camera which is mounted on a pan-tilt unit. Prediction of visual data is carried out on the level of camera images. In analogy to the sensor distribution on the human retina, a retinal mapping is carried out which decreases the resolution of the camera images from center to border. We use this mapping to make the prediction task more difficult; we do not intend to develop, implement, or test a model of the human visual pathway. The input of the visual FWM is a "retinal image" at time step t (called "input image" in the following) and a motor command $\mathbf{m}_t$. The output is a prediction of the retinal image at the next time step $t + 1$ (called "output image" in the following; see left part of Fig. 1).

The question is how such an adaptive visual FWM can be implemented and trained by exploration of the environment. A straight-forward approach is the use of function approximators which predict the intensity of single pixels. For every pixel $\langle x_{\mathrm{Out}}, y_{\mathrm{Out}} \rangle$ of the output image, a specific forward model $\mathrm{FWM}_{\langle x_{\mathrm{Out}}, y_{\mathrm{Out}} \rangle}$ is acquired which forecasts the intensity of this pixel (see right part of Fig. 1). Together, the predictions of these single FWMs form the output image as in Fig.

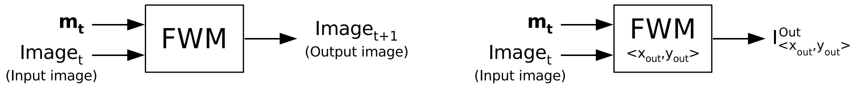

Fig. 1. Left: Visual forward model (FWM). Right: Single component of a visual forward model predicting the intensity of a single pixel $\langle x_{\mathrm{Out}}, y_{\mathrm{Out}} \rangle$ of the output image.

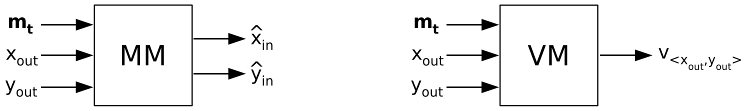

Fig. 2. Left: Mapping model (MM). Right: Validator model (VM) (for details see text).

1 (left). Unfortunately, this simple approach suffers from the high dimensionality of the input space (the retinal image at time step t is part of the input), and does not produce satisfactory learning results [4].

Hence, in this study we pursue a different approach. Instead of forecasting pixel intensities directly, our solution is based on a "back" prediction of where a pixel of the output image was in the input image before the camera's movement. The necessary mapping model (MM) is depicted in Fig. 2: As input, it receives the motor command $\mathbf{m}_t$ and the location of a single pixel $\langle x_{\mathrm{Out}}, y_{\mathrm{Out}} \rangle$ of the output image; as output it estimates the previous location $\langle \widehat{x}_{\mathrm{In}}, \widehat{y}_{\mathrm{In}} \rangle$ of the corresponding pixel (or region) in the input image. The overall output image is constructed by iterating through all of its pixels and computing each pixel intensity as $\widehat{\mathrm{I}}^{\mathrm{Out}}_{\langle x_{\mathrm{Out}}, y_{\mathrm{Out}} \rangle} = \mathrm{I}^{\mathrm{In}}_{\langle \widehat{x}_{\mathrm{In}}, \widehat{y}_{\mathrm{In}} \rangle}$ (using bilinear interpolation).[1] Moreover, an additional validator model (VM) generates a signal $v_{\langle x_{\mathrm{Out}}, y_{\mathrm{Out}} \rangle}$ indicating whether it is possible at all for the MM to generate a valid output for the current input. This is necessary because even for small camera movements parts of the output image are not present in the input image. In this way, the overall FWM (Fig. 1, left) is implemented by the combined application of a mapping and a validator model.

The basic idea of the learning algorithm for the MM is outlined in the following for a specific $\mathbf{m}_t$ and $\langle x_{\mathrm{Out}}, y_{\mathrm{Out}} \rangle$. During learning, the motor command is carried out in different environmental settings. Each time, both the actual input and output image are known afterwards, thus the intensity $\mathrm{I}^{\mathrm{Out}}_{\langle x_{\mathrm{Out}}, y_{\mathrm{Out}} \rangle}$ is known as well. It is possible to determine which of the pixels of the input image show a similar intensity. These pixels are candidates for the original position $\langle x_{\mathrm{In}}, y_{\mathrm{In}} \rangle$ of the pixel $\langle x_{\mathrm{Out}}, y_{\mathrm{Out}} \rangle$ before the movement. Over many trials, the pixel in the input image which matches most often is the most likely candidate for $\langle x_{\mathrm{In}}, y_{\mathrm{In}} \rangle$

[1] In this study, pixel intensities of the retinal input and output images are three-dimensional vectors in RGB color space.

and chosen as MM output $\langle \widehat{x}_{\text{In}}, \widehat{y}_{\text{In}} \rangle$. When none of the pixels matches often enough, the MM output is marked as non-valid (output of VM).

3 Method

To acquire such a MM and VM as in Fig. 2, the following steps are executed. First, a grid of points is defined in the input space of the MM and VM (composed of $\mathbf{m}_t$ and $\langle x_{\text{Out}}, y_{\text{Out}} \rangle$), ranging from the minimum to the maximum value in each input dimension. For each grid point, the most likely estimate $\langle \widehat{x}_{\text{In}}, \widehat{y}_{\text{In}} \rangle$ is determined by collecting candidate pixels in many different visual surroundings. Along the way, the VM output $v_{\langle x_{\text{Out}}, y_{\text{Out}} \rangle}$ is determined as well. Thereafter, one radial basis function network (RBFN) is trained to interpolate the MM output between the grid points, and another RBFN to interpolate the VM output. The resulting networks can be applied to image prediction afterwards. In the following, the methods are outlined in more detail.

3.1 Setup

The robot setup is shown in Fig. 3 (left). Only the right camera is used. A central quadratic region of the original camera image (captured in RGB color) with a resolution of 240 × 240 pixels is used for further processing (and called "camera image" in the following for simplicity). The horizontal and the vertical angle of view of this region amount to 48.5 degrees. The camera is mounted on a pan-tilt unit with two degrees of freedom. In this study, the valid range for the pan angle is between -60.4 and 23.8 degrees, for the tilt angle between -42.9 and 21.4 degrees. In this range, the camera image always captures at least a small part of the white table shown in Fig. 3 (left) below the cameras.

The pan and tilt axes cross in close vicinity to the nodal point of the camera-lens system. For this reason, the effect of changing the pan and tilt position by a certain amount Δpan/Δtilt is almost independent of the current camera position. Accordingly, the motor input $\mathbf{m}_t$ of the FWM just consists of Δpan and Δtilt. Both values can vary between -29 and $+29$ degrees. For the same reason, object displacements in the camera images during camera movements are virtually independent from the object distance to the camera. Thus, depth information is irrelevant for our learning task.

3.2 Retinal Mapping

As mentioned before, the input and output images of the FWM are "retinal" images with decreasing resolution from image center to border. Camera images are converted to such retinal images by a "retinal mapping". The effect of this conversion is depicted in Fig. 3 (right). The basic idea of this mapping is best outlined in polar coordinates. The origins of the coordinate systems are located at the image centers. They are scaled in a way that in both images the maximum radius (along the horizontal/vertical direction) amounts to 1.0. r_R is the radius

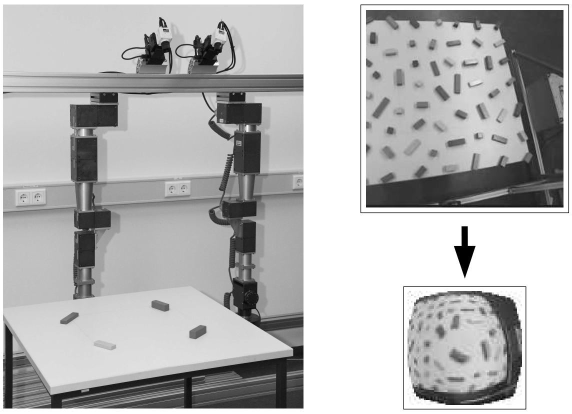

Fig. 3. Left: Setup used as basis for the visual prediction task. Right: Retinal mapping. Upper right image: Original image. Lower right image: Retinal image (scaled up by factor two).

of a point in the retinal image, r_C is the radius of the corresponding point in the camera image, the angle of the polar representation is kept constant. r_C is computed by $r_C = \lambda r_R^{\gamma} + (1-\lambda) r_R \;,\; \gamma > 1 \;,\; 0 \leq \lambda \leq 1$. Here we use $\gamma = 2.5$ and $\lambda = 0.7$. The resolution of the final retinal image is 69 × 69 pixels. To avoid aliasing artifacts in the heavily subsampled outer regions of the original image, adaptive smoothing is applied.

While the input image of the FWM is an unmodified retinal image, the output image is a center crop with a size of 53 × 53 pixels. This is necessary to clip the white corners of the retinal image without valid information (see Fig. 3, right) which are just a technical artifact and would spoil the learning algorithm.

3.3 Grid of Cumulator Units

The input space of the MM and VM consists of four dimensions: Δpan, Δtilt, x_{Out}, and y_{Out}. In this space, a four-dimensional grid $\mathbf{P}$ of points $\mathbf{p}_{ijkl} = \left(\Delta\text{pan}^{(i)}, \Delta\text{tilt}^{(j)}, x^{(k)}_{\text{Out}}, y^{(l)}_{\text{Out}},\right)'$ is inscribed, with $i, j = 1, .., 7$ and $k, l = 1, .., 11$. $\Delta\text{pan}^{(i)}$ and $\Delta\text{tilt}^{(j)}$ vary from -29 to $+29$ degrees with constant step size (covering the whole valid Δpan/Δtilt range), while $x^{(k)}_{\text{Out}}$ and $y^{(l)}_{\text{Out}}$ form an equally spaced rectangular grid covering the whole output image.

To each point $\mathbf{p}_{ijkl}$, a so-called "cumulator unit" C_{ijkl} is attached. Such a unit is basically a single-band image with the same size as the input image. Each "pixel" of this unit can hold any positive integer value including zero. They are used to collect candidate pixels for the MM output $\langle \widehat{x}_{\text{In}}, \widehat{y}_{\text{In}} \rangle$.

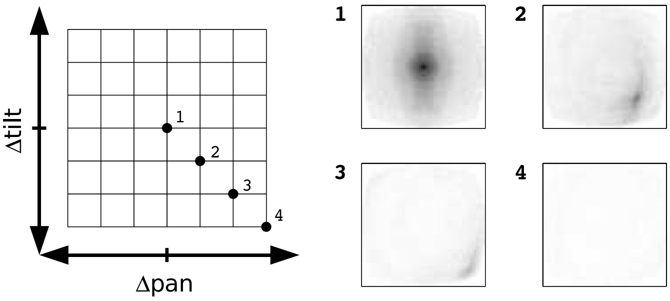

Fig. 4. Cumulator units for the center pixel for four different Δpan/Δtilt positions. All depicted cumulator units were normalized by the same scaling factor so that a pixel value of zero corresponds to white and the overall maximum pixel value to black.

3.4 Learning Process

The goal of the learning process is to accumulate activations in the cumulator units. At the beginning, all pixels of these units are set to zero. In each learning trial, the pan-tilt unit is first moved into a random pan/tilt position. The input image for the FWM is recorded and processed. Afterwards, the algorithm iterates through all points of the grid $\mathbf{P}$, the corresponding motor command is executed (relative to the initial random position), and the output image is generated from the camera image after the movement. For each point $\mathbf{p}_{ijkl}$, the intensity of the output image at the coordinates $\langle x_{\text{Out}}^{(k)}, y_{\text{Out}}^{(l)} \rangle$ is compared to the intensities of all pixels $\langle x_{\text{In}}, y_{\text{In}} \rangle$ in the current input image. Whenever the intensity difference is below a certain threshold α, the value of pixel $\langle x_{\text{In}}, y_{\text{In}} \rangle$ in cumulator unit C_{ijkl} is increased by one. The intensity difference is computed as Euclidean distance in RGB color space. The threshold α is set to 3.5% of the overall intensity range in a single color channel.

In the present study, 100 trials were carried out, each with $7 \times 7 \times 11 \times 11 = 5929$ iteration steps (size of the grid $\mathbf{P}$). Each trial took place in a slightly different visual environment because the initial camera position varied. 42 colored wooden blocks were placed on the table to enhance the visual richness of the environment (see Fig. 3, right).

Figure 4 illustrates four final cumulator units C_{ijkl} in the grid $\mathbf{P}$. Their positions along the Δpan and Δtilt dimensions are marked on the two-dimensional grid on the left (camera movements to the lower right of increasing length, starting at position 1 with zero movement). Their position $\langle x_{\text{Out}}^{(k)}, y_{\text{Out}}^{(l)} \rangle$ in output image coordinates is the center pixel. The pixel color in the cumulator units reflects the size of the accumulated sum from white (zero) to black (maximum sum). Unit 1 with zero camera movement shows a clear maximum exactly in the center. Thus, the most likely origin of the center pixel in the output image is the center pixel in the input image. This is exactly what is expected when no camera movement takes place. Unit 2 is associated with a small camera movement to the lower right. The intensity maximum is no longer in the center of the unit, but in the lower right corner: When the camera moves into a certain direction, the new image center has its origin in the direction of the movement. Because of the retinal mapping, the intensity maximum moves far to the border of the cumulator unit although the corresponding camera movement is rather small. Unit 3 with a larger camera movement shows a similar effect. Moreover, its maximum intensity is obviously weaker than in unit 1. This is mainly caused by the retinal mapping with its heavy subsampling in the outer image regions (causing fewer matches with the correct candidate pixel). Finally, unit 4 shows no visible maximum in print at all. Actually, the corresponding camera movement is so large that the center pixel of the output image has no valid counterpart in the input image, therefore it is unpredictable.

3.5 Generating a Raw Version of the MM and VM

After the cumulator units have been acquired in the learning process, raw versions of the MM and VM can be created whose output is defined at the grid positions $\mathbf{p}_{ijkl}$ in input space. The output $\langle \widehat{x}_{\mathrm{In}}, \widehat{y}_{\mathrm{In}} \rangle$ of the MM at grid point $\mathbf{p}_{ijkl}$ are the coordinates of the pixel with maximum intensity in the cumulator unit C_{ijkl}. The outputs $v_{\langle x_{\mathrm{Out}}, y_{\mathrm{Out}} \rangle}$ of the VM at point $\mathbf{p}_{ijkl}$ is set to 1 (signalling valid output of the MM at this point) whenever the maximum pixel intensity in unit C_{ijkl} is above a certain threshold. Otherwise, $v_{\langle x_{\mathrm{Out}}, y_{\mathrm{Out}} \rangle}$ is set to 0. The threshold is computed as the product of the maximum pixel intensity of all cumulator units and a factor $\beta = 0.45$. This proved to be the value resulting in the most correct separation.

Figure 5 shows the output of the MM and VM for nine different motor commands $\Delta\mathrm{pan}^{(i)}/\Delta\mathrm{tilt}^{(j)}$. For each motor command, the pixel coordinate space of the input image is shown in a single panel. The two-dimensional grid in each panel connects points along the $x_{\mathrm{Out}}^{(k)}$ and $y_{\mathrm{Out}}^{(l)}$ directions of $\mathbf{P}$. The position of each grid point corresponds to the output $\langle \widehat{x}_{\mathrm{In}}, \widehat{y}_{\mathrm{In}} \rangle$ of the MM at this point. Only points with valid output are shown (determined by the VM). The central panel with no movement shows an identity mapping between $\langle x_{\mathrm{Out}}^{(k)}, y_{\mathrm{Out}}^{(l)} \rangle$ and $\langle \widehat{x}_{\mathrm{In}}, \widehat{y}_{\mathrm{In}} \rangle$ (as expected). The other panels reflect the relationship between the camera movement and the pixel shift between input and output image. The strong curvature of the grid is mainly caused by the retinal mapping.

3.6 Network Training

The output of the raw versions of the MM and the VM is only defined at the grid points $\mathbf{p}_{ijkl}$. To get the output in-between, function interpolation is necessary. For this purpose, the raw versions of the MM and the VM were replaced by radial basis function networks (RBFN) (for details, see [1]) in the final step of the learning algorithm. These networks have the same input/output structure as the MM and the VM, respectively (see Fig. 2). The training data for both

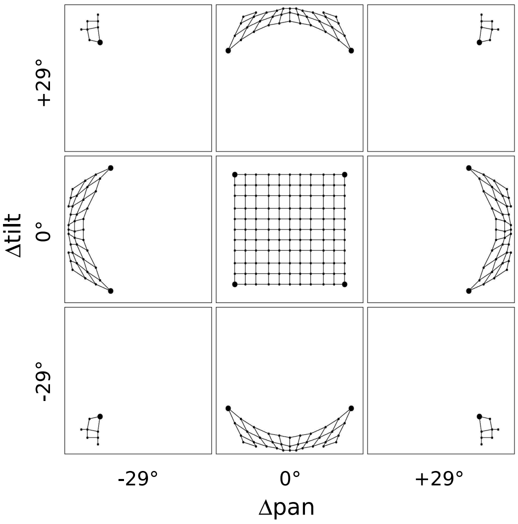

Fig. 5. Mapping from pixel coordinates $\left\langle x_{\mathrm{Out}}^{(k)}, y_{\mathrm{Out}}^{(l)} \right\rangle$ (grid points) in the output image to pixel coordinates $\langle \widehat{x}_{\mathrm{In}}, \widehat{y}_{\mathrm{In}} \rangle$ in the input image for 3×3 different Δpan/Δtilt positions.

networks was generated from the output of the raw versions of the MM and the VM at the grid points $\mathbf{p}_{ijkl}$ (overall, there are $7 \times 7 \times 11 \times 11 = 5929$ grid points). For the MM network, training data was restricted to the 2935 grid points with valid output (signalled by the raw version of the VM).

For both networks, the hyperbolic tangent was used as activation function for the output units. Both the MM and the VM network were initialized with the k-means algorithm, afterwards they were trained for 1000 epochs with gradient descent. Input and output values were scaled to the range $[-0.6; 0.6]$.

The MM network is a RBFN with 200 Gaussians for each output unit (x_{Out} and y_{Out}). The training set consisted of the 2935 valid input-output pairs of the raw MM. The mean squared error per pattern per output unit amounted to $2.3 \cdot 10^{-4}$ after the last epoch.

The VM network has 250 Gaussians in the hidden layer for its single output unit. It basically had to learn a classification task with a training set covering all 5929 grid points. While the mean squared error per pattern per output unit still amounted to $5.3 \cdot 10^{-2}$ after the last epoch, only 1.3% of the grid points were misclassified.

It is possible to use alternative methods for function interpolation, e.g., to construct the RBFNs directly from the grid points without learning (even during the acquisition of the cumulator units as a kind of “online” method), or to use other non-linear regression methods.

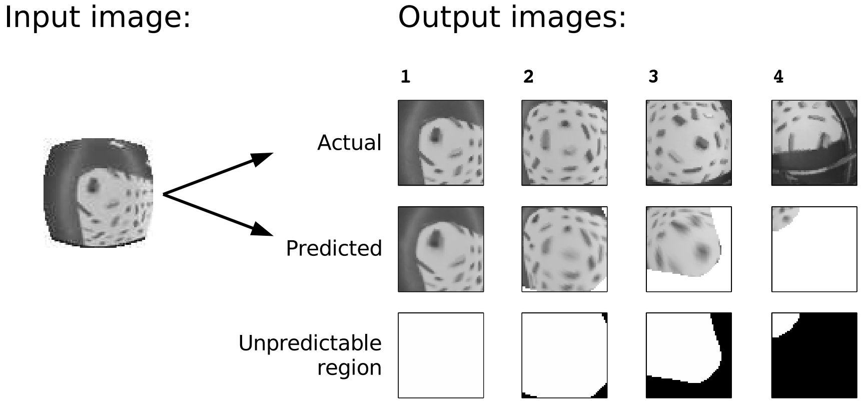

Fig. 6. Comparison of actual and predicted output images at four different Δpan/Δtilt positions (the same as in Fig. 4).

4 Results

The MM and VM network are used to implement the overall visual FWM for predicting the output image as explained in Sect. 2. Especially, non-predictable regions of the output image are marked by the VM network. The prediction works rather precise as shown exemplary in Fig. 6. The actual and the predicted output image are compared for four different motor commands Δpan/Δtilt (camera movements to the lower right of increasing length as in Fig. 4). Moreover, the region of each output image which is marked as non-predictable by the VM is shown in black color in the third row of images. The input image (the same for all four movements) is displayed as well. Movement 1 is a zero movement. The actual and the predicted output image are very similar and show the center crop from the input image. Movements 2 and 3 are of increasing size. The non-predictable regions mask parts of the output images which have no correspondence in the input image. The center of the predicted images is slightly blurred and distorted because the mapping generated by the MM network has to enlarge a region of a few pixels in the input image to a much larger area (especially for movement 3). Movement 4 is so large that the center of the output image is non-predictable. Nevertheless, the small upper left part of the output image which is predicted corresponds closely to the actual output.

This visual inspection of a few exemplary camera movements demonstrates the learning success. At the current stage of development, the additional application of quantitative evaluations is not meaningful because of the lack of competing learning algorithms for visual FWMs. Furthermore, quantitative measures like the Euclidean distance in pixel space are difficult to interprete because the FWM has to enlarge parts of the input image while the actual output maintains the optimum resolution in the image center.

We pointed out in Sect. 3.1 that depth information is irrelevant for our learning task because of the camera geometry. Therefore, it is possible to rearrange objects in the field of view of the camera without any harm to the prediction performance of the visual FWM.

5 Discussion and Conclusions

The proposed learning algorithm overcomes the problem that visual FWMs have a high-dimensional input and output space due to the size of visual data. Forecasting pixel intensities is replaced by forecasting a mapping between output and input pixel locations. The only restriction regarding image size is imposed by the size of the computer memory because it has to hold the cumulator units during the learning process.

The learning process relies on matching pixels between the output and input image. By imposing a retinal mapping, it is demonstrated that this learning principle even works when strong image distortions are involved (including color changes caused by smoothing and subsampling in the outer areas of the camera images). Future research will show to which extent the performance of the learning algorithm deteriorates in response to even more ambiguous visual data (e.g., by using monochrome images).

The distinction between cumulator units with a large and a small maximum pixel intensity offers a natural solution for the detection of unpredictable image regions. A small maximum signals that no correct pixel match exists, while an existing correct match accumulates to a large maximum during the learning process.

At the current stage of development, the application of a grid of cumulator units spanned in the input space of the MM and VM only allows low-dimensional motor commands $\mathbf{m}_t$ because of the storage requirements of these units. To overcome this problem, the next step of development is an online learning scheme to adapt to the maximum (the modal value) of the intensity distribution in each cumulator unit without the need to store the units. This would allow more dimensions in motor space. Even further, the goal is to replace the fixed grid structure in motor space by random movements (while maintaining the grid in $\langle x_{\mathrm{Out}}, y_{\mathrm{Out}} \rangle$ space with the appropriate spacing for the distortions caused by the imaging system).

The basic ideas of the proposed learning algorithm might even offer an explanation for the acquisition of visual FWMs in biological organisms: First, learning the input-output relationship by matching low-level visual features, and second, identifying predictable regions by detecting that a good match emerges during the learning process. In robotics applications, visual FWMs can be used to explore the various functions of FWMs stated in the introduction. Moreover, they may become an important functional building block of truly autonomous systems, both for motor control and for perceptual competences.

In our current research, which also includes motor learning, we plan to use the visual FWM of this study (or a successor) in a saccade learning task. Through the predictions of the FWM, it will be possible to track objects between camera movements efficiently. This allows the computation of the sensory error which is needed in many motor learning schemes [11].

References

1. Bishop, C.M.: Neural Networks for Pattern Recognition. Oxford University Press, UK (1995)
2. Blakemore, S.J., Wolpert, D., Frith, C.: Why can't you tickle yourself? NeuroReport **11**(11) (2000) R11–R16
3. Gross, H.M., Heinze, A., Seiler, T., Stephan, V.: Generative character of perception: A neural architecture for sensorimotor anticipation. Neural Networks **12**(7-8) (1999) 1101–1129
4. Große, S.: Visuelle Vorwärtsmodelle für einen Roboter-Kamera-Kopf (2005) Diploma Thesis. Computer Engineering Group, Faculty of Technology, Bielefeld University.
5. Hoffmann, H., Möller, R.: Action selection and mental transformation based on a chain of forward models. In Schaal, S., Ijspeert, A., Billard, A., Vijayakumar, S., Hallam, J., Meyer, J.A., eds.: From Animals to Animats 8, Proceedings of the Eighth International Conference on the Simulation of Adaptive Behavior, Los Angeles, CA, MIT Press (2004) 213–222
6. von Holst, E., Mittelstaedt, H.: Das Reafferenzprinzip. Die Naturwissenschaften **37**(20) (1950) 464–476
7. Jordan, M.I., Rumelhart, D.E.: Forward models: Supervised learning with a distal teacher. Cognitive Science **16**(3) (1992) 307–354
8. Kawato, M.: Internal models for motor control and trajectory planning. Current Opinion in Neurobiology **9** (1999) 718–727
9. Miall, R.C., Weir, D.J., Wolpert, D.M., Stein, J.F.: Is the cerebellum a smith predictor? Journal of Motor Behavior **25**(3) (1993) 203–216
10. Möller, R.: Perception through anticipation—a behavior-based approach to visual perception. In Riegler, A., Peschl, M., von Stein, A., eds.: Understanding Representation in the Cognitive Sciences. Plenum Academic / Kluwer Publishers, New York (1999) 169–176
11. Schenck, W., Möller, R.: Learning strategies for saccade control. Künstliche Intelligenz **Iss. 3/06** (2006) 19-22
12. Tani, J.: Model-based learning for mobile robot navigation from the dynamical systems perspective. IEEE Transactions on Systems, Man, and Cybernetics—Part B **26**(3) (1996) 421–436
13. Wolpert, D.M., Kawato, M.: Multiple paired forward and inverse models for motor control. Neural Networks **11** (1998) 1317–1329
14. Ziemke, T., Jirenhed, D.A., Hesslow, G.: Internal simulation of perception: A minimal neuro-robotic model. Neurocomputing **68** (2005) 85–104