Constructing Adaptive Complex Cells for Robust Visual Tracking

Dapeng Chen [†] Zejian Yuan [†] Yang Wu [§] Geng Zhang [†] Nanning Zheng[†]
[†] Institute of Artificial Intelligence and Robotics, Xi'an Jiaotong University
[§] Academic Center for Computing and Media Studies, Kyoto University

Abstract

*Representation is a fundamental problem in object tracking. Conventional methods track the target by describing its local or global appearance. In this paper we present that, besides the two paradigms, the composition of local region histograms can also provide diverse and important object cues. We use **cells** to extract local appearance, and construct **complex cells** to integrate the information from cells. With different spatial arrangements of cells, complex cells can explore various contextual information at multiple scales, which is important to improve the tracking performance. We also develop a novel template-matching algorithm for object tracking, where the template is composed of temporal varying cells and has two layers to capture the target and background appearance respectively. An adaptive weight is associated with each complex cell to cope with occlusion as well as appearance variation. A fusion weight is associated with each complex cell type to preserve the global distinctiveness. Our algorithm is evaluated on 25 challenging sequences, and the results not only confirm the contribution of each component in our tracking system, but also outperform other competing trackers.*

1. Introduction

Object tracking is a fundamental problem in computer vision with applications in a wide range of domains. Meanwhile it is one of the most challenging vision tasks, due to many factors like appearance variation, heavy occlusion, illumination changes and cluttered background. To overcome these challenges, a representation should be robust enough to identify the object under motion deformation, while at the same time, the representation should also be distinctive enough to differentiate the target from background clutters. Besides, for real-time applications, computational efficiency is a crucial requirement. However, either a local representation or a global representation can only focus on one aspect of the above requirements. Generally, local representations describe the local regions. They are usually robust to motion variation and can handle partial occlusions [1, 8, 16], but easily fail in background clutters because of ignoring object structures. On the contrary, global representations can capture large object structures [20, 17, 25], but lacking the local flexibility makes them difficult to cope with occlusion and motion deformation. In addition, both kinds of representations only utilize the information inside the object and omit important contextual cues from surrounding background, which can be explored for accurate localization as well as occlusion inference.

In this paper, we propose a hierarchical representation framework. To achieve the local robustness, we extract local feature histograms as the bases of our representation, called *cells*. These cells spread over regular grids covering both object region and neighbouring background. To obtain the global distinctiveness, we integrate specific cells to construct complex cells, which can explore multiple contextual information. According to different spatial arrangements of cells, the complex cells are categorized into four types that encode the object dependencies from local region, block neighbourhood, inter-region relations and surrounding background respectively. The combination of four types achieve some complementary properties. They not only form a multi-scale representation to balance between local robustness and global distinctiveness, but also utilize both inner and outer object information.

For object tracking, we develop a novel template representation and an efficient matching algorithm. The template is composed of temporal varying cells and has two layers that store the appearance of the target and background respectively. In greater detail, the cells are modeled as Gaussian distribution according to their temporal variation and the two-layer template is convenient for context exploitation as well as occlusion inference. We track the object by matching the complex cells from candidates with those from the template. Each complex cell has two weights.

Email: {chendapeng1988, gzhang}@stu.xjtu.edu.cn; yzejian@ gmail.com; nnzheng@mail.xjtu.edu.cn; yangwu@mm.media.kyoto-u.ac.jp

One weight is associated with each complex cell to cope with occlusion as well as appearance variation, and the other weight is associated with complex cell type to preserve global distinctiveness. As the combination of complex cells form a score field with desirable heuristic cues, we utilize a coarse-to-fine search strategy, leading to a more accurate and efficient object localization. In summary, the main contributions are three folds:

- we propose an effective representation that exploits multiple scale, multiple contextual object information by integrating local histograms.
- we develop a novel two-layer template for object tracking, which not only models the temporal varying appearance of both target and background but also encodes spatial-temporal cues for occlusion inference and stability analysis.
- we evaluate the effectiveness of individual components of the proposed tracker on 25 challenging sequences, and demonstrate that the complex cells are the major force to boost the performance.

2. Related Work

Although our representation framework is not completely biological inspired, there are indeed evidences in the neurophysiological literatures to support the rationality of proposed "cells" and "complex cells". First, we choose feature histograms as local descriptors, which stem largely from the works on mammalian primary visual cortex(V1) [10]. Second, we integrate local information to represent higher level object information, which is inspired by the multiple layer-nature and inter-layer connection of visual cortex [26]. Recently, psychophysical studies indicate that generic object tracking might be implemented in a low level neural mechanism [19], and then we propose a template-based tracking method without a complicated high level object model.

Recent work on object tracking explores effective feature representation. Some trackers [31, 12, 30] apply HOG descriptor [3], which captures intrinsic edges and is invariant to illumination changes. Meanwhile, some trackers [17, 20, 32, 13, 25] utilize the representations based on image patch. These representations are easily extracted and can accurately track the target with a proper motion model. In addition, Haar-like features [27] and binary test based descriptors [21] are also employed by many competing trackers [7, 30, 2, 14, 5], as they are computational efficient and can capture large object structures. Inspired by the merits of aforementioned methods, our complex cells integrate local histograms through several simple operations. They are efficiently generated to achieve both local robustness and global distinctiveness.

Tracking methods can be classified as being either generative or discriminative. Generative methods formulate the tracking problem as searching for the regions most similar to the target model. They usually build robust object representations using particular features including superpixels [28], integral histograms [1, 8], local descriptors[9], subspace representation [22] and sparse representation [17, 20], etc. Discriminative methods formulate tracking as a classification problem to distinguish the target from the background. They usually train a dynamic classifier for target with boosting [5, 6, 2], random forest[14] or SVM [7]. Our tracker falls into the first category which searches for the state with maximum likelihood to the template. However, different from other template based schema that only model the object itself [25, 17, 8, 24, 16], we employ a two-layer adaptive template to model the appearance of slow varying target and the fast changing background simultaneously, and utilize spatial-temporal cues in template for occlusion inference and stability analysis.

3. Representation

An object is represented by a bounding box. We write $\mathcal{X}_t$ as a set of bounding boxes at frame t, whose element $\mathbf{x}_t = \{x_t, y_t, s_t\}$ is a three-dimensional vector indicating position and scale. Based on a bounding box, we construct a hierarchical representation architecture, where cells are the bases and complex cells are constructed upon the cells. The overview of our representation is shown in Fig. 1.

3.1. Cells

We divide the region within and around the bounding box into M disjoint rectangular patches. These patches are called *cells*. Among them, the cells inside the target are called *inner cells* while the others are called *outer cells*. We denote the $\mathcal{L}_{in}$ and $\mathcal{L}_{out}$ for the set of inner cell and outer cell respectively, with $\mathcal{L}_{all} = \mathcal{L}_{in} \bigcup \mathcal{L}_{out}$. Each cell is described by an intensity histogram (I) and an oriented gradient histogram (G). Intensities are in terms of gray values with gamma normalization while the gradients are computed similar to HOG [3]. Both kinds of histograms contain 8 bins computed by the methods introduced in [1] and [3]. The descriptor for cell l is a 16 dimensional vector obtained by concatenating the two histograms, denoted as $\mathbf{h}_l(\mathbf{x}_t)$. We use histogram to describe each cell because it can characterize local structures well and is robust to local motion deformation. The histogram of a rectangle can be efficiently computed by means of integral histogram [18].

3.2. Complex Cells

A complex cell is composed of a group of cells. We introduce two basic operators to describe the complex cells, where ***merge*** maintains the histogram sum of participating cells, while ***contrast*** calculates the histogram difference for a selected cell pair. The results of both operators are L_2 normalized within each channel (I and G). Based on different

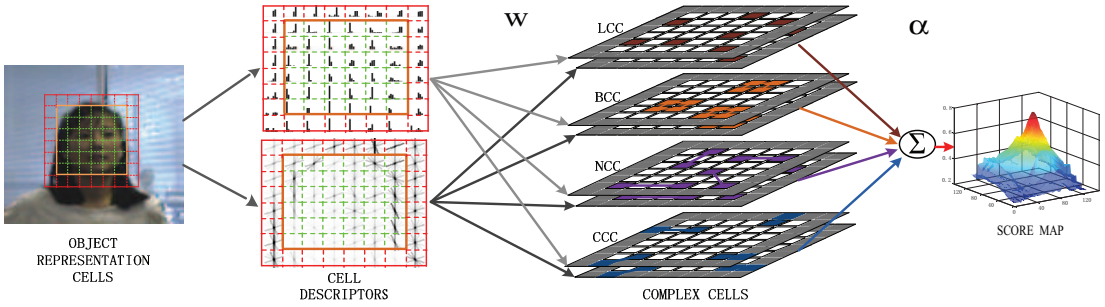

Figure 1. Overview of our tracking representation. **Left:** The spatial layout of cells. **Center left:** Cell descriptors from intensity channel(top) and oriented gradient channel (bottom). **Center right:** Examples of four type complex cells. **Right:** Fusion score. The cells composite to form complex cells, and different complex cells corporate to form the final score.

cell compositions and different operators, we propose four kinds of complex cells, also displayed in Fig. 1.

Local Complex Cell (**LCC**) is constructed by a single inner cell directly, and its descriptor is just the L_2-norm normalized cell descriptor. LCC describes the local appearance and encodes the relative position, so they can handle partial occlusion and local appearance variation effectively. Compared with [1, 8], LCC employs two complementary feature channels (I and G) to acquire extra robustness.

Block Complex Cell (**BCC**) takes neighbouring 2×2 cells to represent larger region of the object, and its descriptor is the merge of the cells. With larger "receptive field", BCC is robust to motion deformation and can provide heuristic cues for object state estimation (detailed in Sec. 4.4). BCC and LCC together form a multi-scale representation to capture geometric structures at different scales.

Non-local Complex Cell (**NCC**) is composed of a randomly selected inner cell pair, and its descriptor is the contrast of the cell pair. NCC encodes the dependency between non-local object parts, which is also useful for localization. e.g., the relationship between eyes and mouth indicates where the face is, and the coherence between coat and trouser also helps to localize a pedestrian. NCC is sensitive to shift and hence highlights the object's position.

Background-Contrast Complex Cell (**CCC**) is composed of a neighbouring inner-outer cell pair, and its descriptor is the contrast of the two cells. CCC describes the dependency between an object and its neighbourhood. It delivers two-fold benefits: (1) It highlights target contours, which are salient cues of the target; (2) It exploits the spatial correlations between a target and its neighbouring background , which in turn serves for localization.

3.3. Template

A two-layer template is proposed to represent the target and background information separately. The target template $\mathbf{T}^{ta}$ is corresponding to inner cells, while the background template $\mathbf{T}^{bg}$ is corresponding to both inner and outer cells as the inner cells may be occupied by background. As an object changes continuously during tracking, we approximate the changing features on the template using the Gaussian model. Specifically, each bin of a cell descriptor is modeled as a single Gaussian, then the cell descriptor is a 16 dimensional Gaussian with mean $\boldsymbol{\mu}$ and variance $\boldsymbol{D}$, where $\boldsymbol{\mu}$ describes the local appearance, and $\boldsymbol{D}$ reflects its temporal variance. For simplicity, the bins of the cell descriptor are assumed to be independently distributed, therefore $\boldsymbol{D}$ is a diagonal matrix. The template $\mathbf{T}$ can be represented as:

$$\mathbf{T}^{ta} = \{\boldsymbol{\mu}_l^{ta}, \boldsymbol{D}_l^{ta} | l \in \mathcal{L}_{in}\}, \quad \mathbf{T}^{bg} = \{\boldsymbol{\mu}_l^{bg}, \boldsymbol{D}_l^{bg} | l \in \mathcal{L}_{all}\} \quad (1)$$

$\mathbf{T} = \mathbf{T}^{ta} \bigcup \mathbf{T}^{bg}$. We use $\boldsymbol{\mu}$ as the cell descriptors for template, and take the inner cells from target template and the outer cells from background template to construct the complex cells. The complex cell descriptors are generated according to Sec. 3.2, which is denoted as $\mathbf{C}_T$.

4. Adaptive Complex Cell based Tracker

We develop a novel template-based tracking algorithm to exhibit the superiorities of proposed complex cells. In this algorithm, tracking is aimed at searching for the state that is most similar to the template. We propose a score function to measure the similarity, which is voted by likelihoods of all the complex cells.

$$\mathbf{S}(\mathbf{x}_t) = \sum_{m \in M} \alpha^m \sum_{j \in J^m} w_j l_j(\mathbf{C}(\mathbf{x}_t), \mathbf{C}_T) \quad (2)$$

we consider two types of weights. α^m is the fusion weight associated with each complex cell type, while w_j is the adaptive weight associated with each complex cell. $M = \{L, B, N, C\}$ are the indexes for complex cell types, and J^m are the complex cell indexes for a specific type m. $\mathbf{C}(\mathbf{x}_t)$ and $\mathbf{C}_T$ are complex cell descriptors for $\mathbf{x}_t$ and template T respectively. The optimal state $\widehat{\mathbf{x}}_t$ is the one with maximal score, namely $\widehat{\mathbf{x}}_t = \arg\max_{\mathbf{x}_t} \mathbf{S}(\mathbf{x}_t)$.

$l_j(\mathbf{C}(\mathbf{x}_t), \mathbf{C}_T)$ is the likelihood of jth complex cell. To measure the likelihood, we introduce a kernel function $\mathbf{k}$. Suppose $\mathbf{f}$ and $\mathbf{g}$ are the corresponding complex cell descriptors, function $\mathbf{k}$ integrates the two channel features by a linear combination:

$$\mathbf{k}(\mathbf{f},\mathbf{g}) = \langle \mathbf{f}^I, \mathbf{g}^I \rangle + \langle \mathbf{f}^G, \mathbf{g}^G \rangle \tag{3}$$

The results of function $\mathbf{k}$ have different ranges depending on the complex cell type. We normalize the results to be within $[0,1]$, and $l_j(\mathbf{C}(\mathbf{x}_t), \mathbf{C}_T)$ takes the normalized value.

4.1. Adaptive Weights

Adaptive weights $\mathbf{w}$ are decided by two factors: variation and occlusion. The appearance variation reflects the inner changes from the object itself, while occlusion is related to the surrounding background. Both factors are spatially related and can severely affect the tracking performance. To reduce the influence of the two factors, we focus more on stable complex cells and exclude occluded complex cells.

$$w_j = s_j \cdot o_j / \sum_{j \in J^m} s_j \cdot o_j \tag{4}$$

where s_j, o_j are the stability factor and occlusion factor associated with complex cell j. We further decompose s_j and o_j into corresponding factors of its subordinated cells.

$$s_j = \prod_{l \in \mathcal{L}_j} s_l, \qquad o_j = \prod_{l \in \mathcal{L}_j} o_l \tag{5}$$

$\mathcal{L}_j$ indexes the subcells of complex cell j. Note that different complex cells share the same weighting factors from cells so that w_j can be efficiently computed.

Stability Spatial stable parts within or around a target are important for tracking because they provide more reliable evidence to predict the target state. The stability of cell l can be directly reflected by the template variance $\boldsymbol{D}_l$. In general, a smaller $\mathbf{Tr}(\boldsymbol{D}_l)$ corresponds to a more stable cell l ($\mathbf{Tr}$ is the trace of a matrix), therefore the stability factors for inner and outer cells can be calculated as:

$$s_l = \begin{cases} \log(A_{in}/\mathbf{Tr}(\boldsymbol{D}_l^{ta})), & l \in \mathcal{L}_{in} \\ \log(A_{out}/\mathbf{Tr}(\boldsymbol{D}_l^{bg})), & l \in \mathcal{L}_{out} \end{cases} \tag{6}$$

where $A_{in} = \sum_{\mathcal{L}_{in}} \mathbf{Tr}(\boldsymbol{D}_l^{ta})$ and $A_{out} = \sum_{\mathcal{L}_{out}} \mathbf{Tr}(\boldsymbol{D}_l^{bg})$.

Occlusion Occlusion handling is also necessary, because it can alleviate the template deterioration and can use valid complex cells for accurate tracking. We provide a scheme to treat occlusion handling as background substraction. Assuming background is consistent in neighbouring cells, we determine if an inner cell is covered by the background through evaluating its affinity to neighbouring background cells. Let o_l be a binary indicator associated with cell l, if a cell is occupied by the background, $o_l = 0$, otherwise $o_l = 1$. The occlusion state of the next frame is predicted based on the current optimal state $\widehat{\mathbf{x}}_t$. Suppose cell j is adjacent to cell l, we only change the occlusion state when:

$$o_l = \begin{cases} 1 \to 0, & \text{if} \quad (\exists o_j = 0) \wedge (r(l,j) > \theta_{occ}) \\ 0 \to 1, & \text{if} \quad \mathbf{k}(\mathbf{h}_l(\widehat{\mathbf{x}}_t), \boldsymbol{\mu}_l^{ta}) > \theta_{deocc} \end{cases} \tag{7}$$

where $\mathbf{h}_l(\widehat{\mathbf{x}}_t)$ is the current cell descriptor extracted from the optimal state, and $r(l,j) = \frac{\mathbf{k}(\mathbf{h}_l(\widehat{\mathbf{x}}_t), \boldsymbol{\mu}_j^{bg})}{\mathbf{k}(\mathbf{h}_l(\widehat{\mathbf{x}}_t), \boldsymbol{\mu}_l^{ta})}$ is the ratio of its affinities with the neighbouring background template and the affinity with the its target template. We occlude the cell when it is more similar to the neighbouring background cell, and de-occlude the cell when it is similar to its target template again ($\theta_{occ} = 1.25$ and $\theta_{deocc} = 0.8$). Once the cell l is changed to be occluded, we initialize its background template with a Gaussian $(\mathbf{h}_l(\widehat{\mathbf{x}}_t), \boldsymbol{D}_0)$, where $\boldsymbol{D}_0$ is a default diagonal matrix. To guarantee sufficient number of valid cells, we apply two criteria: (1) If more than 60% of the inner cells are occluded, we de-occlude all the cells. (2) If an inner cell is occluded for more than 15 frames, we de-occluded the cell.

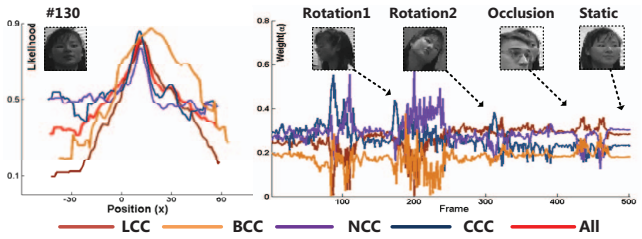

Figure 2. Fusion of four types of complex cells. **Left**. Sectional view of observation likelihood according to x axis at #130. **Right**. The time-varying curve of the fusing weights $\boldsymbol{\alpha}$ for the four types of complex cells, where the fusion weights automatically adjust to different challenges. Rotation 1 and Rotation 2 are out-of-plane and in-plane rotation respectively.

4.2. Fusion Weights

The fusion weights $\boldsymbol{\alpha}$ balance between different complex cell types to preserve global distinctiveness. For m type complex cells, α^m is computed based on the samples in the previous frame $t-1$, using a kind of "score normalization"[11]:

$$\alpha^m \propto \frac{\mathbf{S}^m(\widehat{\mathbf{x}}_{t-1}) - median}{MAD} \tag{8}$$

where $\mathbf{S}^m(\mathbf{x}_t) = \sum_{j \in J^m} w_j l_j(\mathbf{C}(\mathbf{x}_t), \mathbf{C_T})$ is the score for m-type complex cells. For all the samples collected in the frame $t-1$, $median$ is the median $\mathbf{S}^m$ and MAD measures their deviation defined as $\text{median}(|\mathbf{S}^m(\mathbf{x}_{t-1}^k) - median|)$.

α^m reflects the discriminate ability of m-type complex cells. With fusion weight α^m, we can weight more on distinctive complex cell types, which are less prone to be confounded by the background and improve the global distinctiveness of object model. The four types of complex cells are complementary for both representation and optimal state estimation, see Fig. 2. Since complex cells of different types are responsible for different structures, when a certain challenge happens, some types will degenerate their discriminate abilities, while other types are still be distinctive. They can track the objects collaboratively. Besides, combining the complex cells with different receptive field forms a score distribution with "high peak" and "heavy tail", which is desirable for a heuristical search strategy .

Algorithm 1 Complex Cell Tracker

Input: $\widehat{\mathbf{x}}_{t-1}$, $\mathbf{w}$, $\mathbf{T}$

sample $\mathbf{x}_t^{1,i} \sim q(\mathbf{x}_t^1|\widehat{\mathbf{x}}_t)$
compute the score $S(\mathbf{x}_t^{1,i})$ and $\eta_t^{1,i} \propto S(\mathbf{x}_t^{1,i})$
resample $\{\eta_t^{1,i}, \mathbf{x}_t^{1,i}\}$ to get N particles $\{\frac{1}{N}, \overline{\mathbf{x}}_t^{1,i}\}$
for $r = 2$ to R **do**
  sample $\mathbf{x}_t^{r,i} \sim q_n(\mathbf{x}_t^{r,i}|\overline{\mathbf{x}}_t^{r-1,i})$
  compute the score $S(\mathbf{x}_t^{r,i})$ and $\eta_t^{r,i} \propto S(\mathbf{x}_t^{r,i})$
  resample $\{\eta_t^{r,i}, \mathbf{x}_t^{r,i}\}$ to get N particles $\{\frac{1}{N}, \overline{\mathbf{x}}_t^{r,i}\}$
end for
estimate the optimal state $\widehat{\mathbf{x}}_t = \max_{r,i} S(\mathbf{x}_t^{r,i})$
determine the fusion weight $\boldsymbol{\alpha}$ using (8).
update the adaptive weight $\mathbf{w}$ using (4)(5)(6)(7).
update the template according to Sec. 4.3.

Output: $\widehat{\mathbf{x}}_t$, $\mathbf{w}$, $\mathbf{T}$

4.3. Updating with Occlusion

As cell descriptors in $\mathbf{T}^{ta}$ and $\mathbf{T}^{bg}$ are modeled as Gaussian distribution, we incrementally update the parameters $(\boldsymbol{\mu}_l^{ta}, \boldsymbol{D}_l^{ta})$ and $(\boldsymbol{\mu}_l^{bg}, \boldsymbol{D}_l^{bg})$ by current cell descriptor $\widehat{\boldsymbol{h}}_l(\mathbf{x}_t)$, which is also modeled as a Gaussian distribution $G(\widehat{\boldsymbol{h}}_l, \boldsymbol{D}_0)$. The template updating is therefore operated as Gaussian merging. We first update the target template $\mathbf{T}^{ta}$:

$$\begin{aligned}
&\boldsymbol{\mu}_l^* = \lambda^{ta}\boldsymbol{\mu}_l^{ta} + (1-\lambda^{ta})\widehat{\boldsymbol{h}}_l \\
&\boldsymbol{D}_l^* = \lambda^{ta}(\boldsymbol{D}_l^{ta} + \boldsymbol{\mu}_l^{ta}\boldsymbol{\mu}_l^{ta\top}) + (1-\lambda^{ta})(\boldsymbol{D}_0 + \widehat{\boldsymbol{h}}_l\widehat{\boldsymbol{h}}_l^\top) - \boldsymbol{\mu}^*\boldsymbol{\mu}^{*\top} \\
&\boldsymbol{\mu}_l^{ta} = o_l\boldsymbol{\mu}^* + (1-o_l)\boldsymbol{\mu}_l^{ta} \quad \boldsymbol{D}_l^{ta} = o_l\boldsymbol{D}^* + (1-o_l)\boldsymbol{D}_l^{ta}
\end{aligned} \tag{9}$$

where $0 < \lambda^{ta} < 1$ is a learning rate parameter. The update rule for background template $\mathbf{T}^{bg}$ is similar but with two significant differences: (1) we only update $(\boldsymbol{\mu}_l^{bg}, \boldsymbol{D}_l^{bg})$ for the cell with $o_l = 0$, which is opposite to target template; (2) since the background changes much faster than the target on the template, the learning rate λ^{bg} should be much smaller than λ^{ta} to crop the real-time background information ($\lambda^{ta} = 0.98, \lambda^{bg} = 0.4$).

4.4. Coarse-to-Fine Search

To effectively search for the optimal state $\widehat{\mathbf{x}}_t$, we propose a coarse-to-fine search strategy based on SMC (Sequential Monte Carlo) [4] to gradually approximate the high score region. Specifically, we sample $N = 50$ candidates each time and iterate $R = 9$ times. Let $\mathbf{x}_t^{r,i}$ be the i^{th} candidate at r^{th} iteration and $\eta_t^{r,i}$ be the corresponding sample's weight. The searching procedures are summarized in Alg. 1, where $q(\mathbf{x}_t^1|\widehat{\mathbf{x}}_t) = G(\widehat{\mathbf{x}}_t, \Sigma_t^0)$ is a Gaussian distribution with the mean $\widehat{\mathbf{x}}_{t-1}$ and the variance $\Sigma_t^0 = diag(\sigma_{x,t}^2, \sigma_{y,t}^2, \sigma_{s,t}^2)$. The transition probability is $q_n(\mathbf{x}_t^{r,i}|\overline{\mathbf{x}}_t^{r-1,i}) = G(\overline{\mathbf{x}}_t^{r-1,i}, \Sigma_t^{r-1})$, whose variance Σ_t^{r-1} gradually decreases as time r increases [1].

5. Experiment

Datasets We evaluate our Complex Cell based Tracker (CCT) on 25 challenging sequences, which include all the sequences in MIL benchmark [2] (*tiger2*, *tiger1*, *david*, *dollar*, *twinings*, *cliffbar*, *surfer*, *faceocc1*, *faceocc2*, *sylv*, *girl*, *coke*), all the sequences in Prost benchmark [23] (*board*, *box*, *lemming*, *liquor*) and additional 9 frequently used sequences (*shaking*, *football*, *singer1*, *animal*, *basketball* [15], *woman*, *panda* [32], *car4*, *bolt* [30]). These sequences contain different challenging situations listed in Tab. 1.

Table 1. The main challenges observed in 25 sequences.

Main Challenges	Sequence
Background Clutter	*dollar*, *basketball*, *liquor*, *football*, *bolt*
Fast motion	*tiger1*, *tiger2*, *animal*, *panda*
Rotation	*cliffbar*, *girl*, *twinnings*, *surfer*, *faceocc2*, *panda*
Illumination	*tiger1*, *tiger2*, *david*, *sylv*, *coke*, *box* *shaking*, *singer1*, *car4*, *basketball*
Partial or full occlusion	*tiger1*, *tiger2*, *girl*, *coke*, *faceocc1*, *faceocc2* *box*, *lemming*, *liquor*, *woman*, *basketball*
Pose and scale variation	*tiger1*, *tiger2*, *david*, *twinings*, *cliffbar*, *surfer*, *sylv* *girl*, *coke*, *board*, *box*, *lemming*, *liquor*, *bolt* *shaking*, *football*, *singer1*, *woman*, *car4*, *basketball*

Setup The proposed CCT tracker is implemented in MATLAB/C and runs average 10 frames per second with a 3.07 GHz CPU. The most computationally expensive procedures are the extraction of the cell descriptors and the computation of score values. The configuration of the cells depends on the shape of the initial bounding box, where the number of inner cells is around 25 and the outer cells are generated around the bounding box. The number for different types of complex cells are set as follows: LCC takes every inner cell; BCC covers every possible 2×2 cell region; 60 inner cell pairs are randomly selected for NCCs; 30 inner-outer cell pair are selected as CCCs across the bounding box boundary. It is important to note that the parameters in our method are fixed through the experiments.

Evaluation Metric For quantitative comparison, we use two evaluation criteria. Firstly, the mean center location error (CLE) is calculated for each tracker. Secondly, we report the Pascal VOC overlap ratio (VOR), that VOR $=$ Area$(B_T \bigcap B_G)$/Area$(B_T \bigcup B_G)$, where B_T is the tracking bounding box and B_G is the ground truth bounding box. Based on CLE and VOR, we also employ the Precision plot and Success plot to demonstrate the global properties. The definition of the two plots can be found in [29].

We report the most important the findings in the paper, while other results as well as source code will be available on the author's webpage: http://dapengchen.com .

[1] For Σ_t^0, $\sigma_{x,t}, \sigma_{y,t}$ equal to 1/3 diagonal length of $\widehat{\mathbf{x}}_{t-1}$, $\sigma_{s,t} = 0.03$. $\Sigma_t^1 = 0.5\Sigma_t^0$ and $\Sigma_t^r = 0.9\Sigma_t^{r-1}$ when $r \geq 2$.

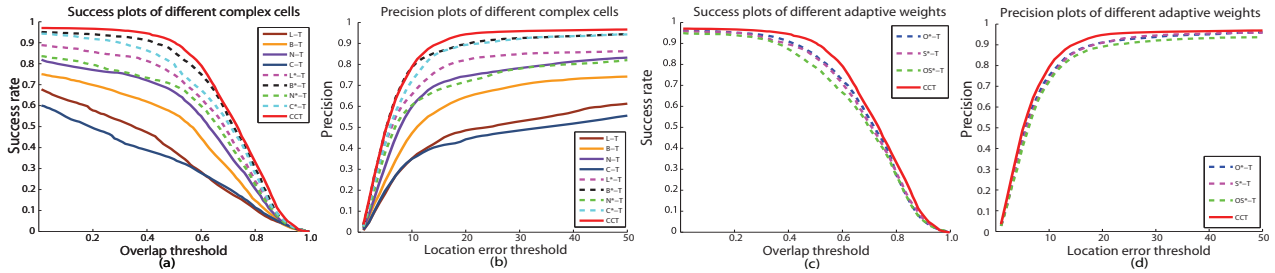

Figure 3. The success plot and the precision plot for trackers with different complex cells and different adaptive weights.

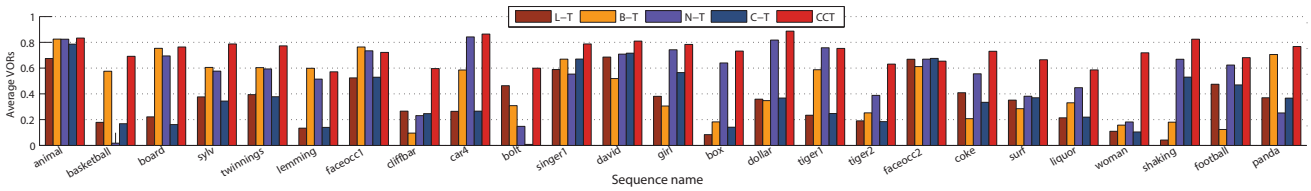

Figure 4. Comparing the performance of independent complex cells based trackers and the proposed CCT using the Pascal VOC overlap ratio.

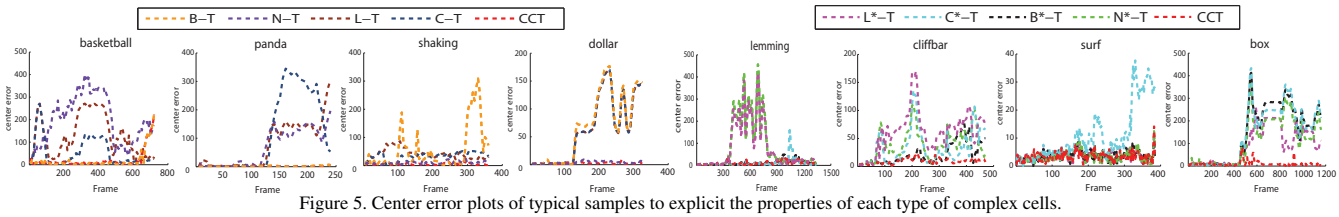

Figure 5. Center error plots of typical samples to explicit the properties of each type of complex cells.

Table 2. The average VORs and CLEs of constructed trackers with different complex cells.

Trackers	L-T	B-T	N-T	C-T	L*-T	B*-T	N*-T	C*-T	CCT
Ave CLE	65.44	53.29	29.82	89.57	25.57	16.90	35.79	17.55	**9.77**
Ave VOR	0.35	0.46	0.53	0.32	0.60	0.67	0.55	0.64	**0.71**

Table 3. The average VORs and CLEs of trackers with different adaptive weights.

Trackers	OS*-T	O*-T	S*-T	CCT
Ave CLE	14.58	12.29	15.31	**9.77**
Ave VOR	0.64	0.67	0.66	**0.71**

Table 4. The average VORs and CLEs of the trackers with and without fusion weights.

Trackers	α^*-T	CCT
Ave CLE	15.19	**9.77**

Trackers	α^*-T	CCT
Ave VOR	0.67	**0.71**

5.1. Analysis of our Method

Performance of complex cells We investigate the properties of complex cells by building the trackers L-T, B-T, N-T, C-T based on the four different types of complex cell independently. We also verify the necessity of each complex cell by constructing the L*-T, B*-T, N*-T, C*-T which cast the corresponding complex cells away from CCT. Together with CCT, we run the 9 trackers on all the sequences. The average CLEs and VORs over the frames are given in Tab. 2. The Success plots and Precision plots of the trackers over these frames are reported in Fig. 3 (a)-(b). We also gives the average VORs for each sequence in Fig. 4 and the CLE plots for some example sequences in Fig. 5.

We found that the tracking performance is significantly improved by the combination of different complex cells. The more types of complex cells the tracking system integrates, the better performance it achieves, see Fig. 3 (a)-(b) and Tab. 2. The results also indicate some complementarities among different complex cells, as shown in Fig. 4. In particular, B-T and N-T are typical complementary trackers, and they perform more reliably than L-T and C-T. However, LCC and CCC are also indispensable. If we discard either complex cell from CCT, the overall performance will decrease. Here, we investigate the performance of the each type of complex cells one by one.

BCC is more flexible than NCC on handling large deformation because of its large receptive field. Besides, its "heavy tail" property also enables our search strategy to follow fast moving objects. In *basketball*, BCC shows its superiority to NCC on handling large pose variation (Fig. 5). In *panda*, BCC also performs well on handling fast motion and in-plane rotation.

NCC is more stable than BCC for tracking rigid objects, because it emphasizes the spatial constraints between object parts. NCC is distinctive, and it can distinguish the difference between the two similar objects in *dollar* (Fig. 5). Furthermore, the contrast operator for NCC can also offset some influence caused by illumination change, so the NCC can track the target in *shaking* very well.

LCC is the basis for constructing other complex cells, it is not as reliable as NCC and BCC when used independently. However, the performance difference between L*-T and C-CT also indicates that LCC is indispensable (Fig. 3 (a)-(b)). Examples can refer to *lemming* and *cliffbar* in Fig. 5.

CCC can't track the object independently because it only

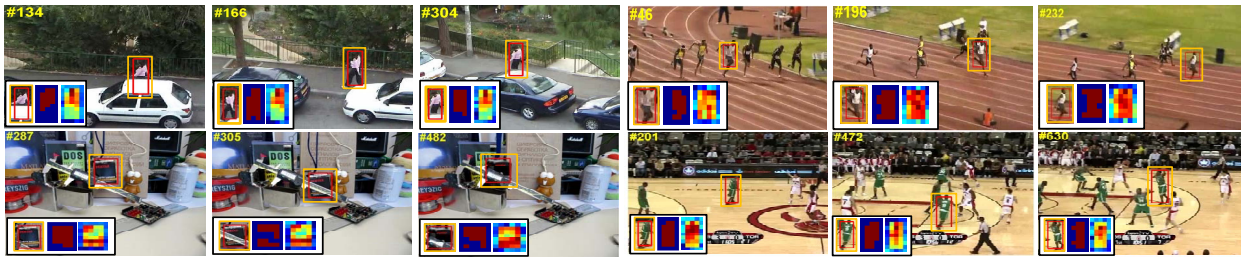

Figure 6. Qualitative results over representative frames of four sequences(i.e. *woman*,*bolt*,*box*,*basketball*), we show the occlusion mask and stability weights at the bottom left.

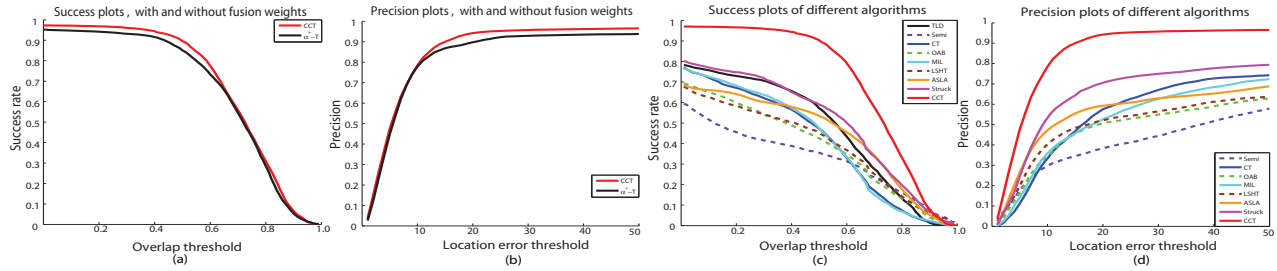

Figure 7. The success plots and the precision plots for investigating the effect of fusion weights and for the comparison of different algorithms respectively.

focuses on the object contours, which are not stable when background changes a lot or occlusion happens. Even so, CCC is still an effective cue to identify the target. Examples can refer to *surf* and *box*. Especially in *surf*, when the surfer's head rotates in the image plane, its content changes but its contours and background are relative invariant. If we don't employ CCC, the tracker C*-T drifts heavily.

Performance of adaptive weights To verify the effectiveness of adaptive weights $\mathbf{w}$, we also construct three trackers S*-T, O*-T, OS*-T that drop the stability weights s, occlusion weights o, and the two weights from CCT, respectively. The quantitative results are given in Tab. 3 and Fig. 3(c)-(d), where the results demonstrate that weighting the complex cells with occlusion and stability factors can cooperatively improve the tracking performance. Stability weights emphasize on the importance of temporal stable parts, which is particular useful for semi-deformable objects, such as human, animal, etc. Occlusion weights force CCT only use un-occluded cell to track the object, and they protect the occluded content from updating the background. We display stability weights and occlusion masks for some representative frames in Fig. 6.

Performance of fusion weights To justify the effectiveness of adaptive weights $\boldsymbol{\alpha}$, we construct a tracker $\boldsymbol{\alpha}^*$-T ignoring the fusion weights $\boldsymbol{\alpha}$, which combines the score of four types of complex cells equally. The quantitative results in Tab. 4 and Fig. 7 (a)-(b) prove the usefulness of adaptive fusion weights. Although their contribution is not as significant as other components, they provide a reasonable way to balance between difference complex cell types.

5.2. Empirical comparison of other trackers

We compare CCT with eight competing trackers named Semi [6], OAB [5], MIL [2], TLD [14], CT [30], LSHT [8], ASLA [13] and Struck [7]. The tracking results are obtained by running their publicly available source codes with default parameters[2]. Tab. 5 and Tab. 6 summarize the average VORs and average CLEs of the compared tracking algorithms [3]. Fig. 7 (c)-(d) display the success plots and precision plots of these algorithms.

The results reveal the potential benefits of integrating multiple complex cells, which enable CCT to be more universal to handle various challenges. Different from other trackers that may severely fail on certain types of videos, CCT tracks well on almost all the listed data. It is important to note that we employ all the videos in MIL Dataset and Prost Dataset without discrimination. Furthermore, if we only use a single type of complex cells (compare Fig. 3 (a)-(b) with Fig. 7 (c)-(d)), the performance may be similar to or even worse than other existing methods, which again confirms the importance of complex cell combination.

6. Conclusion

In this paper, we have presented a novel representation framework for single object tracking. We constructed complex cells from local descriptors to represent multiple scale and multiple contextual object information. Equipped with a two-layer template, the complex cells were further weighted by adaptive weights and fusion weights to cope

[2] For ASLA, we evaluate them using a fixed motion model as [29].

[3] The entry "- -" for TLD indicates the value is not available as the algorithm loses to track the target object.

Table 5. The average VORs of the nine trackers on the 25 sequences.

	Semi	CT	OAB	MIL	LSHT	ASLA	Struck	TLD	CCT
animal	0.28	0.02	0.80	0.39	0.04	0.03	**0.84**	0.51	0.83
basketball	0.17	0.29	0.04	0.26	0.40	0.20	0.02	0.02	**0.69**
board	0.24	0.61	0.17	0.37	0.63	0.70	0.65	0.57	**0.78**
sylv	0.58	0.50	0.60	0.45	0.63	0.59	0.72	0.60	**0.79**
twinings	0.48	0.59	0.62	0.61	0.46	0.65	0.64	0.36	**0.77**
lemming	0.26	0.42	0.41	0.46	0.25	0.13	0.43	0.48	**0.60**
faceocc1	0.85	0.51	0.57	0.62	0.66	0.83	**0.87**	0.53	0.71
cliffbar	0.12	0.55	0.24	0.56	0.08	0.29	0.20	**0.67**	0.61
car4	0.22	0.24	0.41	0.21	0.26	**0.87**	0.55	0.03	0.86
bolt	0.06	0.54	0.40	0.50	0.32	**0.60**	0.17	0.06	**0.60**
singer1	0.16	0.34	0.34	0.27	0.34	0.75	0.34	0.52	**0.79**
david	0.29	0.50	0.37	0.48	0.53	0.43	0.24	0.59	**0.80**
girl	0.59	0.56	0.63	0.49	0.14	0.55	0.66	0.63	**0.78**
box	0.11	0.25	0.19	0.15	0.30	0.21	0.34	0.61	**0.73**
dollar	0.36	0.69	0.45	0.61	0.86	0.83	0.65	0.31	**0.89**
tiger1	0.22	0.64	0.16	0.46	0.09	0.19	0.63	0.48	**0.74**
tiger2	0.16	0.42	0.16	0.64	0.13	0.14	0.51	0.34	**0.65**
faceocc2	0.55	0.62	0.55	0.57	0.68	0.54	**0.71**	0.63	0.65
coke	0.33	0.43	0.71	0.34	0.56	0.62	0.60	0.54	**0.73**
surf	0.05	0.07	0.39	0.23	0.04	0.05	0.38	**0.71**	0.67
liquor	0.64	0.21	0.55	0.21	0.23	0.21	0.60	**0.67**	0.59
woman	0.24	0.13	0.15	0.15	0.14	0.67	**0.73**	0.65	0.72
shaking	0.07	0.63	0.19	0.58	0.53	0.64	0.23	0.03	**0.83**
football	0.25	0.45	0.20	0.53	0.52	0.48	0.50	0.47	**0.68**
panda	0.04	0.06	0.03	0.32	0.12	0.25	0.33	0.58	**0.77**
Average	0.29	0.41	0.37	0.42	0.36	0.46	0.50	0.46	**0.73**

Table 6. The average CLEs of the nine trackers on the 25 sequences.

	Semi	CT	OAB	MIL	LSHT	ASLA	Struck	TLD	CCT
animal	78.8	246.7	6.0	54.0	100.1	140.5	**3.1**	- -	4.4
basketball	111.5	90.6	183.1	87.3	67.3	137.7	132.4	- -	**17.6**
board	128.2	29.8	140.9	60.8	26.3	17.2	24.2	30.0	**11.1**
sylv	16.2	16.6	13.1	29.7	15.0	17.8	6.9	10.3	**4.5**
twinings	20.0	12.0	8.3	10.4	20.8	7.8	7.3	15.9	**3.2**
lemming	146.0	41.8	47.6	68.6	95.3	197.4	40.6	- -	**10.7**
faceocc1	7.5	41.0	35.2	24.3	28.7	7.2	**5.8**	27.5	9.9
cliffbar	76.6	9.0	42.4	10.6	80.2	42.5	74.6	**3.7**	9.8
car4	107.3	72.2	29.4	62.6	56.7	**3.2**	6.3	- -	3.5
bolt	48.5	8.6	31.2	9.7	35.8	**8.2**	46.8	- -	8.4
singer1	89.0	14.3	13.5	28.9	16.5	**4.0**	15.2	26.6	6.7
david	36.2	11.4	24.0	10.5	6.1	18.4	64.5	19.3	**3.7**
girl	18.8	20.1	11.6	26.8	96.6	22.8	**5.8**	- -	6.6
box	140.1	123.0	131.3	117.6	106.7	145.8	140.8	- -	**10.3**
dollar	64.7	14.3	24.7	18.9	4.2	3.4	17.0	69.1	**2.5**
tiger1	46.7	8.5	47.6	23.4	79.4	41.7	7.2	- -	**4.4**
tiger2	39.3	19.5	56.8	6.0	43.0	39.7	14.9	- -	**7.6**
faceocc2	22.3	15.3	23.6	19.0	9.8	22.7	**8.0**	12.7	11.4
coke	17.9	14.1	3.8	15.5	7.2	5.4	5.5	7.8	**3.1**
surf	71.7	29.4	14.1	25.4	54.4	56.1	7.8	3.5	**3.3**
liquor	51.8	164.4	**25.5**	165.8	107.5	225.2	51.4	- -	33.6
woman	57.6	112.0	119.8	126.6	123.8	10.4	**3.5**	6.2	**3.5**
shaking	68.4	10.2	80.2	14.5	17.7	11.6	49.6	- -	**4.8**
football	30.0	15.4	149.0	13.7	19.1	19.0	14.0	17.0	**4.7**
panda	81.5	172.6	112.5	138.0	95.2	67.2	88.1	- -	**1.8**
Average	63.1	52.5	55.8	46.7	52.5	50.1	33.7	- -	**7.5**

with tracking challenges in different situations. Experiments over 25 sequences confirmed the complementarity between different complex cells and showed that the combination of them would significantly improve the tracking performance. The computation cost of our tracker lies on the feature extraction, which are desirable to adopt parallel-processing schemes to make our tracker realtime. Our representation method is flexible so that it can be simply transplanted to other model-free trackers, and it is likely to improve the performance of them as well.

Acknowledgement

This work was supported by the National Basic Research Program of China under Grant No. 2012CB316402 and the National Natural Science Foundation of China under Grant No. 91120006.

References

[1] A. Adam, E. Rivlin, and I. Shimshoni. Robust fragments-based tracking using the integral histogram. In *CVPR*, 2006. 1, 2, 3

[2] B. Babenko, M.-H. Yang, and S. Belongie. Visual Tracking with Online Multiple Instance Learning. In *CVPR*, 2009. 2, 5, 7

[3] N. Dalal and B. Triggs. Histograms of oriented gradients for human detection. In *CVPR*, 2005. 2

[4] A. Doucet and A. M. Johansen. A tutorial on particle filtering and smoothing: fifteen years later, 2011. 5

[5] H. Grabner, M. Grabner, and H. Bischof. Real-time tracking via on-line boosting. In *BMVC*, 2006. 2, 7

[6] H. Grabner, C. Leistner, and H. Bischof. Semi-supervised on-line boosting for robust tracking. In *ECCV*, 2008. 2, 7

[7] S. Hare, A. Saffari, and P. H. S. Torr. Struck: Structured output tracking with kernels. In *ICCV*, 2011. 2, 7

[8] S. He, Q. Yang, R. W. Lau, J. Wang, and M.-H. Yang. Visual tracking via locality sensitive histograms. In *CVPR*, 2013. 1, 2, 3, 7

[9] W. He, T. Yamashita, H. Lu, and S. Lao. Surf tracking. In *ICCV*, 2009. 2

[10] D. Hubel and T. Wiesel. Receptive fields and functional architecture of monkey striate cortex. *J Physiol*, pages 215–43, 1968. 2

[11] A. Jain, K. Nandakumar, and A. Ross. Score normalization in multimodal biometric systems. *Pattern Recogn.*, 38(12):2270–2285, Dec. 2005. 4

[12] S. JamesSteven and D. Ramanan. Self-paced learning for long term tracking. In *CVPR*, 2013. 2

[13] X. Jia, H. Lu, and M.-H. Yang. Visual tracking via adaptive structural local sparse appearance model. In *CVPR*, 2012. 2, 7

[14] Z. Kalal, K. Mikolajczyk, and J. Matas. Tracking-learning-detection. *TPAMI*, 34(7):1409–1422, 2012. 2, 7

[15] J. Kwon and K. M. Lee. Visual tracking decomposition. In *CVPR*, 2010. 5

[16] T. Lee and S. Soatto. Learning and matching multiscale template descriptors for real-time detection, localization and tracking. In *CVPR*, 2011. 1, 2

[17] B. Liu, J. Huang, C. Kulikowski, and L. Yang. Robust visual tracking using local sparse appearance model and k-selection. *TPAMI*, 2012. 1, 2

[18] T. Liu, Z. Yuan, J. Sun, J. Wang, N. Zheng, X. Tang, and H.-Y. Shum. Learning to detect a salient object. *TPAMI.*, 33(2):353–367, 2011. 2

[19] V. Mahadevan and N. Vasconcelos. On the connections between saliency and tracking. In *NIPS*. 2012. 2

[20] X. Mei and H. Ling. Robust visual tracking using l1 minimization. In *ICCV*, 2009. 1, 2

[21] M. Özuysal, M. Calonder, V. Lepetit, and P. Fua. Fast keypoint recognition using random ferns. *TPAMI*, 32(3):448–461, 2010. 2

[22] D. A. Ross, J. Lim, R.-S. Lin, and M.-H. Yang. Incremental learning for robust visual tracking. *IJCV*, 77(1-3):125–141, May 2008. 2

[23] J. Santner, C. Leistner, A. Saffari, T. Pock, and H. Bischof. PROST Parallel Robust Online Simple Tracking. In *CVPR*, 2010. 5

[24] G. G. Scandaroli, M. Meilland, and R. Richa. Improving ncc-based direct visual tracking. In *ECCV*, 2012. 2

[25] L. Sevilla-Lara. Distribution fields for tracking. In *CVPR*, 2012. 1, 2

[26] M. Usher, Y. Bonneh, D. Sagi, and M. Herrmann. Mechanisms for spatial integration in visual detection: a model based on lateral interactions. *Spat Vis*, 12(2):187–209, 1999. 2

[27] P. Viola and M. Jones. Rapid object detection using a boosted cascade of simple features. In *CVPR*, 2001. 2

[28] S. Wang, H. Lu, F. Yang, and M.-H. Yang. Superpixel tracking. In *ICCV*, 2011. 2

[29] Y. Wu, J. Lim, and M.-H. Yang. Online object tracking: A benchmark. In *CVPR*, 2013. 5, 7

[30] K. Zhang, L. Zhang, and M.-H. Yang. Real-time compressive tracking. In *ECCV (3)*, 2012. 2, 5, 7

[31] L. Zhang and L. van der Maaten. Structure preserving object tracking. In *CVPR*, 2013. 2

[32] W. Zhong, H. Lu, and M.-H. Yang. Robust object tracking via sparsity-based collaborative model. In *CVPR*, 2012. 2, 5