Analysis on a local approach to 3d object recognition

Elisabetta Delponte, Elise Arnaud, Francesca Odone, Alessandro Verri

Analysis on a local approach to 3D object recognition

Elisabetta Delponte, Elise Arnaud, Francesca Odone, and Alessandro Verri

DISI - Università degli Studi di Genova - Italy

Abstract. We present a method for 3D object modeling and recognition which is robust to scale and illumination changes, and to viewpoint variations. The object model is derived from the local features extracted and tracked on an image sequence of the object. The recognition phase is based on an SVM classifier. We analyse in depth all the crucial steps of the method, and report very promising results on a dataset of 11 objects, that show how the method is also tolerant to occlusions and moderate scene clutter.

1 Introduction

This paper proposes a method based on local keypoints that allows us to recognize 3D objects in real environments, under different viewing and scene conditions. The method falls into the view-based approaches that have been widely used in the past for object recognition, as they offer a simple but principled way to model viewpoint variation.

Our method can be summarized as follows: for each object we acquire an image sequence that describes the 2D appearance of the object from different viewpoints. We extract local keypoints from the sequence, describe them with SIFT descriptors, and track them over the sequence with an Unscented Kalman filter. For each SIFT trajectory we compute a compact representation, or *virtual feature*, that becomes the delegate for all the keypoints of the trajectory and hopefully for the same feature belonging to yet to be seen images of the same object. The collection of virtual features form a model of the object, or *vocabulary*. The actual recognition is based on learning from examples: we represent all the images of a training set with respect to the vocabulary estimating the degree of similarity between each image and the vocabulary, and train a binary SVM classifier. The multiclass nature of the problem is captured with a one-vs-all approach. We carry out our analysis on a dataset of 11 objects (see Figure 1).

The paper is organized as follows. Section 2 reviews related work on object recognition with local approaches. Section 3 describes how the object model is built, while Section 4 discusses how to represent an image with respect to the model. Section 5 describes the training stage, and Section 6 reports the recognition experiments. Section 7 is left to a final discussion.

Fig. 1. The 11 objects of the dataset. From top left: *bambi, box, duck, biscuit, book1, book2, dino, teddy, pino, tele, tommy.*

2 Related work

In the last few years there has been an increasing interest on object recognition systems based on local keypoints. Among the many possible approaches we focus on those combining local descriptions with learning from examples.

Statistical learning methods have been often coupled to local keypoints through the design of *ad hoc* kernels. Since local representations are variable-length and usually they do not carry internal ordering, local kernels are derived from the studies on kernels for sets. One of the first works proposed is the *matching kernel* [10], that has been proved very effective for object recognition. Unfortunately, it has been demonstrated that it is not a Mercer kernel [2]. More recently a modification of the matching kernel, called *intermediate matching kernel* has been proposed [2]. The new kernel is based on the concept of *virtual features*, which is reminiscent of the ones we will define.

An alternative approach to combining local descriptors with learning from examples is the so called *bags of keypoints* approach. The idea is inspired to the bag of words used for text categorization, and it was first proposed in [3] for visual categorization. The method can be summarized as follows: (1) extract interesting points for all images of the training set mixing keypoints from all the classes; (2) cluster all the keypoints and identify the keypoints "bags", or equivalence classes; the collection of bags form a vocabulary of keypoints; (3) represent all images with respect to the bags with a histogram-like approach. With this approach keypoints belonging to different objects may fall in the same equivalence class, as long as they are more similar to it than to other classes.

Our method is close to bags of keypoints, but it is also somewhat related to the intermediate matching kernel. Similarly to [3] we look for keypoints equivalence classes, but since our input datum (an image sequence) is more informative than a single image, our classes will only contain different instances of the same feature. To do so, we exploit the image sequence temporal coherence. A similar idea can be found in [4], where a local spatio-temporal description is computed using SIFT and the KLT tracker; since their tracker has no prediction ability a more complex trajectory selection is used.

3 The object model

For each object of interest the first stage of our method consists of finding a model based on local keypoints taken from an image sequence of the object. First, we extract local keypoints from all the images of the training sequence and exploit temporal coherence by tracking them, obtaining a list of trajectories, or *trains of keypoints*. Second, we represent the trains in a compact way that we call a *virtual feature* and build a *vocabulary* of such compact representations. This vocabulary is the model of the object, since it is the information about the object that we will use for recognition.

3.1 The local keypoints

For each image we locate interesting points on a difference of Gaussians pyramid. These points are centers of blob-like structures. We represent them with SIFT descriptors [6]. This descriptor contains the following information about the keypoint: (a) position $\mathbf{p}$, (b) scale $\mathbf{s}$, (c) main orientation $\mathbf{d}$, and (d) a vector $\mathbf{H}$ containing local orientation histograms around the keypoint position. We will use the first three elements for SIFT tracking and the orientation histograms $\mathbf{H}$ for computing the similarities. Scale and main orientation are also implicitly used for computing the similarities, as $\mathbf{H}$ is built on an image patch centered at the keypoint position, and scaled and rotated according to scale and main orientation.

3.2 The SIFT tracker

The selected keypoints are tracked over time with an Unscented Kalman filter [5, 11]. This method belongs to the filtering algorithms family that are well-known for their simplicity and robustness. Such an algorithm allows us to cope with temporal detection failures, and as a consequence avoids redundancies in the vocabulary.

Filtering methods consist of a dynamic system tracked by a hidden Markov process. The goal is to estimate the values of the state $\mathbf{x}_k$ from a set of observations $\mathbf{z}_{1:n} = \{\mathbf{z}_1, \mathbf{z}_2, ..., \mathbf{z}_n\}$. The system is described by a dynamic equation $p(\mathbf{x}_k|\mathbf{x}_{k-1})$ modeling the evolution of the state and a measurement model $p(\mathbf{z}_k|\mathbf{x}_k)$ that links the observation to the state. The goal is then to estimate the *filtering distribution* $p(\mathbf{x}_k|\mathbf{z}_{1:k})$, that carries the whole information on the process to be estimated. The choice of the estimation algorithm (Kalman filter, Extended Kalman filter, Particle filter, etc.) depends on the characteristics of the model (if it is linear, Gaussian, etc.).

The system we consider here – whose unknown state is $\mathbf{x}_k = \{\mathbf{p}_k, \mathbf{s}_k, \mathbf{d}_k\}$, where $\mathbf{p}_k$ is the SIFT position, $\mathbf{s}_k$ its scale and $\mathbf{d}_k$ its main orientation – is composed of the following dynamic and measurements models.

The **dynamic equation** describes the evolution in time of the keypoint. A constant model is associated to $\mathbf{s}_k$ and $\mathbf{d}_k$:

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

where γ_k is a zero-mean Gaussian white noise of covariance matrix Γ_k (set *a priori*). As for $\mathbf{p}_k$, its dynamic model has to describe the motion of the keypoint along the image sequence. Since no *a priori* information is available, and in order to be reactive to any change of speed and direction, we define the state equation as [1]:

$$\mathbf{p}_k = \mathbf{p}_{k-1} + \mathbf{u}_k(\mathbf{p}_{k-1}) + \psi_k, \tag{2}$$

where ψ_k is assumed to be zero-mean Gaussian white noise of covariance Ψ_k (set *a priori*). The variable $\mathbf{u}_k(\mathbf{s})$ denotes the motion vector associated to a pixel $\mathbf{s}$. It is estimated with a robust parametric technique [7] that computes a 2D parametric model representing the dominant image motion within a considered support $\mathcal{R}$. For the computation of $\mathbf{u}_k(\mathbf{p}_{k-1})$ between images $\mathbf{I}_{k-1}$ and $\mathbf{I}_k$, $\mathcal{R}$ is chosen as a small region around $\mathbf{p}_{k-1}$, introducing a non linearity in the system.

Given a search window, the **measurement** $\mathbf{z}_k$ is the keypoint that is nearest to the prediction. The measurement and the state are defined in the same space, then the following linear observation model can be set:

$$\mathbf{z}_k = \mathbf{x}_k + \mathbf{v}_k, \tag{3}$$

where $\mathbf{v}_k$ is a zero-mean Gaussian white noise of covariance R_k (set *a priori*). If no keypoint is detected in the search window, R_k is set to $\infty \times Id$ (Id is the identity matrix) so that the current estimation only relies on the dynamic equation.

As the dynamic equation is non linear because of Eq. (2), the Kalman filter is not appropriate. Recently Particle filters have been extensively used to deal with the non linearity of a system. These methods are very interesting because they enable an accurate approximation of the distribution of interest even if it is highly multimodal. However, their interest can decrease if the system under consideration is weakly non linear as the one we propose here. In this case, the use of an algorithm that assumes a Gaussian approximation of the filtering density can be both sufficient and efficient. We choose the Unscented Kalman filter that describes the Gaussian approximation of the posterior density by carefully selected weighted sample points. These points capture the mean and covariance of the approximation accurately to the 3rd order and are propagated through the non linear system. To implement our SIFT tracker we apply the Unscented Kalman Filter to Eq. (1, 2, 3).

3.3 The vocabulary

All keypoints linked by a tracking trajectory, or train, belong to the same equivalence class. A *virtual feature* $\mathcal{V}_i$ is the average of all local orientation histograms $\mathbf{H}_k$, with k running through the train. We use the virtual feature as a delegate for the train. Average values are good representatives of the original keypoints as the tracking procedure is robust and leads to a class of keypoints with a small variance. Being an average, some histogram peculiarities are smoothed or suppressed, but empirical evidence shows that the information that it carries is enough to describe the object.

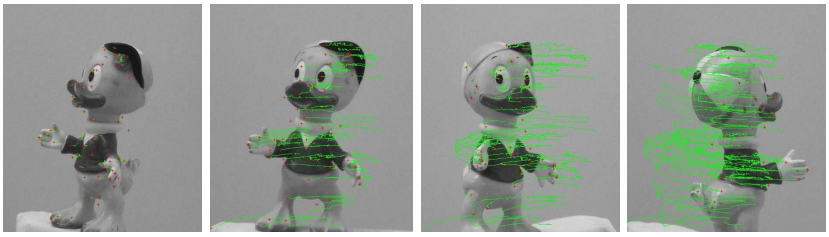

Fig. 2. SIFT tracking on the image sequence of object *duck*: SIFT point trajectories are displayed at different steps of the image sequence.

The set of virtual features form a vocabulary of keypoints for the object: $\mathcal{V} = \{\mathcal{V}_1, \ldots, \mathcal{V}_N\}$. *The vocabulary $\mathcal{V}$ is the model for the object.*

4 Image representation

Once the model for the object has been computed, each image is represented with respect to the model. This representation carries information on how much related the image is related to the object. It is based on first extracting local keypoints from the image, then comparing this list of keypoints with the object vocabulary.

4.1 The choice of the similarity measure

A crucial point is to decide how to compare the local orientation histogram of a keypoint with the average orientation histogram of a virtual feature. Then, a comparison criterion for histograms seems to be appropriate. We consider (1) Euclidean distance D, (2) Chi-square distance χ^2, (3) Kullback-Leibler divergence $\mathcal{K}$, (4) Histogram intersection $\cap$ [9].

Since 1-3 are distance measures we will use the exponent version: $D_{exp} = \exp(-D), \chi^2_{exp} = \exp(-\chi^2), \mathcal{K}_{exp} = \exp(-\mathcal{K})$. Also, since the keypoint descriptions may not be normalized, instead than measure 4 we will use

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

If the histograms are normalized, this similarity measure is equivalent to histogram intersection.

Let us reason on what we would ask to a similarity measure: high scores on similar keypoints, low scores on different keypoints. Figure 4 shows the results of comparing two similar images (Figure 3, left and center) and two very different images (Figure 3, center and right), with the four similarity measures. The plots are obtained as follows: for each keypoint of the first image we compute the highest match value with respect to keypoints of the other image. The results

Fig. 3. Example images used for the comparative analysis of similarity measures.

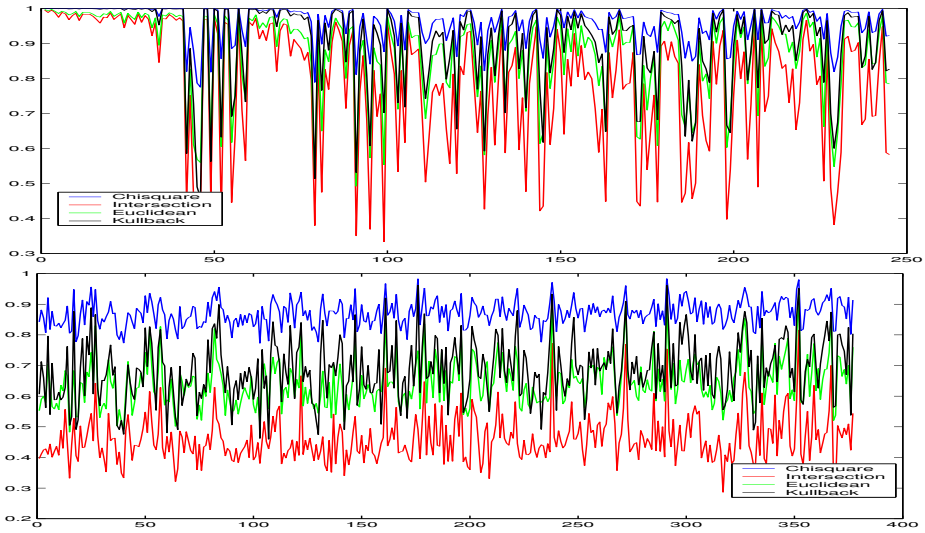

Fig. 4. Match values obtained comparing 2 images with the 4 similarity measures. Top: results from two similar images (Fig. 3 left and center). Bottom: results from two different images (Fig. 3 center and right). On the x axis are the indices of keypoints in the first image, on the y axis the corresponding match values with the most similar keypoints of the second image.

show that Chi-square returns uniformly high scores in both cases. The best compromise between intraclass and interclass keypoints is obtained with normalized histogram intersection $\bigcap_{norm}$, which will be used in the rest of the experiments.

4.2 Building the representation

An image F_i, after we extract local interest points, can be seen as a collection of keypoints $F_i = \{\mathcal{F}^i{}_1, \ldots, \mathcal{F}^i_M\}$, where M will vary. The vocabulary helps us to avoid the problem of variable length representations: each image F_i is represented with a vector R_i of length N.

Each entry k of R_i carries the contribution of the keypoint $\mathcal{F}^l_j$ most similar to $\mathcal{V}_k$, if there is one. Possible choices on how to build R_i include:

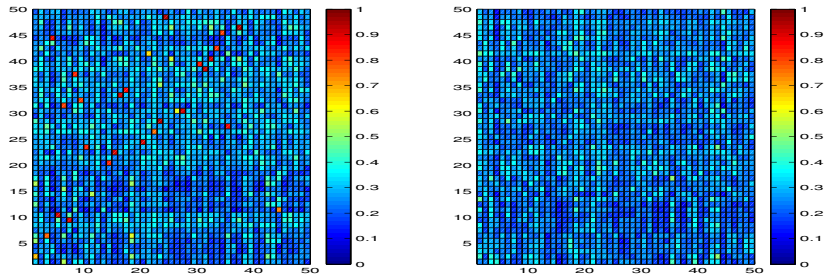

Fig. 5. Similarity matrices obtained comparing the vocabulary of object $\mathcal{O}$ with one image of object $\mathcal{O}$ (on the left) and one image of another object (on the right). On the rows: the image keypoints, on the columns: the virtual features (see text).

1. **Binary entries**, with $R_i^k = 1$ if there exist a keypoint $\mathcal{F}_j^l$ closer to $\mathcal{V}_k$ than a threshold.
2. **Real value entries** describing the degree of similarity between $\mathcal{V}_k$ and the most similar keypoint $\mathcal{F}_j^l$.
3. **SIFT entries**, with $R_i^k = \mathcal{F}_j^l$, where $\mathcal{F}_j^l$ is the most similar keypoint to $\mathcal{V}_k$.

Our image representations will be based on choice 2, as it is the best compromise between effectiveness and simplicity. It is worth mentioning that choice 3 corresponds to an explicit mapping of the intermediate matching kernel [2].

We compute the similarity values between all keypoints of image F_i and all virtual features of the vocabulary $\mathcal{V}_k$. An explicit computation would lead to a similarity matrix as the ones shown in Figure 5. The final description is obtained by taking the maximum values column-wise. While finding the association between $\mathcal{V}_k$ and $\mathcal{F}_l^j$, keypoint that appear similar to more than one virtual feature are penalized. Figure 5 considers a vocabulary for object $\mathcal{O}$ and includes the comparison with one image of object $\mathcal{O}$ (on the left) and one image of another object (on the right). On the left matrix are clearly visible the high match values corresponding to the most similar keypoint.

5 Object representation

We acquired a dataset of 11 different objects (Figure 1) that include examples of similar objects (5 plastic toys, 2 books), but at the same time are variable enough to represent a possible selection of things of a real indoor environment.

Each object is represented by an image sequence of about 200 frames acquired by placing the object on a turntable. 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 but real environment. No segmentation or background subtraction is applied.

For each object we acquired six different test sets: (1) similar conditions to the training, (2) moderated illumination changes, (3) different scale, (4) allowing

Fig. 6. The different conditions under which the test data have been acquired (see text).

for severe occlusions of the object (5) placing the object against a plain, but different background, (6) placing the object against a complex and highly textured background (see Figure 6). We also acquired background images, and images of other objects to be used as negative examples. For each object we use about 200 positive training examples, 300 negative training examples. Each object has about 18 000 images of test examples. For each object we build the vocabulary and then represent the training data with respect to it. We then train a binary SVM classifier with a histogram intersection kernel [8], as it was proved effective on a number of applications and does not depend on any parameter. We deal with the multiclass nature of the problem with a *one against all* approach.

6 Experiments on object recognition

The recognition rates obtained over test sets (1-4) are summarized in Table 1. The column **simple** refers to the results obtained on test sets (1) and (2), the column **scale** refers to test set (3), while the column **occlusions** refers to test set (4). The very good results confirm how SIFT keypoints combined with a robust feature tracking produce a model which is robust to illumination and scale changes, and to occlusions. The description proposed captures the peculiarity of objects, and allows us to recognize them correctly even if the possible classes contain many similar objects. The drop obtained for object "book2" is due to the fact that in many test images the object was entirely hidden.

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).

	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.

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 keypoints detected (of which only a small number belong to the object).

7 Discussion

We proposed a method for 3D object recognition which is robust to scale, illumination changes and viewpoint variation. The results we presented show that the method is also tolerant to occlusions and moderate clutter. We are currently dealing with the presence of multiple objects. Preliminary results indicate that the local approach coupled with analysis on image sub-regions allows us to focus on one object at a time and achieve good recognition results. Figure 7 shows a test image with two objects and the keypoints that match 3 different object vocabularies. The high scores (in red) are positioned on the correct object. We also considered embedding additional information in the keypoint description, such as color information, but the modest increase in the performance does not justify the choice.

Acknowledgements This research is partially supported by the FIRB project RBIN04PARL *Learning Theory and Applications*. The authors thank Francesco Isgrò for proofreading.

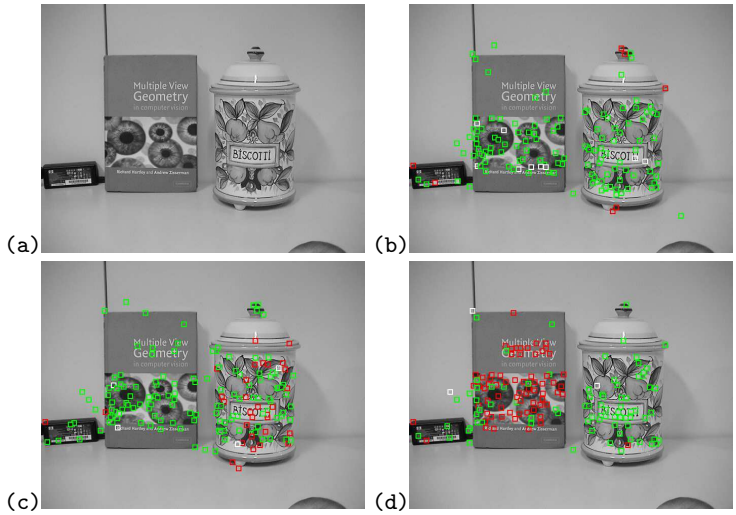

Fig. 7. A test image (a), and the SIFT points matched with vocabularies of different objects : (b) duck, (c) biscuits, (b) book1 (high similarity score are red).

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. S. Boughorbel, J. P. Tarel, and N. Boujemaa. The intermediate matching kernel for image local features. In *International Joint Conference on Neural Networks*, pages 889–894, Montreal, Canada, 2005.
3. G. Csurka, C. Dance, L. Fan, J. Willamowsky, and C. Bray. Visual categorization with bags of keypoints. In *Int. Work. on Stat. Learn. in CV, ECCV*, 2004.
4. M. Grabner and H. Bischof. Object recognition based on local feature trajectories. In *I Cognitive Vision Work.*, 2005.
5. S. Julier and J. Uhlmann. A new extension of the kalman filter to non linear systems. In *Int. Symp. Aerospace/Defense Sens., Sim. and Cont.*, 1997.
6. D. Lowe. Distinctive image features from scale invariant keypoints. *IJCV*, 60(2):91–110, 2004.
7. J.-M. Odobez and P. Bouthemy. Robust multiresolution estimation of parametric motion models. *Journal of Vis. Comm. and Image Rep.*, 6(4), 1995.
8. 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.
9. M. J. Swain and D. H. Ballard. Color indexing. *IJCV*, 7(1):11–32, 1991.
10. C. Wallraven, B. Caputo, and A. Graf. Recognition with local features: the kernel recipe. In *ICCV*, page 257ff, 2003.
11. E.A. Wan and R. van der Merwe. The Unscented Kalman filter for nonlinear estimation. In IEEE *Symp. on Adapt. Sys. for Sig. Proc., Communication and Control*, 2000.