in: Proceedings of the SSPR 1998, Springer Verlag, 1998.

Tracking of Rotating Objects

Gabriele Peters, Christian Eckes, Christoph von der Malsburg

Institut für Neuroinformatik, Ruhr-Universität Bochum,
Systembiophysik, Universitätsstr. 150, D-44780 Bochum, Germany

Abstract. A representation of a three-dimensional object is autonomously learned from a sequence of the rotating object. The representation consists of single views in form of graphs and is achieved by performing a segmentation-based tracking of the object. First we apply a segmentation algorithm which is based on gray level values. This provides the location of the object in the images and a rough shape of it. Then we position landmarks on the object in the first frame of the sequence. These landmarks are tracked throughout the sequence on the basis of Gabor wavelet responses and guided by the segmentation result. During rotation landmarks are lost and new landmarks are added when object parts vanish or come into sight, respectively.

1 Introduction

In this paper we describe the learning of a viewpoint-invariant object representation. In our system each view of an object is represented by a single graph. Each node of a graph is labeled with features which describe the local surroundings of the node position. A graph for a particular view is generated autonomously by performing a segmentation-based tracking with the nodes of the graph of the previous view.

2 Segmentation-Based Tracking

The segmentation-based tracking of a rotating object consists of two parts. In the first part we perform a segmentation of each image of the sequence based on graylevel values, which provides the region of the image occupied by the object (see subsection 2.1). In the second part we track landmarks on the object guided by the result of the segmentation.

2.1 Segmentation

The segmentation method is described in [1] and is based on the system of [2]. The segmentation model contains Potts spins with coarse-to-fine dynamics comparable to real-space renormalisation methods often used in theoretical physics. Average intensity is used as the only low-level cue, although the system

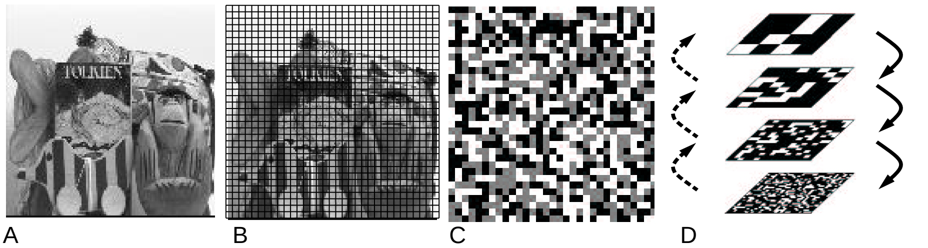

Fig. 1. Overview of the segmentation model. The complex scene shown in **A** is divided into $32 \times 32 = 1024$ patches in **B**. **C** shows the corresponding randomly initialized spin image for $k = 3$, in which each spin value is displayed as the appropriate gray level. **D** illustrates the renormalisation of the interaction between spins on different resolution levels (arrows on left) and the coarse-to-fine dynamics (arrows on right).

is able to make use of additional cues if they become available. The segmentation model [2] on which we have based our work divides an incoming image of some fixed resolution into N small patches I_i, $i = \{1, 2, ..., N\}$. Each surface receives a label $S_i, i = \{1, 2, ..., N\}$ that encodes its membership of one of several possible segments (see figure 1**A**–**C**). Because on the analogy between this label-based model and an interacting spin system in solid state physics, we call such a label a "spin". The range of values, k, allowed for a spin, $S_i \in \{1, 2, ..., k\}$, is a parameter of the system and is set to $k = 2$, because we want to separate only one object from the background. The images in our video sequences have been reduced to a resolution of 256×256 pixels and we use $N = 32 \times 32 = 1024$ spins, resulting in patches of $8 \times 8 = 64$ pixels per spin. The aim now is to find that spin configuration which encodes the "correct" segmentation of the given scene. Each spin S_i interacts with all other spins S_j via an interaction matrix W_{ij}. The difference in mean intensity $\left|\overline{I_i} - \overline{I_j}\right|$ at the corresponding image regions is used to compute the interaction W_{ij} between the two spins S_i, S_j assigned to these positions - the desired segmentation is mapped onto the global minimum of the following energy function:

$$E\left(S\right) = -\frac{1}{2} \sum_{i=1}^{N} \sum_{j=1 j \neq i}^{N} W_{ij} \cdot \delta_{S_i, S_j} \qquad \text{with} \tag{1}$$

$$W_{ij} = \max\left(1 - \left|\overline{I_i} - \overline{I_j}\right|/\alpha, 0\right) - \overline{W} \tag{2}$$

The parameter α – we use $\alpha = 100$ – in combination with the maximum function ensures that the difference in average intensity on the interval $[0, \alpha]$ is mapped to $[0, 1]$. To stress the *Gestalt* law of neighborhood [3], we restrict the interaction to spins with distances below 7.1 spins. In order to map low similarity to negative interaction and high similarity to positive one, we subtract the mean interaction $\overline{W}$ from all similarity values to obtain the used interaction W_{ij}.

We use the Metropolis [4] algorithm at zero temperature with a coarse-to-fine dynamics to let the system relax to a local energy minimum (see figure 1**D**

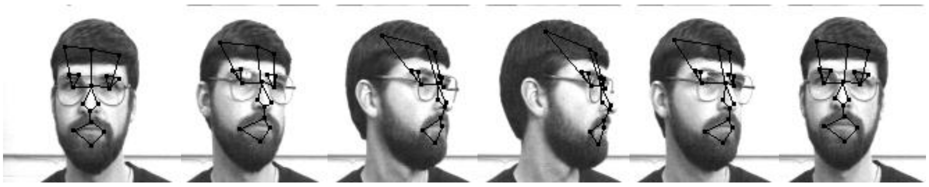

Fig. 2. Tracking of facial landmarks. The frames 1, 10, 20, 30, 40, and 50 of the sequence are shown here.

and [1] for the details). We have used 3 stages and $N(1) = 1024$ as the number of spins in the highest resolution. The number of spins in each resolution is given by $N(n) = N(1) \cdot 2^{-2(n-1)}$.

2.2 Tracking

The tracking procedure we use is described in [5] and based on [6] and [7]. Given a sequence of a moving object and the pixel position of a landmark of the object for frame n, the aim is to find the corresponding position of the landmark in frame $n+1$. As a visual feature, which describes the local surroundings of a landmark, responses of Gabor wavelets of different frequencies and orientations are used. These filter responses form the *jet* $\mathbf{J}$. The jth component of the jet can be written in terms of amplitude a_j and phase ϕ_j. Thus, a similarity function between two jets $\mathbf{J}$ and $\mathbf{J'}$ can be defined as

$$S\left(\mathbf{J}, \mathbf{J'}, \mathbf{d}\,\right) := \frac{\sum_j a_j a_j^{'} \cos\left(\phi_j - \phi_j^{'} - \mathbf{d} \cdot \mathbf{k}_j\right)}{\sqrt{\sum_j a_j^2 \sum_j a'^2_j}} \tag{3}$$

with $\mathbf{d}$ being the displacement vector of the two jets and $\mathbf{k}_j$ being the wave vectors of the Gabor filters. If $\mathbf{J}$ and $\mathbf{J'}$ are extracted at same pixel positions in the frames n and $n+1$, $\mathbf{d}$, and thus the new position of the landmark, can be found by maximizing S with respect to $\mathbf{d}$ (see figure 2 for tracking facial landmarks).

2.3 Combining Segmentation and Tracking

With each image we perform a segmentation as described in subsection 2.1. But this may provide also regions which are regarded as belonging to the object due to their gray levels, but in fact do not belong to it (see figure 3). We get rid of them by simply choosing that segment as object, which is closest to the center of the image. Given this result we mask the original images with it (see again figure 3). Now we can start the segmentation-based tracking by covering the whole masked image of the start view with a grid graph. Then those nodes are deleted which lie on the background or which lie on the object but are too close to the background. The reason for this is to prevent nodes from clinging to the

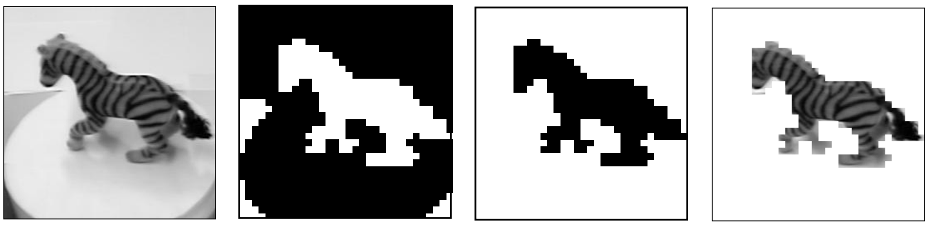

Fig. 3. Segmentation. Original image, result of the segmentation described in subsection 2.1, result after getting rid of wrong segments, masked image.

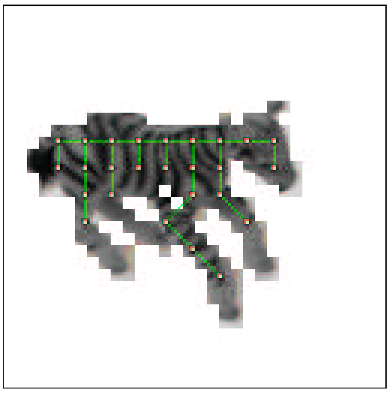

Fig. 4. Masked image with its graph. The first masked frame of a sequence with its start grid graph on the object.

outline contour of the object during tracking. (Nodes on contours are tracked very well in the sense that they stay on the contour, but they do not keep track of the same object landmark.) The minimal allowed distance to the background segment is determined by a certain fraction p of the radius of the largest Gabor kernel used for tracking. For display purposes the remaining nodes are connected by a minimal spanning tree (see figure 4). Given this reduced grid graph for the first frame of the sequence we perform with each node the tracking we have described in subsection 2.2. During the rotation of the object some parts of it will vanish while others become visible. Therefore it is necessary that some nodes will be lost and some other nodes have to be added. So, after a tracking step for a new frame we delete and add some of the nodes.

1. Deletion of nodes: It is again checked whether the new node positions are still on the object and whether the nodes did not move too close to the contour of the object. A third deletion condition is that nodes come too close to other nodes. In this case both nodes would contribute nearly the same information to the representation of the view. The minimal allowed distance of two nodes is the same as the minimal allowed distance to the background segment.
2. Addition of nodes: Here we check whether new nodes can be added according

to the result of the segmentation and consistent with the already existing nodes. We first modify the masked image in such a way that a rectangular surroundings of each already existing node is set to "background". The side length of these surrounding region is the same as the distance of the nodes in the initial grid graph of the first frame.
For this modified masked image we again create a new grid graph. All of the nodes of this new grid graph which lie on the background now or which are too close to the object's outline contour are deleted. In case that there are still nodes in the grid graph after this procedure, these nodes are added to the already existing nodes, but only if they do not come to close to other (already existing) nodes.

After the deletion and addition of nodes the new set of nodes is connected by a new minimal spanning tree for display purposes. The same procedure (tracking, deletion, and addition of nodes) is repeated for every frame.

3 Simulations and Results

For our simulations we used 100 frames per sequence, each of them with a resolution of 256^2. For the tracking we use Gabor wavelets with four different frequencies and eight different orientations. For the grid graphs we chose a grids of 15×15 nodes, and the fraction of the radius of the largest Gabor kernel was $p = 0.2$. We present here the result of a simulation with a toy zebra performing a full rotation on a turntable. In figure 5 you see some frames of the tracked sequence. Results for other sequences can be seen at *http://www.neuroinformatik.ruhr-uni-bochum.de/ini/PEOPLE/gpeters/top.html*.

4 Discussion and Outlook

We have shown that our system is capable of learning a representation for a rotating object by a novel combination of algorithms for segmentation on the one hand and for tracking on the other hand. Tracking of rotating objects means to catch new aspects of the object while it rotates and to realize where old parts of it vanish. Our method meets these conditions. In addition the learning works autonomously with little a priori knowledge about the object. The only assumptions are that the object has to be close to center of the images, that it has to be separable from the background by the segmentation, and that the surface structure is textured enough to allow the nodes to be tracked properly.

Up to now our representation belongs to the category of multiple-view based models, i.e., each view is independent. We plan to make a transition to a representation of a 3D-object in form of an *aspect graph* (e.g., [8], [9]). For this purpose it is necessary to define similarities between the graphs of successive views. We believe that the proposed representation is well suitable to define such similarities by using cluster techniques, with the number of deleted and added nodes as criterions.

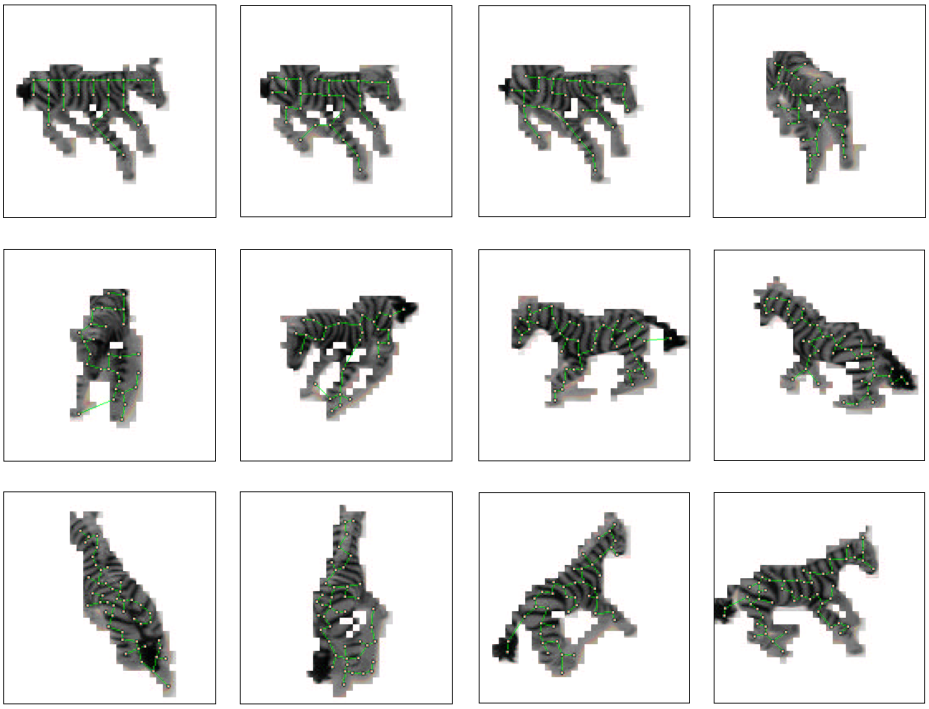

Fig. 5. Results for the zebra sequence. The first three images of the first row show the tracked graphs of the first three frames of the sequence. From the first to the second frame three nodes have been added (on the legs of the zebra), from the second to the third frame two nodes have been deleted (at the backside of the zebra and one of the previously added ones) and two nodes have again been added (at the mouth and in the middle of the body). The rest of the images show the representations for the frames 10, 20, 30, . . . , 90. Some of these images show the problem that the gray level segmentation has to deal with shadows, especially the first image of the second row and the second image of the third row.

Acknowledgments

We would like to thank Michael Neef for system administration, Torsten Wolf for fruitful discussions, and Rolf Würtz for proofreading this paper.

References

[1] C. Eckes, J. C. Vorbrüggen, Combining Data-Driven and Model-Based Cues for Segmentation of Video Sequences, *Proceedings WCNN96*, INNS Press & Lawrence Erlbaum Ass., San Diego, CA, USA, 16–18 September, pp. 868–875, 1996.

[2] J. C. Vorbrüggen, Zwei Modelle zur datengetriebenen Segmentierung visueller Daten, *PhD-thesis*, Ruhr-Universität Bochum, 1994.

[3] M. Wertheimer, Untersuchungen zur Lehre von der Gestalt. II, *Psychol Forschung*, vol 4, pp. 301–350, 1923.

[4] N. Metropolis, A. Rosenbluth, M. Rosenbluth, A. Teller, E. Teller, Introduction of the Metropolis Algorithm for Molecular-Dynamics Simulation, *J. Chem. Phys*, vol. 21, p. 1987, 1953.

[5] T. Maurer, C. von der Malsburg, Tracking and Learning Graphs and Pose on Image Sequences of Faces, *Proc. 2nd Int. Conf. on Automatic Face- and Gesture-Recognition*, Killington, Vermont, USA, IEEE Comp. Soc. Press. Los Alamitos, California, pp. 176-181, 1996.

[6] D. J. Fleet, A. D. Jepson, Computation of Component Image Velocity from Local Phase Information, *Int. Journal of Computer Vision*, vol. 5(1), p. 77, 1990.

[7] W. M. Theimer, H. A. Mallot, Phase-Based Binocular Vergence Control and Depth Reconstruction using Active Vision, *CVGIP: Image Understanding*, vol. 60(3), p. 343, 1994.

[8] J. J. Koenderink, A. J. van Doorn, The Singularities of the Visual Mapping, *Biological Cybernetics*, vol.24, pp. 51-59, 1976.

[9] M. F. Roy, T. van Effelterre, Aspect Graphs of Algebraic Surfaces, *Proceedings of the 1993 International Symposium on Symbolic and Algebraic Computation*, pp. 135-143, 1993.