

Appearance-based 3D object recognition with time-invariant features

E. Delponte N. Noceti F. Odone A. Verri
DISI - Università degli Studi di Genova, Italy
{delponte, noceti, odone, verri}@disi.unige.it

Abstract

In this paper we explore the interlink between temporally dense view-based object recognition and sparse image representations with local keypoints. The temporal component is an add on that allows us to extract information which is distinctive of a given object in a given view-point range. We use temporal descriptions both for training and for testing. In the training phase each image sequence contains one object only, observed at different view points. At run time video shots are analyzed looking for known objects. Train and test video shots are represented by a structure of scale-space keypoints selected so that they are robust to view-point changes. In the matching phase we emphasize co-occurring keypoints and attenuate the importance of isolated points, both in the model and in the test representation. With our prototype recognition system we obtained very good results in controlled and unconstrained environments, setting the ground for real world applications such as automatic place recognition, or robot object grasping.

1 Introduction

The problem of describing 3D objects from visual information without explicitly computing their 3D structure has been extensively studied in the last decade. Among the better established methods it is worth mentioning multiple-view approaches and local approaches.

In this paper we explore in depth the interlink between temporally dense view-based object recognition and sparse image representations with local keypoints. In the training phase we acquire image sequences containing one object only, observed at continuous view points. At run time video shots are analyzed looking for known objects. Train and test video shots are represented by a structure derived from scale-space keypoints selected so that they are robust to view-point changes. In the matching phase we emphasize co-occurring keypoints and attenuate the importance of isolated points, both in the model and in the test representation. Our work is motivated by the fact that an image se-

quence does not just carry multiple instances of the same scene, but also information on how the appearance of objects evolves when the observation point changes smoothly. Since in many applications image sequences are available and often under exploited, our aim is to fill this gap. We believe that, in the appropriate application setting, observing an object from slightly different viewpoints, and looking for features that are distinctive in space, stable and smooth in time, can greatly help object recognition.

The popularity of local approaches for object recognition [7, 9, 12, 4] is due to the fact that, unlike global methods [10, 11], they produce relatively compact descriptions of the image content and do not suffer from the presence of cluttered background and occlusions. The main problem with local methods is that, while observing minute details, the overall object's appearance may be lost. Also, small details are more likely to be common to many different objects [13]. For this reason local information is usually summarized in global descriptions of the object, for instance in codebooks [2, 7]. Alternatively, closeness constraints can be used to increase the quality of matches [4]. Continuity along the sequence may help to cluster related keypoints observed at continuous view-points, in the case a dense sequence of the object of interest is available, as suggested in [5], where a simple tracking procedure is used to extract temporal information on the training phase only — at run time only one image is used. In [3] we started exploring descriptions based on image sequences, only on the training phase, obtaining an object description that we then used in the statistical learning framework. The limits of our previous approach are that the rather complex learning procedure made it difficult to grow the number of learned object, and the difference between training model (based on image sequences) and test models (based on single images) caused many false positives, as the scene complexity grew. The use of image sequences for the test phase has been explored both in computer vision and robotics [14, 4], in most cases, though, test data are analyzed looking for representative frames and then test is performed on one image. Extracting temporal information to the purpose of object recognition is not common, instead it is a natural choice in other

recognition tasks, such as gesture recognition. Local spatio-temporal features for action recognition are described in [6], based on finding sharp changes in the temporal direction that are distinctive of a given motion. Our time-invariant features are different since they are designed to recognize appearance and not dynamics.

The main contribution of our work is an appearance-based recognition method that exploits temporal coherence *both in training and test*, and the balance between the two descriptions is crucial to obtain good matches. As a consequence of this, the method fits naturally in the video analysis framework. In spite of the redundancy of the data that we use, the descriptions we obtain are compact since they only keep information which is very distinctive across time. A second important contribution is the matching procedure: simple nearest neighbor is strengthened with a strategy that favors matching between groups of coeval features, i.e. features belonging to similar fields of view. Our procedure allows us to identify groups of very strong matches, thus minimizing the false positives rate. Recognition is performed following a one-class at a time scheme. We report very good results on test sequences of increasing difficulty, in the presence of clutter, illumination and scene changes, occlusions, similar objects.

2 Our approach

We look for models of 3D objects starting from visual cues. Given an object of interest, in the training phase, we capture its 2D appearance at different temporally dense viewpoints. At the end of the training phase we obtain a model of an object as a *set of time-invariant features*, i.e., features obtained from keypoints trajectories longer than a fixed number of frames. Also, since we do not make any assumption neither on the camera motion, nor on the distance between camera and observed objects, we choose scale-invariant descriptions.

At run time, given a video shot (that could be the entire sequence, a part of it, or the buffer content in an online application) we extract time-invariant features from it, and then we match the description of the test sequence against the various training models. Matching is performed at feature level with a nearest neighbor approach, looking for groups of coeval features both in training and test.

2.1 Time-invariant feature detection

Given an image sequence, we detect corners in scale-space following Lindeberg [8], that is, choosing the scale where the response of the corner to a derivative-based operator is higher. We also estimate the corner principal direction, similarly to [9]. Then we track the corners over the sequence with a non-linear dynamic filter — the *unscented*

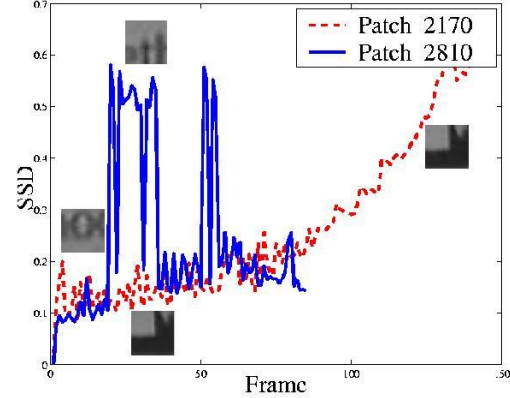


Figure 1. The SSD distance computed between the patch extracted in the first frame and the following ones (see text).

Kalman filter [15], that allows us to cope with non linearities in the system [1]. The unknown state of the system is $x_k = \{p_k, s_k, d_k\}$, where p_k is the keypoint position at time k , s_k its scale, d_k its principal direction. At the end of this extraction procedure, short trajectories are discarded.

2.2 Time-invariant feature descriptors

Each keypoint of the feature trajectories is first represented as a SIFT [9]. Then we apply a cleaning procedure that eliminates noisy trajectories: first, we compute the variance of the scale and of the principal direction of each trajectory. Then, trajectories with a high variance are further analysed in order to check whether they contain abrupt changes that could be caused by tracking errors. To do so, we perform a SSD correlation test between the first gray-level patch of the trajectory and the subsequent ones. In the presence of an abrupt change, the trajectory is discarded. Figure 1 compares two trajectories: the dashed one refers to a good feature, whose appearance varies slowly and smoothly in time; the solid one refers to an unstable trajectory, containing tracking errors (a visual inspection of the gray-level patches confirms this hypothesis). After the cleaning procedure, a time-invariant feature is described by (a) a spatial appearance descriptor, that is the average of all SIFT vectors of its trajectory, and (b) a temporal descriptor, that contains information on when the feature first appeared in the sequence and on when it was last observed. To increase the stability of the appearance descriptor we discard the keypoints lying outside an hyperball centered at the keypoints appearance centroid, whose radius is proportional to the keypoints standard deviation.

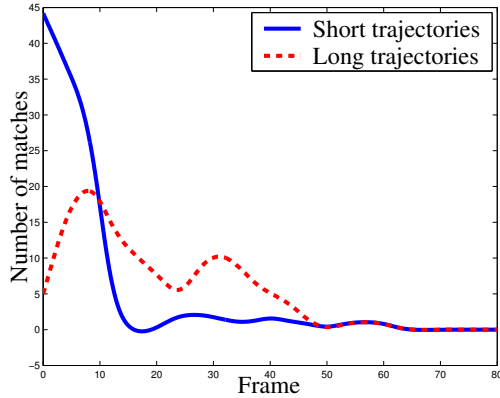


Figure 2. Matches between training and test sequences for different choices of N : solid line is for $N = 10$, dashed line is for $N = 50$.

2.3 Choosing the temporal range size

In the feature extraction phase, features robust to view-point variations are preferred. Therefore we extract long trajectories. Instead, on the description phase, long trajectories may contain too much information since they are the result of observing a feature on a long period of time (and possibly a high range of views). Descriptions generated from these long trajectories tend to oversmooth the appearance information, and may lead to a high number of false positives. This is especially true if the object is deeply 3D and its appearance changes dramatically over the sequence. To this purpose we apply, before computing the time-invariant feature descriptions, a cutting phase that cuts a trajectory into many sub-trajectories of length N . The choice of N is not crucial, and common sense rules may be applied: if N is too small we lose efficiency and compactness, as the model becomes very big. If N is too big, the information within a feature is oversmoothed. Figure 2 shows the effect of changing N : a small training sequence containing a small range of viewpoints is compared with a test sequence of the same object. A similar viewpoint of the training appears in the first part of the test video. The solid line shows the matches between training and test models obtained with $N=10$, the dashed line is for $N=50$. The smoothing effect as N grows is apparent. It is important to choose similar N both on the training and on the test sequences, so that we obtain descriptions carrying a similar amount of information. On this respect, we noticed that this procedure is extremely important when comparing sequences acquired with different modalities (e.g., still camera on the training phase and hand-held camera on the test phase, as in our experiments). In our experiments, unless otherwise stated, N is set to 10.

2.4 A model for an image sequence

The content of an image sequence is redundant both in space and time. We obtain compressed descriptions for the purpose of recognition, extracting a collection of time-invariant features over the sequence, that we call a *model* of the sequence, and discarding all the other information. We do not keep any information on the relative motion between the camera and the object, as it is not informative for the recognition purpose. In the training phase the sequence model is also a *model for the object*.

The space occupancy of our model is limited: considering that the keypoints detected in each frame are approximately 300. Thus, if every keypoint of the sequence participates to build the model, in the average case (a sequence of length 250) the model is made of approximately 75000 features. Our approach grants to obtain a compact representation of typically 1500 time-invariant features.

2.5 Matching sequence models

Let us consider a test sequence represented by a model T , and a training sequence, represented by an object model M . The problem we address is to see whether the test model T contains the object M . Then, for each feature in M , we check whether T contains a similar feature to it.

To compare features we apply a two-stages comparison: first comparing appearance, then exploiting temporal constraints. On the first stage, since the appearance descriptor is an average SIFT (that is, it is histogram-like), we compare similarities between features by *histogram intersection*. On the second stage we use the temporal description of each feature to look for groups of *coeval features*, i.e., groups of features that appear in a given view-point range, both in the training and in the test model. If a feature f_M matches with a feature f_T , we look for other features of the model that are coeval to f_M and see whether they did match with features of the test (Figure 3), while isolated features are discarded. This procedure increases the proportion of high scores when groups of coeval features appear both in training and test. This is more likely to happen if training and test models contain features coming from the same object. Experiments will show that this two-stages strategy is especially useful when the acquisition setting is very different between training and test.

Thanks to the compact representation of visual information the matching phase is efficient even when the sequence is long and there are several detected keypoints.

3 Experiments

In this section we report some experiments carried out on different video sequences acquired under various conditions. For each object that we want to recognize we perform

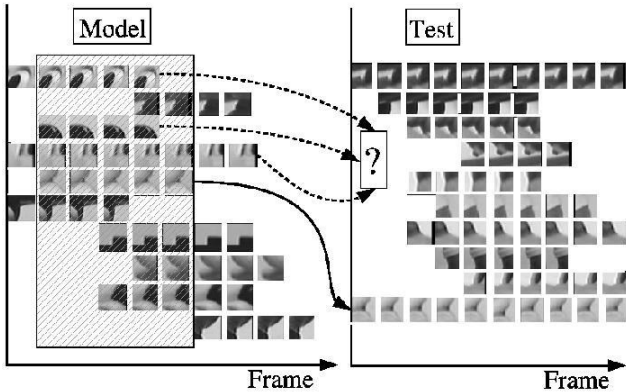


Figure 3. The matching procedure with temporal constraints. Coeval features overlap of at least 4 frames.

a training phase, first acquiring an image sequence of about 250 frames of the object on a turntable, to the purpose of observing it from all the view-points. Then we compute a model as described in previous section. A model from a video of 250 frames is usually made of approximately 1000 features. A first set of experiments is accomplished to assess the matching method and evaluate its robustness with respect to time delays between training and test, scene changes, variations in the camera motion. In a second set of experiments we test the robustness of the recognition system to localize the presence of object in a video, using various sequences containing some of the objects that were previously modeled, as well as many other objects of a similar type.

Recognition is performed following a one-class at a time scheme: for each object model we test its presence against the test video, identifying in what video chunks the object appeared. Notice that, if the number of models grows the system may slow down but there is no price to pay in terms of performance.

3.1 Matching

For the matching experiments we consider test sequences containing one or two objects from the first to the last frame, and compute the match between training and test models according to Sec. 2.5. We consider four objects, 2 planar objects (*book* and a ring binder: *winnie*), 2 complex 3D objects with many concavities (*dewey* and *goofy*).

Changes in scale, light and scene background

We use various test sequences for each object of interest. The results are shown in Table 1: on the columns are the objects models, on the rows various kinds of test models. Rows marked with T1 refer to test videos with light and

	Book	Goofy	Dewey	Winnie
1. Book T1	91	2	0	7
2. Book T2	97	0	0	3
3. Goofy T1	0	95	5	0
4. Goofy T2	1	95	2	2
5. Goofy zoom	8	64	12	16
6. Dewey T1	0	0	100	0
7. Dewey T2	0	5	90	5
8. Dewey opp	4	10	84	2
9. Winnie T1	9	0	0	91
10. Winnie T2	0	2	0	98

Table 1. A summary of matching experiments: the columns are the models, the rows different types of tests. The table report hit percentages of one type of test against the different models (see text).

scale variations, rows T2 correspond to sequences also containing other objects or clutter in the background.

Changes in the relative motion

To verify the robustness of our approach with respect to changes in the relative motion between the camera and the scene we set up an experiment in which the motion in test sequence is in the opposite direction with respect to the training sequence. The matches obtained for one object are shown in row (8) of Table 1, and they confirm that there is little influence of the type of motion on the results. Similarly we test the robustness of the method when test sequence zooms in: row (5) of Table 1 shows the percentage of matches obtained with a zooming sequence where the size of one object doubles, from the first to the last frame. This experiment corroborates the robustness of the descriptors to strong scale variation.

Multiple objects

In the second type of test we consider the presence of two objects, for both of which we have a model. In some frames one object occludes the other. Figure 4¹ highlights the features matching the models.

Acquisition of training and test in different days

Table 2 shows the results of comparing training models with test models acquired in a different room, with a different illumination. Training and test sequences are acquired in different days and daytime. In these cases using the two-stages matching strategy noticeably improves results. M1 refers to the matching based only on the first stage while M2

¹The features drawn here and in Fig. 6 are the projection on a single frame of a temporal feature; they are not drawn at their actual size to keep the figure clear. Circles: *dewey*'s features; squares: *book*, crosses: *winnie*; X's: *goofy*.

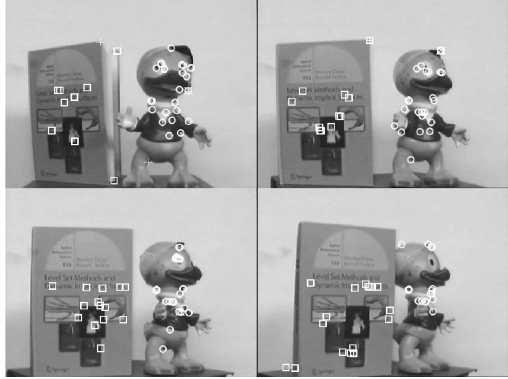


Figure 4. Four frames from a test sequence with 2 known objects. Only similarity scores above 0.6 are shown.

	Book		Goofy		Dewey		Winnie	
	M1	M2	M1	M2	M1	M2	M1	M2
Book	58	67	7	4	6	8	1	0
Goofy	14	7	66	75	34	28	13	4
Dewey	9	5	19	12	52	59	6	2
Winnie	19	21	8	9	8	5	80	94

Table 2. Hit percentages between training and tests video acquired in different places and days, with and without the analysis of coeval features (M2 and M1 respectively). Models are on the rows.

columns refer to the two-stages matching. Test sequences are on the columns, models on the rows.

Acquisition with a hand-held camcorder

The last set of experiments of this model assessment is devoted to checking the method’s robustness when the camera moves discontinuously, for instance it is hand-held by the user. It is worth mentioning that, since the ego-motion is shaky, features are more difficult to track, and, as a result, trajectories tend to be shorter and more unstable. Table 3 shows the results. Columns M1 show the results obtained by simple appearance matching (the first stage of our matching strategy), while columns M2 show how the results have been increased with the two-stages strategy. The advantage of this approach is apparent, for this complex case.

3.2 Recognition in complex scenes

We use our system to acquire and store objects models and to acquire test sequences. To assess our recognition method we performed an extensive analysis on various

	Book		Goofy		Dewey		Winnie	
	M1	M2	M1	M2	M1	M2	M1	M2
Book	60	93	17	4	9	2	15	12
Goofy	21	4	67	91	13	2	13	11
Dewey	10	3	16	5	69	86	11	9
Winnie	0	0	0	0	9	9	61	68

Table 3. Hit percentages between training and test videos acquired with a hand-held camcorder, with and without the analysis of coeval features (M2 and M1 respectively). Models are on the rows.

test shots of variable length, where the objects that we previously modeled were put in cluttered environments with many other similar objects (examples frames are shown in figure 6). The videos have been acquired at a variable speed, slowing the camera down in the presence of possible objects of interest (not necessarily the ones belonging to our models set). When the speed is higher, the video quality decreases and very few time-invariant features are detected. The objective of these experiments is to test the videos against the presence of our objects of interest. Matches are kept when the similarity value is higher than 0.5; groups of coeval features bigger than 5 are kept. Figure 5 shows the results obtained on a *tracking shot* video: the plot above shows the number of matches obtained per frame, the intermediate horizontal colored bars are the ground truth, the plot below shows the total number of trajectories detected along the sequence. When the camera speed grows the number of detected features decreases and so does the number of matches. All the objects are detected correctly in the high quality sections of this video with the exception of the book, because it appeared only briefly in a low quality section of the video. Figure 6 shows matching examples on sample frames from our video shots.

4 Discussion

This paper proposed a 3D object recognition method that exploits temporal information to obtain a compact description of the scene content. The time-invariant local features that we proposed are based on tracking keypoints over the sequence, keeping only stable trajectories. Thanks to our approach the overall appearance of 3D objects can be represented with a relatively small set of features. Also, we proposed a simple two stages matching strategy that favors groups of features co-occurring in the same range of view. Our description strategy may be applied also to relatively simple poor-textured objects, since few stable features are enough for recognition. Currently we are in an advanced

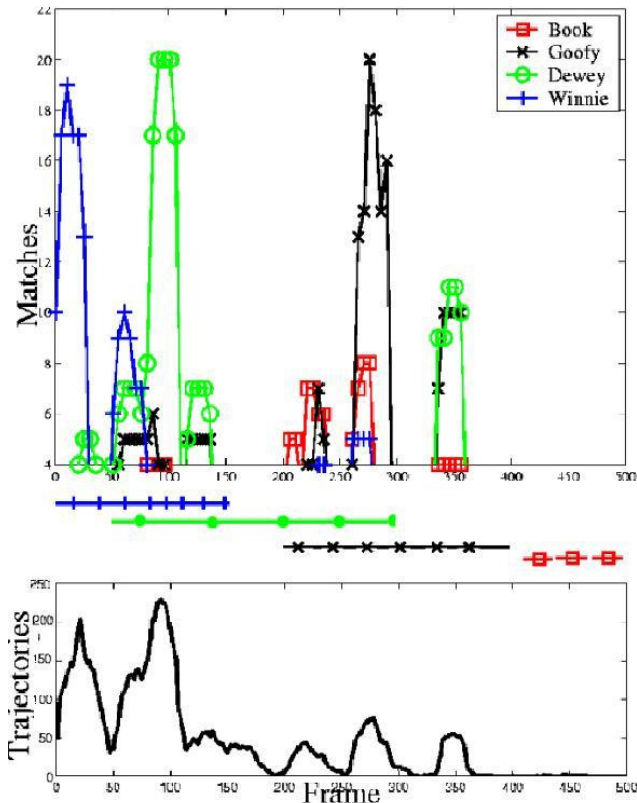


Figure 5. Number of matches visualized per frame on one tracking shot. Below, a plot of the number of time-invariant features detected per frame (see text).

stage of development of the real-time version of the system.

To our knowledge no benchmarks are available for view-based 3D object modeling (with the exception of the COIL dataset, where the image size is 32×32). Thus, the data we are acquiring to build the models will be soon available at <http://slipguru.disi.unige.it/>.

References

- [1] E. Arnaud, E. Mémin, and B. Cernuschi-Frías. Conditional filters for image sequence based tracking - application to point tracking. *IEEE Tr. on Im. Proc.*, 1(14), 2005.
- [2] G. Csurka, C. Dance, L. Fan, and C. Bray. Visual categorization with bag of keypoints. In *The 8th European Conference on Computer Vision - ECCV*, Prague, 2004.
- [3] E. Delponte, E. Arnaud, F. Odone, and A. Verri. Trains of keypoints for 3d object recognition. In *IEEE International Conference on Pattern Recognition*, Hong Kong, 2006.
- [4] V. Ferrari, T. Tuytelaars, and L. V. Gool. Simultaneous object recognition and segmentation from single or multiple model views. *International Journal of Computer Vision*, 67(2), 2006.

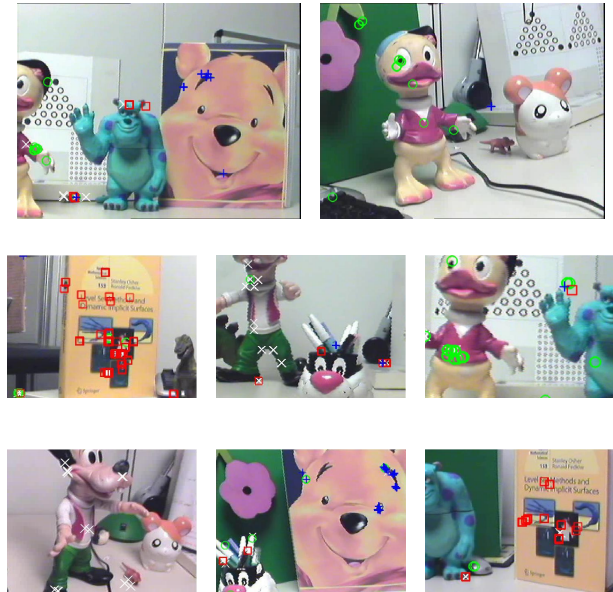


Figure 6. Objects identified in sample frames (from top left): (1) winnie, (2) dewey (3) book (4) goofy (5) dewey (6) goofy (7) winnie (8) book

- [5] M. Grabner and H. Bischof. Object recognition based on local feature trajectories. In *I Cogn. Vision Work.*, 2005.
- [6] I. Laptev and T. Lindeberg. Space-time interest points. In *ICCV*, 2003.
- [7] B. Leibe, K. Mikolajczyk, and B. Schiele. Efficient clustering and matching for object class recognition. In *British Machine Vision Conference*, 2006.
- [8] T. Lindeberg. Feature detection with automatic scale selection. Technical report, CVAP, Department of numerical analysis and computing science, 1998.
- [9] D. Lowe. Object recognition from local scale-invariant features. In *International Conference on Computer Vision*, pages 1150–1157, Corfú, Greece, 1999.
- [10] H. Murase and S. Nayar. Visual learning and recognition of 3d objects from appearance. *IJCV*, 14(1), 1995.
- [11] M. Pontil and A. Verri. Support Vector Machines for 3D object recognition. *IEEE PAMI*, 20:637–646, 1998.
- [12] F. Rothganger, S. Lazebnik, C. Schmid, and J. Ponce. 3D object modeling and recognition using local affine-invariant image descriptors and multi-view spatial constraints. *International Journal of Computer Vision*, 66(3), 2006.
- [13] A. Torralba, K. Murphy, and W. Freeman. Sharing visual features for multiclass and multiview object detection. *Trans. on PAMI*. in press.
- [14] A. Ude, C. Gaskett, and G. Cheng. Support vector machines and gabor kernels for object recognition on a humanoid with active foveated vision. In *Proc. IEEE/RSJ Int. Conf. Intelligent Robots and Systems.*, pages 668–673, 2004.
- [15] E. Wan and R. van der Merwe. The unscented kalman filter for nonlinear estimation. In *IEEE Symp. on Adaptive Systems for Signal Processing, Communication and Control*, 2000.