in: *Proceedings of the Workshop Self-Organization and Emergence, 18th Int. Conf. on Architecture of Computing Systems*, Innsbruck, Austria, March 14-17, 2005.

Adaptive Object Acquisition

Gabriele Peters, Thomas Leopold, Claus-Peter Alberts, Markus Briese, Sebastian Entian, Christian Gabriel, Zhiqiang Gao, Alexander Klandt, Jan Schultze, Jeremias Spiegel, Jürgen Thyen, Martina Vaupel, Peter Voß, Qing Zhu

Universität Dortmund, Informatik VII, D-44227 Dortmund, Germany

Abstract

We propose an active vision system for object acquisition. The core of our approach is a reinforcement learning module which learns a strategy to scan an object. The agent moves a virtual camera around an object and is able to adapt dynamically to different conditions of its environment such as different objects and different purposes of the data acquisition by means of a reinforcement signal which rewards a chosen action with respect to the intendet purpose. The purpose of the acquisition we consider here is the reconstruction of non-acquired views. The learned scan path allows the generation of a sparse, view-based object representation which consists of some key views of the scan path. We present preliminary results from a project conducted with undergraduate students and show that the scan pattern obtained with the proposed method allows a better reconstruction of unfamiliar views than random scan paths. As the reward signal is based on local information at the current position of the camera this approach is an example for an organic-computing system with the emergence of a global strategy from local rules.

1 The Problem

Computer vision, as well as computer graphics, deals with the visual appearance of objects. Among the topics of computer graphics is the generation of 3d models from real world objects for geometric modeling. One of the major problems in Computer Vision is the recognition of objects from single views. For both purposes internal object representations, either 3d model-based or 2d view-based, have to be acquired.

Up to now in both fields of research data acquisition is separated for the most part from the processing of the acquired data. This often implicates that the acquired data are either insufficient or redundant for a future application.

2 Adaptive Approach

In this article we concentrate on the view-based acquisition of objects. We approach this problem by an active vision system whereby the processing of the data directly influences the data acquisition. Data are demanded only when they are purposeful for the intended application. To be more concrete, we put this principle into practice with the example of learning view-based object representations. Our system learns a strategy to scan an object. The learned scan path on the view sphere of the object allows the generation of a sparse, view-based object representation which consists of some key views of the scan path. The strategy for scanning is learned dynamically in the sense that different scan paths would result for different applications. The application we consider here is the view-based reconstruction of non-acquired views. View reconstruction is done by 2d view morphing from key views selected from the scan path.

The core of our approch is a *reinforcement learning* module which implements an autonomous system with a *sensor* (a camera moving around the object) and an *actuator* which moves the camera to the next view. The system, i.e., the *agent*, interacts with its *environment* in a perception-action loop. In each step of interaction the agent receives information on the current *state* of its environment. A state is defined by the current camera parameters (i.e., the current perceived view of the object) and information on the object learned so far. Then the agent chooses an *action*, i.e., moves the camera to a next view and updates the representation learned up to this time. This changes the state of the environment, which again is perceived in the next step of interaction.

The agent is able to *adapt dynamically* to different conditions of its environment (i.e., different objects and different purposes of the data acquisition) by means of a reinforcement signal, which rewards a chosen action with respect to the intended purpose (here the reconstruction of unfamiliar views). The goal is the maximization of the longterm sum of the reinforcement signals for a sequence of actions. By a systematic trial-and-error approach over several scanning episodes, i.e., by exploration and exploitation of its environment, the agent learns his behavior, thus improving his scanning strategy slowly from episode to episode. By rewarding only actions which have proven to be useful for a specific purpose meaningful behavior of the agent emerges. We show that a scan pattern learned in the described way results in a better ability to reconstruct non-

acquired views than a random scan path.
The described method has parallels to principles observed in natural systems, where learning by exploration and reward can create advantageous behavior. As the learned scan pattern depends on the goal of the data acquisition as well as the structure of the object the proposed system is *context-sensitive*. It is also *self-organizing* in the sense that meaningful behavior, i.e., performing only those actions which are required, emerges without an external organizer. The role of the user is restricted to the definition of the *high-level goal* of the object acquisition, such as the generation of an object representation that allows for the reconstruction of non-acquired views or the recognition of the object on the one hand, or the generation of a 3d model on the other hand.

3 Related Work

The term *viewpoint planning* summarizes techniques of deciding the optimal viewpoint distribution which captures all relevant information about an object or a scene for a specific task. Within the last decade a variety of methods for viewpoint planning have been proposed. But in the field of computer vision it is usually not employed until the level of object recognition [1], instead of utilizing it also for object acquisition. In [3] an approach to the acquisition of view-based object representations is proposed where key-frames for the representation are chosen from an image sequence. But the scan path as well as the strategy for the choice of key-frames are given. Also for more adaptive systems, which try to adapt the scan path to the object or the application, holds true that the strategies for scanning an object or a scene are mostly given by the developer [4, 6, 5, 2, 7]. Only recently these strategies are also learned automatically, for example with methods of reinforcement learning. This approach is chosen, e.g., by [8, 9, 10] for the autonomous emergence of strategies for object recognition. We do not know any approach to object acquisition by active learning up to now and propose a method which adaptively learns a view-based object representation without a given strategy.

4 Components of the System

In this section we introduce the components which constitute the object acquisition system. In section 5 is described how they work together.

4.1 Data Base and View Representation

Up to now the proposed scanning system is working virtually only, i.e., it is not implemented on a hardware scanner yet. We simulate an eye-in-hand camera setup with the object on a table. The camera rotates around the object at a fixed distance and is oriented to the center of the object base. The observed object views are represented in a data base which contains views for 100 lines of longitude and 25 line of latitude on the upper view hemisphere resulting in 2500 views for one object (figure 3).

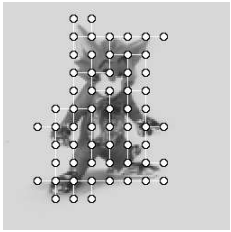

Figure 1 Labeled grid graph placed on the image after segmentation. Each node of the graph is labeled with the corresponding Gabor wavelet responses.

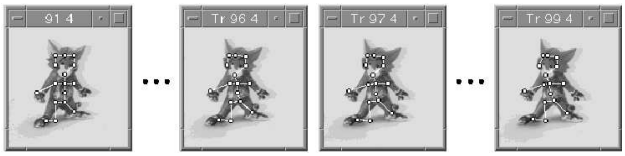

Figure 2 The grid graph shown in the left view is tracked along the sequence to the view shown on the right. The nodes stay on corresponding object points.

Each of the recorded views is preprocessed by a *Gabor wavelet transform*, which is biologically inspired because Gabor wavelets approximate response patterns of neurons in the visual cortex of mammals [14, 15]. A simple *segmentation* based on [16] utilizing gray level values follows. It separates the object from the background. A regular grid graph is placed on the object segment and the nodes of the graph are labeled with the corresponding Gabor wavelet responses. This results in a representation of each view in form of a *labeled grid graph* (figure 1). Each node label is a feature vector which describes the local surroundings of the node. It consists of the amplitude and phase components of the convolution of the image with the Gabor filter bank at the node position. A filter bank with wavelets of 8 orientations and 4 frequencies is used.

4.2 Correspondences by Tracking

The view-based reconstruction of non-acquired views by morphing requires the existence of corresponding points on the object between scanned views. They are obtained by tracking the nodes of a graph from frame to frame within a local area of the view hemisphere. This is realized by utilizing the information obtained from the Gabor transform at each node of the graph [17] (figure 2). A similarity function between two graphs based on the Gabor wavelet responses is defined reflecting the similarity between the particular views [18].

4.3 Sparse Object Representation

A sparse, view-based object representation consists of original grid graphs and tracked graphs of only some key views of the scanned path. We obtain it in the following way. Given a scan path on the view hemisphere we start with

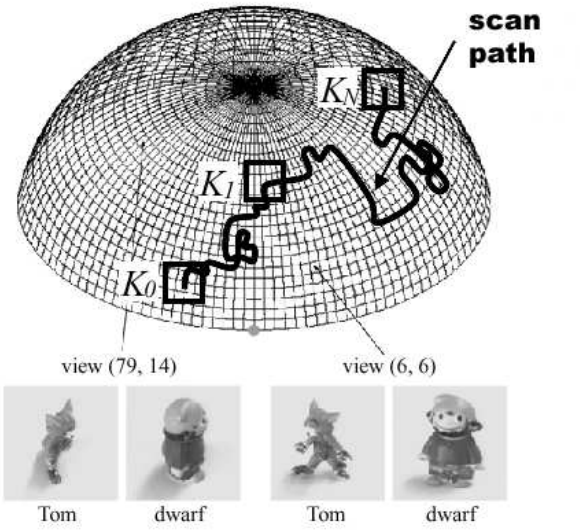

Figure 3 View hemisphere with some images of sample views of two objects. The illustration shows a possible scan path with three key views.

its first view (key view K_0) and incorporate its original grid graph $\mathcal{G}_{orig}^{K_0}$ in the object representation. This graph is tracked according to section 4.2 along the scan path until the similarity between the tracked graph at the current view of the scan path and $\mathcal{G}_{orig}^{K_0}$ drops below a preset threshold. The tracked graph $\mathcal{G}_{track}^{K_1}$ for this second key view K_1 is also stored in the object representation. For K_1 a new grid graph $\mathcal{G}_{orig}^{K_1}$ is generated and incorporated into the representation as well. Then it is also tracked until the similarity to $\mathcal{G}_{orig}^{K_1}$ drops again below the threshold, and so on. Thus, for the first and the last key view of the scan path only one graph is stored ($\mathcal{G}_{orig}^{K_0}$ and $\mathcal{G}_{track}^{K_N}$, respectively), whereas for each other key view $K_j, j = 1, \ldots, N-1$ of the scan path two graphs $\mathcal{G}_{track}^{K_j}$ and $\mathcal{G}_{orig}^{K_j}$ are stored in the object representation, ensuring piecewise correspondences for local areas of the view hemisphere (figure 3).

4.4 Reconstruction of Non-Acquired Views

To test whether the relevant information on the object has been captured by the learned scan path we reconstruct non-acquired views from the key views. An unfamiliar view is morphed from those two consecutive key views which are closest to it, using the correspondences provided by the tracking procedure (section 4.2). For view morphing we use a standard technique described in [19]. A morphed view can then be compared to its original version by an error function also described in [19]. This yields an error e_{recon} for a reconstructed view (figure 4).
This technique is used for the calculation of the reward signal after each step of a scan episode as well as for the calculation of the total reconstruction error after each episode during the learning phase.

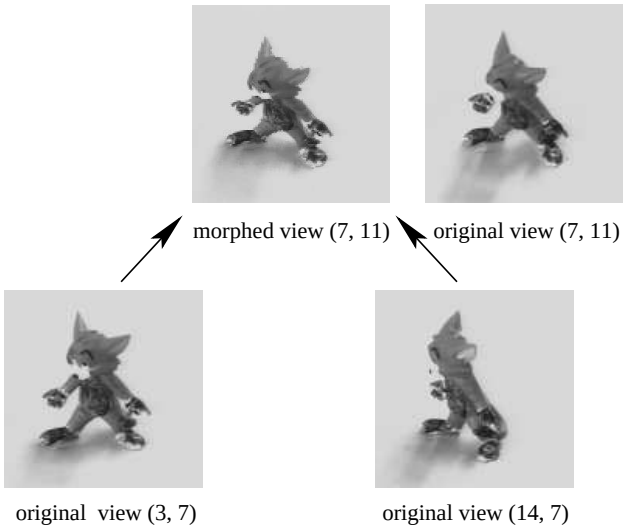

Figure 4 Reconstruction of non-acquired views. In this example the non-acquired view $(7, 11)$ is reconstructed from the key views $(3, 7)$ and $(14, 7)$. It can be compared to the original view $(7, 11)$.

5 Adaptive Object Acquisition with Reinforcement Learning

Several methods have been proposed for the control of reinforcement learning designs many of which are summarized in [11, 12, 13]. We apply Q-learning with a learning rate $\alpha = 0.85$ and an ϵ-greedy policy with $\epsilon = 0.1$. This means that the agent chooses a random action in 10 percent of all steps (exploration) and an action based on the learned information in 90 percent of all steps (exploitation).The Q-values

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

with s_t state, a_t action, and r_{t+1} reward at step t, are stored in a table. This requires the number of state-action-pairs to be reasonably small.
On the one hand, the current position of the camera only is not sufficient to define a state of the environment. On the other hand, the complete path which has been scanned would yield too many states to be stored in the Q-table (all possible paths). For this reason we define a state as a vector which contains the current position of the camera and four values which describe the degree of unfamiliarity of the area to the north, east, south, and west of the current position on the view hemisphere, respectively. This has the advantage that those scan paths which differ only slightly but yield the same information on the object are mapped to the same state (figure 5).
The degree of unfamiliarity of an area is calculated in the following way. We assign a value to each unfamiliar position of an area. This value is the distance from this unfamiliar position to the next familiar position (i.e. one that

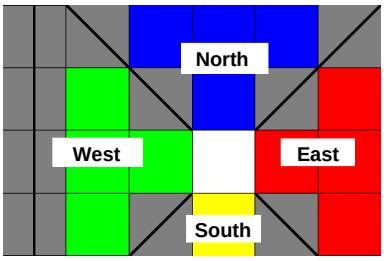

Figure 5 Areas of the view hemisphere used for the definition of the states. In this illustration the hemisphere is quantized and projected to a plane. The white position in the center is the current position of the agent. For the areas to the north, east, south and west of the current position the degrees of unfamiliarity define the state of the agent. Positions on the diagonals which separate the areas are assigned to both adjacent areas.

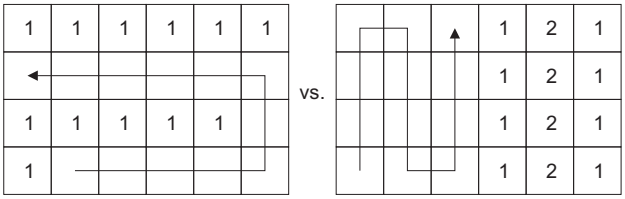

Figure 6 Two examples for the calculation of unfamiliarity. The arrows depict the scan paths. The numbers are values of single positions within either of the four areas.

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 possible values of an area are quantized into five bins, 0 encodes very familiar areas, 4 encodes very unfamiliar areas. For a further reduction of the number of states we also quantize 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 (5 possible values), unfamiliarity of the areas in the four directions (5 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 quantized 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 he would perceive if he performed the action. The prediction is calculated according to the morphing technique described in section 4.4 from the last two key views he 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, $r_{t+1} = e_{recon,t+1} - 1$.

We carry out 50 steps per episode. Each episode starts at position $(0, 0)$ on the view hemisphere. In each step the camera is moved one position on the quantized hemisphere, i.e. we track the current graph on the unquantized hemisphere to the next position on the coarser raster according to section 4.2. While tracking from step to step key views are determined as described in section 4.3. Each episode provides a scan path with associated key views. To assess the quality of the acquired data of one episode we calculate a total reconstruction error in the following way. We preselect a set of 10 test views, which are distributed uniformly on the hemisphere (figure 8). These views are reconstructed from the acquired key views as described in section 4.4. Then the total reconstruction error for the episode is the mean of the reconstruction errors e_{recon} of all test views.

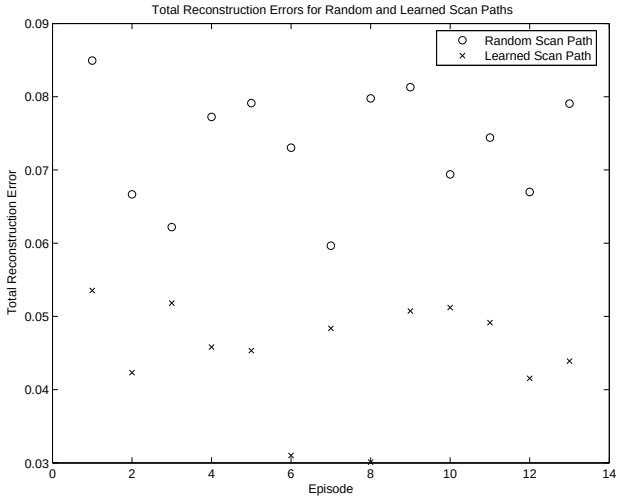

Figure 7 Total reconstruction errors for random and learned scan paths.

6 Results

We carried out 13 episodes for the "Tom" object (figure 3) with the method described above and calculated the total reconstruction errors. We also performed independent random walks with 50 steps per episode (by choosing $\epsilon = 1$) and calculated the total reconstruction errors for those. The results are shown in figure 7. We obtain significantly smaller total errors for scan paths learned with the proposed method than for the random scan paths. In the figures 8, 9, and 10 the key views of some learned scan paths are depicted.

7 Conclusion

We have introduced an active vision system which learns a strategy to scan objects. This strategy adapts to the purpose of the data acquisition, which is the reconstruction of non-acquired views here. 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. However, this has been demonstrated for only one object and only some episodes of the learning module up to now.

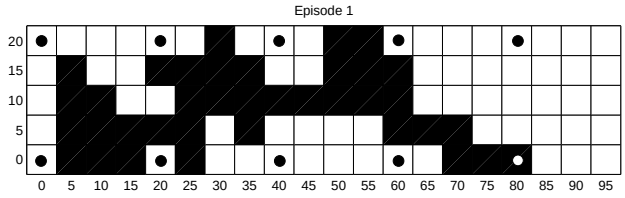

Figure 8 The black squares represent the key views of the learned scan path after the first episode. The total reconstruction error for this path is 0.053. The dots mark the test views used to calculate this error.

Figure 9 Key views of the learned scan path after the fifth episode. The total reconstruction error for this path is 0.045.

For a longer learning phase with more scan episodes we expect the gain of the proposed method to be even more obvious. We also hope to learn characteristic scan paths for different object classes in the future. An analysis of the system with respect to different scan purposes such as object recognition or the acquisition of a 3d model also remains to be done.
Nevertheless we believe that the proposed concept, if realized on appropriate hardware, can result in an intelligent scanner which allows a more efficient acquisition and storage of objects. Possible applications are learning, recognition, and grasping of objects in the area of service robotics or finding 3d models in data bases. In addition, it is an example for learning strategies for problem solving in the area of computer vision.

8 References

[1] Callari, F. G. and Ferrie, F. P.: Autonomous Recognition: Driven by Ambiguity. Proceedings of the Conference on Computer Vision and Pattern Recognition, 701–707, 1996.

Figure 10 Key views of the learned scan path after episode 8. This path yields the smallest total error of 0.030.

[2] Dickinson, S. J., Christensen, H. I., Tsotsos, J. K., and Olofsson, G.: Active Object Recognition Integrating Attention and Viewpoint Control. Computer Vision and Image Understanding, 67(3), 239–260, 1997.

[3] Wallraven, C. and Bülthoff, H. H.: Automatic Acquisition of Exemplar-Based Representations for Recognition from Image Sequences. CVPR 2001 - Workshop on Models vs. Exemplars, IEEE CS Press, 2001.

[4] Maver, J. and Bajcsy, R.: Occlusions as a Guide for Planning the Next View. IEEE Transactions on Pattern Analysis and Machine Intelligence, 15(5), 417–433, 1993.

[5] Hlaváč, V., Leonardis, A., and Werner, T.: Automatic Selection of Reference Views for Image-based Scene Representations. Proceedings of the European Conference on Computer Vision (ECCV), Lecture Notes in Computer Science, No. 1064, vol.1, 526–535, Springer, 1996.

[6] Wixson, L. E.: Gaze Selection for Visual Search. TR 512 and Ph.D. Thesis, Computer Science Dept., U. Rochester, May 1994.

[7] Chen, S. Y. and Li, Y. F.: Optimum Viewpoint Planning for Model-Based Robot Vision. IEEE 2002 World Congress on Computational Intelligence (WCCI) / Congress on Evolutionary Computation, 634–639, 2002.

[8] Paletta, L. and Pinz, A.: Active Object Recognition by View Integration and Reinforcement Learning. Robotics and Autonomous Systems, 31(1–2):1–18, 2000.

[9] Reinhold, M., Deinzer, F., Denzler, J., Paulus, D., and Pösl, J.: Active Appearance-Based Object Recognition Using Viewpoint Selection. In Girod, B., Greiner, G., Niemann, H., and Seidel, H.-P. , editors, Vision, Modeling, and Visualization 2000, 105–112, infix, Berlin, 2000.

[10] Deinzer, F., Denzler, J., and Niemann, H.: On Fusion of Multiple Views for Active Object Recognition. Pattern Recognition – 23rd DAGM Symposium, 239–245, Springer, Berlin, 2001.

[11] Sutton, R. S. and Barto, A. G.: Reinforcement Learning: An Introduction. MIT Press, Cambridge, 1998.

[12] Kaelbling, L. P., Littman, M. L., and Moore, A. P.: Reinforcement Learning: A Survey. Journal of Artificial Intelligence Research, vol. 4, 237–285, 1996.

[13] Russell, S. and Norvig, P.: Artificial Intelligence - A Modern Approach. Prentice Hall, 2003.

[14] Burr, D. C., Morrone, M. C., and Spinelli, D.: Evidence for Edge and Bar Detectors in Human Vision. Vision Research, 29(4):419–431, 1989.

[15] Jones, J. P. and Palmer, L. A.: An Evaluation of the Two-Dimensional Gabor Filter Model of Simple Receptive Fields in Cat Striate Cortex. Journal of Neurophysiology, 58(6):1233-1258, 1987.

[16] Eckes, C. and Vorbrüggen, J. C.: Combining Data-Driven and Model-Based Cues for Segmentation of

Video Sequences. Proceedings WCNN96, 868–875, Press & Lawrence Erlbaum Ass., 1996.

[17] Maurer, T. and von der Malsburg, C.: Tracking and Learning Graphs and Pose on Image Sequences of Faces. Proceedings of the 2nd International Conference on Automatic Face- and Gesture-Recognition, 176–181, 1996.

[18] Lades, M., Vorbrüggen, J. C., Buhmann, J., Lange, J., von der Malsburg, C., Würtz, R. P., and Konen, W.: Distortion Invariant Object Recognition in the Dynamic Link Architecture. IEEE Trans. Comp., vol. 42, 300–311, 1993.

[19] Peters, G.: A View-Based Approach to Three-Dimensional Object Perception. PhD thesis, Shaker Verlag, Aachen, Germany, 2002.