Dynamic learning of action patterns for object acquisition

Gabriele Peters* and Thomas Leopold

Department of Computer Graphics, University of Dortmund,
Otto-Hahn-Str. 16, D-44221 Dortmund, Germany
Fax: (+49) 231 755 6321
E-mail: peters@ls7.cs.uni-dortmund.de
E-mail: thleopold@hotmail.com
*Corresponding author

Abstract: We propose an active vision system for the acquisition of internal object representations. The core of the approach is an agent which learns goal-directed action patterns depending on the perceived environment via reinforcement learning. The user supervision is restricted to the definition of this goal in the form of a reward function. We demonstrate this approach by means of learning a strategy to scan an object. The agent moves a virtual camera around an object and is able to adapt her scan path dynamically to different conditions of the environment such as different objects and different goals of the data acquisition. The purpose of the acquisition which we consider here is the view-based reconstruction of non-acquired views. The scan pattern obtained after the learned path has stabilised allows a better reconstruction of unfamiliar views than random scan paths.

Keywords: active vision; object acquisition; view reconstruction; object representation; reinforcement learning; learning action patterns; goal-directed behaviour; adaptive control.

Reference to this paper should be made as follows: Peters, G. and Leopold, T. (2007) 'Dynamic learning of action patterns for object acquisition', *Int. J. Intelligent Systems Technologies and Applications*, Vol. 2, Nos. 2/3, pp.113–124.

Biographical notes: Gabriele Peters received her Diploma in Mathematics from the Ruhr-University Bochum in 1996 and her PhD from the Faculty of Technology of the Bielefeld University in 2002. From 1996 to 2001 she was employed at the Institute of Neural Computation, Ruhr-University Bochum. She was a Visiting Professor at the California Institute of Technology in 2004 and 2005. Currently she is a Research Assistant at the Department of Graphical Systems of the University Dortmund.

Thomas Leopold is a student of Computer Science and Mathematics in the 14th semester at the University Dortmund. His main areas of interest lie in the fields of artificial intelligence/artificial life and astronomy/astro-physics. In his free time he enjoys watching Star Trek and introducing young people to science. Since his fourth semester he has been regularly supervising training groups of novice students at the University Dortmund.

1 Introduction

The visual appearance of objects is a concern of computer vision as well as computer graphics. Both fields of research utilise internal representations of objects. One main topic of computer graphics is the generation of 3D models from real world objects for geometric modelling, and one of the major problems in computer vision is the recognition of objects from single views. The internal object representations that have to be acquired can be 3D model-based or 2D view-based. Until now one of the problems concerning object acquisition has been its separation from the processing of the acquired data, especially from the specific goal of a future application. This often implies that the acquired data are either insufficient or redundant for the application. Thus, there is an increasing demand for learning methods which allow the extraction of only the relevant information with respect to a defined goal. Among the principles of learning, agents to be striven for are the learning of goal-directed behaviour, adaptivity to the environment, and as little supervision by the user as possible.

In this paper, we propose a learning scheme which follows these principles. An agent autonomously adapts to her environment, resulting in a learned action pattern that depends on the environment and the goal of the action only.

We implement these principles considering the learning of view-based object representations as examples. Our agent simulates a scanner which moves a camera around an object. The action pattern to be learned is the scan path on the view sphere which is optimal with respect to the object and the goal of the data acquisition. The learned scan path allows for the generation of a sparse, view-based object representation in the form of some selected key views of the path. The goal of the agent is to find that scan path which best enables the view-based reconstruction of non-acquired views from key views of the scan path. The only user interaction consists in the definition of this goal in the form of a reward signal which guides the learning process. The appropriate behaviour emerges autonomously then by interaction of the agent (the moving camera) with the environment (the object). Thus, different scan paths would result for different object classes and for different goals of the data acquisition (such as learning 3D models vs. 2D view-based representations).

The core of our approach is a reinforcement module. Its principles are briefly sketched. An agent interacts with the environment by perception and action. In an interaction step the agent receives information on the current state of the environment as input via perception. A state is defined by the current camera parameters and information on the object learned so far. Then the agent chooses an action according to a policy function, i.e., the camera is moved to the next view and the representation learned up to this time is updated. The action is carried out and changes the state of the environment. The agent is able to adapt her behaviour dynamically to certain conditions. For this purpose the agent receives direct feedback for the last action by a reward signal which supports the intended goal (here the reconstruction of unfamiliar views). The behaviour of the agent should maximise the long term sum of the reward signals. Thus, the agent learns her behaviour by systematic trial-and-error over several scanning episodes.

2 Related work

Recently more attention is paid to the importance of joining object learning and action. In Fitzpatrick et al. (2003) action-specific movement patterns of objects are statistically learned while actions such as pushing are carried out on them. But the acquisition phase (i.e., the learning or training) is still disconnected from the application without feedback between perception and action.

Another field of research related to this work is denoted by the term viewpoint planning. It describes a bunch of techniques used to determine viewpoint distributions of objects or scenes which are optimal with respect to the information necessary for a specific task. In computer vision these techniques are not utilised at the level of object acquisition up to now, rather they are employed first on the level of recognition (Callari and Ferrie, 1996).

The concept of key-frames is another issue related to the acquisition of objects. In Wallraven and Bülthoff (2001) key-frames are chosen from an image sequence to represent an object, but still with a given scan path and a given strategy for their choice. Other systems exist which are more adaptive. They try to adjust the scan path to the object or the application (Wixson, 1994; Dickinson et al., 1997; Maver and Bajcsy, 1993; Hlaváč et al., 1996; Chen and Li, 2002), but here the strategies for scanning an object or a scene are mostly given by the developer as well. Only recently has an effort been made to learn the strategies as well, for example with methods of reinforcement learning. But here again the autonomous emergence of strategies is not explored until the level of object recognition (Paletta and Pinz, 2000; Reinhold et al., 2000; Deinzer et al., 2001). To our knowledge no approach to object acquisition by active learning has been proposed up to now. The system we describe in this paper learns a view-based object representation adaptively without a given strategy via reinforcement learning. Methods for the control of reinforcement learning designs are summarised, e.g., in Sutton and Barto (1998), Kaelbling et al. (1996) and Russell and Norvig (2003).

3 Components of the system

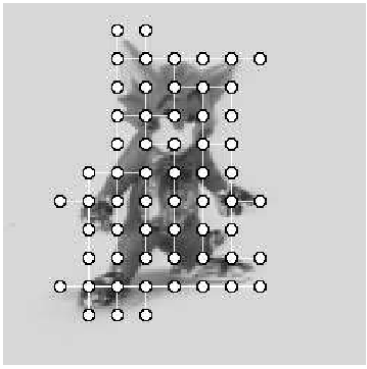
In this section we describe the preprocessing of the acquired views, the calculation of correspondences between frames by tracking local feature descriptors, the data structure for the object representation, and the reconstruction of unfamiliar views which have not been scanned. These are the basic components of our system. The learning of a scan strategy is treated in Section 4.

3.1 Preprocessing and view representation

The results described in this paper have been obtained with a scanning system which is virtual only, i.e., which is not implemented on a hardware scanner yet. We simulate a scanner which rotates the camera around the object at a fixed distance oriented to the centre of the object base. For that purpose we recorded views of objects in distances of 3.6° in both longitudinal and latitudinal directions on the upper hemisphere of the object, resulting in 2500 views per object (see Figure 3). Each view is represented by a graph, which covers the object in the image. The nodes of a graph are labelled with

Gabor wavelet responses, which describe the local surroundings of the node in the image. For the Gabor transform we use a set of wavelets with eight directions and four frequencies. The graphs are generated automatically from the images: first, the object is separated from the background by a *segmentation* algorithm described in Eckes and Vorbrüggen (1996), which is based on the grey level values of the image. Then a *grid graph* (Figure 1) is put on the resulting object segment.

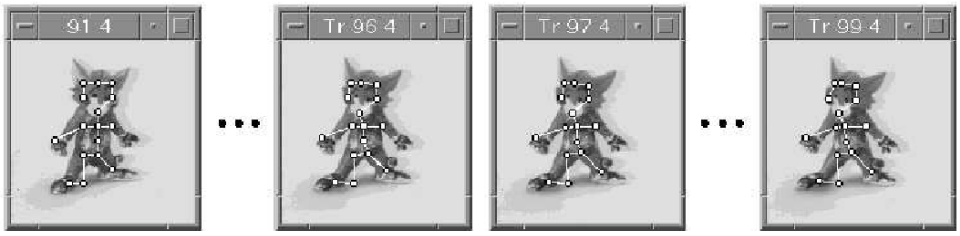
Figure 1 Grid graph



3.2 Tracking local object features

Corresponding object points between scanned views are obtained by tracking the nodes of a graph from frame to frame. They are required later for the view-based reconstruction of non-acquired views by morphing. The information stored at a node in the form of Gabor wavelet responses enables the node to be tracked to the next frame (Maurer and von der Malsburg, 1996). The grid graph shown in the left view of Figure 2 is tracked along the sequence to the view shown on the right. The similarity between two views can be expressed by the result of a similarity function between two graphs, which is based on the Gabor wavelet responses (Lades et al., 1993).

Figure 2 Tracking of object points

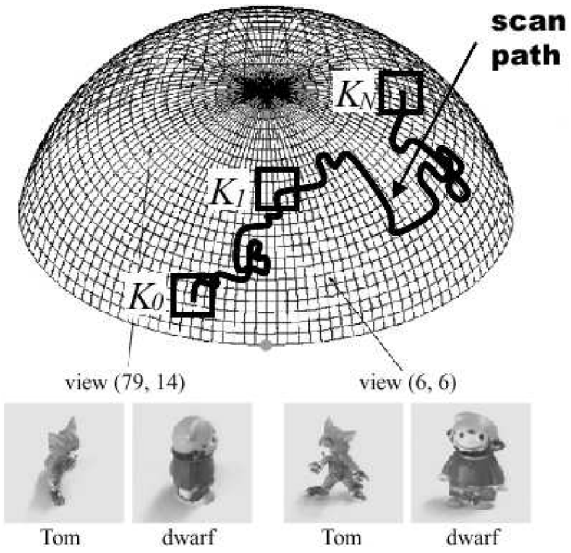


3.3 Object representation

Assume a given scan path of an object. To obtain a sparse, view-based object representation, we select key views from this path and store either one original grid graph or one original and one tracked graph per key view. We start with the first view of the scan path. This is the first key view K_0. Its original grid graph $\mathcal{G}_{\text{orig}}^{K_0}$ is incorporated in the

object representation. Then it is tracked according to Section 3.2 along the scan path until the similarity between the tracked graph at the current view of the scan path and $\mathcal{G}_{\text{orig}}^{K_0}$ drops below a preset threshold. The tracked graph $\mathcal{G}_{\text{track}}^{K_1}$ for this second key view K_1 is also stored in the object representation. For K_1 a new grid graph $\mathcal{G}_{\text{orig}}^{K_1}$ is generated and incorporated into the representation as a second graph for this view as well. Then this graph is also tracked until the similarity to $\mathcal{G}_{\text{orig}}^{K_1}$ drops again below the threshold, and so on. This means that for the first and the last key view of the scan path only one graph is stored ($\mathcal{G}_{\text{orig}}^{K_0}$ and $\mathcal{G}_{\text{track}}^{K_N}$, respectively), whereas for each other key view $K_j, j = 1, ..., N-1$ of the scan path two graphs $\mathcal{G}_{\text{track}}^{K_j}$ and $\mathcal{G}_{\text{orig}}^{K_j}$ are stored in the object representation. This ensures piecewise correspondences for local areas of the view hemisphere. The illustration in Figure 3 shows sample views of two objects and a possible scan path with three key views.

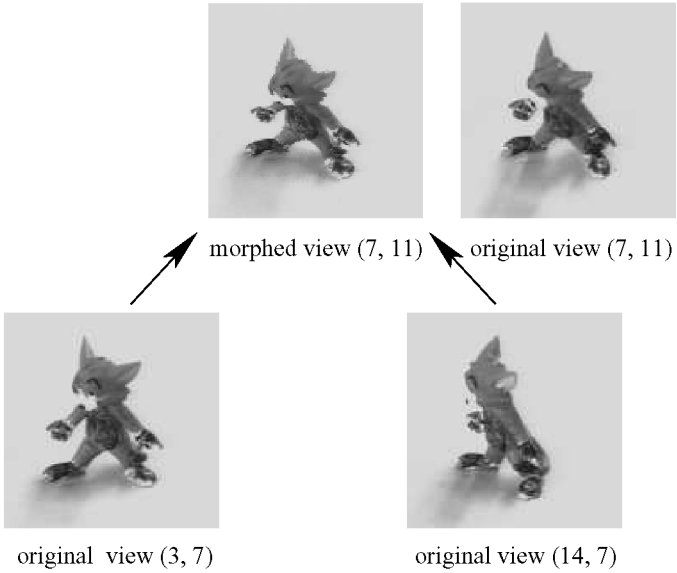
Figure 3 View hemisphere with key views



3.4 Reconstruction of non-acquired views

The reconstruction of non-acquired views from the key views of a scan path has two functions. On the one hand, it serves as a test whether the relevant information on the object has been captured after the scan path has been learned. On the other hand, it is used for the calculation of the reward signal after each step of a scan episode. The correspondences provided by the tracking procedure (Section 3.2) enable us to apply a standard view morphing technique described in Peters (2002). An unfamiliar view is morphed from those two consecutive key views which are closest to it. To compare a morphed view to its original version an error $e_{\text{recon}} \in [0, 1]$ also described in Peters (2002) can be defined. In the example illustrated in Figure 4 the non-acquired view (7, 11) is reconstructed from the key views (3, 7) and (14, 7). It can be compared to the original view (7, 11).

Figure 4 Reconstruction of non-acquired views



4 Learning action patterns

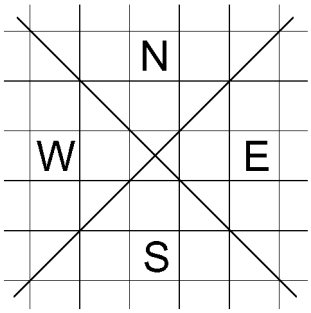
We apply ***Q***-learning in our simulations. It works by estimating the values of state-action pairs. The ***Q***-value is the expected discounted sum of future payoffs obtained by taking a particular action in a current state and following an optimal policy thereafter. Once these values have been learned, the optimal action from any state is the one with the highest ***Q***-value. We apply ***Q***-learning with a learning rate $\alpha = 1/3$ and an ϵ-greedy policy with $\epsilon = 1/3$, annealed by the factor 1/1.000001. In the beginning the agent chooses *exploration*, i.e., a random action, in one third of all steps and *exploitation*, i.e., an action based on the learned information, in two thirds of all steps. With ongoing processing we slowly decrease the probability for exploration for the benefit of exploitation. The ***Q***-values are defined as follows:

$$Q(s_t, a_t) = Q(s_t, a_t) + \alpha(r_{t+1} + \max_a Q(s_{t+1}, a) - Q(s_t, a_t)).$$

with s_t the state, a_t the action, and r_{t+1} the reward at step t. As we currently store them in a table, the number of state-action-pairs has to be reasonably small. The definition of the *state* as the current position of the camera would yield a sufficiently small number of states. But this definition would not be effective enough for learning, because information of the scan history is lost. The scan history could be retained by defining a state of the environment by the complete path. This, in turn, would yield too many states to be stored in the ***Q***-table (all possible paths). For these reasons we define a *state* as a vector of five values. The first value encodes the current position of the camera and the remaining four values describe the degree of unfamiliarity of the areas to the north, east, south, and west of the current position on the view hemisphere, respectively. By this definition similar

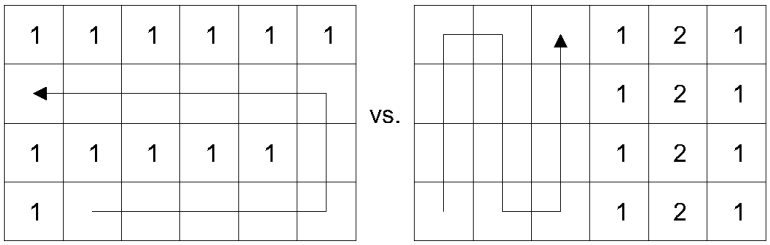
scan paths, which provide almost the same information on the object, are mapped to the same state. In the illustration in Figure 5 the hemisphere is quantised and projected to a plane. The position in the centre is the current position of the agent. For the areas to the north, east, south and west of the current position the degrees of unfamiliarity define the state of the agent. Positions on the diagonals which separate the areas are assigned to both adjacent areas.

Figure 5 Hemisphere areas used for state definitions



We calculate the degree of unfamiliarity of an area in the following way. To each unfamiliar position of an area we assign the distance from this unfamiliar position to the next position that has already been scanned. Then the value of an area is the sum of all values of unfamiliar positions in this area (Figure 6). The arrows depict the scan paths. The numbers are values of single positions within any of the four areas.

Figure 6 Examples for unfamiliarity calculation



The possible values of an area are quantised into five bins; 0 encodes very familiar areas, 4 encodes very unfamiliar areas. For a further reduction of the number of states we also quantise the original view hemisphere, resulting in a raster of 20×5 views. Thus, a state of the reinforcement learning module consists of six components: x-position on the hemisphere (20 possible values), y-position (five possible values), and unfamiliarity of the areas in the four directions (five possible values each), resulting in a total of 2000 states.

Possible *actions* are the movement of the camera in one of the four above mentioned directions on the quantised view hemisphere.

The *reward signal* r_{t+1} is calculated in the following way. Before the choice of the next action the agent predicts the view she would perceive if she performed the action. The prediction is calculated according to the morphing technique described in Section 3.4 from the last two key views she has experienced so far. After the prediction the action is

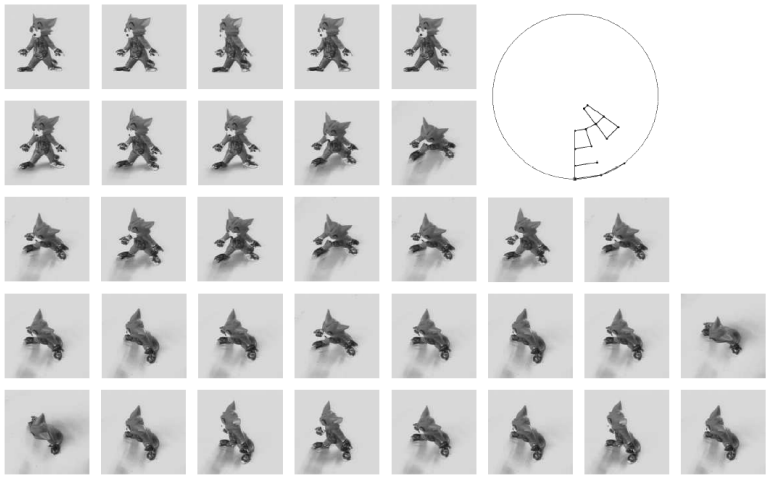
carried out. The reward for this action is higher for smaller similarities between the predicted and the actual view. More concrete, the reward is calculated according to $r_{t+1} = (-(e_{\text{recon},t+1} - 1)^{16}$. The total return for one episode is the sum of the rewards received for each step of the episode.

Each episode starts at position (0, 0) on the view hemisphere, which can be regarded as a canonical view. While the camera is moved, one position on the coarser raster of the quantised hemisphere the current graph is tracked according to Section 3.2. Key views are determined along the way as described in Section 3.3 providing a scan path with associated key views for each episode. An episode consists of 32 steps. This learning process is stopped when the scan path has stabilised. Finally, the quality of the learned path has to be assessed. To this end we randomly choose a set of 25 test views on the unquantised hemisphere. These views are reconstructed from the key views of the learned path as described in Section 3.4. Then a total reconstruction error, which is the mean of the single reconstruction errors e_{recon} of all test views, gives information about the quality of the learned scan path.

5 Results

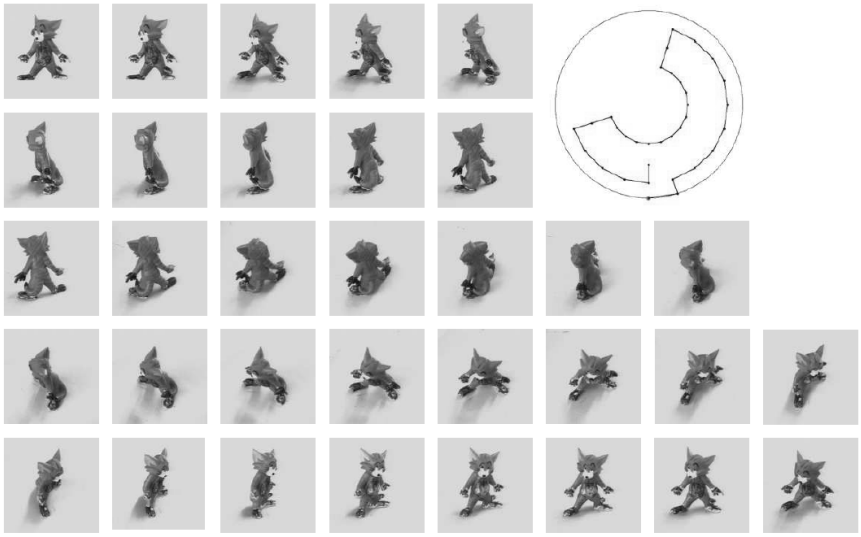
The method described above has been carried out for the 'Tom' object (Figure 3). The learned scan path stabilised after 2 million episodes and yielded a significantly lower total reconstruction error than achieved with random scan paths of equal length. The mean reconstruction error for 100 random paths is 9.2, whereas the error for the learned path is 5.8. A typical random path with 32 steps is shown in Figure 7. The inset shows the view hemisphere seen from above with view (0, 0) at the bottom. Only the key views of the path are displayed. Random paths have been generated using the proposed method with $\varepsilon = 1$.

Figure 7 Key views of a random scan path



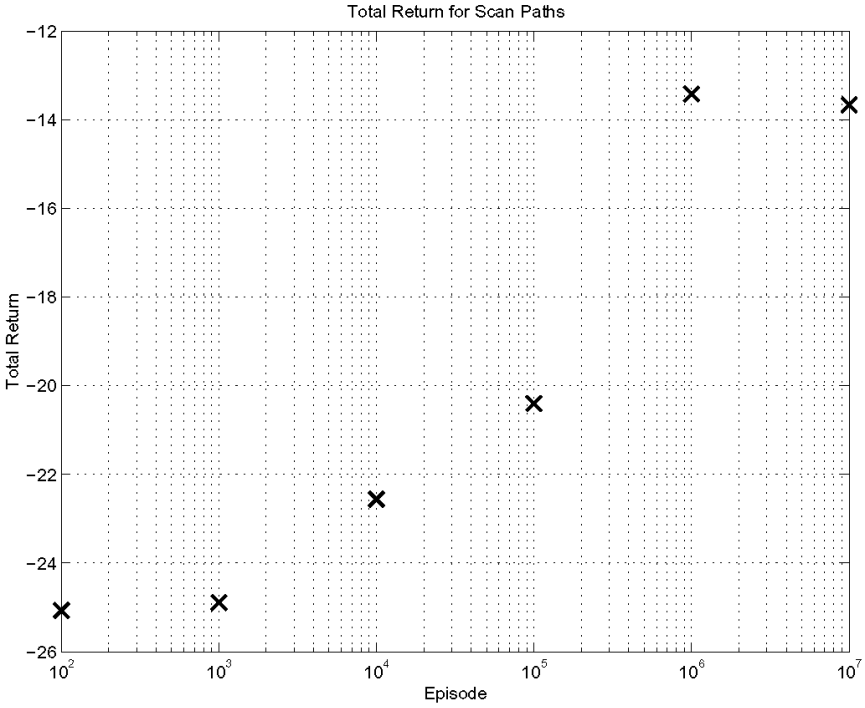
In Figure 8 the key views of the stabilised, learned scan path are depicted.

Figure 8 Key views of the learned scan path



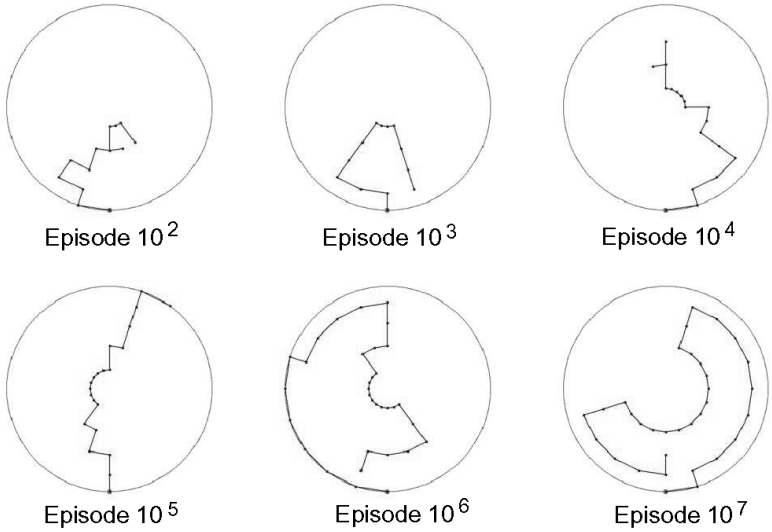
In Figure 9 the total returns obtained for one episode are plotted on a logarithmic scale vs. the number of episodes that have been carried out so far. The returns seem to be monotonously increasing until the scan path has stabilised between episodes 10^6 and 10^7.

Figure 9 Total returns for scan paths



The scan paths learned up to these episodes are displayed in Figure 10 illustrating the learning process.

Figure 10 Scan paths learned up to certain episodes



The resulting path has an even shape, going around the lower part of the view hemisphere from the front to the back, turning up and moving back to the front in the upper part of the hemisphere. Those views of the back of the object that have not been covered by the learned path are rather similar to the views where the agent turned up towards the top of the hemisphere. Thus, it seems to make sense not to incorporate these redundant views into a sparse object representation.

We carried out experiments with an episode length of 36 steps as well. The shape of the resulting scan path for these experiments is similar to the one with a length of 32 steps with the exception that it alternates its direction once more at the top of the hemisphere. But for the episodes with 32 steps the difference between learned and random paths in terms of the total reconstruction error is more obvious than for the episodes with 36 steps.

6 Conclusions

We have introduced an active vision system which automatically learns internal object representations for defined purposes. By adaption to its environment it develops goal-directed behaviour in the form of a strategy to scan an object in such a way that the reconstruction of non-acquired object views is possible. This results in an object-specific movement pattern of the scanner. Up to now we have demonstrated for only one object that the learned scan strategy is more suitable for the reconstruction of unfamiliar views of the scanned object than any of the tested random scan paths. We will test our system with other objects with different shapes in the future and also hope to learn characteristic scan paths for different object classes.

The system, as described, does not work in real-time. But we believe that the basic idea of the approach will enable real-time applications in the future. For that purpose we will, for example, replace the table-based ***Q***-learning by an appropriate function approximation. (Then the restriction to paths of a preset length will also be superfluous.) In addition, we believe that once characteristic scan strategies for different object classes can be learned, the inspection of objects, e.g., for the purpose of recognition will be possible in real-time. Currently we are working on the transfer of our approach to a hardware system. We use an anthropomorphic robot with a manipulator arm which moves a camera in its gripper around an object placed on a table. In addition, we investigate the influence of different goals (such as the acquisition of a 3d model) on the resulting scan strategy. We believe that the proposed concept will result in an intelligent scanner which allows a more efficient acquisition and storage of objects. Possible applications are finding 3D models in data bases and learning, recognition, and grasping of objects in the area of service robotics.

Acknowledgements

This research was funded by the German Research Association (DFG) under Grant PE 887/3-1.

References

Callari, F.G. and Ferrie, F.P. (1996) 'Autonomous recognition: driven by ambiguity', *Proceedings of the Conference on Computer Vision and Pattern Recognition*, pp.701–707.

Chen, S.Y. and Li, Y.F. (2002) 'Optimum viewpoint planning for model-based robot vision', *IEEE 2002 World Congress on Computational Intelligence (WCCI)/Congress on Evolutionary Computation*, pp.634–639.

Deinzer, F., Denzler, J. and Niemann, H. (2001) 'Fusion of multiple views for active object recognition', *Pattern Recognition – 23rd DAGM Symposium*, Springer, Berlin, pp.239–245.

Dickinson, S.J., Christensen, H.I., Tsotsos, J.K. and Olofsson, G. (1997) 'Active object recognition integrating attention and viewpoint control', *Computer Vision and Image Understanding*, Vol. 67, No. 3, pp.239–260.

Eckes, C. and Vorbrüggen, J.C. (1996) 'Combining data-driven and model-based cues for segmentation of video sequences', *Proceedings WCNN96*, Press & Lawrence Erlbaum Ass., pp.868–875.

Fitzpatrick, P., Metta, G., Natale, L., Rao, A. and Sandini, G. (2003) 'Learning about objects through action – initial steps towards artificial cognition', *Proceedings of the 2003 IEEE International Conference on Robotics and Automation (ICRA)*, pp.3140–3145.

Hlaváč, V., Leonardis, A. and Werner, T. (1996) 'Automatic selection of reference views for image-based scene representations', *Proceedings of the European Conference on Computer Vision (ECCV), Lecture Notes in Computer Science*, Vol. 1, No. 1064, Springer, pp.526–535.

Kaelbling, L.P., Littman, M.L. and Moore, A.P. (1996) 'Reinforcement learning: a survey', *Journal of Artificial Intelligence Research*, Vol. 4, pp.237–285.

Lades, M., Vorbrüggen, J.C., Buhmann, J., Lange, J., von der Malsburg, C., Würtz, R.P. and Konen, W. (1993) 'Distortion invariant object recognition in the dynamic link architecture', *IEEE Trans. Comp.*, Vol. 42, pp.300–311.

Maurer, T. and von der Malsburg, C. (1996) 'Tracking and learning graphs and pose on image sequences of faces', *Proceedings of the 2nd International Conference on Automatic Face- and Gesture-Recognition*, pp.176–181.

Maver, J. and Bajcsy, R. (1993) 'Occlusions as a guide for planning the next view', *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 15, No. 5, pp.417–433.

Paletta, L. and Pinz, A. (2000) 'Active object recognition by view integration and reinforcement learning', *Robotics and Autonomous Systems*, Vol. 31, Nos. 1–2, pp.1–18.

Peters, G. (2002) *A View-Based Approach to Three-Dimensional Object Perception*, PhD Thesis, Shaker Verlag, Aachen, Germany.

Reinhold, M., Deinzer, F., Denzler, J., Paulus, D. and Pösl, J. (2000) 'Active appearance-based object recognition using viewpoint selection', in Girod, B., Greiner, G., Niemann, H. and Seidel, H-P. (Eds.): *Vision, Modeling, and Visualization 2000*, infix, Berlin, pp.105–112.

Russell, S. and Norvig, P. (2003) *Artificial Intelligence – A Modern Approach*, Prentice-Hall, Englewood Cliffs, NJ, USA.

Sutton, R.S. and Barto, A.G. (1998) *Reinforcement Learning: An Introduction*, MIT Press, Cambridge.

Wallraven, C. and Bülthoff, H.H. (2001) *Automatic Acquisition of Exemplar-Based Representations for Recognition from Image Sequences*, CVPR 2001 – Workshop on Models vs. Exemplars, IEEE CS Press, Kauai, HI, USA.

Wixson, L.E. (1994) *Gaze Selection for Visual Search*, TR 512 and PhD Thesis, Computer Science Dept., U. Rochester, May.