Two Methods for Comparing Different Views of the Same Object

Gabriele Peters [*] Barbara Zitova [†] Christoph von der Malsburg [*]

Abstract

The viewing hemisphere of a 3-dimensional object can be partitioned into areas of similar views, termed *view bubbles*. We compare two procedures of generating view bubbles: *tracking* of object features, i.e., Gabor wavelet responses, by utilizing the continuity of successive views and *matching* of features in different views, which are assumed to be independent. Both procedures proved to be appropriate to detect canonical views. We found no difference concerning the size of the view bubbles, but tracking provides more precise correspondences than matching. Tracking is more appropriate for recognizing *changes* of features, whereas matching is more suitable if features of the *same* appearance are to be found.

1 Subject of Investigation

For pose-invariant object recognition and pose estimation of objects it is necessary to utilize an appropriate object representation. A naive representation might consist of densly spaced views of an object's viewing sphere. Our aim is to reduce such a "full" representation to only some representative views and the relations between them. Such a sparse representation belongs to the *aspect graph* approaches proposed by, e.g., [1], [5]. To choose representative views (*aspects*) for a final representation our plan is first to determine for each view of a "full" representation a surrounding area of similar views, termed *view bubble* (see figure 1). Later the aspects for the final representation of the object can be derived from the overlaps of the view bubbles.

In this paper we describe the generation of the view bubbles. We restrict our investigation to the upper hemisphere of an object's viewing sphere. We compare two procedures of determining the similarity of two views: *matching* a representing graph of one view to another view, and *tracking* object features, i.e., Gabor wavelet responses, from one view to another view. During the matching procedure each view is treated independently, whereas the tracking procedure utilizes the continuity of neighbouring views. Our investigations were guided by the question which procedure - matching or tracking - is more appropriate to find for each view of the hemisphere view bubbles of *maximal size* containing views of *maximal similarity*?

[*] Institut für Neuroinformatik, Systembiophysik, Ruhr-Universität Bochum, Universitätsstr. 150, D-44780 Bochum, Germany. E-mail: Gabi.Peters@neuroinformatik.ruhr-uni-bochum.de

[†] Department of Image Processing, Institute of Information Theory and Automation, Academy of Sciences of the Czech Republic, Pod vodárenskou věží 4, 182 08 Praha 8, Czech Republic. E-mail: zitova@utia.cas.cz

BMVC 1999 doi:10.5244/C.13.49

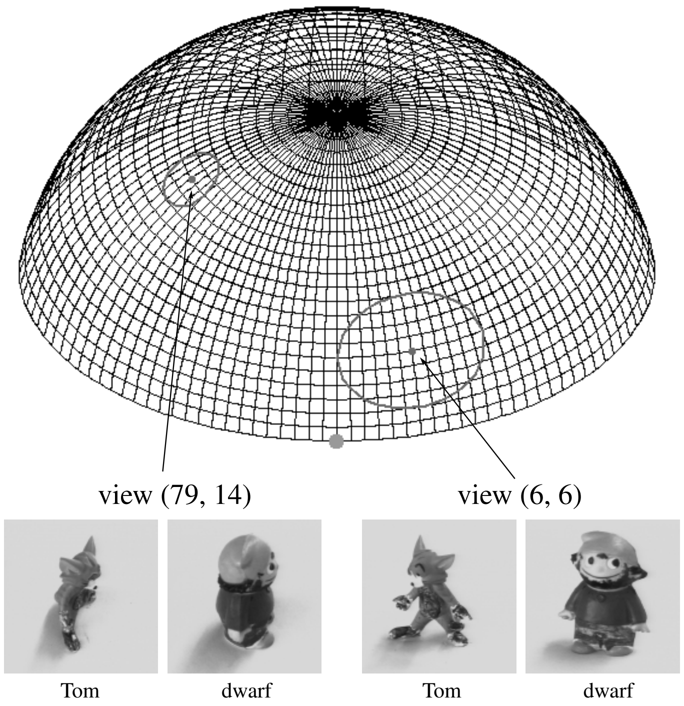

Figure 1: Viewing Hemisphere with Examples of View Bubbles. The representation of a viewing hemisphere consists of 100 × 25 views. Each crossing of the grid stands for one view. The angle between two neighbouring views is 3.6 degrees in either direction. The dot in front marks view (0, 0). The depicted view bubbles have been determined with the tracking procedure for object "Tom". View (79, 14) provides a small view bubble. It includes views which cover a range of 21.6 degrees in x-direction and 14.4 degrees in y-direction. View (6, 6) provides a larger view bubble, which covers a range of 43.2 degrees and 28.8 degrees, respectively.

2 Description of the System

2.1 Preprocessing

For each recorded view of an object we first perform a segmentation based on gray levels which separates the object from the background. Then we put a grid graph onto the segment of the image which has been assigned to the object. At each vertex of the graph we extract features which describe the surroundings of the vertex, i.e., local features of the special view of the object. Thus we derive a representation for each view in form of a *model graph* which provides the basis of both, the matching and the tracking procedure (see figure 2).

Segmentation: The segmentation method is based on the system of [6]. An image is divided into small patches. Each patch receives a label that encodes its membership of one of several segments. We need two segments, because we want to separate one object from the background. The aim is to find the label configuration which encodes the "correct" segmentation of the given scene. Each label interacts with neighbouring labels via an

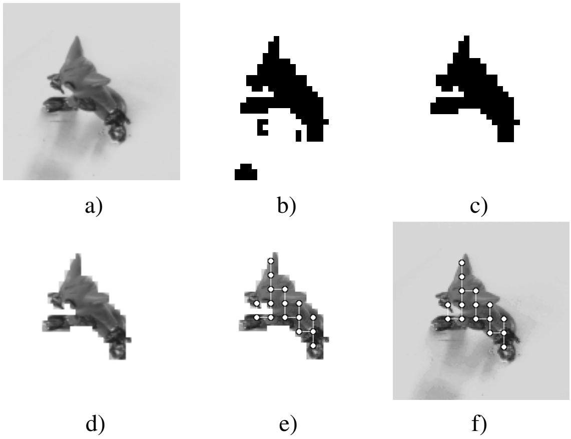

Figure 2: Preprocessing. a) original image, b) gray level segmentation, c) centered segmentation, after eliminating wrong segments, d) original image masked with result of centered segmentation, e) grid graph on object segment, f) grid graph on original image.

interaction matrix. The interaction between two labels is computed from the difference in mean intensity at the corresponding image regions. The desired segmentation results from coarse-to-fine dynamics which relax to a local energy minimum. The segmentation as described may also provide regions, which are regarded as belonging to the object due to their gray levels, but in fact do not belong to it, like shadows or reflections. We get rid of them by simply choosing that segment as object, which is closest to the center of the image (see figure 2 c), centered segmentation).

Gabor Transform and Similarity Function: The original image is convolved with a family of Gabor kernels, which differ in wavelength and orientation. We chose 4 wavelenghts and 8 orientations. The kernels take the form of a plane wave restricted by a Gaussian envelope function. At each image coordinate we obtain filter responses for each Gabor wavelet. Filter responses at one image coordinate form a *jet* $\mathcal{J}$. We can express the ith component of a jet in terms of amplitude a_i and phase ϕ_i: $\mathcal{J}_i = (a_i,\ \phi_i)$. Thus, a similarity function $\mathcal{S}$ between two jets $\mathcal{J}$ and $\mathcal{J}'$ can be defined as $\mathcal{S}(\mathcal{J},\ \mathcal{J}') = \frac{1}{2} \cdot \left(\frac{\sum_i a_i a'_i \cos(\phi_i - \phi'_i)}{\sqrt{\sum_i a_i^2 \sum_i a'^2_i}} + 1 \right)$. $\mathcal{S}$ is the similarity function we used for our simulations, for the matching as well as for the tracking procedure.

Grid Graphs: Given the result from the centered segmentation we mask the original image of size 128×128 pixels with it. We cover the whole masked image with a grid graph of 13×13 vertices. All vertices 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 vertices from incorporating too much information of the background. Then each vertex is labeled with the jet, which corresponds to the position of the vertex. For display purposes the remaining vertices are connected by a minimal spanning tree (see figure 2 e) and f)).

2.2 Matching Object Features

Elastic Graph Matching is described in detail in [2]. Given a graph with vertices labeled with jets, the aim of matching this graph to an image is to find new vertex positions which optimize the similarity of the vertex labels to the features extracted at the new positions. In a first stage (global move) the rigid graph is shifted across the image. For each position we calculate the total similarity of the new positioned graph to the original graph. The total similarity is just the average similarity taken for each vertex by using the similarity function $\mathcal{S}$. The position which provides the highest similarity is the starting position for the second stage, which permits small graph distortions. The vertices are shifted in a small surroundings of their starting position. After this local move the optimal position of the graph is found at the position which provides the highest total similarity.

2.3 Tracking Object Features

The tracking procedure we use is described in [3]. 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 $n+1$. A similarity function S' between two jets $\mathcal{J}$ and $\mathcal{J}'$ is defined, which differs slightly from $\mathcal{S}$: $S'\left(\mathcal{J}, \mathcal{J}', \vec{d}\right) := \frac{\sum_i a_i a_i^{'} \cos\left(\phi_i - \phi_i^{'} - \vec{d}\vec{k}_i\right)}{\sqrt{\sum_i a_i^2 \sum_i a'^2_i}}$ with $\vec{d}$ being the displacement vector of the two jets and $\vec{k}_i$ being the wave vectors of the Gabor filters. If $\mathcal{J}$ and $\mathcal{J}'$ are extracted at same pixel positions in the frames n and $n+1$, $\vec{d}$ (and thus the new position of the landmark) can be found by maximizing S' in its Taylor expansion. For each vertex of the graph of frame n the displacements are calculated for frame $n+1$. Then a graph is created with its vertices at the new corresponding positions in frame $n+1$, and the labels of the new vertices are extracted from the new positions. To compensate for a subpixel error $\Delta\vec{d}$ the phases of the Gabor filter responses are shifted according to $\Delta\phi_i = \Delta\vec{d} \cdot \vec{k}_i$.

2.4 Generation of View Bubbles

For each view (i, j) of the hemisphere an affiliated view bubble is created, with the view as its center. To determine the view bubble, we compare neighbouring views of (i, j) in all directions (east, west, north, and south). We match (or track) the grid graph of view (i, j) onto the neighbouring views. If the similarities to the graph of view (i, j) are still sufficiently high we depart one step further from (i, j). We begin with the views $(i-1, j)$ and $(i+1, j)$ (taking the continuity from the views $(0, j)$ to $(99, j)$ into account). If both views provide a sufficiently high similarity to the start graph we go on with the views $(i-2, j)$ and $(i+2, j)$. We stop this procedure if one of both tested views becomes too dissimilar. By doing the same for the vertical direction we obtain four views $(i-n, j)$, $(i+n, j)$, $(i, j-m)$, and $(i, j+m)$, which define the view bubble for view (i, j). To depict it we draw an ellipse through these four views with view (i, j) as its center. Figure 1 shows two ellipses projected onto the viewing hemisphere. For both procedures we used the same similarity threshold 0.77. A difference between tracking and matching concerning the similarity of views lies in the fact that during matching the similarity is calculated always in reference to the starting view, whereas during tracking the similarity refers to the preceding view.

3 Methods of Comparision

We ran our simulations with two objects, a simple one (the "dwarf") and a more complex one (the cat "Tom") (see figure 1). "Simple", in contrast to "complex", means that the views of the object do not change rapidly while the object rotates. The "dwarf" is a relatively convex object with rather similar shape for all viewing directions, whereas "Tom" is a more irregular object with faster changing views. We used two different methods to compare the view bubbles generated by the matching (resp. tracking) procedure. With statistical analyses we made a quantitative comparison, and by judging the correspondences, which were found by both procedures, we compared both procedures qualitatively.

For the quantitative statistics, we determined the area of each view bubble by calculating the area of the ellipse, described in section 2.4. (In [4] we also counted for each view, in how many other view bubbles it is contained. The results for this condition, however, resemble the results published here.) For both objects we carried out a t-test to proof the hypothesis of different means of the areas of view bubbles for the tracked versus the matched view bubble samples.

For the qualitative comparison we chose several sequences on the hemisphere (from a starting view to a destination view) for both objects, and show the results for two of them in section 4. For each sequence we performed the matching and the tracking procedure. To assess the correspondences by visible inspection we displayed for each view of the sequences the resulting matched and tracked graphs and plotted the calculated similarities in diagrams. The sequences had an average size of about 8 frames which means a covered rotation angle of 25,2 degrees.

4 Results

The diagrams in figures 3 and 4 show results from the quantitative comprisons. Figure 3 shows the distributions of areas of view bubbles for the object "Tom", figure 4 for object "dwarf". For both figures the first diagram depicts the results from the tracking procedure, the second the ones from the matching procedure. Lighter colors encode larger areas of view bubbles. To compare the results for tracking and matching the third diagram shows the difference between the first and second diagram.

From the diagrams we get following results. For both objects, "Tom" and "dwarf", the distribution of areas of view bubbles is qualitatively similar for the tracking procedure and for the matching procedure. The back view seen from slightly above and the front view provide the largest bubbles, and they can be regarded as *canonical views*.

These results hold for both objects, "Tom" and "dwarf". But there is a difference between the objects. The tracking procedure provides larger view bubbles than the matching procedure for the majority of views for the more complex object "Tom", whereas for the more simple object "dwarf" it is the other way around: here the matching procedure provides larger view bubbles than the tracking procedure for the majority of views. The one-tailed t-test, with which we compared the mean values, was significant with $\alpha = 1\%$ for each case.

The figures 5 (object "Tom") and 6 (object "dwarf") show results for the qualitative comparison, i.e., the assessing of the correspondences. The first part of both figures displays views with graphs resulting from tracking (first row) and matching (second row). Both rows start with the starting view of the sequence. The next two images show the

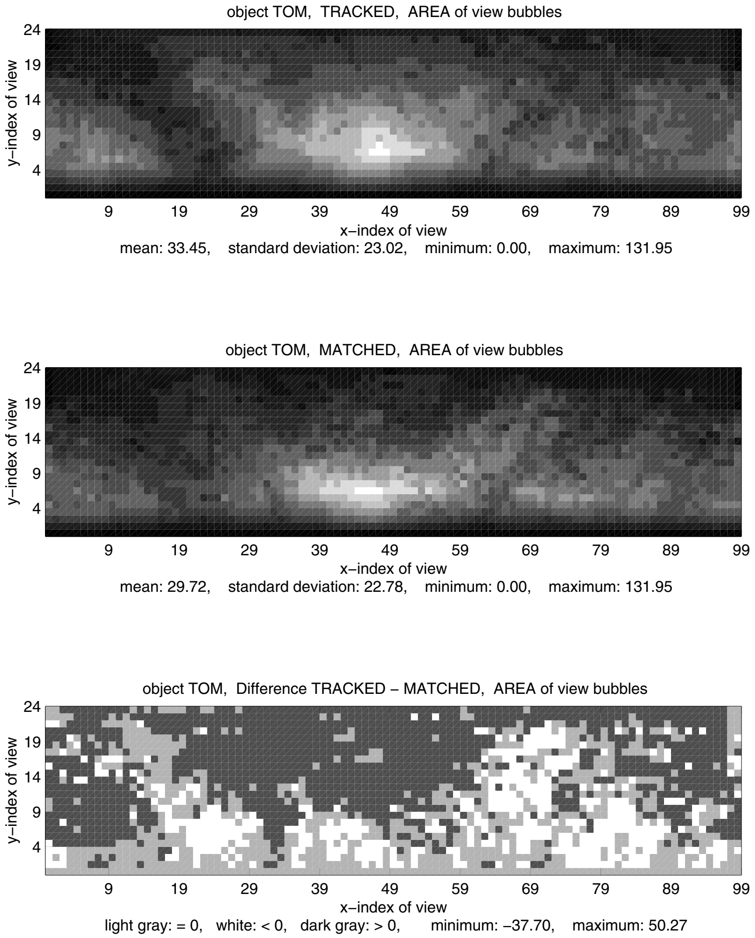

Figure 3: Distribution of View Similarities for Object “Tom”.

views where matching provided the last successfully matched and first mismatched graph in the sequence. Arrows point to mismatched vertices. The last images of the rows show the last views of the sequence where tracked graphs still keep the corresponding points, whereas the matched graphs do not. In the headers of the images the indices of the views are printed. (“Tr” means “tracked”, “Mt” means “matched”.) For each view of both sequences tracking provides the same or better correspondences than matching. The second part of both figures shows a diagram where the similarities for each view of the sequence to the starting view is plotted for tracking as well as matching. From the similarity diagrams of these and other analyzed sequences reported in [4] we get the following result. At the beginning of a sequence the tracking procedure always provides higher similarities than the matching procedure. This relationship is reversed at that point of the sequence where the matching starts to provide poor correspondences, whereas tracking provides good correspondences until the end of the sequence (see figure 7).

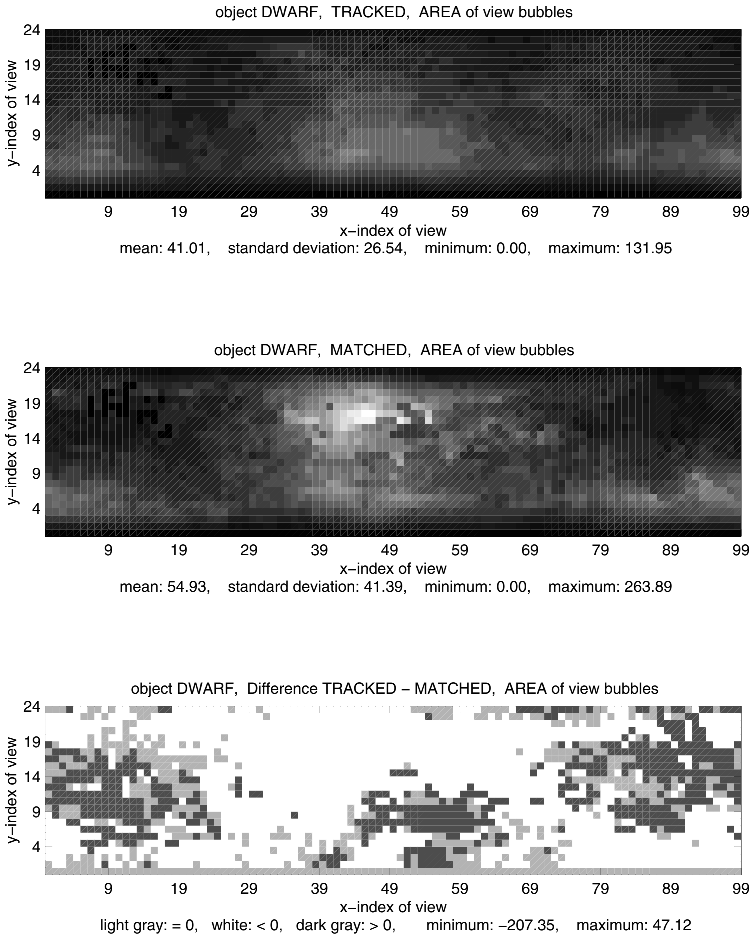

Figure 4: Distribution of View Similarities for Object “Dwarf”.

5 Discussion and Conclusion

Both procedures, matching and tracking of object features, are suitable to generate a distribution of view similarities on the viewing hemisphere of a 3–dimensional object. On the hemisphere areas of large and of small view bubbles arise. Centers of areas of large view bubbles can be regarded as *canonical views* (see figure 8).

From both test objects no statement was possible about the superiority of one procedure in terms of size of view bubbles, because for the more complex object “Tom” tracking provided larger view bubbles, whereas matching outperformed tracking for the simpler object “dwarf”. A possible explanation for this result could be that the rapidly changing views of object “Tom” cannot be matched over larger distances, because the matching procedure is looking for the *same* appearance of the object features. We assume that the tracking procedure leads to larger view bubbles for “complex” objects, whereas matching is superior for “simple” objects. But this hypothesis has to be verified for more

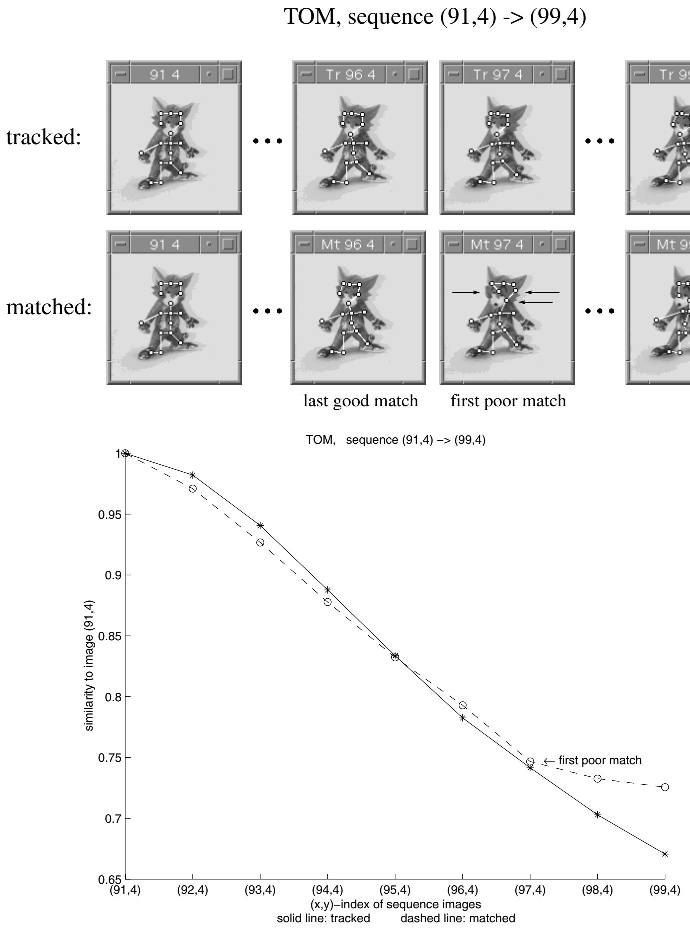

Figure 5: Correspondences for Object "Tom".

examples.

A reason for the more precise correspondences found by tracking could be the fact that an object feature changes its appearence while the object rotates. The feature in the tracking procedure adapts to this change, whereas the matching procedure always searches for the same starting feature. The more the rotation proceeds the more difficult it is for the matching procedure to find the correct point, whereas the tracking utilizes *continuous information*. Matching is the more appropriate method if the task is to find features with the *same* appearance, tracking is the more appropriate method if *changes* of the features should be followed. Even if it would turn out that matching is superior to tracking for simple objects in terms of size of the view bubbles we suggest that precise correspondences should take priority over larger view bubbles, particularly for further processing. For a view interpolation, e.g., precise correspondences are necessary, and to establish these correspondences, the continuity information of successive views has to be utilized. Accordingly, our final conclusion is that tracking of object features is superior

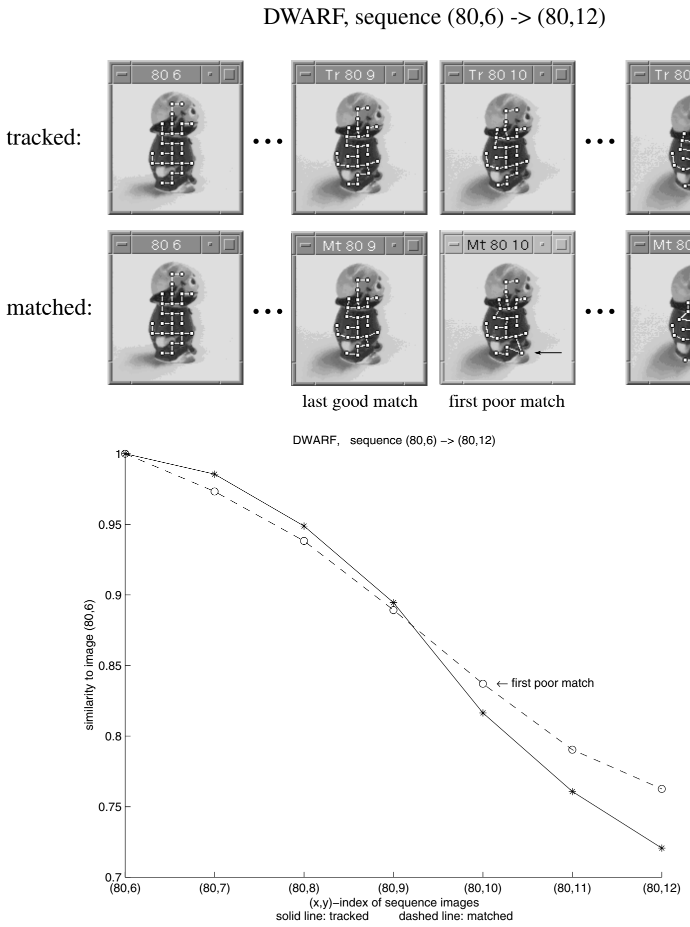

Figure 6: Correspondences for Object “Dwarf”.

to matching for estimating similar view areas of 3–dimensional objects, especially for complex objects.

References

[1] J. J. Koenderink and A. J. v. Doorn. The Internal Representation of Solid Shape with Respect to Vision. *Biological Cybernetics*, 32:211–216, 1979.

[2] M. Lades, J. C. Vorbrüggen, J. Buhmann, J. Lange, C. v. d. Malsburg, R. P. Würtz, and W. Konen. Distortion Invariant Object Recognition in the Dynamic Link Architecture. *IEEE Transactions on Computers*, 42:300–311, 1993.

[3] T. Maurer and C. v. d. Malsburg. Tracking and Learning Graphs and Pose on Image Sequences of Faces. In *Proceedings of the 2nd International Conference on Auto-*

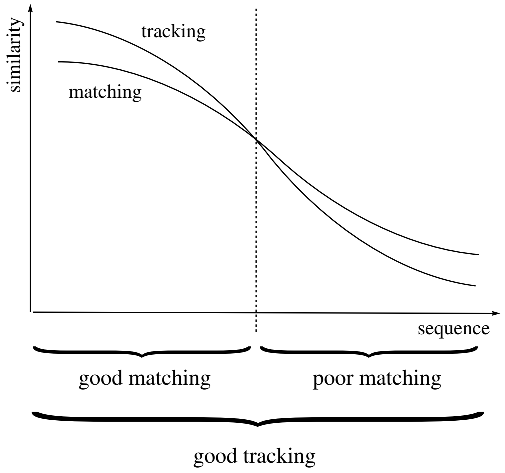

Figure 7: Qualitative Similarity Diagram. “Good” and ”poor” is meant in the sense of correct, respectively incorrect, correspondences. See description in the text for details.

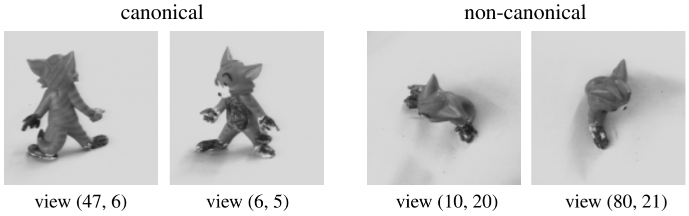

Figure 8: Canonical and Non-Canonical Views for Object “Tom”. View (47, 6) is the view with the largest area of its view bubble (generated by tracking). Compare with the first diagram of figure 3.

matic Face- and Gesture- Recognition, pages 176–181, Killington, Vermont, USA, October 1996.

[4] G. Peters, B. Zitova, and C. v. d. Malsburg. A Comparative Evaluation of Matching and Tracking Object Features for the Purpose of Estimating Similar-View-Areas of 3-Dimensional Objects. Internal Report IRINI 99-06, Institut für Neuroinformatik, Ruhr-Universität Bochum, D-44780 Bochum, Germany, April 1999.

[5] M. Seibert and A. M. Waxman. Adaptive 3-D Object Recognition from Multiple Views. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 14(2):107–124, 1992.

[6] Jan C. Vorbrüggen. *Zwei Modelle zur datengetriebenen Segmentierung visueller Daten*, volume 47 of *Reihe Physik*. Verlag Harri Deutsch, Thun, Frankfurt am Main, 1995.