On Feature Combination and Multiple Kernel Learning for Object Tracking

Huchuan Lu[1], Wenling Zhang[1], and Yen-Wei Chen[1,2]

[1] School of Information and Communication Engineering, Dalian University of Technology, Dalian, China
[2] College of Information Science and Engineering, Ritsumeikan University, Kusatsu, Japan

Abstract. This paper presents a new method for object tracking based on multiple kernel learning (MKL). MKL is used to learn an optimal combination of χ^2 kernels and Gaussian kernels, each type of which captures a different feature. Our features include the color information and spatial pyramid histogram (SPH) based on global spatial correspondence of the geometric distribution of visual words. We propose a simple effective way for on-line updating MKL classifier, where useful tracking objects are automatically selected as support vectors. The algorithm handle target appearance variation, and makes better usage of history information, which leads to better discrimination of target and the surrounding background. The experiments on real world sequences demonstrate that our method can track objects accurately and robustly especially under partial occlusion and large appearance change.

1 Introduction

Visual tracking is an important task in many computer vision applications like visual surveillance, human computer interaction, and traffic monitoring or sports analysis. Tracking is significantly challenging because of the intrinsic appearance variability, and extrinsic illumination change. In order to ameliorate the adaptability of the algorithm to scene change, many different methods have been proposed for it. To adapt to the changes of object and background during tracking process, it is necessary to introduce on-line learning mechanism into trackers. Avidan's ensemble tracker [1] replaces some old weak classiers with new ones by AdaBoost at each frame. Grabner *etal* [2] adopt online AdaBoost firstly for feature selection and introduce selectors which paved the way for application on visual object tracking. In the paper [3, 4], different online SVM algorithms are proposed and applied to object tracking. However, due to using results which the tracker gives to update the tracker itself, slight inaccuracies in the tracker can therefore lead to incorrectly labeled training examples, which degrades the classifier and may cause further drift. To resolve the problem, Avidan [1]adopts a simple outliers rejection scheme, and Babenko *etal.* [5]introduce the concept of Multiple Instance Learning into visual tracking. Also these methods are inclined to fail in case of partial or complete occlusion, for they rely on the global template model focusing on the integrated appearance information of the object.

R. Kimmel, R. Klette, and A. Sugimoto (Eds.): ACCV 2010, Part III, LNCS 6494, pp. 511–522, 2011.

Yin *etal.* [6] consider visual object tracking as a numerical optimization problem by making a numerical hybrid local and global mode-seeking tracker that combines detection and tracking. “Frag-Track” [7] extracts the integral histogram of multiple image fragments or patches to represent the template, calculates each patch voted by comparing its histogram with the corresponding image patch histogram and then combines voted maps of the multiple patches. Experiments demonstrate that the approach can deal with partial and swift occlusion well. In addition, some works are also tried to resolve the occlusion problem in the multiple objects tracking.

Recently multiple kernel learning (MKL) methods [8, 9]have shown great advantages in various classification(e.g. Learning the discriminative power-invariance trade-off; Support kernel machines for object recognition). Instead of using a single kernel in support vector machine (SVM) [10], MKL learns an optimal kernel combination and the associated classifier simultaneously, and provides an effective way of fusing informative features and kernels. However, these methods basically adopt a uniform similarity measure over the whole input space. When a category exhibits high variation as well as correlation with other categories in appearance, they are difficult to cope with the complexity of data distribution. Varma *etal.* [11] proposed combining multiple descriptors using Multiple Kernel Learning(MKL) and showed impressive results on varied object classification tasks. So in this paper, we propose Multiple Kernel Learning for Tracking (MKLT) approach uses an easily obtained training data as input, and then tunes itself to the classification for tracking at hand. It simultaneously updates the training examples to tailor them towards the objects in the scene. It also updates the weights that determine the optimal combination of different kernels, while allowing different combinations to be chosen for different objects. Finally, it tunes the classifier to the updated training data. Our final system is obtained by combining the outputs of this online classifier with the high probability outputs of the original classifier trained on the first frames.

We firstly describe the MKL formulation of Bach*etal.* [12]. More efficiency in multiple kernel learning known as SimpleMKL [13], which we use to obtain a classifier for initial training frames. SimpleMKL carries out this optimization in an SVM framework to learn the SVM model parameters as well as kernel combination weights simultaneously. Our tracking procedure with MKL is an exacting online solution that allows us to update the Lagrangian multipliers of the training data, as well as the kernel combination weight, five or ten frames at a time. The main contribution is that we adopt spatial pyramid [14] to obtain the features' spatial information which is inspired by the remarkable ability of “bag of words” to handle intra-class pose variant and occlusion. The spatial pyramid works by computing rough geometric correspondence on a global scale using an efficient approximation technique adapted from the pyramid matching scheme of Grauman and Darrel [15]. However, the satisfactory results are obtained by our approach when occlusions occur or the object' length and scale change severely, as well as the ability to recapturing the object.

2 Related Work

Early works on an object feature extracting used global features such as color or texture histogram [16]. However, these features were not robust to view-point changes, clutter and occlusion. Over the years, more sophisticated approaches such as part-based [17]and bag of features [18]methods have become more popular. Increased interest in object recognition has resulted in new feature descriptors and a multitude of classifier. Inspired by the pyramidal feature matching approach of [15], Bosch *et al.* proposed two new region descriptors - the Pyramid Histogram of Oriented Gradients (PHOG) and Pyramidal Histogram of Visual Words (PHOW) [19]zhang *etal.* used the Geometic Blur(GB) feature [20]and proposed using a discriminative nearest neighbor classification for object recognition. Wu *etal.* [21]used edge features to capture the local shape of objects.

Kernel based method is one of attractive research areas for object categorization in recent years. Diverse kernels such as pyramid matching kernel(PMK) [15], spatial pyramid matching kernel(SPK) [14], distribution kernel (PDK) [22] and chi-square kernel are delicately designed to compute the similarity of image pair on certain features that represent particular visual characteristics. Multi-kernel based classifiers have been introduced into object categorization yielding promising results. And multiple features (e.g. appearance, shape) are employed and kernels (e.g. PMK and SPK with different hyper-parameters) are linearly combined in MKL framework.

Lanckriet *etal* [23]introduced the MKL procedure to learn a set of linear combination weights, while using multiple features of information with a kernel method, such as an SVM. It can result in a convex but non-smooth minimization problem. The algorithm worked for hundreds of examples or hundreds of kernels [12] provided the additional advantage of encouraging sparse kernel combinations. Our initial object classifier built the object's features from the first few frames. Our work builds on MKL and fits well into the SVM framework. And also we provide the online updating process to retrain the classifier that accounts for appearance changes and allows reacquisition of an object after total occlusion.

3 Algorithms

3.1 The Multiple Kernel Learning

Kernel based learning methods have proven to be an extremely effective discriminative approach to classification as well as regression problem. One approach performing kernel selection is to learn a kernel combination during the training phase of the algorithm. One prominent instance of this class is MKL. Its objective is to optimize jointly over a linear combination of kernel(equation1). Given multiple features, one can calculate multiple basis kernels, one for each feature. So the kernel is often computed as a convex combination of the basis kernels,where x_i are objects' some samples, $k_m(x_i, x_j)$ is the m^{th} Kernel, and d_m are the weights given to each kernel.

$$K(x_i, x_j) = \sum_{m=1}^{M} d_m \, k_m(x_i \, , x_j) \; , \; \sum_{m=1}^{M} d_m = 1 \; , d_m \geqslant 0 \tag{1}$$

Learning the classifier model parameters and the kernel combination weights in a single optimization problem is known as the Multiple Kernel Learning problem [23]. There are a number of formulations for the MKL problem, our approach builds on the MKL formulation of [13], known as SimpleMKL. This formulation enables the kernel combination weights to be learnt within the SVM framework. The optimization equation is given by equation 2,3,4:

$$\min \; \sum_m \frac{1}{d_m} w_m \, w_m^T + C \sum_i \xi_i \tag{2}$$

$$such\,that \;\; y_i \sum_m \varphi_m(x_i) + y_i \, b \geqslant 1 - \xi_i \;\; \forall i \tag{3}$$

$$\xi_i \geqslant 0 \; \forall i, \;\; d_m \geqslant 0 \; \forall m, \; \sum_m d_m = 1 \tag{4}$$

Where b is the bias, ξ_i is the slack afforded to each sample data and C is the regularization parameter. The solution to the MKL formulation is based on a gradient descent on the SVM objective value.

The final binary decision function of MKL is of the following form:

$$F_{MKL}(x) = sign(\sum_{m=1}^{M} \beta_m (k_m \, (x)^T \, \alpha + b)) \tag{5}$$

The only free parameter in the MKL approaches is the regularization constant C, which is chosen using Cross Validation (CV). In this paper we study a class of kernel classifiers that aim to combine several kernels into a single model. We associate object features (color features, spatial pyramid histogram features) with different parameters of kernels (Gauss kernel, χ^2 kernel) functions, kernel combination/selection translates naturally into feature combination/selection.

$$k(x_1, x_2) = \chi^2(x_1, x_2) \;\;\; \chi^2(x_1, x_2) = \sum \frac{(x_1 - x_2)^2}{(x_1 + x_2)} \tag{6}$$

$$k(x_1, x_2) = \exp((-(x_1 - x_2)^2)/(2\sigma^2)) \tag{7}$$

A conceptually simple approach is the use of CV to select the different parameters of kernels.

3.2 Spatial Pyramid Histogram Feature

The traditional bag of features methods, which represent an image as an orderless collection of local features, have severely limited descriptive ability, because there methods disregard all information about the spatial layout of features. Lazebink *etal.* [14] provides repeatedly subdividing the image and computing histograms of local features at increasingly fine resolutions. The spatial pyramid framework suggests a possible way to address the issue: namely, the best results may be achieved when multiple resolutions are combined in a principled way.

We describe the original formulation of pyramid matching [15]. Consider matching two images each consisting of a 2D point set, where we wish to determine matches between the point sets when the images are over laid for a particular point the strength of the match depends on the distances from its position to points in the other set. Each image is divided into a sequence of increasingly finer spatial grids by repeatedly doubling the number of divisions in each axis direction. The number of points in each grid cell is then recorded. This is a pyramid representation because the number of points in a cell at one level is simply the sum which is divided into four cells at the next level. The cell counts at each level of resolution are the bin counts for the histogram representing that level. The correspondence between the two point sets can then be computed as a weighted sum over the histogram intersections at each level. Similarly, the lack of correspondence between the point sets can be measured as a weighted sum over histogram differences at each level. It is illustrated in Fig.1.

A spatial pyramid histogram is the single histogram intersection of "long" vectors which is formed by concatenating the appropriately weighted histograms of all cells at all resolutions. In the long histogram, the three color bins denote features' bins of image and the height expresses occurrence which extracted features fall in each bin. When L=0, the histogram is the first part (level 0 - there are only three bins). We can see details in [14].

3.3 Online Updating

There is a set of data samples $(x_1, x_2, \ldots, x_n)$ with corresponding class labels $(y_1, y_2, \ldots, y_n)$. Let $\Phi_k(x_i, x_j)$ be the set of K kernels. The MKL solution for the

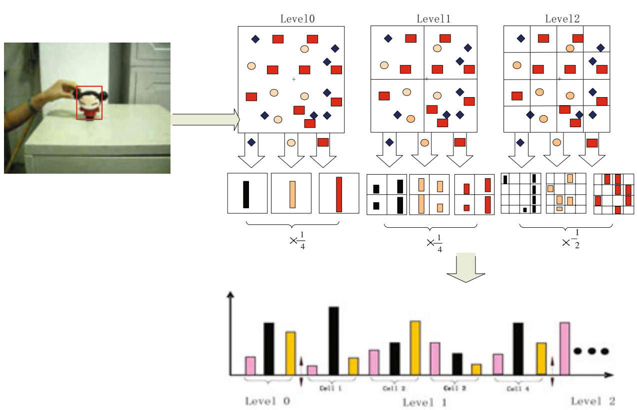

Fig. 1. Example of constructing a three-level spatial pyramid histogram. We weight each spatial histogram and concatenate them to form a long histogram.

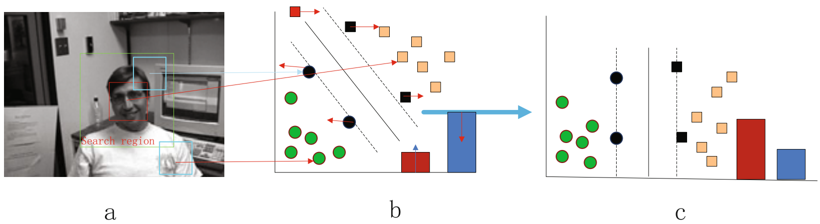

Fig. 2. (a):The object region,search region and the context region.(b):The black circles and rectangles are support vectors.kernel 1 (brown bar) kernel 2(red bar).(b)shows the effect of adding a new sample (shown in red) on the original samples and weights. Some samples change.(c)shows the final classifier after adding a new sample; and the corresponding weights are changed.

given data is obtained by SimpleMKL [13]. The data samples are divided into three disjoint sets based on their Lagrange multipliers: lying on the correct side of the margin vectors ($\alpha_i = 0$) support vectors ($0 < \alpha_i < C$) and lying on the wrong side of the margins ($\alpha_i = C$).In this paper, we adopt the online updating the tracker to make it adapt to the appearance change of the target. Through the above, a classifier based MKL is formed. When a new object sample is added to the solution, we need to calculate its Lagrange multiplier ($0 \leqslant \alpha_t \leqslant C$) such that the KKT conditions are satisfied once again by using the support vectors last time and the new samples. In the process of solving the problem, the kernel weights and the bias will be changed to maintain the constraints in KKT. It is shown in Fig.2 in short. A new point marked in red is added to the system. In order to adopt the KKT conditions, the margin changes, while some of the other points change set membership. At the same time, the kernel combination weights also change. The KKT conditions for our problem are derived from the Lagrangian function corresponding to equation 8.

$$\begin{aligned} L &= \frac{1}{2}\sum_m \frac{w_m\, w_m}{d_m} + C\sum_i \zeta_i - \sum_i v_i\, \zeta_i - \mu_m\, d_m - A - \lambda(\sum_m d_m - 1) \\ A &= \sum_i \alpha_i (y_i\, w_m\, \phi_m(x_i) + y_i\, b - 1 + \zeta_i) \end{aligned} \tag{8}$$

4 Our Tracking Framework

Within the context of object tracking, we define the object region and its surroundings as positive samples and negative samples respectively, as shown in Fig.2(a). Our target is to learn a MKL classifier which can classify the positive sample and negative sample in the new frame. Starting from first few frames, the positive and the negative samples are used to training the MKL classier. Then the search region can be estimated in the next frame. Finally, the target region in next frame is located with local maximum score within the search region.The incremental tracking [24] performs as guidance in our whole tracking process.

In order to improve the discriminative power, we utilize higher dimensional "strong features"–spatial pyramid histogram and color histogram. We use a dense regular grid instead of interest points' detection to extract SIFT features because the former can capture uniform regions such as forest, face. And the SIFT descriptors are then vectors quantized into visual "words" for the dictionary. The vector quantization is carried out by K-means clustering algorithm. Therefore, an image can be represented as a histogram which is equivalent to the occurrence frequency of dictionary of a sample. SIFT features are extracted by a dense regular grid technique, and are multi-image representations of an image neighborhood. They are Gauassian derivatives computed at 8 orientation planes over a 4×4 grid of spatial locations, giving a 128-dimension vector. The color histogram is well known so we don't need to introduce it.

One of the most difficult tasks for a tracker is how to online update the tracker to make it adapt to the appearance change of the target. Here we propose a

---

Algorithm 1. Online MKL Tracking & Updating

Input: I_n Video frames for processing
Output: Rectangles of target object's region
Training with the first frames $I_n(n = 10+)$:
(1) Manually initializing parameters describing the property of region of interest (center, size and rotation angle) in the first frame.
(2) Parameters sampling and considering the optimization produced by IVT[19] model as positive samples in the first few frames. Also obtaining negative samples around the positive samples.
(3) Extracting features (color histogram and spatial pyramid histogram) from the positive and negative samples.
(4) Train the MKL classifier (in the processing of kernel calculation, the Gaussian kernels are computed on the color features and χ_2 kernels are computed on the spatial pyramid histogram features) to get $F_{MKL}(x) = sign(\sum_{m=1}^{M} \beta_m (k_m (x)^T \alpha + b))$, and its support vectors $V_1 = \{x_i, y_i\}_{i=1}^{M}$
Online tracking: When a new frame comes:
(1) Randomly sampling 300 candidates with different affine parameters around the object's position obtained in the previous frame.
(2) Extract features, using the trained MKL classifier to find the maximum score (the object's location) given by $F_{MKL}(x)$ and go to the next frame.
(3) Approximately accumulate five or ten frames, update the MKL classifier (refresh positive samples $p_n = V^+ \cup C^+$ and negative samples $N_n = V^- \cup C^-$ V is the support vectors, and C are the new positive and negative samples by tracking).
(4) Retrain the MKL classifier using new samples for updating to $F_{MKL}(x) = sign(\sum_{m=1}^{M} \beta_m (k_m (x)^T \alpha + b))$
Go to next frame.
End

---

simple yet effective way for on-line updating the linear MKL classifier which is inspired by the [25]. And our tracker can not only record the "Key Frames"(first few frames) of the target as the history information, but can also update online to decrease the risk of drift. By online updating, the MKL tracker can adjust its hyper-plane for the maximum margin between the new positive and negative samples. The support vectors transferred frame by frame contain important "Key Frames" of the target object in the previous tracking process. The details are described as Algorithm 1.

We adopt the random sampling for improving efficiency rather than sliding window technique: in the following frame, a number of candidates with different affine parameters are generated according to a Gaussian distribution centered at the central position of the target in the previous frame. Incremental PCA tracking [24] is applied in order to collect sufficient training samples in the first few frames. In each frame, we obtain one optimal tracking result as the positive sample according to incremental PCA [24]. At the same time, a few negative samples are randomly selected around the optimal target. Note that the sampling radius should be properly predefined to guarantee that negative samples cannot overlap with the positive ones. While combining features is very beneficial, the gain obtained by MKL over simple one kernel or averaging is modest. However MKL determines a sparse selection of features, which helps improve the efficiency of inference.

5 Experimental Results

Our tracking framework is implemented on a Pentium Dual Core 1G PC with 1G memory. All the test sequences are 320×240 resolutions. A 128-dim SIFT feature is extracted to represent objects at first. The size of dictionary which we have formed is 20. And the spatial pyramid's level is 3. Thus an object is represented as a 420-dim vector (spatial pyramid histogram) by the method proposed. The

Fig. 3. "sail"sequence Top row: results of RGBSVM. Middle row: results of pyramidSVM. Bottom row: results of our approach.

Fig. 4. “singer1”sequence Top row: results of RGBSVM. Middle row: results of pyramidSVM. Bottom row: results of our approach.

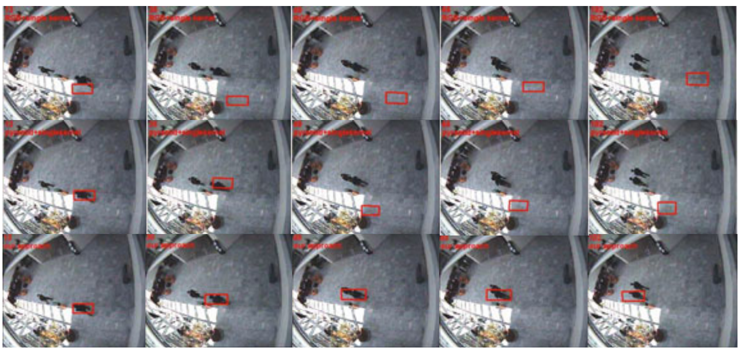

Fig. 5. “Meetwalksplit”sequence Top row: results of RGBSVM. Middle row: results of pyramidSVM. Bottom row: results of our approach.

object’s color histogram is a 256-dim vector. We sample 300 candidates randomly around the tracked object in the previous frame in the tracking. All participating kernels are Gaussian kernels and χ^2 kernels. There are 50 different parameters of Gaussian kernels (in the range from 5 to 250) and χ^2 kernels respectively (in the range from 10 to 500). For MKL we fix C = 1000 which yields best results. In our experiments, we compare the results of the color information with the single kernel (RGBSVM), spatial pyramid information with the single kernel (pyramidSVM) and our approach (color+spatial information with MKL)on publicly available datasets and our own datasets. In Fig.3(sequence “sail”–our datasets), the boy’s face is occluded by a book constantly. At the beginning all the three approaches can keep track of the face, but RGBSVM gradually fails when encountering complex and long time occlusion. At the same time, pyramidSVM can track the object reluctantly. But it is not stable and it drifts away the face. On the contrary, our approach successfully tracks the face during the entire tracking process. Fig.4 shows the tracking results for the “Singer1”sequence

Fig. 6. "EnterExitCrossingPaths1cor"sequence Top row: results of RGBSVM. Middle row: results of pyramidSVM. Bottom row: results of our approach.

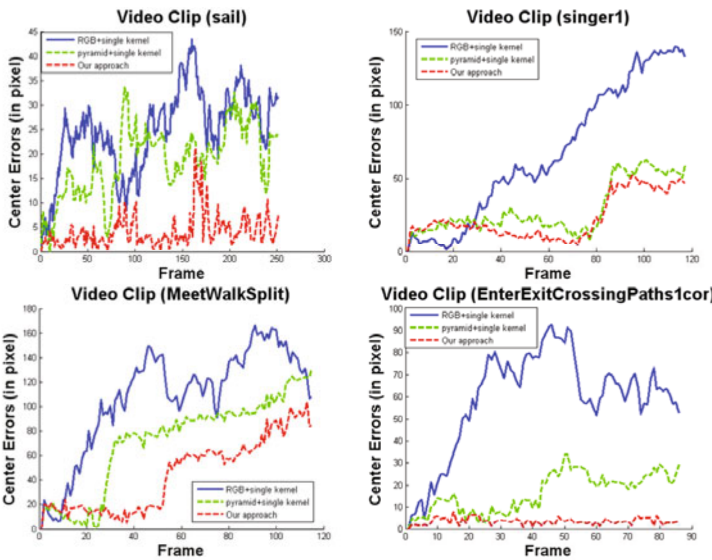

Fig. 7. The four videos' quantitive comparisons among the results of the three approaches –our approach(red), RGBSVM(blue) and pyramidSVM(green)

which is from Junseok Kwon *etal.* [26]. This sequence includes scale change and large illumination changes. The RGBSVM can not lock the object completely when there are illumination changes but pyramidSVM track the object all the same. We deduce that the spatial pyramid histogram feature displays better than the generally features. Our approach combines color, global and local feature (spatial pyramid histogram), adds updating mechanism, keeps the target still and is more robust. Fig.5, in the "Meetwalksplit"video which is from the

CAVIAR database1, both RGBSVM and pyramidSVM display badly, looses the target completely after frame 30. The video exhibits challenges, including occlusions, the small target and fast motion (which causes motion blur). But our approach performs the best. Another result -"EnterExitCrossingPaths1cor"(It is also from the CAVIAR database1) Fig.6 also shows the compared results. Though the experiment results, the approach which uses multiple kernels can be performed better than the single kernel approach, because we combine different features and different kernels.Different from existing approaches that use a linear weighting scheme to combine different features, our approach does not require the weights to remain the same across different samples, and therefore can effectively handle features of different types with different kernels. Fig.7 is the previous four videos' quantitive comparisons among the results of the three approaches–our approach, RGBSVM and pyramidSVM. RGBSVM and pyramidSVM perform not well mostly. In contrast, our approach is robust in handling occlusion,scaling and illumination changing.

6 Conclusions

In this paper, we propose a tracking framework successfully incorporating "spatial pyramid histogram" and Multiple Kernel Learning(MKL) to deal with occlusions and illumination changes, scale changes. We adopt IVT algorithm to collect training samples to training the MKL classifier using color feature and spatial pyramid histogram in the first few frames. The most important part in our paper is that we apply the spatial pyramid method to partition the image into increasingly fine sub-regions to construct long weighted and jointed histogram which includes the images' spatial information. The spatial pyramid can represent the object globally and locally. An updating mechanism –online Multiple Kernel Learning classier is adopted to deal with pose and appearance changes of object. Experiments show that our approach outperforms RGBSVM in handling occlusions and pyramidSVM in handling scaling and rotation. In a word, our approach is more robust in various situations than other methods.

Acknowledgement. The work was supported by the Fundamental Research Funds for the Central Universities, No. DUT10JS05, and the National Natural Science Foundation of China (NSFC), No.61071209.

References

1. Avidan, S.: Ensemble tracking. In: CVPR (2005)
2. Grabner, H., Bischof, H.: On-line boosting and vision. In: CVPR (2006)
3. Tian, M., Zhang, W., Liu, F.: On-line ensemble SVM for robust object tracking. In: Yagi, Y., Kang, S.B., Kweon, I.S., Zha, H. (eds.) ACCV 2007, Part I. LNCS, vol. 4843, pp. 355–364. Springer, Heidelberg (2007)
4. Tang, F.F., Brennan, S.: Co-tracking using semi-supervised support vector machines. In: ICCV (2007)

5. Babenko, B., Yang, M.-H.: Visual tracking with online multiple instance learning. In: CVPR (2009)
6. Yin, Z., Robert, T.: Collins object tracking and detection after occlusion via numerical hybrid local and global mode-seeking. In: CVPR (2008)
7. Adam, A., Rivlin, E., Shimshoni, I.: Robust fragmentsbased tracking using the integral histogram. In: CVPR (2006)
8. Bach, F.R., Lanckriet, G.R.G., Jordan, M.I.: Multiple kernel learning, conic duality, and the smo algorithm. In: ICML (2004)
9. Bosch, A., Zisserman, A., Munoz, X.: Representing shape with a spatial pyramid kernel. In: CIVR (2007)
10. Platt, J.: Fast training of support vector machines using sequential minimal optimization. In: Advances in Kernel Methods - Support Vector Learning. MIT Press, Cambridge (1998)
11. Varma, M., Ray, D.: Learning the discriminative power invariance trade off. In: ICCV (2007)
12. Rakotomamonjy, A., Bach, F., Canu, S.: More efficency in multiple kernel learning. In: ICML (2007)
13. Rakotomamonjy, A., Bach, F.R., Canu, S., Grandvalet, Y.: Simplemkl. Journal of Machine Learning Research 9, 2491–2521 (2008)
14. Lazebnik, S., Schmid, C., Ponce, J.: Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories. In: CVPR (2006)
15. Grauman, K., Darrell, T.: Pyramid match kernels: Discriminative classification with sets of image features. In: ICCV (2005)
16. Pontil, M., Verri, A.: Support vector machines for 3d object recognition (PAMI)
17. Fergus, R., Perma, P., Zisserman, A.: Object class recognition by unsupervised scale-invariant learning. In: CVPR (2003)
18. Sivic, J., Russell, B., Efros, A., Zisserman, A., Freeman, W.: Discovering objects and location in images. In: ICCV (2005)
19. Bosch, A., Zisserman, A., Munoz, X.: Representing shape with a spatial pyramid kernel. In: CIVR (2007)
20. Bosch, A., Zisserman, A., Munoz, X.: Geometric blur for template matching. In: CVPR (2001)
21. Wu, B., Nevatia, R.: Simultaneous object detection and segmentation by boosting local shape feature based classifier. In: CVPR (2007)
22. Haibin, L., Soatto, S.: Proximity distribution kernels for geometric context in category recognition.In: ICCV (2007)
23. Lanckriet, G., Cristianini, N., El Ghousi, L., Bartlett, P., Jordan, M.: Learning the kernel matrix with semi-definite programming. JMLR (2004)
24. Lim, J., Ross, D., Lin, R.-S., Yang, M.: Incremental learning for visual tracking. In: NIPS (2004)
25. Matthews, I., Ishikawa, T., Baker, S.: The template update problem. PAMI 26, 810–815 (2004)
26. Kwon, J., Lee, K.M.: Visual tracking decomposition. In: CVPR (2010)