Trains of keypoints for 3d object recognition

Elise Arnaud, Elisabetta Delponte, Francesca Odone, Alessandro Verri

Trains of keypoints for 3D object recognition

Elise Arnaud Elisabetta Delponte Francesca Odone Alessandro Verri
DISI Università di Genova, Italy
{arnaud, delponte, odone, verri}@disi.unige.it

Abstract

This paper presents a 3D object recognition method that exploits the spatio-temporal coherence of image sequences to capture the object most relevant features. We start from an image sequence that describes the object's visual appearance from different view points. We extract local features (SIFT) and track them over the sequence. The tracked interest points form trains of features *that are used to build a vocabulary for the object. Training images are represented with respect to that vocabulary and an SVM classifier is trained to recognize the object. We present very promising results on a dataset of 11 objects. Tests are performed under varying illumination, scale, and scene clutter.*

1. Introduction

An ideal 3D object recognition system is able to recognize many different objects and spot their presence in various environments, no matter the viewing position or distance. The research on this field has been very active in the last decades, but the ideal system has still to come. View-based approach to object recognition has been widely used in the last few years [11, 12, 13, 9], mostly because it offers a simple but principled way to model objects variations with respect to the viewing angle, and also because the intuition behind it is supported by psychophysical evidence [2]. Other causes of difficulties, such as scene illumination, viewing distance changes but also clutter and occlusions have been successfully addressed with the local feature approach [7, 8, 3]. In this paper we present a technique that allows us to recognize 3D objects in realistic environments, under different viewing and scene conditions. We combine a view-based approach with the use of local features, obtaining object descriptions that model how the features evolve over time. In our approach each object is represented by an image sequence acquired in a controlled environment. Local interest points are extracted and tracked over the sequence with a filtering algorithm. All trajectories (trains of features) that are stable over the sequence are used to describe the appearance of the object under varying viewing conditions. The idea of building trains of keypoints is loosely inspired to the *bag of words* used for example-based text categorization [6]. Recently this idea has been reformulated for the case of image categorization, leading to the *bag of keypoints* [3]. The bag of keypoints approach uses a simple clustering technique to group features that carry some visual similarity. Our method exploits both visual similarity and the image sequence temporal coherence, as we group the features that are connected by a temporal trajectory. Similarly to bags of keypoints, each train of features constitute a word in a vocabulary. We build a vocabulary for each object of interest. We model 3D objects with a learning from examples scheme, representing training images with respect to the object vocabulary and training a binary SVM for each object. The multiclass nature of the problem is captured with a one-vs-all approach. We report very promising results on a dataset of 11 objects (Figure 1).

The paper is organized as follows. Section 2 describes how we compute the object vocabulary, Section 3 illustrates how we represent images with respect to the vocabulary. Section 4 reports the results of our experiments, while Section 5 is left to the final discussion.

2. The features vocabulary

For a given object, the first stage of our method consists in building its features vocabulary. We consider an image sequence acquired observing the object from different viewpoints. For each image we locate interesting points on a difference of Gaussians pyramid, and then represent each point with a SIFT descriptor [7], that is, a vector containing local orientation histograms around the keypoint position.

2.1. SIFT tracker

The selected features are tracked over time with an Unscented Kalman filter [14]. This method belongs to the filtering algorithms family that are well-known for their simplicity and robustness to difficult situations. Such an algorithm allows us to cope with temporal detection failures,

Figure 1. The 11 objects of our dataset. (From top left: tele, teddy, dino, box, book1, duck, pino, bambi, tommy, book2, biscuits).

and as a consequence avoids redundancies in the vocabulary. The filtering methods consist in modeling the dynamic system to be tracked by a hidden Markov process. The goal is to estimate the values of the state $\mathbf{x}_k$ from observations $\mathbf{z}_k$ obtained at each instant. The system is described by a dynamic equation modeling the evolution of the state and a measurement model that links the observation to the state. The unknown state is $\mathbf{x}_k = \{\mathbf{p}_k, \mathbf{s}_k, \mathbf{d}_k\}$, where $\mathbf{p}_k$ is the location of the SIFT, $\mathbf{s}_k$ its scale and $\mathbf{d}_k$ its main direction. The final system consists in:

• A state equation of $\mathbf{s}_k$ and $\mathbf{d}_k$ defined as a simple constant model:

$$\begin{pmatrix} \mathbf{s}_k \\ \mathbf{d}_k \end{pmatrix} = \begin{pmatrix} \mathbf{s}_{k-1} \\ \mathbf{d}_{k-1} \end{pmatrix} + \gamma_k, \quad (1)$$

where γ_k is a zero-mean Gaussian white noise.

• A state equation of $\mathbf{p}_k$ specified online by an instantaneous motion vector of the tracked point $\mathbf{p}_k = \mathbf{p}_{k-1} + \mathbf{u}_k(\mathbf{p}_{k-1}) + \psi_k$, where ψ_k is the zero-mean Gaussian white noise. The variable $\mathbf{u}_k(\mathbf{p}_{k-1})$ denotes the motion vector associated to a pixel $\mathbf{p}_{k-1}$. This vector is estimated with a robust parametric motion estimation technique on a small region around $\mathbf{p}_k$ (introducing a non linearity in the system)[1].

• An observation $\mathbf{z}_k$ defined as the nearest detected SIFT from the prediction on a given window. The observation is then linearly linked to the state: $\mathbf{z}_k = \mathbf{x}_k + \mathbf{v}_k$ where $\mathbf{v}_k$ is the zero-mean Gaussian white noise.

Since the dynamic equation is non linear, Kalman filter is not appropriate. Recently, Particle Filters [4] have been extensively used to deal with the non linearity of a system, but in our case since the system is weakly non linear, the use of the Unscented Kalman filter is both sufficient and efficient.

2.2. Virtual features

All features linked by a tracking trajectory form a train of elements belonging to an equivalence class; we compute the average value of all the elements, that we call a *virtual feature* $\mathcal{V}_i$, and use it as a delegate for the train. The average values are good representatives of the original features as the tracking procedure is robust and leads to a group of features with a small variance. The set of virtual features form a *vocabulary of features for the object*: $\mathcal{V} = \{\mathcal{V}_1, \ldots, \mathcal{V}_N\}$.

3. Data representation

An image F_i, after we extract local interest points, can be seen as a collection of SIFT features $F_i = \{f^i{}_1, \ldots, f^i_m\}$. Once the vocabulary $\mathcal{V}$ is available, F_i can be represented with respect to the vocabulary, with a vector $\mathcal{F}_i$ of N elements, that contains at each entry j the degree of similarity between $\mathcal{V}_j$ and the feature f^i_k most similar to $\mathcal{V}_j$. While finding the association between $\mathcal{V}_j$ and f^i_k, features that appear similar to more than one virtual feature are penalized.

Since SIFT description is obtained as a concatenation of direction histograms, the similarity between features and virtual features can be estimated as follows:

$$\cap_{norm}(H, H') = \frac{\bigcap(H, H')}{\bigcup(H, H')} = \frac{\sum_{i=1}^{n}(\min(H_i, H'_i))}{\sum_{i=1}^{n}(\max(H_i, H'_i))}.$$

where H_i is the description associated with $\mathcal{V}_j$ and H'_i is associated with f^i_k. Therefore we obtain a similarity measure which takes values in $[0, 1]$ and if the histograms are normalized, this similarity measure is equivalent to histogram intersection.

4. Object recognition experiments

4.1. The dataset

We acquired a dataset of 11 different objects [1] (see Figure 1). The selected objects include examples of similar

[1]The dataset is available on demand.

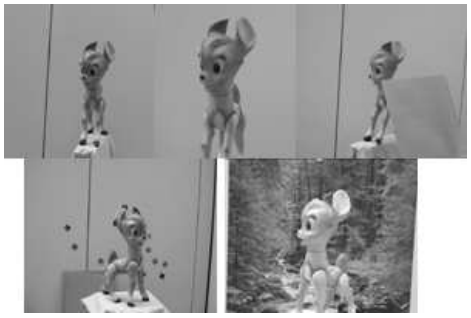

Figure 2. From top left: examples of test sets (2-6), see text.

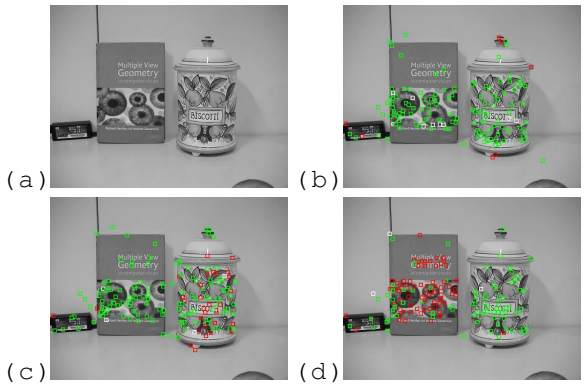

Figure 3. A test image (`(a)`), and the SIFT points matched with vocabularies of different objects : `(b)` duck, `(c)` biscuits, `(b)` book1 (high similarity score are red).

objects (5 plastic toys, 2 books), but at the same time are variable enough to represent a good selection of things that can be found in a real indoor environment. Each object is represented by an image sequence of about 200 frames acquired by placing the object on a turntable and observing it from different viewpoints. We use these sequences both for building the vocabulary and as positive examples for training the recognition system. The training set is acquired in a neutral environment. For each object we acquired six different test sets: (1) acquired in similar conditions to the training from a similar viewpoints, (2) under different illumination conditions, (3) at a different scale, (4) allowing for severe occlusions(5) the object against a plain with different background, (6) the object against a complex and highly textured background (see Figure 2 for samples of test sets (2-6) of an example object). We also acquired clutter images (including various indoor and oudoor scenes) or images of other objects to be used as negative examples for training. The negatives set has also been enriched with images downloaded from the Web. For each object we use about 200 images as positive training examples, 300 images as negative training examples. Each object has about 18 000 images of test examples.

4.2. The classifiers

For each object we build the vocabulary and then represent the training data (both positive and negative) with respect to it. We then train a binary SVM classifier with a histogram intersection kernel [10], that was proved effective on a number of applications and does not depend on any parameter. Preliminary results that we do not report here showed that its performances are superior to standard kernels for the problem at hand. We deal with the multiclass nature of the problem with a *one against all* approach. The results obtained over test sets (1-4) are summarized in Table 1. The column **simple** refers to the results obtained both under similar conditions and under some illumination variation (tests (1) and (2)), the column **scale** refers to the results obtained on test set (3), that contains the objects at different scale. Finally, the column **occlusions** report the results obtained on test set (4), containing the objects with various degrees of occlusions. As the recognition rates are very high, we do not report details on the confusion among different objects (in the case of object *book2*, the misses are spread among objects *tele*, *box*, and *dino*). The description proposed captures the peculiarity of objects, and allows us to recognize them correctly even if the possible classes contain many similar objects. In the case of more complex backgrounds, instead, it is worth showing the confusion matrices (Tables 2 and 3). They show how, if the amount of clutter is small, the recognition rates are still very satisfactory. In the case of very complex and textured backgrounds the performance drops because of the high number of features detected (of which only a small number belong to the

objects	simple	scale	occlusions
bambi	99.50	92.34	100.00
box	100.00	100.00	100.00
duck	100.00	98.90	100.00
biscuit	100.00	100.00	100.00
book1	100.00	100.00	100.00
book2	100.00	95.53	77.78
dino	100.00	100.00	100.00
teddy	100.00	98.86	100.00
pino	100.00	99.59	92.96
tele	100.00	100.00	100.00
tommy	100.00	100.00	100.00

Table 1. Hit percentages of the 11 classifiers against test sets (1-4) (see text).

	bambi	box	duck	biscuit	book1	book2	dino	teddy	pino	tele	tommy
bambi	71.46	0.00	13.69	0.00	0.00	0.00	0.00	3.48	2.55	0.23	8.58
box	0.34	98.65	1.01	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
duck	0.00	0.00	100.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
biscuit	0.00	0.00	0.22	99.78	0.00	0.00	0.00	0.00	0.00	0.00	0.00
book1	0.00	0.00	0.00	0.00	100.00	0.00	0.00	0.00	0.00	0.00	0.00
book2	5.22	0.00	0.37	0.00	0.00	91.04	0.00	1.49	0.00	1.49	0.37
dino	9.48	0.00	13.73	0.00	0.00	0.00	58.17	0.33	0.00	0.00	18.30
teddy	0.00	0.00	3.13	0.00	0.00	0.00	0.00	96.87	0.00	0.00	0.00
pino	15.66	0.00	15.93	0.00	0.00	0.00	7.42	1.92	41.48	0.00	17.58
tele	0.93	0.93	6.48	0.00	0.00	0.00	0.00	0.93	0.00	90.28	0.46
tommy	4.86	0.00	2.86	0.00	0.00	0.00	4.29	0.57	2.29	0.00	85.14

Table 2. Confusion matrix for test set (5), with moderate quantities of clutter on the background.

	bambi	box	duck	biscuit	book1	book2	dino	teddy	pino	tele	tommy
bambi	2.11	0.00	5.15	19.67	2.11	11.01	0.00	11.94	0.00	8.90	39.11
box	0.00	85.81	0.65	0.65	0.00	8.06	0.65	0.00	0.00	0.65	3.55
duck	0.53	0.00	40.74	9.52	1.06	0.53	0.53	4.23	0.53	4.23	38.10
biscuit	0.00	0.00	0.00	100.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
book1	0.00	0.00	0.00	0.00	100.00	0.00	0.00	0.00	0.00	0.00	0.00
book2	0.37	0.00	0.74	0.00	0.37	96.68	0.00	0.37	0.00	0.74	0.74
dino	1.08	0.00	0.65	16.85	33.69	3.46	2.38	11.45	1.08	2.81	26.57
teddy	1.24	0.00	3.96	0.25	1.73	4.70	0.50	36.14	7.43	14.60	29.46
pino	0.00	0.63	8.15	25.08	13.48	7.52	0.00	4.70	0.63	10.34	29.47
tele	0.00	0.47	0.47	0.00	0.94	12.21	0.00	0.00	0.00	81.22	4.69
tommy	2.07	0.00	0.00	1.38	7.24	6.55	0.00	33.79	1.72	6.55	40.69

Table 3. Confusion matrix for test set (6), with a very complex background.

object). This suggest that for very complex environments a local search with a window sliding over the image to mark the test area should be performed. We started investigating the case of test images with multiple objects. Figure 3 shows a test image with two objects and the features that match 3 different object vocabularies. The high scores (in red) are positioned on the correct object.

5. Discussion

We presented a learning from examples procedure to represent and recognize 3D objects, based on local representations. Each object is modeled using an image sequence. SIFT features are extracted and tracked over the sequence, and the obtained trains of features are used to build a vocabulary of virtual features. The actual classification is performed with an SVM classifier. The results reported show how the method behaves nicely in the case of illumination and scale changes, allow for occlusions, and the presence of moderate quantities of clutter in the background. To deal with more complex backgrounds or multiple objects, a region-based search procedure is currently being implemented. Recently a view-based approach to recognition that exploits a feature tracking has been proposed [5]. The possible connections of our approach with this work are under investigation.

References

[1] E. Arnaud, E. Mémin, and B. Cernuschi-Frías. Conditional filters for image sequence based tracking - application to point tracking. IEEE *Trans. on Image Proc.*, 1(14), 2005.

[2] H. H. Bulthoff, S. Y. Edelman, and M. J. Tarr. How are three-dimensional objects represented in the brain? *Cerebral Cortex*, 5(3), 1995.

[3] G. Csurka, C. Dance, L. Fan, J. Willamowsky, and C. Bray. Visual categorization with bags of keypoints. In *Int. Work. on Statistical Learning in Comp. Vis., ECCV*, 2004.

[4] A. Doucet, S. Godsill, and C. Andrieu. On sequential Monte Carlo sampling methods for Bayesian filtering. *Statistics and Computing*, 10(3):197–208, 2000.

[5] M. Grabner and H. Bischof. Object recognition based on local feature trajectories. In *I cognitive vision works.*, 2005.

[6] T. Joachims. Text categorization with support vector machines: learning with many relevant features. In *Proc. of ECML*, pages 137–147, 1998.

[7] D. Lowe. Distinctive image features from scale invariant keypoints. *IJCV*, 60(2):91–110, 2004.

[8] K. Mikolajczyk and C. Schmid. Indexing based on scale invariant interest points. In *ICCV*, pages 525–531, 2001.

[9] H. Murase and S. K. Nayar. Visual learning and recognition of 3d object from appearance. *IJCV*, 14:5–24, 1995.

[10] F. Odone, A. Barla, and A. Verri. Building kernels from binary strings for image matching. *IEEE Trans. on Image Processing*, 14(2):169–180, 2005.

[11] M. Pontil and A. Verri. Support vector machines for 3d object recognition. *IEEE Trans. on PAMI*, 6:637–646, 1998.

[12] P. Sinha and T. Poggio. The role of learning in three-dimensional form perception. *Nature*, 384, 1996.

[13] M. Turk and A. Pentland. Eigenfaces for recognition. *Journ. of Cognitive Neuroscience*, 3:71–86, 1991.

[14] 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.