# Object Tracking in Sports Fields

Kun Huang, Wendong Li, YikangLiao, LequanYu, YutongZhang.
College of Computer Science, Zhejiang University
...@zju.edu.cn

## Abstract

*Object tracking is among the most important computer vision problems, which has numerous applications ranging from human-computer interaction to video surveillance. In this paper, we pay attention to a practical applicaitions, tracking on sports fields. It is used widely in NBA and soccer match to obtain the senior stastics about the game, the running distance for each players during a game, the preferred position and even the team tactics. However, due to the unexpected abrupt motion and appearance changing of the target, object tracking turns out to be a difficult challenge. In this paper, we introduce 5 different object tracking algorithms and evaluate them under different circumstances.*

## 1 Introduction

Object tracking is a very important component in computer vision. Despite significant progress, tracking is still considered to be a very challenging task. It is important to evaluate the performance of state-of-the-art methods, therefore, we must develop confidence benchmark of object tracking.

When a new method is proposed, the author will define some metrics to measure the performance of his own method. However, these metrics are restricted as they only focus on specific scenario, considering few features, where their method will perform well. So, we must develop some general measures.

As we focus on the sports fields, we try to distinguish the trackers by the charactics on tracking on sports fields. Two aspect, dataset and metric must be considered. Numerous factors affect the performance of a tracking algorithm, such as illumination variation, occlusion, as well as background clutters, therefore, the benchmark dataset must contain different scenarios - the illumination in the target region is significantly changed, the target is partially or fully occluded, the target rotates out of the image plane, etc.

The metric can be divided into two aspects: the metric of each frame and the metric of the whole sequences. As for frame metric, we can define successful tracked frame or failed tracked frame. Besides, we can define the degree of success, for example the overlap with ground truth. As for the whole sequences, we must consider the frame metrics.

## 2 Algorithm Description

### 2.1 Struck

Struck is an adaptive tracking-by-detection algorithm applicable for tracking of arbitrary objects. It does not need off-line labeled data. The only labeled example is the first frame of the video, which specifies the object to be tracked. This makes Struck a very flexible tracking method.

The popularity of tracking-by-detection, which treats the tracking problem as a detection task applied to every frame, is partly due to the development of object detection, like [5]. Previous tracking-by-detection methods typically maintain a classifier trained online. During tracking, these methods choose the position to which the classifier gives the maximum classification score to be the next estimated object position. After that, a set of negative samples generated along with the positive sample(estimated position) are used to update the classifier. In a word, previous tracking-by-detection methods mainly consists of two phases: (i) generation and labeling of samples and (ii) updating classifier. (see *fig.1*) Struck, on the other hand, incorporates the two phases together using a kernel-based structured support vector machine. By adopting the framework in [4], Struck formalizes the tracking problem into a discriminant function $F : \mathcal{X} \times \mathcal{Y} \rightarrow \mathbb{R}$ that can be used for prediction according to

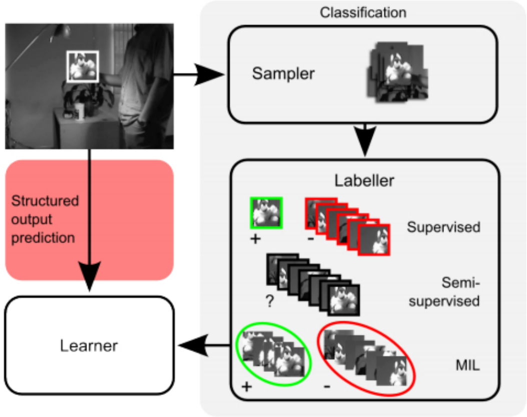


Figure 1: Previous track algorithm

$$\mathbf{y_t} = f(x_t^{p_{t-1}}) = argmax_{\mathbf{y}\in\mathcal{Y}} F(x_t^{p_{t-1}}, \mathbf{y}). \quad (1)$$

To improve the speed, Sruck introduces a budget approach with makes it suitable for online tracking.

## 2.2 TLD

The Tracking-Learning-Detection(TLD)frame is developed by Zdenek Kalal during his PhD thesis supervised by Krystian Mikolajczyk and Jiri Matas. It is a framework designed for long-term tracking of an unknown object in a video stream. The starting point of this frame is combining tracking with detection in solving the long-term tracking task. A tracker can provide weakly labeled training data for a detector and thus improve it during run-time. A detector can re-initialize a tracker and thus minimize the tracking failures.

The TLD block diagram is shown in *fig.2*. The components of the framework are characterized as follows: Tracker estimates the object's motion between consecutive frames under the assumption that the frame-to-frame motion is limited and the object is visible. The tracker is likely to fail and never recover if the object moves out of the camera view. Detector treats every frame as independent and performs full scanning of the image to localize all appearances that have been observed and learned in the past. As any other detector, the detector makes two types of errors: false positives and false negative. Learning observes performance of both, tracker and detector, estimates detector's errors and generates training examples to avoid these errors in the future. The learning component assumes that both the tracker and the detector can fail.

The Learning component of the TLD framework is P-N learning. The key idea of P-N learning is that the detector errors can be identified by two types of experts. P-expert identifies only false negatives, N-expert identifies only false positives. In fig.1, the author models the P-N learning as a discrete dynamical system and finds conditions under which the learning

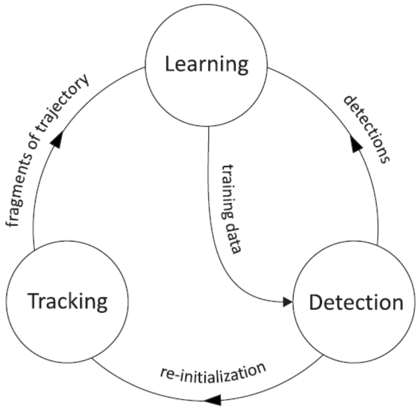


Figure 2: TLD block diagram

guarantees improvement of the detector.

The Detector component of the TLD framework is scaning the input image by a scanning-window and for each patch decides about presence or absence of the object. The main technologies is that Scanning-window grid and Nearest Neighbor Classifier. The Tracker component of the TLD framework is based on Median-Flow tracker extended with failure detection. The block diagram of author's implement is shown in *fig.3*.

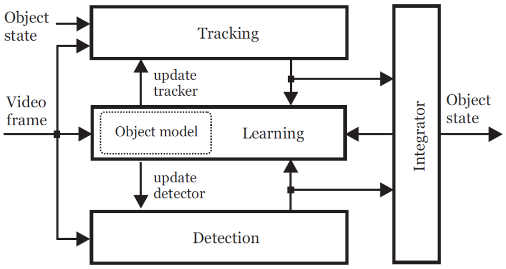
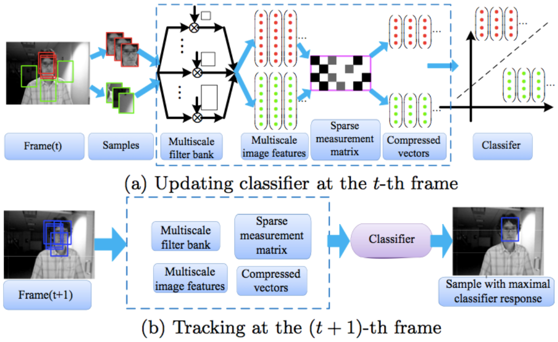


Figure 3: implementation block diagram

## 2.3 CT

The objective of this tracking algorithm is to develop an effective, efficient and robust tracker regardless of the factors such as pose variation, illumination change, occlusion, and motion blur, etc.

According to the Johnson-Lindenstrauss lemma, a sparse high-dimensional matrix x can be reduced to a low-dimensional matrix v which preserves almost all the information in x.$V = M \times H$ where M is the fixed transformation matrix. At the beginning of the tracking process, after we determining the object patch of the first frame, we then sample some positive samples near the current target location as well as some negative samples far away from the object center. These extracted patches are then dimensional reduced and used to train a naive Bayes classifier. Then in the following frames, we sample patches with different sizes near the former patch and then extract the features with low dimensionality. We then use the classifier to each feature vector v and find the tracking location with the maximal classifier response. Again, we sample the positive samples and negative samples near the current target patch now and update the naive Bayes classifier.


Frame(t)
Samples
Multiscale filter bank
Multiscale image features
Sparse measurement matrix
Compressed vectors
Classifer
Frame(t+1)
Multiscale filter bank
Sparse measurement matrix
Multiscale image features
Compressed vectors
Classifier
Sample with maximal classifier response

(a) Updating classifier at the $t$-th frame

(b) Tracking at the $(t+1)$-th frame

Figure 4: main components of CT algorithm.

### 2.4 CSK

There are three key points of CSK, as proposed in [2], which are ”Circulant Structure”, ”Tracking-by-detection”, and ”kernels”. Tracking-by-detection is a very successful approach for object tracking, but all of the traditional methods have one thing in common: a sparse sampling strategy. In each frame, several samples are collected in the target's neighborhood, where typically each sample characterizes a subwindow the same size as the target. Clearly, there is a lot of redundancy, since most of the samples have a large amount of overlap. The fact that the training data has so much redundancy means that we are probably not exploiting its structure efficiently. ”Circulant Structure” is a new theoretical framework to address this, using Fast Fourier Transform (FFT) to quickly incorporate information from all subwindows, without iterating over them. These developments enable new learning algorithms that can be orders of magnitude faster than the standard approach. We also show that classification on non-linear feature spaces with the Kernel Trick can be done as efficently as in the original image space.

### 2.5 SCM

SCM is a robust object tracking algorithm with an effective and adaptive appearance model developed by Wei Zhong , Huchuan, Ming-Hsuan Yang[9]. They use intensity to generate holistic templates and local representations in each frame.

SCM uses a collaborative model which exploits both holistic templates and local representations. Motivated by the success of sparse coding for image classification in [8, 6, 1] as well as object tracking in [3], they present a generative model for object representation that considers the location information of patches and takes occlusion into account. Their model are sparsity-based discriminative classifier (SDC) and a sparsity-based generative model (SGM). In the SDC module, it incorporates an effective method to compute the confidence value that assigns more weights to the fore-ground than the background. In the SGM module, it uses an novel histogram-based method that takes the spatial information of each patch into consideration with an occlusion handing scheme.

The update scheme considers both the latest observations and the original template in order to enabling the tracker to deal with appearance change effectively and alleviate the drift problem.

## 3 Evaluation Criterion

Having read several paper, we find it difficult to come up with new criteria. After all, this area has been studied for decades, almost all reasonable and possible measurements have been proposed. Therefore we changed our mind to find a specific application of object tracking and examine what criterion is suitable for it.

After some initial thoughts, we find an interesting and concrete application: tracking the players in the sports fields, including basketball, football and others, to compute the running distance, the favor positions and other senior statistics of each players. So our subsequent work all revolves around this application.

### 3.1 Criterion

We adopted several work from Benchmark[7].

**Success Rate**

One of the most popular metric for object tracking is success rate. We define the $r_t$ as the tracked bounding box and $r_a$ as the ground truth bounding box, the overlap score is given as

$$\frac{|r_t \cap r_a|}{|r_t \cup r_a|}$$

where $r_t$ and $r_a$ are regions of true and estimated object

When $S$ is greater than some given threshold $t_0$, we think it is a hit.Counting the total number of success frame we could get the ratio of successful frames, as the success rate. Varying the value of the threshold from 0 to 1, we could plot the success rate figure for each video and each algorithm.

**Reset Frequence**

Through the test for our video dataset, we found that it is very frequent that the tracker lost the object. Obviouly, it is meanless to measure success rate from when we lost the object as the tracker cannot find the initial object again.

Thus, we come up with a new idea, named as Reset Frequence. The object position generated by our algorithm should be reset to the ground truth if it lost the real object. We assume that there is a loss for the tracker when

$$r_t \cap r_a = 0$$

, on the contrary, if

$$r_t \cap r_a > 0$$

, we anotated it as a hit. Counting the overall reset frequence in a video, we are able to judge whether the tacker perform well in it.

**Mean Distance**

Another intuitive merit for evaluating the performance is the distance between the position detected by trackers and the ground truth. To measure the overall performance, we count the mean distance for a video.

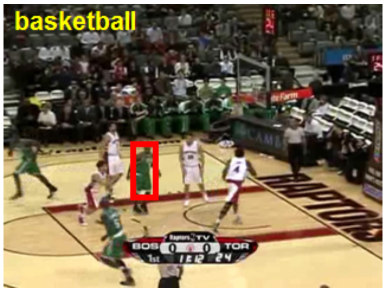


Figure 5: Basketball.

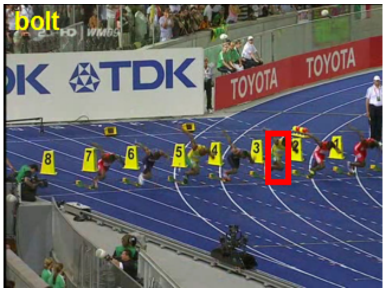


Figure 6: Bolt.

### 3.2 Dataset

We use 4 sequences listed in Benchmark[7] which could represent our application, on the sports field. Each sequences is accompanied with a grouth truth for the object in each frame(*fig.5*,*fig.6*,*fig.7*and *fig.8*).

One of the intuitive characteristics in the sequences from sports fields is that, there are so many players, which are difficult to distinguish, in each frame. Further, the players would run around, occlude each other and collide each other. It could be even more difficult if we would like to track multiple objects.

**Sequence attribute**

In Benchmark[7], the authors listed a lot sequence attributes, among which we choose 4 important ones for our application:

**Occlusion** : the target is partially or fully occluded

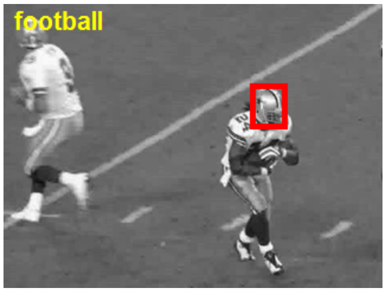


Figure 7: Football.

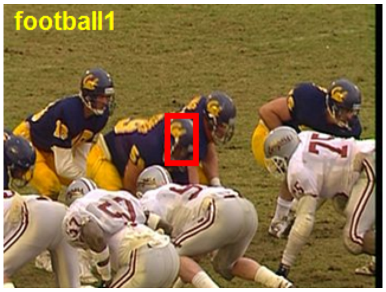


Figure 8: Football1.

**Deformation** : non-rigid object deformation

**Background Clutters** : the background near the target has the similar color or texture as the target

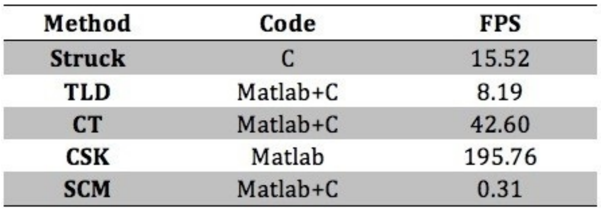
**Low Resolution** : the number of pixels inside the ground-truth bounding box is less than some $t_r$

# 4 Evaluation Result

For each tracker, the default parameters with the source code are used in all evaluations. *fig.9* shows the average FPS of each tracker on a Mac with 1.3GHz dual-core Intel Core i5 (Turbo Boost up to 2.6GHz) with 3MB shared L3 cache.

## 4.1 Overview

Then we gather experiments results together and make an overall evaluation. As mentioned above, we aim at solving real-time tracking problems in sports fields so we have four tracking sequences that are related to sports for evaluation, namely BASKETBALL, BOLT, FOOTBALL and FOOTBALL1. We will report our most significant findings in the following part.



| Method | Code | FPS |
|---|---|---|
| Struck | C | 15.52 |
| TLD | Matlab+C | 8.19 |
| CT | Matlab+C | 42.60 |
| CSK | Matlab | 195.76 |
| SCM | Matlab+C | 0.31 |

Figure 9: Football1.

We summarize the overall performance of these four trackers by Success Plots, Mean Distance, Reset Frequencies as shown in the following figures. Since these three evaluation methodologies are related to each other, we will compare and analyze the performance of the five trackers by grouping the three evaluation methodologies together.

Typically, we will use Area Under Curve (AUC) scores to rank the trackers. In this case, however, it does not work because we will reset the tracking result according to the groundtruth of image sequences when error plots occur. As we focus on the moving of sports players, we consider the number of Reset the most important factor to rank our algorithms. The less an algorithm needs to be reset, the better this tracker performance in this sequences. On the other hand, in situation where Reset to groundtruth is acceptable while we put more weight on the precision of plots, we may think highly of the AUC in the success rate figures (*fig.11*)as well as the Mean Distance Figures(*fig.10*).

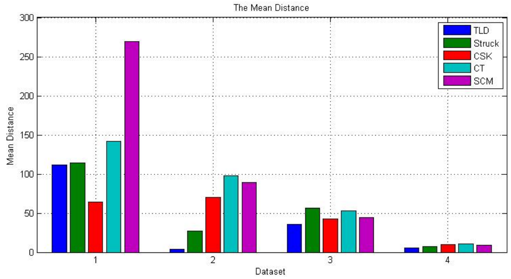


Figure 10: Mean Distance.

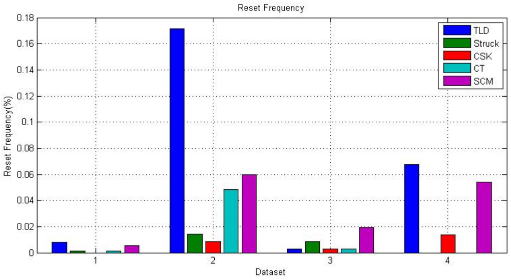


Figure 11: Reset Frequency.

### 4.2 Performance On Basketball

To begin with, let us take a look at the tracking results of BASKETBALL. As mentioned in the Dataset section, BASKETBALL is a sequences with properties of occlusion and background clutters. And we can see that our five trackers all perform not bad in this case except that TLD needs to be reset 6 times so as to hit the basketball player all the time. It is easy to find that CSK does quite a good job in this case. It ranks first because it does not need to be reset as well as it is with the least Mean Distance and largest AUC. CT and Struct consume once reset time separately and performance similar good precision and distance result. Strictly speaking, Struct may be slightly better than CT with regard to Mean Distance and Precision. As for TLD, though it has been initialized to groundtruth for 6 times, its Precision and Distance measurements are still not as good as CSK. SCM performs not that well in this case with 4 reset time and worst Mean Distance and small AUC especially when the overlap threshold is small. But when the overlap threshold is large, the success plots of SCM is higher than CT, TLD and Struct.

All in all, CSK is the best tracker for this sequences with respect to all evaluation factors.

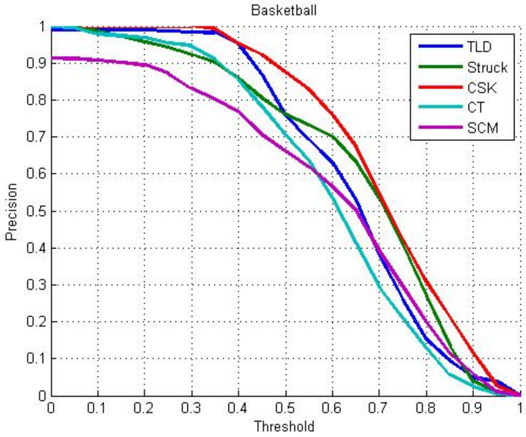


Figure 12: Basketball.

### 4.3 Performance On Bolt

The Bolt sequences, which is considered to be the most difficult testing sample among the four sequences, make some of our trackers yield really bad results. The number of reset time of TLD, SCM and CT is 60, 21 and 17 separately while Struct needs 5 times and CSK needs to be adjusted only for 3 times. So the performances of Struct and CSK are acceptable in this case. As for TLD, though its AUC in the precision plots is very large, the large number of its reset number is responsible for its outstanding performance. So it is not meaningful at all. The success rate of CT is very small as shown in the figure so it is not very precise. SCM, with similar reset number has higher success rate than CT but larger Mean Distance.

In conclusion, when an object moves fast like Bolt, CSK and Struct outperform other trackers. On one hand, CSK needs to be reset only for three times in order to track Bolt from beginning to end. On the other hand, Struct, with five reset number, has very good success rate and Mean Distance. So which one is better may depend on which factor we consider more.

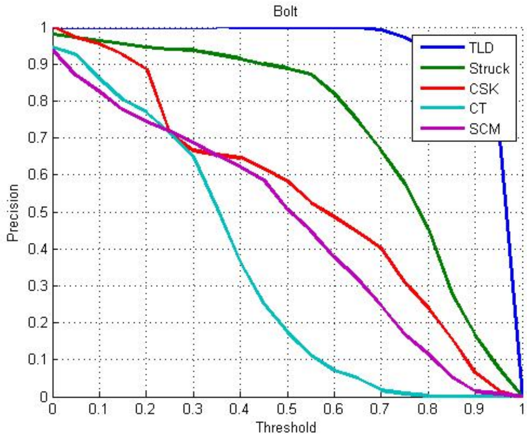


Figure 13: Bolt.

## 4.4 Performance On Football

The next sequences are Football. Our five trackers all perform well in this case. CSK, CT, TLD only needs to be adjusted once while the reset number for Struct and SCM is 3 and 7 separately. It is worth to mention that all trackers need to be reset when the tracked football player is occluded by another player. It is a common situation in real-time sports and should be handled well. So that may be the future work we ought to do to improve our trackers. Except from that, the results are satisfying when tracking the football player.

In a word, CT and CSK may be considered to be the most appropriate among the five trackers in this case. CT and CSK have least reset frequency and best success rate and Mean Distance.

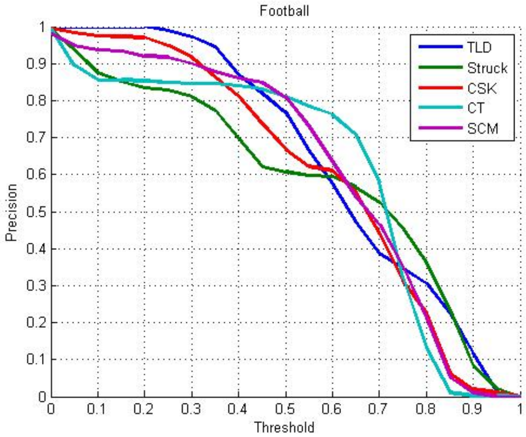


Figure 14: Football.

## 4.5 Performance On Football1

On our last testing sequence, Football1, all trackers have very satisfying Mean Distance. Struct and CT can keep tracking the object through the sequence without lost. However, the success rate of CT is not that high especially when the overlap threshold is large. TLD and SCM, though need to be adjusted to groundtruth for several times, have good success rate overall. CSK does not have bright spot in this case, but it is a good tracker overall. Also, this sequence is very short but TLD performs better in long sequences with a redetection module while there are numerous short segments in this case.

Generally speaking, Struct may be considered to be the best tracker in this case due to the reason that it can hit the football player all the time without lost and very good success rate and Mean Distance as well.

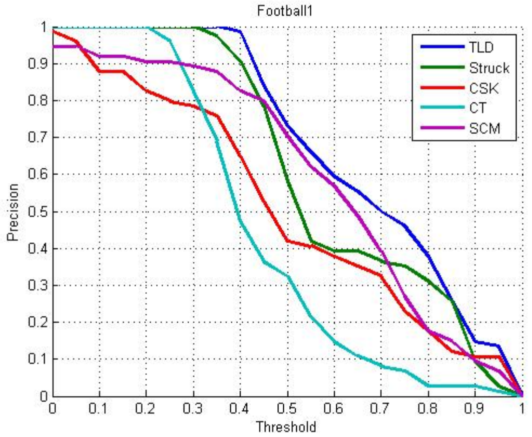


Figure 15: Football1.

### 4.6 Evaluation Conclusion

In this section, we evaluate our five state-of-art trackers with respect to Reset Frequence, Success Rate and Mean Distance and they have performed not bad in most frames of the chosen sequences. Besides, every tracker may have its preferred sequences. However, in some case such as Bolt, the results may not be that ideal. For example, TLD needs to be reset for about 60 times in the Bolt sequence due to the fast moving object and the deformation of the running person. We think it is because that these trackers are aiming at tracking all common things in the world rather than focus on just sports fields.

## 5 Conclusion

In this paper, we focused on the comparisons of 5 state-of-art object tracking approaches in sports scenario. One common application using these approaches is to tracking athletes in sports fields to measure their running distance.

We carry out large 5 experiments to evaluate the performance of 5 recent online tracking algorithms namely Struck, TLD, CT, CSK and SCM. To overcome the problem of losing target when measuring the tracking performance, we propose a new method, reset frequency, which is defined as the reset number divided by the frames number. We also use precision plot and average center location error to measure the overall performance. Since every time the tracker lost the target, we will reset the tracker to the correct location, our data is not suffered from lack of accuracy due to the losing of target.

Quantitative and qualitative comparisons with this algorithms on 4 challenging image sequences showed that even though this 5 state-of-art approaches achieve good performance in some classic datasets, but they are generally not acceptable when applied to sports video because most of the video in sports field is suffered from illumination variation, occlusion, deformation, fast motion. Based on our evaluation results and observations, we found some tracking components which are essential for improving tracking performance. First, background information is critical for effective tracking. It can be exploited by using advanced learning techniques to encode the background information in the discriminative model implicitly(e.g., Struck). Second, because when the motion of target is large or abrupt, motion model or dynamic model is crucial for object tracking, especially. Good location prediction based on the dynamic model could reduce the search range and thus improve the tracking efficiency and robustness. Improving these components will further advance the state of the art of online object tracking.

While much progress has been made in recent years, it is still a challenging problem to develop a robust algorithm for object tracking in sports field. Our ongoing work focus on how to improve the performance in complex and dynamic complex and dynamic scenes and deal with the problem of varying illumination, fast motion, occlusions shape deformation.

## References

[1] S. Gao, I. W. Tsang, L.-T. Chia, and P. Zhao. Local features are not lonely–laplacian sparse coding for image classification. In *Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on*, pages 3555–3561. IEEE, 2010.

[2] J. F. Henriques, R. Caseiro, P. Martins, and J. Batista. Exploiting the circulant structure of tracking-by-detection with kernels. In *Computer Vision–ECCV 2012*, pages 702–715. Springer, 2012.

[3] B. Liu, J. Huang, L. Yang, and C. Kulikowsk. Robust tracking using local sparse appearance model and k-selection. In *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on*, pages 1313–1320. IEEE, 2011.

[4] I. Tsochantaridis, T. Joachims, T. Hofmann, Y. Altun, and Y. Singer. Large margin methods for structured and interdependent output variables. *Journal of Machine Learning Research*, 6(9), 2005.

[5] P. Viola and M. J. Jones. Robust real-time face detection. *International journal of computer vision*, 57(2):137–154, 2004.

[6] J. Wang, J. Yang, K. Yu, F. Lv, T. Huang, and Y. Gong. Locality-constrained linear coding for image classification. In *Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on*, pages 3360–3367. IEEE, 2010.

[7] Y. Wu, J. Lim, and M.-H. Yang. Online object tracking: A benchmark. In *Computer Vision and Pattern Recognition (CVPR), 2013 IEEE Conference on*, pages 2411–2418. IEEE, 2013.

[8] J. Yang, K. Yu, Y. Gong, and T. Huang. Linear spatial pyramid matching using sparse coding for image classification. In *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*, pages 1794–1801. IEEE, 2009.

[9] W. Zhong, H. Lu, and M.-H. Yang. Robust object tracking via sparsity-based collaborative model. In *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*, pages 1838–1845. IEEE, 2012.