Semi–supervised learning with constraints for multi–view object recognition

Stefano Melacci, Marco Maggini, and Marco Gori

Department of Information Engineering,
University of Siena,
Via Roma 56, 53100 Siena, Italy
{mela,maggini,marco}@dii.unisi.it

Abstract. In this paper we present a novel approach to multi–view object recognition based on kernel methods with constraints. Differently from many previous approaches, we describe a system that is able to exploit a set of views of an input object to recognize it. Views are acquired by cameras located around the object and each view is modeled by a specific classifier. The relationships among different views are formulated as constraints that are exploited by a sort of collaborative learning process. The proposed approach applies the constraints on unlabeled data in a semi–supervised framework. The results collected on the COIL benchmark show that constraint based learning can improve the quality of the recognition system and of each single classifier, both on the original and noisy data, and it can increase the invariance with respect to object orientation.

Key words: semi–supervised learning, constraints, multi–view object recognition, kernel methods.

1 Introduction

Object recognition from static images is a wide and challenging research topic in the fields of computer vision and pattern recognition. In the last few years several systems and techniques have been proposed for this task [1–12]. Some of them are *single–view*, in the sense that they process a single viewpoint of an object. Objects are captured in different conditions of illumination, with occlusions or in presence of noise [1]. In those contexts the focus is posed in finding a compact, discriminative and robust representation of the objects in the feature space [1, 2].

When multiple viewpoints are introduced, object recognition usually performs more accurately [3–12]. In this scenario, referred to as *multi–view* object recognition, a single object is represented by a set of views captured at different angles. Some existing approaches use local feature representations to exploit the correspondences among the available views [4]. The generation of 3D models from local image features for viewpoint invariant object recognition has been studied in [5]. Other authors jointly modeled object appearance and viewpoint

or extended single–view techniques, such as the Implicit Shape Model (ISM) [13], to the multi–view scenario [6]. However, many of these approaches assume that a single image is available at test time [8–12].

In this paper we investigate the problem of object recognition from multiple views. In this case, a set of views of an object is fed as input to the system at test time. In a real scenario this model corresponds to the situation in which a set of cameras acquire images of a given object from different viewpoints. The recognition system must be able to exploit the availability of multiple views to enhance its discriminative power.

In our approach, we adopt kernel machines [14] to model each view and then we reinforce the classifiers by combining the single decisions in a constraint based framework, requiring coherence in the decision among different views. In particular, unlabeled data is exploited in a semi–supervised fashion to force the fulfillment of coherence constraints. In a wider context, our method could be applied also with other kind of classifiers and in every situation when there is a relationship among corresponding decisions on different representations of the same object.

This paper is organized as follows. In Section 2 the multi–view object recognition scenario is formalized. Section 3 describes constraint based learning in the semi–supervised framework. Experimental results are collected in Section 4 and concluding remarks are presented in Section 5.

2 Multi–view object recognition

In multi–view object recognition, each object is represented by a set of images acquired from different viewpoints. Given a collection of known objects, the goal is to correctly classify the input element into one of the known object categories. The information contained in multiple views is more informative than the one in a single image and it can increase the accuracy of the classifier but it can also contain redundant data due to, for example, the overlapping regions among different images.

In details, given a set D of objects, we consider k cameras c_i, $i = 1, \ldots, k$ that simultaneously acquire k pictures of the same object $\boldsymbol{x} \in D$ from k different points of view. Each camera produces a bidimensional representation of $\boldsymbol{x}$, indicated with $\boldsymbol{x}_i$. Such process can be modeled by an unknown function $g_i : D \rightarrow \mathbb{R}^d$, where d is the number of pixels of each acquired image, and $g_i(\boldsymbol{x}) = \boldsymbol{x}_i$. The functions g_i describe a complex relationship that maps the object $\boldsymbol{x}$ in the three dimensional object space to a planar image belonging to $\mathbb{R}^d$. A collection of k views is referred to as *viewset* and it is indicated with $X = \{\boldsymbol{x}_1, \ldots, \boldsymbol{x}_k\}$. Viewsets belong to the cartesian product of k sets in $\mathbb{R}^d$, $V = \mathbb{R}^d \times \mathbb{R}^d \times \cdots \times \mathbb{R}^d$. In particular, we can define a distribution $\mathcal{P}$ on V of the viewsets representing objects from D. The distribution $\mathcal{P}$ expresses the correlation between different views of the same object, and regions with zero probability correspond to unknown objects.

Given a collection of q viewsets representing the objects in D, acquired in different conditions of illumination or with slight orientation/position changes, we define the set of labeled instances as $L = \{(X_{j,h}, t_j) \mid X_{j,h} \in V;\ j = 1,\ldots,n;\ h = 1,\ldots,v_j\}$, where t_j is the actual label of the j–th object described by the viewset $X_{j,h}$, and v_j is the number of viewsets available for that object (note that $q = \sum_{j=1}^{n} v_j$).

We model the system using n binary multi–view classifiers, in a one–against–all strategy [15]. Moreover, we indicate with the function $o_j : V \rightarrow [0,1]$ the output of each classifier.

First, as baseline approach, we use a single discriminating function $f_j : I\!R^d \rightarrow [0,1]$ as base of the j–th classifier, that makes no distinctions among the views of an object, since it does not include any information on viewpoints. The output of such classifier for a generic input X is then

$$o_j(X) = \frac{1}{k}\sum_{i=1}^{k} f_j(\boldsymbol{x}_i), \tag{1}$$

where the k outputs are averaged to obtain a single combined output given the k input images.

Secondly, we separately model the data $\boldsymbol{x}_i$ acquired by the camera c_i with a specific function $f_{j,i} : I\!R^d \rightarrow [0,1]$. The output function becomes

$$o_j(X) = \frac{1}{k}\sum_{i=1}^{k} f_{j,i}(\boldsymbol{x}_i). \tag{2}$$

In both cases, the output of each binary classifier is compared with a reject threshold $\tau_j \in (0,1]$. If all $o_j(X)$, $j = 1,\ldots,n$, are less than their corresponding thresholds, the object is classified as not belonging to the set D. Otherwise, the predicted class label $c(X)$ corresponds to the index of the binary classifier with the highest confidence, as formalized in

$$c(X) = \begin{cases} \arg\max_j \frac{o_j(X)-\tau_j}{1-\tau_j} & if\ \exists j\ (o_j(X) \geq \tau_j) \\ \text{unknown} & otherwise. \end{cases} \tag{3}$$

We exploit kernel machines [14] to model the functions f_j and $f_{j,i}$. Focusing on the second approach, given a positive definite Kernel function $K_j : I\!R^d \times I\!R^d \rightarrow I\!R$, we indicate with $\mathcal{H}$ the Reproducing Kernel Hilbert Space (RKHS) corresponding to it, and with $\|\cdot\|_{\mathcal{H}}$ the norm of $\mathcal{H}$. From Tikhonov regularization in a RKHS, when the loss function $\mathcal{L}$ is the classic squared loss, the problem becomes an instance of ridge regression [15]. In details, for each of the k functions of the j–th classifier we have $\mathcal{L}_{j,i} = \sum_{r=1}^{q}(y_r - f_{j,i}(\boldsymbol{x}_i^r))^2$, where $\boldsymbol{x}_i^r$ indicates the r–th instance of the i–th view and $y_r \in \{0,1\}$ is the corresponding label. The k functions $f_{j,i} \in \mathcal{H}$ are chosen such that

$$\min_{f_{j,i}\in\mathcal{H}} \sum_{i=1}^{k}\sum_{r=1}^{q}(y_r - f_{j,i}(\boldsymbol{x}_i^r))^2 + \lambda_j \sum_{i=1}^{k}\|f_{j,i}\|_{\mathcal{H}}^2, \tag{4}$$

where λ_j is the weight of the regularization term.

From the Representer Theorem [14] the form of functions $f_{j,i}$, solution to the Tikhonov minimization problem, is given by

$$f_{j,i}(\boldsymbol{x}_i^{\cdot}) = \sum_{r=1}^{q} w_{j,i}^r K_j(\boldsymbol{x}_i^{\cdot}, \boldsymbol{x}_i^r), \tag{5}$$

where $w_{j,i}^r$ are the function weights and $\boldsymbol{x}_i^{\cdot}$ is a generic input. Using this representation when minimizing Eq. 4 with respect to the function $f_{j,i}$, is equivalent to solving a linear system of equations in the weights $w_{j,i}^r$, $r = 1, \ldots, q$ [15]. In matrix notation, $\boldsymbol{w}_{j,i} \in I\!R^q$ is the weight vector that collects the q weights $w_{j,i}^r$, $G_{j,i} \in I\!R^{p,p}$ is the Gram matrix associated to the selected kernel function, $\boldsymbol{y}_j \in \{0,1\}^q$ is the vector that collects the q labels y_r and $I \in I\!R^{p,p}$ is the identity matrix. Finally,

$$\boldsymbol{w}_{j,i} = (\lambda_j I + G_{j,i})^{-1} \boldsymbol{y}_j. \tag{6}$$

The solution for the baseline approach (Eq. 1) is straightforward, since it is a just simplified case of the described one. Note that the number of parameters for the j–th classifier in both the approaches is exactly the same. In particular each of the k functions $f_{j,i}$ is composed by q weights for a total of $k \cdot q$, that is equivalent to the number of weights of f_j since its representation includes all the $k \cdot q$ training views.

3 Semi–supervised learning with constraints

Each input viewset X belongs to the space V, and in particular to regions of V where the distribution $\mathcal{P}$ is non–zero. The classification approach described by Eq. 2 models different views with independent functions, that share only the selected kernel function and regularization weight. The set L of labeled training instances implicitly includes the information on the data distribution, since views of the same object are marked with the same label. If the classifier accurately approximates training data, it is assured to model the distribution $\mathcal{P}$ but only in regions of V that correspond to such data.

When unlabeled data is available, the correlation among the k views expressed by $\mathcal{P}$ can be exploited as prior knowledge to improve the discriminative power of the classifier. In particular, it introduces a dependency among the functions $f_{j,i}$ that can be modeled by constraining the learning process. Each function can benefit by taking into account the shape of the others in different, but corresponding, regions of the space.

Ideally the functions should produce exactly the same output for the k views of a given viewset X, since they belong to the same object. More formally, we require the fulfillment of the following constraints

$$\begin{cases} f_{j,1}(\boldsymbol{x}_1) = f_{j,2}(\boldsymbol{x}_2) \\ f_{j,2}(\boldsymbol{x}_2) = f_{j,3}(\boldsymbol{x}_3) \\ \quad \cdots \\ f_{j,k-1}(\boldsymbol{x}_{k-1}) = f_{j,k}(\boldsymbol{x}_k). \end{cases} \tag{7}$$

Given a collection of m unlabeled viewsets $U = \{X_u \in V \mid u = 1, \ldots, m\}$, a penalty term is added to the cost function of Eq. 4 to bias the learning process by the described constraints, leading to the following new cost

$$\sum_{i=1}^{k} \sum_{\boldsymbol{x}_i^r \in L} (y_r - f_{j,i}(\boldsymbol{x}_i^r))^2 + \lambda_j \sum_{i=1}^{k} \|f_{j,i}\|_{\mathcal{H}}^2 + \mu \sum_{i=1}^{k-1} \sum_{\boldsymbol{x}_i^u \in U} (f_{j,i}(\boldsymbol{x}_i^u) - f_{j,i+1}(\boldsymbol{x}_{i+1}^u))^2. \tag{8}$$

The parameter μ is the weight associated to the penalty term and it determines how strictly the system is forced to fulfill the given constraints. The accurate selection of the value of μ is crucial for the system performances. In fact, high values of μ could result in a worse fitting of the labeled data, and the overall accuracy could degenerate, moving the system towards a trivial solution where all the functions assume values close to zero.

We solved the minimization problem of Eq. 8 by gradient descent. Since labeled data already fulfill the constraints, training the unconstrained classifiers by solving the linear system of Eq. 6 will lead to a solution that is probably close to the constrained one. Exploiting this consideration, the solution of Eq. 6 is a promising starting point for the gradient descent, in order to reduce the number of iteration required to achieve convergence.

4 Experimental results

The COIL-100 database [16] is one the most used benchmarks for object recognition algorithms. It consists of a collection of multiple views of 100 objects. Each object was placed on a turntable and every 5° an image was acquired, generating a total of 72 views for object. The database is composed by the collection of 7200 color images at the resolution of 128x128 pixels (Fig. 1).

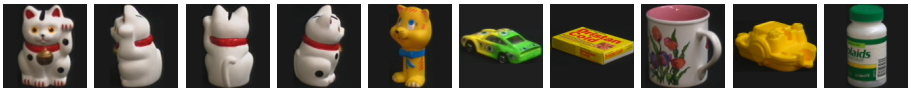


Fig. 1. Sample images from the COIL-100 database.

In the last decade, a large number of experiments have been performed on this collection [7–12]. As in many previous approaches [8–10] we rescaled each image to 32x32 gray scale pixels in the interval $[0, 1]$, since it has been shown that the information coming from color is highly discriminative among objects and it makes the learning task quite trivial [9, 11].

In a multi–view scenario we consider four cameras c_i, $i = 1, \ldots, 4$, equally spaced around the object, that simultaneously acquire four images at $90° \cdot (i - 1)$ considering the reference angles provided in the COIL-100 database. Each viewset $X = \{\boldsymbol{x}_1, \boldsymbol{x}_2, \boldsymbol{x}_3, \boldsymbol{x}_4\}$ is identified by the degree of rotation of the image acquired by the first camera, c_1, that falls in the range $[-45°, 45°]$.

Differently from the experiments available in the literature, we decided to make the recognition task more challenging by considering only a relatively small amount of views of a sub selection of objects to train the recognizer. We defined a set K of *known objects*, composed by the first 50 ones, and a set U of the remaining 50 *unknown objects*. For each element in K we selected only 3 viewsets (12 images) to train the system, each separated from the previous one by $30°$, starting at $-30°$. Similarly other 3 viewsets where selected to cross–validate the system parameters, alternatively starting at $-15°$ or $-45°$ for each object[1]. The other viewsets were used to test the recognition accuracy in two different scenarios, *test K* and *test KU*. In the former, only the remaining 12 viewsets (48 images) of the known objects K are considered, whereas in the latter, also the 18 ones (72 images) that are available for each unknown object in U are added. In other words we do not only require the ability to recognize and discriminate known objects but also to correctly reject the unknown ones. Table 1 summaries the details of the described experimental framework.

Table 1. The selected experimental setup. The left portion of the table details the list of objects and total number of images in each set, whereas the right one collects information on viewsets for "each" object of the list ($j = 0, \ldots,$Viewsets-1).

Set	Objects	Images
Training	$1, \ldots, 50$	600
Validation	$2, \ldots, 50$ (even only)	300
	$1, \ldots, 49$ (odd only)	300
Test K	$1, \ldots, 50$	2400
Test KU	$1, \ldots, 50$	2400
	$51, \ldots, 100$	3600

Set	Viewsets	Positions
Training	3	$-30° + (30 \cdot j)°$
Validation	3	$-15° + (30 \cdot j)°$
	3	$-45° + (30 \cdot j)°$
Test K	12	The remaining ones
Test KU	12	The remaining ones
	18	All

We trained 50 binary classifiers in a one–against–all strategy and we selected as kernel a Gaussian function of the form $K_j(x, y) = \exp \frac{-\|x-y\|}{2 \cdot \sigma_j^2}$. For every classifier the optimal values of σ_j and of λ_j are determined by varying them in the sets $\{1e{-}3, 1e{-}2, 1e{-}1, 1, 2, 3, \ldots, 12\}$ and $\{1e{-}5, 1e{-}4, \ldots, 1\}$ respectively, in order to maximize the sum of accuracies on training and validation data. The optimal rejection threshold τ_j^* is determined with the same criterion.

We approached the problem using three different methods, in order to show how the new constraints can improve the performances. First, the baseline approach of Eq. 1, where we discarded the information about the four cameras and their positions, modeling each classifier with a single function. In the second approach the output of every classifier is composed by the contribution of 4 functions, one for each image of the viewset, as described in Eq. 2. Finally, we constrained the 4 functions to be coherent in a semi–supervised framework, by minimizing the cost function of Eq. 8.

[1] The views located at $45° \cdot (i-1)$, with $i = 1, \ldots, 4$, were alternatively considered as acquired by camera c_i or by the following one.

We smoothly increased the value of the penalty weight μ, ranging in $[1e-2, 25]$. Constraints were forced on validation data, then the thresholds τ_j^* and, in particular, the optimal value of μ were determined. We selected the value of μ that yields the best performances on both training and validation data first, and, secondly, the value that causes a better accuracy in approximating the given constraints. In Table 2 the resulting macro accuracies of the three described approaches are reported. They are referred as *single* (classifiers with a single function), *multi* (classifiers with four functions), and *constrained* (classifiers with four functions and constraints) respectively. In Fig. 2(a) the accuracy of the complete constraint based learner with respect to the value of μ is shown, and the selected optimal value μ^* is indicated with a vertical line. Similarly, in Fig. 2(b) the average penalty value on the 50 classifiers is reported. The violation of the constraints on the validation data decreases as the value of μ grows but the opposite behavior can be observed on training data, since the contribution of the approximation error becomes less important that the constraint penalty. The optimal value μ^* can be selected in correspondence of a roughly equivalent violation of constraints on the two data sets, as a trade–off between an appropriate labeled data fitting and a good fulfillment of the given constraints.

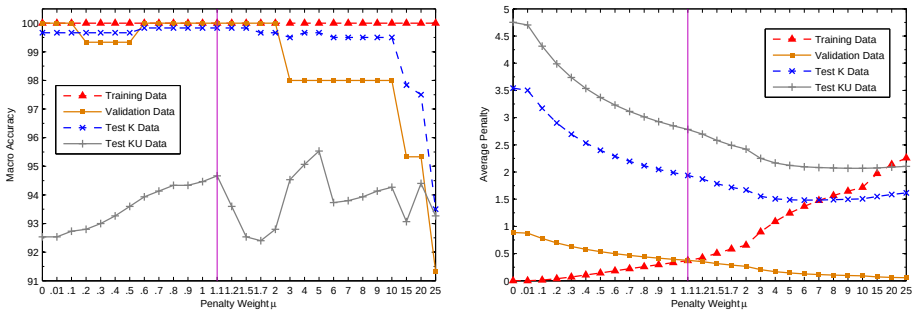
Table 2. Recognition (macro) accuracies of the three proposed approaches (in percentage). The better results on test data are reported in bold.

Technique	Training Data	Validation Data	Test K Data	Test KU Data
Single	100	100	99.67	90.07
Multi	100	100	99.67	92.53
Constrained	100	100	**99.83**	**94.67**

The recognition accuracy of the multiple function approach is equivalent to the single one for known objects, but when unknown objects are introduced the multiple function technique is more robust. This is mainly due to the specific training of each function on a specific view that allows them to achieve a more tight fitting around the positive training instances. The introduction of constraints offers another significant increment of accuracy on such data and a slight increment on the discrimination capability of the system. It can be clearly seen that increasing the weight of the constraints increases the accuracy on the test data. Moreover, beyond a certain value, the contribution of the squared loss on labeled data becomes less significant in the cost function, and performances decrease or become really unstable.

We tested the performances of the constraint based learner also in other different tasks: robustness with respect to object orientation, to noise and to missing cumulative information.

Assuming that an input object is given to the system but its actual orientation is unknown, we checked if the model is still able to correctly recognize it. As a consequence, if the object is rotated by 90° four times and four viewsets are acquired, one of such sets must be oriented consistently with the training data.



(a) (b)

Fig. 2. Recognition (macro) accuracy (a) and average penalty value (b) on training, validation and test data in function of the penalty weight μ. The vertical line represent the selected value of μ accordingly to the described validation criterion.

If the object is highly asymmetric and differs among the four views, then the system should have more confidence only on the viewset aligned with respect to the training data. Following this idea we generated the required four viewsets for each data set in Table 1 and we fed them to the system, selecting, for each classifier, the prediction with the highest confidence on the four "rotated" inputs. The recognition accuracies are reported in Table 3.

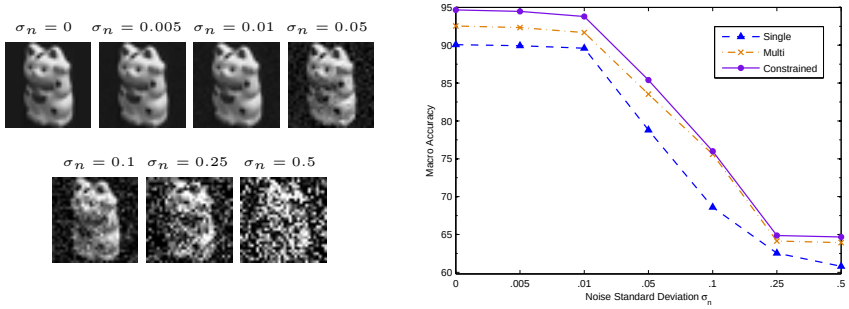
Table 3. Recognition (macro) accuracies of the three proposed approaches (in percentage) discarding information on the right viewset orientation. The better results on test data are reported in bold.

Technique	Training Data	Validation Data	Test K Data	Test KU Data
Single	100	100	99.67	90.07
Multi	100	99.33	99.67	91.87
Constrained	100	99.33	**99.83**	**93.53**

The results for the single function case are obviously the same of Table 2, since we are not differently modeling the four views. The other techniques achieve the same results on test objects with or without the information on viewset position but when unknown objects are introduced, performances are slightly reduced. This indicates that a small portion of unknown objects, under some viewset orientations are wrongly recognized as known ones. The constraint based learner keeps showing better accuracy than the other approaches on test data and, in particular, it is still the most accurate recognizer when unknown objects are introduced.

Another test scenario involves the introduction of noise into the acquired images. In a real scenario this could be due to low quality or damaged cameras or to a noisy transmission channel from cameras to the recognizing software.

We artificially introduced pseudo–random noisy values drawn from a normal distribution, with zero mean and incremental values of the standard deviation σ_n, to each pixel of the images (Fig. 3(a)).



(a) (b)

Fig. 3. (a) An object from COIL-100 with increasing noise ratios – (b) Recognition (macro) accuracy on test data KU in presence of noise.

The recognition accuracies are reported in Fig. 3(b). As expected, while the noise standard deviation increases, the performances of the three techniques degrades gracefully. The constraint based classifier keep showing more robustness to noisy images.

Finally, we investigate how the recognition performances of the functions that model each view are changed after applying the constraints to the four function classifier. We "turned off" three of the four cameras and we tried to recognize the object by a single image. In Table 4 the resulting accuracies are reported.

Table 4. Recognition (macro) accuracies based on only one of the four functions that compose the multi function system, with $(+C)$ and without constraints. The better results on test data between each pair of functions are reported in bold.

Data	$f_{j,1}$	$f_{j,1}$ + **C**	$f_{j,2}$	$f_{j,2}$ + **C**	$f_{j,3}$	$f_{j,3}$ + **C**	$f_{j,4}$	$f_{j,4}$ + **C**
Training	100	100	100	100	100	100	100	100
Validation	85.33	91.33	74.67	75.33	95.33	95.33	62.67	71.33
Test K	94.5	**97.83**	85.5	**92.5**	98.83	**99**	83.5	**89.17**
Test KU	85.87	**87.07**	**87.07**	86.87	86.87	**90.2**	87.07	**90.2**

Interestingly, the role of the constraints appears determinant for the increments of accuracy of the single functions. The improvement of the functions that model each view from the constrained classifier with respect to the ones from the unconstrained system is evident. These results show that the interaction among functions due to the constraints can enhance the cumulative decision of the classifier but also the single power of each $f_{j,i}$. Moreover, the lower performances of the pair of functions $f_{j,2}$ and $f_{j,4}$ with respect to $f_{j,1}$ and $f_{j,3}$

indicates how the frontal and backward views, associated to the former pair, are more discriminative that the side views for the object set of COIL-100.

5 Conclusions and future work

In this paper a multi–view approach to object recognition has been presented. The proposed kernel based method has been proved to increase the accuracy of the classifier by exploiting a set of constraints formulated from prior knowledge on the viewpoints. Moreover, unlabeled data has been used to require their fulfillment in a semi–supervised framework. The experiments on the COIL database have shown robustness to noise, to orientation changes and to missing input views. Finally, the proposed approach is general, and it can be applied when a coherent decision on different representations of the same input is required.

References

1. Lowe, D.G.: Object recognition from local scale-invariant features. In: Proc. of the Int. Conf. on Computer Vision. Volume 2. (1999) 1150
2. Belongie, S., Malik, J., Puzicha, J.: Shape matching and object recognition using shape contexts. IEEE Trans. PAMI **24**(4) (2002) 509–522
3. Mokhtarian, F., Abbasi, S.: Automatic selection of optimal views in multi-view object recognition. In: Proc. of the British Machine Vision Conf. (2000) 272–281
4. Torralba, A., Murphy, K.P.: Sharing visual features for multiclass and multiview object detection. IEEE Trans. PAMI **29**(5) (2007) 854–869
5. Rothganger, F., Lazebnik, S., Schmid, C., Ponce, J.: 3d object modeling and recognition using local affine-invariant image descriptors and multi-view spatial constraints. Int. J. Comput. Vision **66**(3) (2006) 231–259
6. Thomas, A., Ferrari, V., Leibe, B., Tuytelaars, T., Schiele, B., Van Gool, L.: Towards multi-view object class detection. In: Proc. of CVPR. (2006) 1589–1596
7. Christoudias, C., Urtasun, R., Darrell, T.: Unsupervised feature selection via distributed coding for multi-view object recognition. In: Proc. of CVPR. (2008) 1–8
8. Pontil, M., Verri, A.: Support vector machines for 3D object recognition. IEEE Transactions on Pattern Analysis and Machine Intelligence **20**(6) (1998) 637–646
9. Roobaert, D., Van Hulle, M.: View-based 3D object recognition with support vector machines. In: Neural Networks for Signal Processing. (1999) 77–84
10. Wallraven, C., Caputo, B., Graf, A.: Recognition with local features: the kernel recipe. In: Proc. of Int. Conf. on Computer Vision. Volume 1. (2003) 257–264
11. Caputo, B., Dorko, G.: How to Combine Color and Shape Information for 3D Object Recognition: Kernels do the Trick. Advances in NIPS (2003) 1399–1406
12. Lyu, S.: Mercer Kernels for Object Recognition with Local Features. In: Proc. of Int. Conf. on CVPR. Volume 2. (2005) 223–229
13. Leibe, B., Schiele, B.: Scale-invariant object categorization using a scale-adaptive mean-shift search. DAGM (2004) 145–153
14. Shawe-Taylor, J., Cristianini, N.: Kernel Methods for Pattern Analysis. Cambridge University Press, New York, NY, USA (2004)
15. Rifkin, R., Klautau, A.: In defense of one-vs-all classification. Journal of Machine Learning Research **5** (2004) 101–141
16. Nene, S., Nayar, S., Murase, H.: Columbia Object Image Library (COIL-100). Techn. Rep. No. CUCS-006-96, Dept. Comp. Science, Columbia University (1996)