Automatic acquisition of exemplar-based representations for recognition from image sequences

Proceedings of CVPR'01

Christian Wallraven* and Heinrich Bülthoff
Max-Planck-Institute for Biological Cybernetics
72076 Tübingen, Germany
{christian.wallraven, heinrich.buelthoff}@tuebingen.mpg.de

Abstract

We present an exemplar-based object recognition system which is capable of on-line learning of representations of scenes and objects from image sequences. Local appearance features are used in a tracking framework to find 'key-frames' of the input sequence during learning. The representation of the stored sequences which are used for recognition of novel images consists only of the appearance features in these key-frames and contains no further a-priori assumptions about the underlying sequences. The system is able to create sparse and extendable representations and shows good recognition performance in a variety of viewing conditions for databases of natural and synthetic image sequences.

1 Introduction

Many computer vision recognition systems typically followed Marr's approach to vision in building three-dimensional (3D) representations of objects and scenes (e.g., [2]). The view-based or exemplar-based approach, however, has recently gained much momentum due to its conceptual simplicity and strong support from studies on human perception [5, 20]. In this approach, an object is represented by viewer-centered 'snapshots' instead of an object-centered 3D model.

In recent years exemplar-based vision systems based on local image descriptors have demonstrated impressive recognition performance [1, 6, 8, 13, 18]. These systems normally work on a pre-defined database of objects (in the case of [13] more than 1000 images). However, one problem these approaches have hardly addressed is how to acquire such a database. When considering an active agent which has to learn and later recognize objects, the visual input the agent receives consists of a sequence of images. The temporal properties of the visual input thus represent another source of information the agent can exploit [9].

In this work, we therefore want to go one step further and move from a database of static images to image sequences. We present a low-level recognition system which is capable of building on-line image-based scene representations for recognition from image sequences. These sequences are processed by the system to find 'key-frames' - frames where the visual change in the scene is high (related to the idea of 'aspect-graphs' from [7]). These key-frames are then characterized by local image descriptors on multiple scales which are used in the learning and recognition stages. The system uses the same framework for learning and for recognition which leads to an efficient and simple implementation. Furthermore, an inherent disadvantage of the traditional image-based approaches is that they require many training images in order to be able to recognize objects in a variety of viewing conditions thus leading to high memory requirements. Our framework addresses this problem by offering an automatic data-driven way of selecting which image data to keep and by using only a small amount of the image data itself for recognition.

We tested the recognition system on databases of computer rendered sequences of faces and real-world video-sequences of cars. Recognition results demonstrated the robustness of the system under a variety of viewing conditions. We also present preliminary results for an incremental learning strategy which is used to expand the key-frame representation.

2 Overview

During learning (see Figure 1), the input consists of an image sequence, which is processed on-line. In the first frame features are extracted at several scales, which are then tracked in the subsequent frames. Once tracking fails, a new key-frame is added to the representation and a new set of

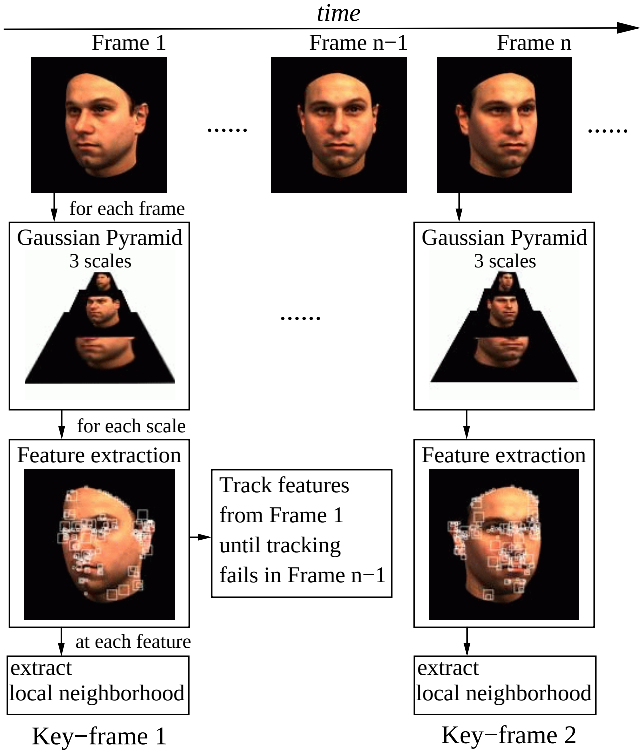


Figure 1. Overview of the learning stage.

features is extracted in this key-frame, and the whole process repeats until the sequence ends. The final representation of the sequence then consists of a number of key-frames containing visual features on multiple scales. For recognition of novel test images, local features are first extracted and then these features and their configurations are matched against the key-frames in all learned representations.

Section 3 presents the databases which were used to test the performance of the system. Section 4 describes the features which are used in the system together with the matching algorithm that is applied both for tracking and recognition. In Section 5 the generation of key-frames is analyzed and recognition results for the system on the databases are presented in Section 6. In Section 7 a scheme to extend already existing models is introduced and preliminary results are given. Section 8 considers the proposed framework as an exemplar-based approach and addresses critical points often brought forward against this approach.

3 Databases

The first database, which was used for tracking and recognition experiments in sections 5.1 and 6.1 contains 6 short video sequences mainly used for optic flow benchmarks[1] and 4 sequences taken with an off-the-shelf camcorder. This database contains sequences with various types of motion in the scene.

The second database consists of 30 sequences of faces turning from -90 degrees (left) profile view to +90 degrees (right) profile view. The sequences were rendered from models of 30 individuals which were recorded with a laser scanner (CYBERWARE^{TM}) to obtain highly realistic 3D structure and texture data. Each of the sequences contains 61 frames at pose intervals of 3 degrees and was rendered with a black background. This database was used to test the system under controlled motion and illumination conditions (for a detailed examination of the performance of the system under illumination see [21]).

The third database consists of sequences, which were taken with an off-the-shelf camcorder (Sony DCR-TRV17). We took 10 video sequences of different cars which were recorded by walking with a video-camera around a car under daylight lighting conditions. No effort was made to control shaking or distance to the car. For recognition we took a test-set of 25 photos of 5 of the cars with a standard digital camera (Olympus C-1400, note that this also means different camera optics) under different viewing conditions. This database together with the photos contains all kinds of variations in motion in the scene and different backgrounds and illumination in the test-set.

All sequences were resized to 512x512 pixels and processing was done on the full color images except for two monochrome sequences in the first database.

4 Feature extraction and matching

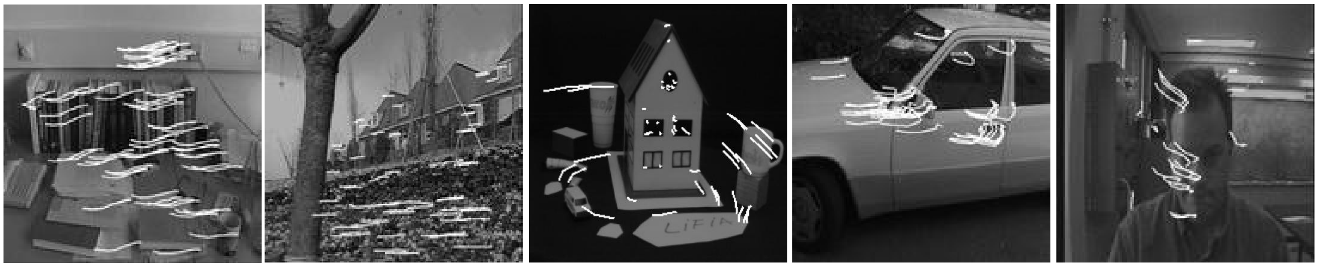
4.1 Visual Features

We decided to use corners as the basis for visual features, since these were found to be good and robust features under many viewing conditions in numerous other works (e.g.,[14]). In order to extract corners we use a standard algorithm [17] modified to integrate information about all three color channels since this further improved robustness in the color image sequences we processed. Corners are found by inspecting the structure in a 9x9 neighborhood $\mathcal{N}$ of each pixel in the following way:

$$\mathbf{H} = \begin{pmatrix} \sum_{\mathcal{N}} \left\langle \frac{\partial \vec{I}}{\partial x}, \frac{\partial \vec{I}}{\partial x} \right\rangle & \sum_{\mathcal{N}} \left\langle \frac{\partial \vec{I}}{\partial x}, \frac{\partial \vec{I}}{\partial y} \right\rangle \\ \sum_{\mathcal{N}} \left\langle \frac{\partial \vec{I}}{\partial x}, \frac{\partial \vec{I}}{\partial y} \right\rangle & \sum_{\mathcal{N}} \left\langle \frac{\partial \vec{I}}{\partial y}, \frac{\partial \vec{I}}{\partial y} \right\rangle \end{pmatrix}$$

with $<,>$ as dot-product and $\vec{I}$ as the vector of RGB-values such that an element of $\mathbf{H}$ is e.g. $H(1,2) = \sum \frac{\partial I_r}{\partial x}\frac{\partial I_r}{\partial y} + \frac{\partial I_g}{\partial x}\frac{\partial I_g}{\partial y} + \frac{\partial I_b}{\partial x}\frac{\partial I_b}{\partial y}$. The smaller of the two eigenvalues λ_2

[1] Available at http://sampl.eng.ohio-state.edu/˜sampl/database.



a) desk − 11 frames b) sflowg − 12 frames c) MOVI − 4 frames d) car01 − 8 frames e) osu−1 − 7 frames

Figure 2. Examples of feature tracking. The first frame of the sequence together with trajectories of the tracked features from all scale levels is shown.

of $\mathbf{H}$ yields information about the structure of the neighborhood.

In the next step, a hierarchical clustering algorithm is used to cluster the values of λ_2 into two sets and use the one with the higher mean-value as the feature set. Using a clustering algorithm has the advantage that one does not need to specify a hard-coded threshold for the values of λ_2 for each image. Furthermore, the hierarchical structure makes a coarse-to-fine strategy possible and thus speeds up the clustering process considerably.

Visual features are then extracted by storing the image location of each corner together with the intensity values in a 11x11 pixel image patch surrounding the corner. This is done on all three scales to get a coarse-to-fine representation of the image consisting of about 200 features in total.

4.2 Feature matching for tracking and recognition

Both tracking and recognition in the proposed system consist of finding correspondences between feature sets in two images. For tracking, matching is done between feature sets from the most recent key-frame of the sequence and the current frame. For recognition, matching is done between the test frame and all the key-frames in the model. As a consequence we use the same matching algorithm for both tracking *and* recognition which leads to an efficient implementation.

The method, which we shortly summarize, is based on [12] (where it was used for stereo matching) and [15] and is a variation on the well-known Procrustes problem of rotating two datasets onto each other.

The algorithm constructs a similarity matrix $\mathbf{A}$ with each entry $A(i,j)$ given by two contributing terms:

$$A(i,j) = e^{\frac{1}{2\sigma^2_{\mathrm{dist}}}(-\mathrm{dist}(\mathrm{f_i},\mathrm{f_j}))} \cdot e^{\frac{1}{2\sigma^2_{\mathrm{NCC}}}(1-\mathrm{NCC}(\mathrm{f_i},\mathrm{f_j}))}$$

where f_i is the position of feature i in one image and f_j of feature j in another image and i,j index all feature pairs in the two images.

The first term measures the image distance (dist) from feature i to feature j with σ_{dist} set to small values thus giving a tendency toward close matches in distance. The second term measures the normalized cross-correlation (NCC) of the neighborhoods from features i and j, with σ_{NCC} set to values larger than 0.5, thus biasing the results toward features that are similar in appearance. The NCC is evaluated in the small image patch surrounding each feature.

From the Singular Value Decomposition of $\mathbf{A}$

$$\mathbf{A} = \mathbf{U} \cdot \mathbf{V} \cdot \mathbf{W}^{\mathbf{T}}$$

the matching algorithm constructs the modified SVD of this matrix, defined by

$$\mathbf{A}' := \mathbf{U} \cdot \mathbf{I} \cdot \mathbf{W}^{\mathbf{T}}$$

where $\mathbf{I}$ is the identity matrix. Features i and j are matched if they have both the highest entrance in the column and row of $\mathbf{A}'$ and if the NCC exceeds a given threshold th_{NCC}. This method effectively provides a least-square mapping between the feature sets and at the same time ensures that there is a one-to-one mapping of features[2]. For recognition and tracking purposes we count the number of matches in a test frame with respect to a given reference frame and use the *percentage of matches* as the decision criterion for matching.

The resulting algorithm is capable of matching under affine feature transformations between two images ensuring reliable and flexible tracking and recognition.

5 Key-frames

5.1 Tracking between Key-frames

As shown in Figure 1, a representation of the sequence is generated by tracking the features found in the first frame

[2] Note, that the feature mapping can occur between sub-sets of features.

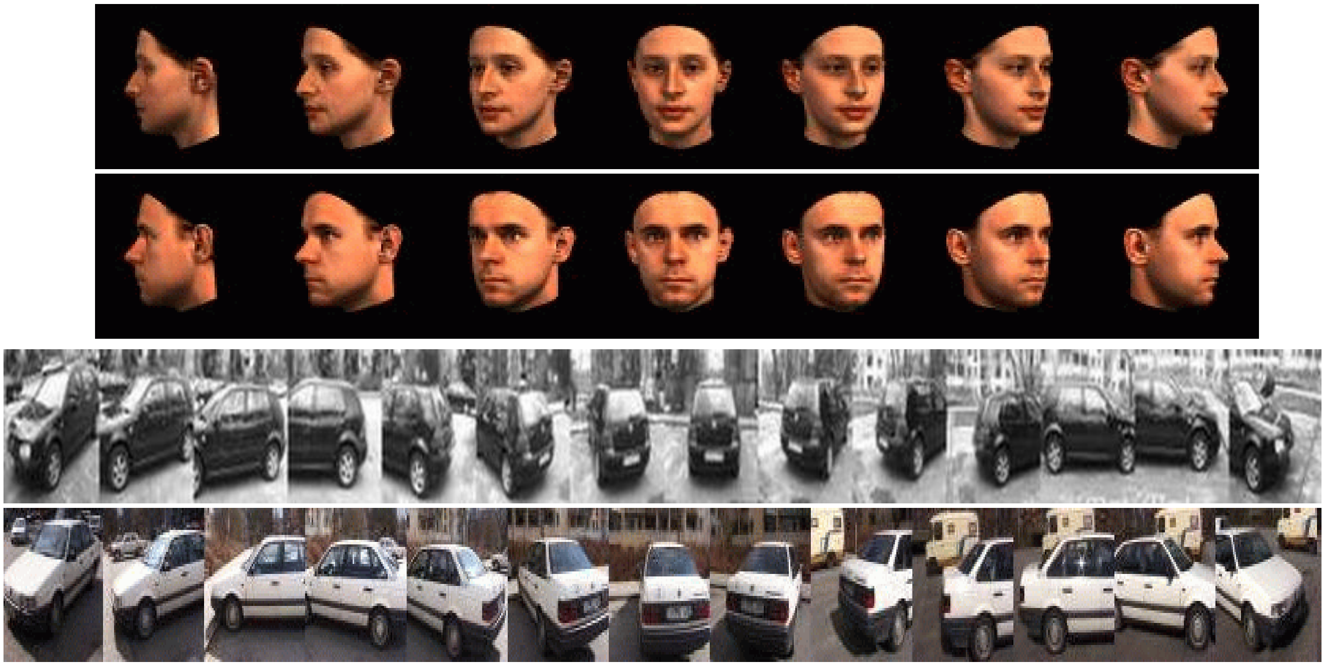


Figure 3. Key-frames for two face sequences and two car-sequences.

until tracking fails at some time. We use the matching algorithm to match features from the first to the current frame. The parameters are $\sigma_{\mathrm{dist}} = 10, 20, 40$ pixels for each of the three scales to restrict feature displacement and $\sigma_{\mathrm{NCC}} = 0.7$ and $th_{\mathrm{NCC}} = 0.7$ to find features which are similar in appearance. Tracking fails once the percentage of matches falls below $th_{\mathrm{track}} = 25\%$; at this time a new key-frame is inserted into the model, a new set of features is found and the process repeats until the sequence ends. In the end, the model of the sequence consists of a number of key-frames with visual features at several scales.

Figure 2 shows examples from the tracking process using five sequences from the first database. The sequence in Figure 2a contains mainly translational motion orthogonal to the viewing direction and Figure 2b contains translational motion in depth. Figure 2c shows tracking for rotational and Figure 2d for rotational and translational motion, while the sequence in Figure 2e contains a more complicated motion of a face getting nearer to the camera. In all cases, the motion in the image is accurately captured by the tracking procedure.

The matching algorithm is capable of handling also larger feature displacements between frames, so that it could even be implemented in real-time, since it does not need to process every single frame.

5.2 Key-frame generation

In the tracking phase, the algorithm has four parameters which control its behaviour: σ_{dist}, σ_{NCC}, th_{NCC} and th_{track}. For the settings given in the previous section we first analyzed the key-frames for the face- and car-database.

For the controlled motion in the face-database the system found 7 key-frames for each of the 30 sequences with each key-frame roughly in the same pose (Table 1 and Figure 3).

Key-frame	1	2	3	4	5	6	7
Pose (deg)	-90	-61	-33	-6	18	44	73

Table 1. Average pose in degrees for each key-frame

The angular distance between key-frames is smallest for the frontal poses (between Key-frames 4 and 5). This is due to the fact that a rotation around the frontal view causes larger variations in features (such as ears disappearing and appearing) which leads to an earlier termination of tracking.

For the car-database with varying amounts of motion we found that a similar number of key-frames was generated overall, with the key-frames roughly covering the same viewpoints of the cars (Figure 3 shows some corresponding key-frames from two car sequences).

In general, we found that increasing th_{track}, σ_{NCC}, th_{NCC} and decreasing σ_{dist} leads to more reliable tracking

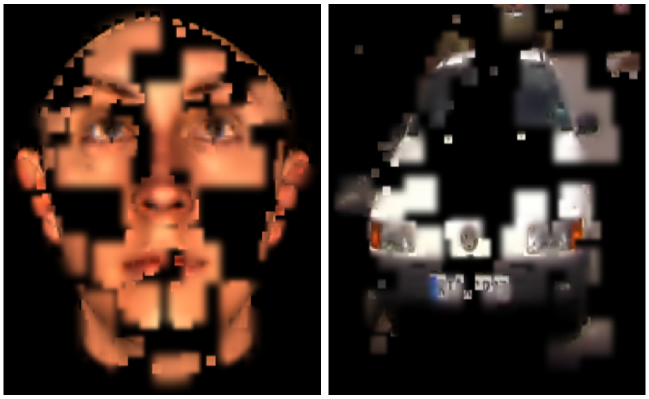


Figure 4. Reconstruction of two images from their feature representations.

but also generates a large number of key-frames. Our choice of parameters reflects a trade-off between generating a very sparse representation reliable recognition between the generated key-frames.

5.3 Size reduction

The final representation of the system consists of a number of key-frames containing small image patches which results in a large size reduction for the processed image sequences. This is an essential property for any image-based system working on sequences since otherwise huge amounts of data would have to be stored.

In order to calculate the size reduction of the representation, we compared the size of the final sequence models for both databases to the raw pixel data and to the size of mpeg2-encoded sequences[3]. For the car-database raw data consisted of 512x512 pixel frames with 25 frames per second and on average 32.2 seconds of image data yielding a total of 603.8MByte, mpeg-2 compressed data size was on average 22.1MByte. For the face-database raw data size was 45.8MByte, mpeg-2 compression resulted in 1.6MByte on average.

The models from the car-database generated 17.2 key-frames on average with each key-frame containing 200 features and 11x11 pixels per feature, yielding a total of 1.2MByte; this leads to size reduction of **99.8%** for raw data and **94.7%** for mpeg2-compressed data. The face models consisted of 7 keyframes and resulted in size reduction of **98.9%** and **69.7%** respectively. The lower size reduction for the mpeg2-compressed face-database was due to the concentration of all 200 features within the region of the face which resulted in larger overlap of features over scales.

While many other approaches characterize features with a lower-dimensional feature vector ([6, 8, 10, 13, 18]) and thus could provide even further size reduction, we want to emphasize that with the proposed framework some of the original image content is still available.

Figure 4 shows two examples of the reconstruction of image data from local features in the representation. Reconstruction is done by drawing all features at the coarsest scale, expanding the image to the next scale and repeating this process until the finest scale is reached. In the reconstructed images, small-scale features appear sharp whereas large-scale features are blurred.

6 Recognition of Images

In the following we will describe three experiments, which tested the recognition performance of the system on single images. The first experiment was done with a variety of image degradations to test the robustness of the system under synthetic noise. In the second experiment, we tested using face-database with faces varying in pose and illumination intensity. The third recognition experiment was done on the car sequences with a test-set of 25 pictures taken under different viewing conditions (with a different camera on different days).

In all experiments again the matching algorithm was used to recognize a given image by matching the features from this image to each frame in the database. For the first experiment matching was done against all individual frames of the *original sequences*, whereas in the second and third experiment the key-frame representation was used. In all cases, the frame with the highest percentage of matches was selected as the best matching frame.

6.1 Image degradations

Forty random images from the first database of short sequences were degraded with 7 types of image degradation (see Table 2) ranging from changes in intensity to warping the image. Also shown in Table 2 is the mean match percentage of the best matching frame and the recognition rate. Larger recognition failures occurred only in the shear, zoom and occlusion conditions, where feature distances were sometimes closer to neighboring frames due to the geometric transformations[4]. To demonstrate the limits of the feature-based approach, we randomly superimposed 12x12 pixel squares in the occlusion condition over the images so that 15% of the image was occluded. This led to a drastic reduction in recognition rate due to the almost destroyed local image statistics.

[3] The encoder is available at `http://www.mpeg.org/MPEG/MSSG/`. For all experiments the default parameters were used.

[4] Note, that the inter-frame distance in the first database is small, making this a hard task.

Figure 5a shows one correctly recognized test image from condition 8.

type	match percentage	recognition rate
1. brightness +60%	66.4%	97.5%
2. contrast +60%	65.6%	97.5%
3. noise +40%	57.6%	95.0%
4. equalized color	53.3%	95.0%
5. shear	31.9%	85.0%
6. zoom x 1.5	32.9%	82.5%
7. occlude 15%	5.8%	47.5%
8. all of 1,2,3,4,5	28.3%	77.5%

Table 2. Recognition results for image degradations

6.2 Face-database

For the face-database we created four test-sets: the first test-set consisted of all frames of the original sequences which were not key-frames. This was done to test whether the system generates a consistent representation for all sequences. In the second set we rendered all 30 sequences again with two different strengths of lighting, and the third set contained pose variation by ±15 degrees orthogonal to the direction of rotation. Finally, the fourth test-set combined all three sets into one. Table 3 lists the match percentage and the recognition rate for all test-sets and Figure 5b,c show recognition results from the second and third test-set.

	between key-frames	lighting variations	pose variations	all three
match percentage	60.2%	33.5%	39.5%	44.5%
recognition rate	100%	90.1%	98.2%	81.2%

Table 3. Recognition results for the face-database

While recognition results for each test-set alone were very good, recognition performance dropped for the combined test-set due to a greater rate of false matches. Almost all false matches for a given face F occurred for images of *other* faces in the same pose as the key-frames for face F. These views led to a higher percentage of matches than views of face F under different lighting in poses, which are far away from key-frame views.

Interestingly, the best matching key-frame corresponded to the correct pose in **99.2%** of all test-images regardless of face identity. This result shows that the system is capable of reliable pose-estimation (see also [21]).

6.3 Car-database

The test-set for the car-database consisted of 25 pictures of cars from the database in different viewing conditions. These can be broadly categorized into changes in lighting conditions (Figure 5d), different viewing distances (Figure 5e) and occlusion by other objects (Figure 5f). In order to reduce false matches due to large feature distances in the image plane, each test image was displaced by 40 pixels in four different directions and the match algorithm was run for each displacement. Table 4 lists again the percentage of matches and the *number* of false matches and Figure 5d-f show recognition results for each of the three viewing conditions.

	lighting variations (12 images)	viewing distance (8 images)	occlusion (5 images)
match percentage	19.1%	17.2%	18.5%
false matches	2	1	1

Table 4. Recognition results for the car-database

Compared to the face-database the average percentage of matches was much lower due to the large variations in appearance. A closer analysis of the recognized frames showed that the percentage of matches dropped very sharply around the best matching key-frame, whereas all other test images gave a nearly constant amount of matches over all key-frames.

7 Incremental Learning

Since the first frame of a sequence always becomes the first key-frame, the resulting set of key-frames will be different for a slightly different starting point. In order to overcome this problem and also to provide a general framework for incremental learning of models, an additional step in the learning phase is introduced.

The first learning phase generates a set of key-frames from a sequence as an initial reference representation. For each new sequence, again, key-frames are extracted. Then, each key-frame is compared to the already learned key-frames with the recognition algorithm. If no match is found[5], the frame is added to the representation, whereas for each match a hit-counter for the corresponding key-frame is increased.

[5] The threshold for rejection of a frame is set to $th_{\mathtt{rej}} = th_{\mathtt{track}}$ to ensure consistency

The resulting representation is a connected graph of characteristic views of the presented sequences with additional information about the frequency of each view. This additional information can then be used to speed up matching since it is more likely that a new frame matches a highly frequented view[6].

Here, we want to present preliminary results for this incremental learning scheme which were obtained with the face-database. We rendered additional sequences of two faces with three different pose animations. Figure 6 shows these sequences for one face together with the connected key-frame representation in its final state in a viewing sphere representation with each key-frame at its corresponding position.

The learning process started with the first sequence (running along the horizontal axis in Figure 6) which created the initial representation, then the second sequence was integrated (running along the vertical axis) which resulted in 6 additional key-frames in the periphery. The third sequence (diagonal from top left to bottom right) again added 6 key-frames. The final test-sequence consisted of 32 images covering a circle in the viewing sphere (the solid circle in Figure 6), which resulted in **9** key-frames on its own. Since the viewing sphere was already covered in many areas, only **3** key-frames were added to the representation (shown in Figure 6 with exclamation marks). The key-frame which received the most matches during the learning process was the frame in the center.

8 The Bigger Picture: The proposed framework as an exemplar-based approach

The proposed framework is purely exemplar-based, i.e. it does not rely on pre-constructed models or high-level a-priori knowledge, but instead constructs a representation closely connected with the original input data.

Some of the main arguments against exemplar-based methods, which are often put forward and which we want to address within the proposed framework, are that

- they require a large number of training examples,
- they use large amounts of storage for the representations,
- indexing into the representation takes a long time.

While the amount of training examples in our framework is presumably still higher to obtain invariant recognition than with an underlying model (e.g., [3] for faces), the combination of configurational information and appearance-based information in the local feature matching process shows increased robustness in many viewing situations (such as occlusion, change in lighting conditions [21], sensor noise). This property thus enables the system to generalize over more cases in comparison to other image-based techniques using whole images, which in turn results in fewer training examples. The matching algorithm essentially combines elements of two prominent approaches which are pursued in object recognition - the feature-based and the appearance-based approach.

In addition, as shown in Section 5.3, our framework creates sparse representations, where the selection of local features over several scales still allows for some actual image information to be present in the representation (see Figure 4). This represents a compromise between the two extremes of pure image-based approaches and highly abstracted model-based approaches.

With regard to matching times for recognition, the complexity of matching is linear with the amount of trained sequences, linear with the amount of key-frames found and cubic in the amount of features in each key-frame. The incremental learning technique described in the previous section, however, allows for a reduction in both training and recognition time since at some point the whole image space will be covered with key-frames. In addition, the concept of the canonical view reduces recognition time by taking into account the presentation statistics of the learned representation. Furthermore, more sophisticated database indexing techniques would offer further improvement for recognition time. A possible solution to overcome the cubic complexity for SVD will be to use sparse matrix techniques (as already suggested in [12]). This is based on the observation that any given feature normally has only a low number of possible match candidates thus making the similarity matrix $\mathbf{A}$ sparse.

The proposed exemplar-based framework thus is able to address these critical points and offer solutions to improve the performance of image-based systems. On the other hand, the representation of this system consisting of local features can be used as the basis from which to construct a higher-level representation. Analysis of the features in key-frames across models belonging to one category (e.g., faces) can be used to find common features or even groups of features (see also [19]) which are characteristic for this category. This would represent a step towards creating *model-based* knowledge from existing exemplar-based models.

9 Conclusion and Outlook

We have presented an exemplar-based recognition system which is capable of on-line learning of sparse scene representations from arbitrary image sequences. It uses a simple and efficient framework both for learning and recognition. The use of sequences together with the key-

[6] In Psychophysics terminology the frame with the highest number of matches would be called the *canonical view* (see [11] and also [4] for a recent study), which plays an important role in human object recognition.

framing technique enables the system to create representations which reflect the amount of visual change in the scenes. Recognition performance on databases containing synthetic and real-world image data demonstrated that the system is capable of recognition and reliable pose-estimation under a variety of different viewing conditions. In addition, results for the incremental learning scheme showed that the key-frame representation can be easily extended.

The system currently operates in closed-loop conditions (i.e. the input consists of pre-cut and labelled sequences for learning). With the use of more globally evaluated features such as color and texture histograms [10, 16] the system will decide autonomously when to begin a new model. In addition, the resulting models will include information about the object and the surroundings since the system uses no segmentation. While the recognition experiments have shown that the system can cope with background variations, in an active vision paradigm an attention module could actively select the features belonging to one particular object, while the others are discarded and not tracked.

References

[1] A. Baerveldt, ”A vision system for object verification and localization based on local features”, *Robotics and Autonomous Systems*, 34(2-3): 83-92, 2001.

[2] P. Beardsley, P. Torr and A. Zisserman, ”3D Model acquisition from extended image sequences”, *In Proc. ECCV’96*, 683-695, 1996.

[3] V. Blanz and T. Vetter, ”A morphable model for the synthesis of 3D faces”, *Proc. ACM SIGGRAPH 99*, 187-194, 1999.

[4] V. Blanz, M. Tarr and H. Bülthoff, ”What object attributes determine canonical views?”, *Perception*, 28, 575-599, 1999.

[5] H. Bülthoff and S. Edelman, ”Psychophysical support for a 2-D view interpolation theory of object recognition, *Proceedings of the National Academy of Science*, vol. 89, 60-64, 1992.

[6] D. Jugessur and G. Dudek, ”Local Appearance for Robust Object Recognition”, *In Proc. CVPR’00*, 834-839, 2000.

[7] J. Koenderink and A. van Doorn, ”The internal representation of solid shape with respect to vision”, *Biological Cybernetics*, 32, 1979.

[8] D. Lowe, ”Toward a Computational Model for Object Recognition in IT Cortex”, *In Proc. BMCV’00*, 20-31, 2000.

[9] A. Massad, B. Mertsching and S. Schmalz, ”Combining multiple views and temporal associations for 3-D object recognition”, *In Proc. ECCV’98*, 699-715, 1998.

[10] B. Mel, ”SEEMORE: Combining color, shape, and texture histogramming in a neurally-inspired approach to visual object recognition”, *Neural Computation*, 9:777–804, 1997.

[11] S. Palmer, E. Rosch and P. Chase, ”Canonical Perspective and the Perception of Objects”, *Attention and Performance IX*, Hillsdale, NJ: Lawrence Erlbaum Associates, 135-151, 1981.

[12] M. Pilu, ”A direct method for stereo correspondence based on singular value decomposition”, *In Proc. CVPR’97*, 261-266, 1997.

[13] C. Schmid and R. Mohr, ”Local Greyvalue Invariants for Image Retrieval”, *IEEE TPAMI*, 19(5), 530-535, 1997.

[14] C. Schmid, R.Mohr and C.Bauckhage, ”Evaluation of Interest Point Detectors”, *International Journal of Computer Vision*, 37(2), 151-172, 2000.

[15] G. Scott and H. Longuet-Higgins, ”An algorithm for associating the features of two images”, *In Proc. Royal Society of London*, B(244):21–26, 1991.

[16] M. Swain and D. Ballard, ”Color Indexing,” *International Journal of Computer Vision*, 7(1), 11-32, 1991.

[17] C. Tomasi and T. Kanade, ”Detection and tracking of point features”, *Carnegie-Mellon Tech Report CMU-CS-91-132*, 1991.

[18] T. Tuytelaars and L. van Gool, ”Content-based image retrieval based on local affinely invariant regions”, *In Proc. Visual ’99*, 493-500, 1999.

[19] S. Ullman, Erez Sali and M. Vidal-Naquet, ”A Fragment-Based Approach to Object Representation and Classification”, *Proc. IWVF 2001*, 85-102, 2001.

[20] G. Wallis and H. Bülthoff, ”Learning to recognize objects”, *Trends In Cognitive Sciences*, 3, 22-31, 1999.

[21] C. Wallraven and H. Bülthoff, ”View-based recognition under illumination changes using local features”, *In Proc. CVPR’01 - Workshop on Identifying Objects Across Variations in Lighting: Psychophysics and Computation*, 2001.

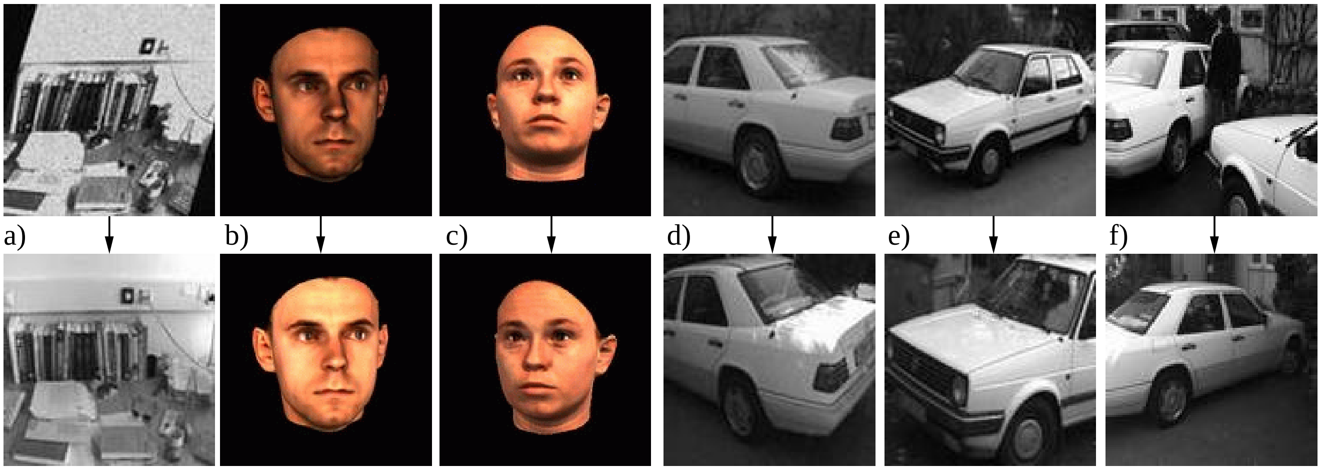


Figure 5. Recognition results with a) degraded and b)-f) novel images. Test images are depicted in the upper row.

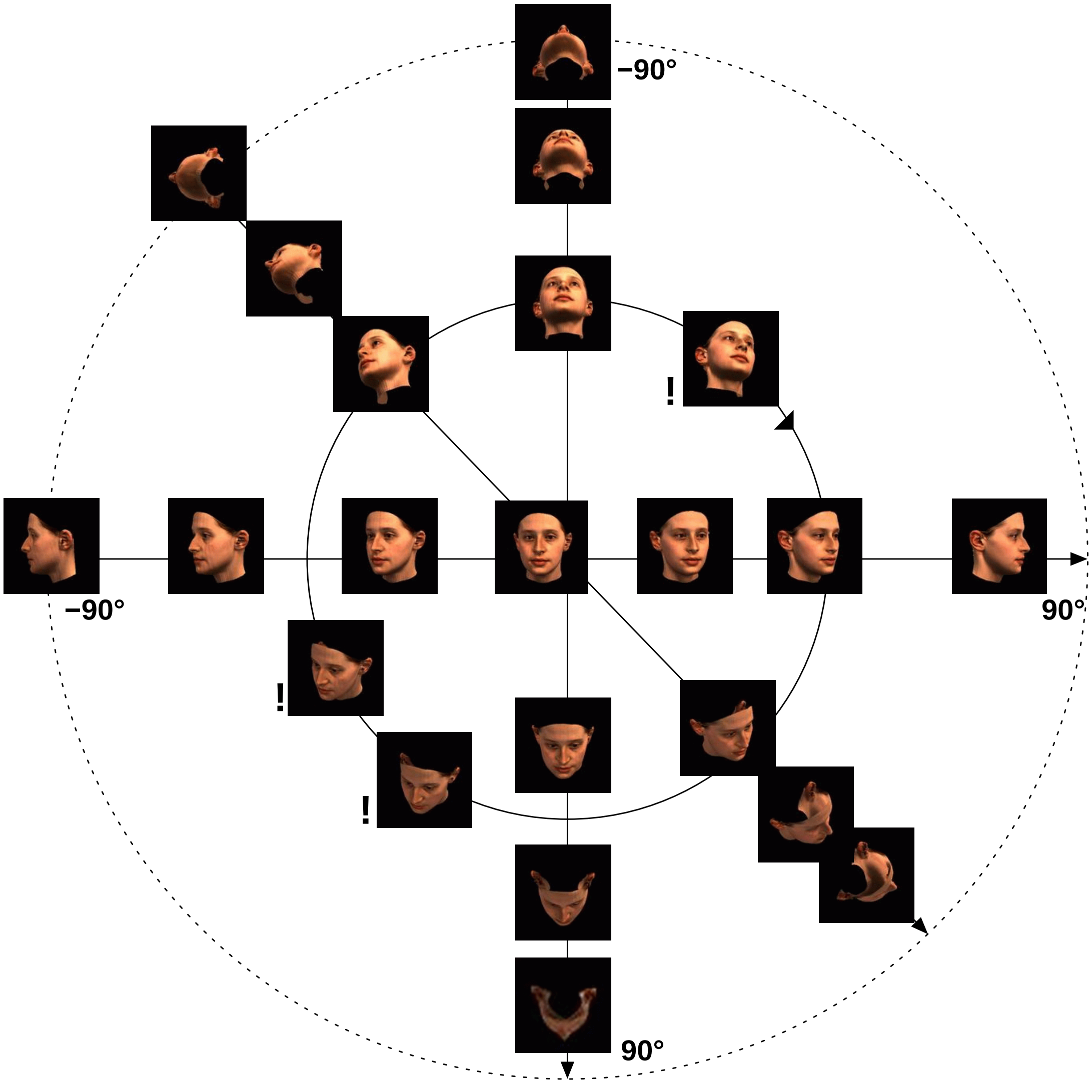


Figure 6. Key-frame representation after three additional sequences were integrated.