Open access • Proceedings Article • DOI:10.1109/CVPR.2010.5539821

Visual tracking decomposition — Source link

Junseok Kwon, Kyoung Mu Lee

Institutions: Seoul National University

Published on: 13 Jun 2010 - Computer Vision and Pattern Recognition

Topics: Eye tracking

Visual Tracking Decomposition

Junseok Kwon and Kyoung Mu Lee
Department of EECS, ASRI, Seoul National University, 151-742, Seoul, Korea
{paradis0, kyoungmu}@snu.ac.kr, http://cv.snu.ac.kr

Abstract

We propose a novel tracking algorithm that can work robustly in a challenging scenario such that several kinds of appearance and motion changes of an object occur at the same time. Our algorithm is based on a visual tracking decomposition scheme for the efficient design of observation and motion models as well as trackers. In our scheme, the observation model is decomposed into multiple basic observation models that are constructed by sparse principal component analysis (SPCA) of a set of feature templates. Each basic observation model covers a specific appearance of the object. The motion model is also represented by the combination of multiple basic motion models, each of which covers a different type of motion. Then the multiple basic trackers are designed by associating the basic observation models and the basic motion models, so that each specific tracker takes charge of a certain change in the object. All basic trackers are then integrated into one compound tracker through an interactive Markov Chain Monte Carlo (IMCMC) framework in which the basic trackers communicate with one another interactively while run in parallel. By exchanging information with others, each tracker further improves its performance, which results in increasing the whole performance of tracking. Experimental results show that our method tracks the object accurately and reliably in realistic videos where the appearance and motion are drastically changing over time.

1. Introduction

Object tracking is a well-known problem in computer vision community. Recently, many researchers have addressed the problem in real-world scenarios rather than a lab environment [13, 19]. In this scenario, it is a very challenging task to track an object since the scenario typically includes severe appearance or motion changes of the object. The appearance changes cover geometric and photometric variations of an object such as occlusion, pose, or illumination changes [8, 17]. Severe motion changes usually occur when a video has a low frame rate or when an object moves abruptly [11, 14]. To deal with all these changes simultaneously, tracking methods need more complex observation and motion model as well as an efficient tracking model. In this paper, we address the problem of efficiently designing complex models and making a proper tracking framework for them. Based on these models, we propose a novel tracking algorithm that can track an object robustly whose appearance and motion are drastically changing as shown in Fig.1.

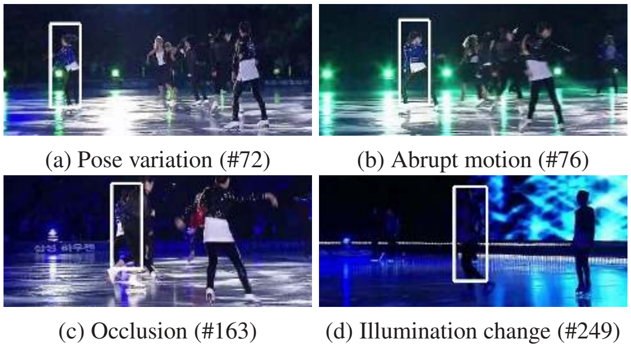

(a) Pose variation (#72) (b) Abrupt motion (#76)
(c) Occlusion (#163) (d) Illumination change (#249)

Figure 1. **Example of our tracking results** in *skating1(low frame rate)* seq. Our tracking algorithm successfully tracks a target even though there are severe pose variations, abrupt motions, occlusion, and illumination changes combinatorially.

The philosophy of our method is to utilize the *basic* distinctive components of the observation, motion, and tracking models to efficiently construct compound models. We call the process of determining these basic models as *Visual Tracking Decomposition*. In visual tracking decomposition, one basic model means a basic appearance, a basic motion, or a basic tracker for an object. To determine the basic observation models, SPCA [4] is first used to find the object models that consist of a different combination of features of an object, and then each object model is mapped to a basic observation model. Our compound observation model is made up of multiple basic observation models, and it is robust to combinatorial appearance changes since the basic models explain most variations of the object's appearances. For the basic motion models, we assume two distinct types of motions, smooth and abrupt motions, and they are mod-

978-1-4244-6985-7/10/$26.00 ©2010 IEEE

eled by different Gaussian perturbations. Similarly, a compound tracker is constructed by employing its basic trackers. For this, we introduce IMCMC [3], which consists of interactive multiple chains. In our tracking system, one chain corresponds to one basic tracker. Although each basic tracker is simple because it utilizes only one pair of observation and motion models among the various ones, by allowing the exchange of information among basic trackers that have different observation and motion models, our tracking method fuses several models efficiently.

The first contribution of this paper is to address the combinatorial and realistic tracking problem and provide an efficient solution of this problem. We test our method using unconstrained videos obtained from broadcast networks such as music concerts, sports events, or documentaries. In these videos, our method shows more accurate and reliable tracking results compared with state-of-the-art tracking algorithms. The second contribution is the proposal of the visual tracking decomposition scheme. This scheme provides an efficient strategy for designing multiple basic observation and motion models that well explain the previous and current status of an object. With these models, our method integrates multiple basic trackers into one robust compound tracker while interactively improving the performance of all basic trackers.

2. Related Works

Tracking methods for combinatorial problems: Ross et al. [17] propose an adaptive tracking method that shows robustness to large changes in pose, scale, and illumination by utilizing incremental principal component analysis. The online multiple instance learning algorithm [1] successfully tracks an object in real time where lighting conditions change and the object is occluded by others. Compared with these two works, we address more challenging scenarios for the tracking problem utilizing unstructured videos captured from broadcast networks.

Tracking methods with feature fusion: Han et al. [6] present a probabilistic sensor fusion technique. The method shows robustness to severe occlusion, clutter, and sensor failures. The method in [5] integrates multiple cues, edge, and color in a probabilistic framework while the method in [18] fuses multiple observation models with parallel and cascaded evaluation. However, these methods do not consider extreme motion changes of an object. Our method explicitly tackles the severe motion changes as well as appearance changes with the visual tracking decomposition scheme, and shows that it increases the tracking performance in a realistic tracking scenario.

Sampling based tracking methods: By handling non-Gaussianity and multi-modality, the particle filter [7] has shown efficiency in conventional tracking problems. The Markov Chain Monte Carlo method is well applied to multi-object tracking problems while rigorously formulating the entrance and exit of an object [9, 20]. As the number of observation and motion models increases, however, these methods need more samples as many times as the number of models. Our method solves this problem by utilizing IMCMC, which requires a relatively small number of samples by exchanging information between chains.

3. Bayesian Tracking Formulation

The goal of our method is to find the best configuration of an object with a given observation. The configuration at time t is represented as a three-dimensional vector, $\mathbf{X}_t = \{X_t^x, X_t^y, X_t^s\}$, where X_t^x, X_t^y and X_t^s indicate the x, y position and scale of the object, respectively. Given the state at time t, $\mathbf{X}_t$ and the observation up to time t, $\mathbf{Y}_{1:t}$, the method estimates the posteriori probability $p(\mathbf{X}_t|\mathbf{Y}_{1:t})$ with the following Bayesian formulation:

$$p(\mathbf{X}_t|\mathbf{Y}_{1:t}) \propto p(\mathbf{Y}_t|\mathbf{X}_t) \int p(\mathbf{X}_t|\mathbf{X}_{t-1}) p(\mathbf{X}_{t-1}|\mathbf{Y}_{1:t-1}) d\mathbf{X}_{t-1}, \tag{1}$$

where $p(\mathbf{Y}_t|\mathbf{X}_t)$ denotes the observation model that measures how much the target object and observation at the proposed state coincide, and $p(\mathbf{X}_t|\mathbf{X}_{t-1})$ represents the motion model that proposes the next state $\mathbf{X}_t$ based on the previous state $\mathbf{X}_{t-1}$. Then the aforementioned best configuration of an object, $\hat{\mathbf{X}}_t$ can be obtained by the Maximum a Posteriori (MAP) estimate over the N number of samples at each time t.

$$\hat{\mathbf{X}}_t = \arg\max_{\mathbf{X}_t^{(l)}} p(\mathbf{X}_t^{(l)}|\mathbf{Y}_{1:t}) \;\; for\; l = 1, \ldots, N, \tag{2}$$

where $\mathbf{X}_t^{(l)}$ indicates the *l*-th sample of the state $\mathbf{X}_t$.

Given a fixed number of samples, the accuracy of the MAP estimate in (2) increases when the posteriori probability $p(\mathbf{X}_t|\mathbf{Y}_{1:t})$ gives an accurate value. Our method obtains the accurate value of the posteriori probability by sophisticating the observation model $p(\mathbf{Y}_t|\mathbf{X}_t)$ and motion model $p(\mathbf{X}_t|\mathbf{X}_{t-1})$ in (1). For this, we design the model as the weighted linear combination of its basic components.

$$p(\mathbf{Y}_t|\mathbf{X}_t) = \sum_{i=1}^{r} w_t^i p_i(\mathbf{Y}_t|\mathbf{X}_t), \;\; \sum_{i=1}^{r} w_t^i = 1, \tag{3}$$

$$p(\mathbf{X}_t|\mathbf{X}_{t-1}) = \sum_{j=1}^{s} w_t^j p_j(\mathbf{X}_t|\mathbf{X}_{t-1}), \;\; \sum_{j=1}^{s} w_t^j = 1, \tag{4}$$

where r and s represent the number of basic components of the observation and motion model, respectively, $p_i(\mathbf{Y}_t|\mathbf{X}_t)$ denotes the *i*-th basic observation model, $p_j(\mathbf{X}_t|\mathbf{X}_{t-1})$ indicates the *j*-th basic motion model, and w_t^i is the weighting variable at time t. The following sections will explain

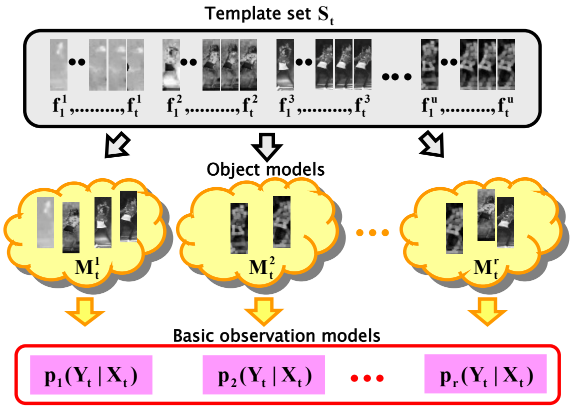

Template set $\mathbf{S}_t$
$\mathbf{f}_1^1,\ldots,\mathbf{f}_t^1$
$\mathbf{f}_1^2,\ldots,\mathbf{f}_t^2$
$\mathbf{f}_1^3,\ldots,\mathbf{f}_t^3$
$\mathbf{f}_1^u,\ldots,\mathbf{f}_t^u$
Object models
$\mathbf{M}_t^1$
$\mathbf{M}_t^2$
$\mathbf{M}_t^r$
Basic observation models
$\mathbf{p}_1(\mathbf{Y}_t\,|\,\mathbf{X}_t)$
$\mathbf{p}_2(\mathbf{Y}_t\,|\,\mathbf{X}_t)$
$\mathbf{p}_r(\mathbf{Y}_t\,|\,\mathbf{X}_t)$

Figure 2. **The process of observation model decomposition** We make the set S_t utilizing image templates up to time t for the u number of different features. Then the SPCA method constructs the object models M_t^i by selecting a certain subset of S_t. With each object model, each basic observation model is defined by (8).

how to determine these basic models efficiently (section 4) and estimate the weight of each model implicitly (section 5). Note that $p_i(\mathbf{Y}_t|\mathbf{X}_t)_{i=1,\ldots,r}$ in (3) form r different basic observation models. Similarly, $p_j(\mathbf{X}_t|\mathbf{X}_{t-1})_{j=1,\ldots,s}$ in (4) build up s different basic motion models. For clarity, we call $p(\mathbf{Y}_t|\mathbf{X}_t)$ and $p(\mathbf{X}_t|\mathbf{X}_{t-1})$ as the *compound* observation and the *compound* motion model, respectively, hereafter.

4. Model Decomposition

4.1. Basic Observation Models

In this paper, we employ the mixture of templates model for object representation. For this, we define a set S_t, which consists of different types of feature templates of an object up to time t:

$$S_t = \{f_m^n | m = 1, \ldots, t, n = 1, \ldots, u\}, |S_t| = tu, \quad (5)$$

where f_m^n denotes the n-th type of the feature template at time m and $|S_t|$ indicates the total number of feature templates in S_t. In (5), different types of feature templates f_m^n are obtained by utilizing different types of feature extractors F^n for the image patch $I(\hat{\mathbf{X}}_m)$ at each time:

$$f_m^n = \frac{F^n(I(\hat{\mathbf{X}}_m))}{\|F^n(I(\hat{\mathbf{X}}_m))\|}, m = 1, \ldots, t, n = 1, \ldots, u, \quad (6)$$

where $I(\hat{\mathbf{X}}_m)$ represents the image patch at time m described by $\hat{\mathbf{X}}_m$ in (2) and F^n indicates the feature extractor for obtaining the n-th type of the feature template.

Each basic observation model $p_i(\mathbf{Y}_t|\mathbf{X}_t)$ in (3) takes one subset of S_t as its own object model M_t^i at time t.

$$M_t^i \subset S_t, i = 1, \ldots, r. \quad (7)$$

Then, it is determined by

$$p_i(\mathbf{Y}_t|\mathbf{X}_t) = exp^{-\lambda DD(\mathbf{Y}_t, M_t^i)}, i = 1, \ldots, r, \quad (8)$$

where λ denotes the weighting parameter [1], and $\mathbf{Y}_t$ represents the u number of observations obtained by feature extractors $F^n, n = 1, \ldots, u$ for the image patch described by $\mathbf{X}_t$. In (8), the DD function returns the diffusion distance between the observation $\mathbf{Y}_t$ and the object model M_t^i at time t. We utilize diffusion distance as a dissimilarity measure, since it is robust to deformation as well as quantization effects of the observation [15]. Because $\mathbf{Y}_t$ and M_t^i consist of multiple observations and multiple templates, respectively, $DD(\mathbf{Y}_t, M_t^i)$ is computed as the sum of dissimilarity between each observation in $\mathbf{Y}_t$ and each template in M_t^i. To complete the designing of $p_i(\mathbf{Y}_t|\mathbf{X}_t)$ in (8), the remaining task is to obtain the r number of different subsets $M_t^i, i = 1, \ldots, r$. This is efficiently done by the sparse principal component analysis method in the next subsection.

4.1.1 Sparse Principal Component Analysis

There are three conditions for the object model M_t^i to be good in terms of tracking performance and efficiency. The first condition is that M_t^i has to cover most appearance changes in an object over time. The second is that the formation of it should be as compact as possible while preserving its good performance. The last condition is that relations between $M_t^i, i = 1, \ldots, r$ should be complementary. To satisfy all of these conditions, our method adopts the SPCA method to construct M_t^i. Given a Gramian matrix A_t, the original SPCA method [4] seeks out sparse principal components c, which only have a limited number of nonzero entries while capturing a maximum amount of variance:

$$\begin{aligned} &maximize \quad c^T A_t c - \rho|c|^2 \\ &subject\ to \quad \|c\|_2 = 1, \end{aligned} \quad (9)$$

where $|c|$ is the number of nonzero entries in c and ρ controls the penalty on the nonzero entries of c. As the ρ value increases, we have more sparse principal components c [2]. For our tracking problem, the Gramian matrix A_t at time t is constructed as

$$\begin{aligned} &A_t = a^T a, \\ &a = \begin{pmatrix} f_1^1 & \ldots & f_t^1 & \ldots & f_1^u & \ldots & f_t^u \end{pmatrix}, \end{aligned} \quad (10)$$

where the size of A_t is $|S_t| \times |S_t|$ since the column size of the matrix a is $|S_t|$.

With the conventional convex optimization tools [4], we can efficiently obtain the approximate solutions of (9).

[1] We set λ to 5 in all of the experiments.

[2] We set ρ to 90 in all of the experiments.

Among these components, we choose the r principal components $c_i, i = 1, \ldots, r$ according to the eigenvalue in descending order. The chosen components compose each object model M_t^i in (7) as follows:

$$M_t^i = \{f_m^n | f_m^n = a(x), c_i(x) \neq 0\}. \tag{11}$$

If the x-th element of c_i has a nonzero value, M_t^i includes the template f_m^n located at the x-th column of the matrix a in (10). By doing this, each object model M_t^i captures the significant appearance changes in an object since each model is constructed by each significant eigenvector. And sparsity of the eigenvector gives compactness to the model while making it have a small number of templates. Since the eigenvectors have orthogonal property, the object models have complementary relationship with each other. Fig. 2 illustrates the whole process of observation model decomposition.

4.2. Basic Motion Models

Each basic motion model $p_j(\mathbf{X}_t|\mathbf{X}_{t-1})$ in (4) describes different types of motions made by a Gaussian perturbation with a different variance.

$$p_j(\mathbf{X}_t|\mathbf{X}_{t-1}) = G(\mathbf{X}_{t-1}, \sigma_j^2), j = 1, \ldots, s, \tag{12}$$

where G represents the Gaussian distribution with mean $\mathbf{X}_{t-1}$ and variance σ_j^2. We assume that the motion of an object can be decomposed into two kinds of motions, smooth and abrupt, and make two motion models, $p_1(\mathbf{X}_t|\mathbf{X}_{t-1})$ and $p_2(\mathbf{X}_t|\mathbf{X}_{t-1})$. $p_1(\mathbf{X}_t|\mathbf{X}_{t-1})$ explains the smooth motion with a small σ_1^2. This kind of the motion model further simulates the seemingly good moves near the local minima (*exploitation*). On the other hand, $p_2(\mathbf{X}_t|\mathbf{X}_{t-1})$ covers the abrupt motion with a large σ_2^2. In this case, the model further simulates moves that have not been explored much (*exploration*). Our method makes full use of the *exploitation* ability with the *exploration* ability by implicitly combining two motion models in section 5.

4.3. Basic Tracker Models

Our compound tracker is composed of $r \times s$ number of basic trackers $T_i^j, i = 1, \ldots, r, j = 1, \ldots, s$ utilizing all pairs of the observation models $p_i(\mathbf{Y}_t|\mathbf{X}_t)_{i=1,\ldots,r}$ and motion models $p_j(\mathbf{X}_t|\mathbf{X}_{t-1})_{j=1,\ldots,s}$ as shown in Fig. 3. Since we choose a few robust basic observation models using SPCA, the number of basic observation models is not increased as many times as the size of the template set S_t in (5). And the number of basic motion models is fixed to 2. Therefore, our method typically maintains a small number of basic trackers and shows good performance in terms of scalability even on a large template set.

Each basic tracker constructs a Markov Chain modeled by one pair of a basic observation and a basic motion model, and produces samples of the state for the MAP estimate in (2) via the Metropolis Hastings algorithm. The algorithm consists of two main steps: the proposal step and the acceptance step. In the proposal step, a new state is proposed by the proposal density function.

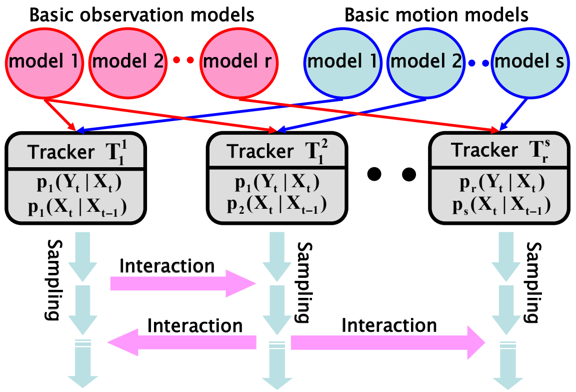


Figure 3. **The process of tracker decomposition** Each pair of the observation and motion model makes each basic tracker. Therefore, the total number of trackers is $r \times s$ if there are r observation and s motion models.

$$Q_j(\mathbf{X}_t^{j^*}; \mathbf{X}_t^j) = p_j(\mathbf{X}_t^{j^*}|\mathbf{X}_t^j), \tag{13}$$

where Q_j denotes the proposal density function which utilizes the j-th motion model in (12) and $\mathbf{X}_t^{j^*}$ represents the new state proposed by Q_j at time t.

Given the proposed state, the tracker T_i^j decides whether the state is accepted or not with the acceptance ratio in the acceptance step:

$$\gamma_{parallel} = min\left[1, \frac{p_i(\mathbf{Y}_t|\mathbf{X}_t^{j^*})Q_j(\mathbf{X}_t^j; \mathbf{X}_t^{j^*})}{p_i(\mathbf{Y}_t|\mathbf{X}_t^j)Q_j(\mathbf{X}_t^{j^*}; \mathbf{X}_t^j)}\right]. \tag{14}$$

These two steps iteratively go on until the number of iterations reaches a predefined value.

5. Integration by Interactive Markov Chain Monte Carlo

While the sampling process goes on, the basic trackers communicate information about the good configuration of an object to other basic trackers as shown in Fig. 3. Since each basic tracker utilizes a different pair of the observation and motion model, exchanging information results in fusing all of these models and estimating the weight w_t^i in (3) and w_t^j in (4) implicitly. To communicate with each other, we introduce IMCMC [3] to our tracking problem. Our method consists of two modes, parallel and interacting. In the parallel mode, the method acts as the parallel Metropolis Hastings algorithms explained in the previous subsection. When the method is in the interacting mode,

Algorithm 1 Visual Tracking Decomposition

```
Input: $\mathbf{X}_{t-1} = (\mathbf{X}^x_{t-1}, \mathbf{X}^y_{t-1}, \mathbf{X}^s_{t-1}), \alpha = 1$
Output: $\hat{\mathbf{X}}_t = (\hat{\mathbf{X}}^x_t, \hat{\mathbf{X}}^y_t, \hat{\mathbf{X}}^s_t)$
rand() returns a random number between 0 and 1.
for 1 to $\frac{N}{rs}$ do
    if rand()$< \alpha$ then
        for 1 to $rs$ do
            Accept the new state with the probability (15).
        end for
    else
        for 1 to $rs$ do
            Propose the new state using (13).
            Accept the new state with the probability (14).
        end for
    end if
    Decrease the $\alpha$ value.
end for
Estimate the MAP state $\hat{\mathbf{X}}_t$ using (2).
Determine basic observation models using (8).
Determine basic motion models using (12).
```

the trackers communicate with the others and make leaps to better states of an object. A basic tracker accepts the state of tracker T_i^j as its own state with the following probability:

$$\gamma_{interacting} = \frac{p_i(\mathbf{Y}_t|\mathbf{X}_t^j)}{\sum_{i=1}^{r}\sum_{j=1}^{s} p_i(\mathbf{Y}_t|\mathbf{X}_t^j)}, \qquad (15)$$

where $p_i(\mathbf{Y}_t|\mathbf{X}_t^j)$ returns the likelihood score of the *i*-th observation model at the state obtained from the *j*-th motion model. Our method operates in an interacting mode with the probability α, which linearly decreases from 1.0 to 0.5 as the simulation goes on. Algorithm 1 illustrates the whole process of our tracking method [3].

6. Experimental Results

6.1. Implementation Details

In the experiment, we utilized hue, saturation, intensity, and edge template for the features. The hue template expresses the chrominance characteristic of an object. The intensity template represents the brightness status of the object [16]. And the edge template gives a relatively consistent information about the shape of the object even when there are severe illumination changes [10]. With four different types of features, we made the set S_t in (5) using five image patches obtained at the initial frame and four recent frames where $|S_t|$ is 20. The parameters of our method were fixed for all of the experiments except the variance of the first motion model, σ_1 [4]. In all the experiments, the number of observation and motion model are set to 4 and 2, respectively. So the total number of the basic trackers is 8. We compared the proposed algorithm (VTD) with four different tracking methods: standard MCMC (MC) based on [9][16]; Mean Shift (MS) [2] based on the implemented function in OpenCV; On-line Appearance Learning (OAL) in [17]; and Multiple Instance Learning (MIL) in [1]. We used the software of authors for testing OAL and MIL. Note that our current implementation is not optimized, and it spends most computational time to get a likelihood score by measuring the diffusion distance in [15]. Thus, by properly optimizing the process of measuring diffusion distance, we can greatly enhance the speed although it takes 1 $\sim$ 5 seconds per frame at the current state. The supplementary material contains videos of tracking results.

	MC	MS	OAL	MIL	VTD	VTD$\tilde{I}$	VTD$\tilde{S}$
tiger1	27	93	65	**15**	**13**	35	54
david	49	88	**4**	23	**7**	24	70
face	19	45	**19**	27	**7**	8	8
shaking	97	241	95	**38**	**5**	7	68
soccer	47	97	151	**41**	**21**	22	96
animal	32	207	**23**	30	**11**	13	40
skating1	111	141	174	**85**	**7**	8	219

Table 1. **Comparison of tracking results.** The numbers indicate average center location errors in pixels. These numbers were obtained by running each algorithm 5 times and averaging the results.

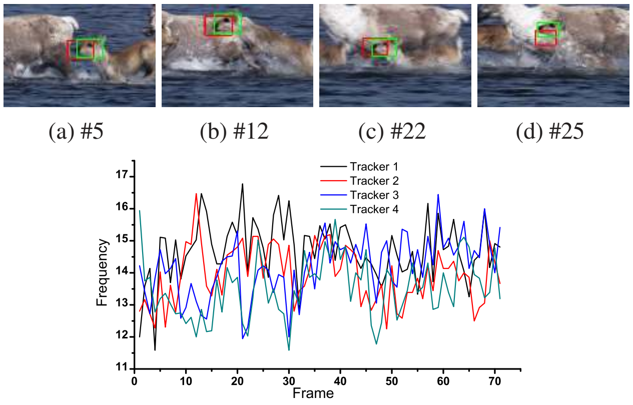
6.2. Quantitative Evaluation

Performance of the tracking algorithms: Table 1 summarizes the tracking results on seven different datasets [5]. In most sequences, our method (VTD) most accurately tracked the targets even though there were severe occlusions, pose variations, illumination changes, and abrupt motions. For the sampling-based methods, which are MC, OAL, and VTD, we used the same number of samples, 800, to track an object. Given the fixed number of samples, our method efficiently searched the solution space and found the best local minimum state of an object. This is because our method exchanged information about the good states between the basic trackers. MIL and OAL showed the second-best performance on average although MIL does not track scale. MIL is good since it was hardly affected by param-

[3] Although IMCMC does not satisfy detailed balance, it produces fair samples from the target posterior in (1) and typically converges [3]. IMCMC needs no burn-in period for the MAP problem.

[4] σ_1 is typically set to $\sigma_1^x = 2, \sigma_1^y = 1.414$, and $\sigma_1^s = 0.0165$ where σ_1^x, σ_1^y, and σ_1^s denote the variance of the x,y translation and scale, respectively. Variance of the second motion model σ_2 is set to $\sigma_2 = 2\sigma_1$. Although the tracking result could be dependent on the variance of the motion model, it is not so sensitive in practice.

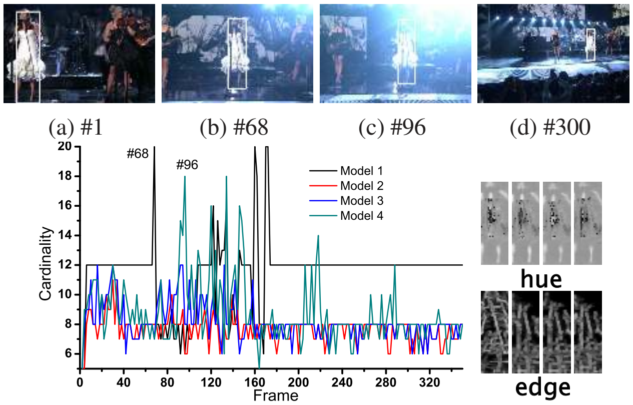
[5] These datasets consist of three publicly available video sequences (*tiger1*, *david indoor*, *occluded face*) in [1] or [17] and four made by us (*shaking*, *soccer*, *animal*, *skating1*). We found the ground truth of our datasets by manually drawing the bounding box of an object.



(a) #5 (b) #12 (c) #22 (d) #25

(e) The frequency of interaction between 4 trackers

Figure 4. **Interaction between multiple trackers** in *animal* seq.



(a) #1 (b) #68 (c) #96 (d) #300

(e) Cardinality of 4 observation models (f) Added templates at #68

Figure 5. **Adaptiveness of observation models** in *singer1* seq.

eter settings. However, MIL failed when severe illumination changes drifted the tracker into a background. OAL was robust to the illumination changes but weak to severe occlusions and viewpoint changes. Our method overcame these changes by designing the compound observation and motion model, and the efficient tracker utilizing the visual tracking decomposition scheme.

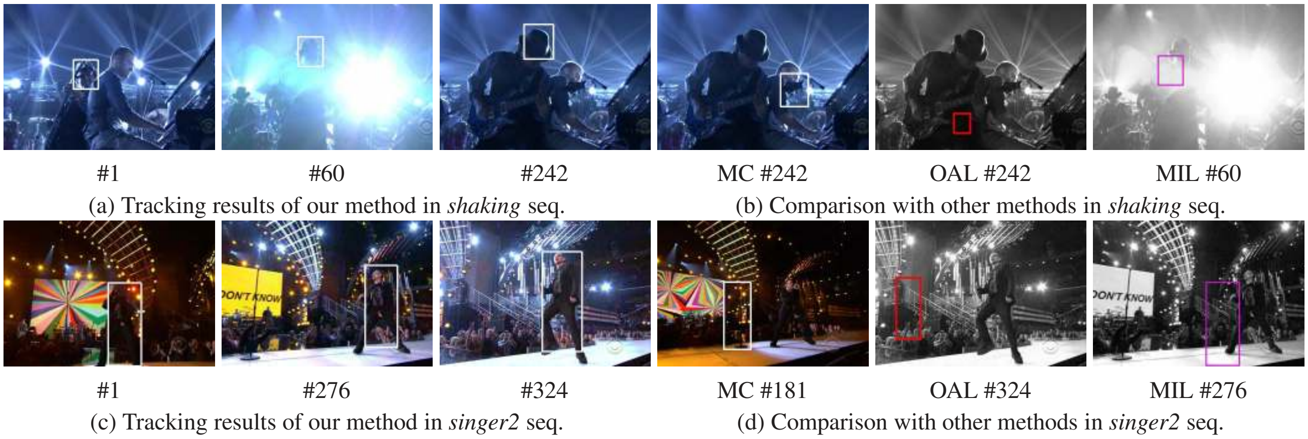
Performance of IMCMC: VTD had a better performance than VTD$\tilde{I}$ as shown in the table 1 where VTD$\tilde{I}$ denotes our method without interaction between trackers. The results show that the interaction process in VTD is important to improve the tracking performance, especially in *tiger1* seq. The sequence contains several kinds of appearance and motion changes. In the VTD method, the proper tracker among multiple ones covered these changes at each time and propagated its state to the other trackers. This is why VTD typically gave more accurate results than VTD$\tilde{I}$. Fig. 4(e) describes how frequently each tracker exchanges information about the state in *animal* seq., which includes drastically abrupt motions of the object. In this sequence, each basic tracker actively interacted with the rest while helping the other basic trackers to make leaps to a better state. Although some basic trackers failed to track the object, our method successfully found the proper state of the object as shown in Fig. 4(a)-(d) where the red rectangle denotes the tracking result of the failed basic tracker. The green rectangle indicates the leapt state of the failed tracker with the help of other good basic trackers.

Performance of SPCA: VTD$\tilde{S}$ indicates our method without sparse principal component analysis. We designed VTD$\tilde{S}$ such that the different observation model employs a different type of features. On the other hand, each observation model of VTD includes several types of feature templates obtained by SPCA. As shown in the table 1, the performance of VTD was drastically improved as compared to VTD$\tilde{S}$. This means that the observation models constructed by SPCA are very useful in our tracking problem. Fig. 5(e) shows how adaptively SPCA constructs object models at each frame under severe illumination changes from frame #60 to #170 in *singer1* seq. The changes of cardinality in each model indicate that SPCA transforms each model into a different one to cover the specific appearance changes in an object. At frame #68, to represent the illuminated object in Fig 5(b), SPCA added hue and edge templates to Model 1 as shown in Fig. 5(e)(f), which are relatively robust to the illumination changes [19]. Similarly, at frame #96, Model 4 is severely modified to deal with these changes. With help of SPCA, VTD tracks the object accurately in spite of severe illumination changes as illustrated in Fig. 5(a)-(d).

6.3. Qualitative Evaluation

Illumination change and pose variation: Fig. 6 presents the tracking results in *shaking* and *singer2* seq. While the stage lighting condition is drastically changed, and the pose of the object is severely varied due to head shaking or dancing, our method successfully tracked the object as shown in Fig. 6(a)(c). Since our observation models evolve themselves by online update, our method efficiently covered the pose variations. Additionally, the method was robust to illumination change because the observation models utilize a mixture of templates. However, other methods failed to track the object when these changes occur combinatorially as illustrated in Fig. 6(b)(d).

Occlusion and pose variation: Fig. 7 demonstrates how the proposed method outperforms the conventional tracking algorithms when the target is severely occluded by other objects. As shown in Fig. 7(a)(c), our method robustly tracked the object in *soccer* and *skating2* seq. The method was robust to the occlusion because it constructed multiple observation models. Each model kept a different history of the object's appearance over time, which includes the oc-



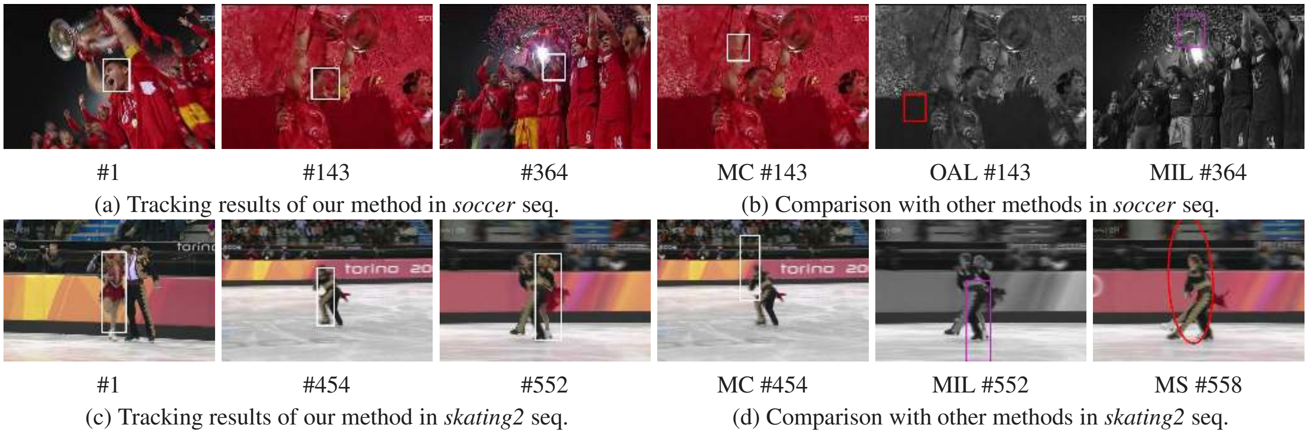
(a) Tracking results of our method in *shaking* seq.

(b) Comparison with other methods in *shaking* seq.

(c) Tracking results of our method in *singer2* seq.

(d) Comparison with other methods in *singer2* seq.

Figure 6. Tracking results when there are **severe illumination changes** and **pose variations**.



(a) Tracking results of our method in *soccer* seq.

(b) Comparison with other methods in *soccer* seq.

(c) Tracking results of our method in *skating2* seq.

(d) Comparison with other methods in *skating2* seq.

Figure 7. Tracking results when there are **severe occlusions** and **pose variations**.

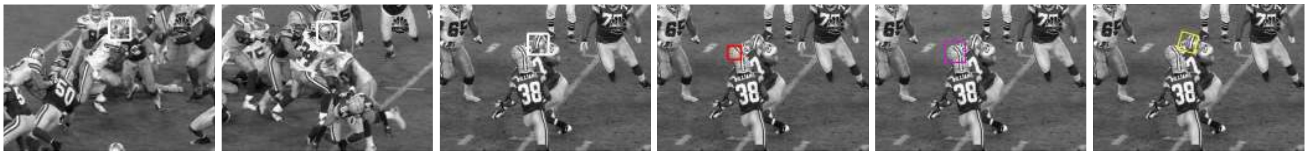
cluded, non-occluded appearance, or mixture of them. And it took charge of a different degree of occlusion. On the other hand, other methods failed to track the object accurately as depicted in Fig. 7(b)(d).

Background clutters: In Fig. 8, we tested *football* seq. that includes severe background clutter, of which appearance is similar to that of the target. In the case of other tracking methods, a trajectory was hijacked by the other football player wearing a similar helmet to the target when two players collided with each other at frame #361 as illustrated in Fig. 8(b). Our method overcame this problem and successfully tracked the target in Fig. 8(a).

Abrupt motion and illumination change: Fig. 9(a) illustrates our tracking results of *tiger1* seq. While the sequence contains abrupt motions as well as illumination changes, our method did not miss the object in all frames. For more tests, we made original videos of *singer1* and *skating1* to have partially low frame rate. In converted videos, the position and scale of an object are drastically changed. At the same time, severe illumination changes translate the appearance of the object into different one. As shown in Fig. 9(c)(e), our method covered these changes and reliably tracked the object. However the other methods failed to track the object as described in Fig. 9(b)(d)(f). Note that WLMC [11] and OIF [12] are the most recent state-of-the-art tracking methods that can cope with abrupt motions and appearance changes, respectively. We used software of authors for WLMC and OIF.

7. Conclusion

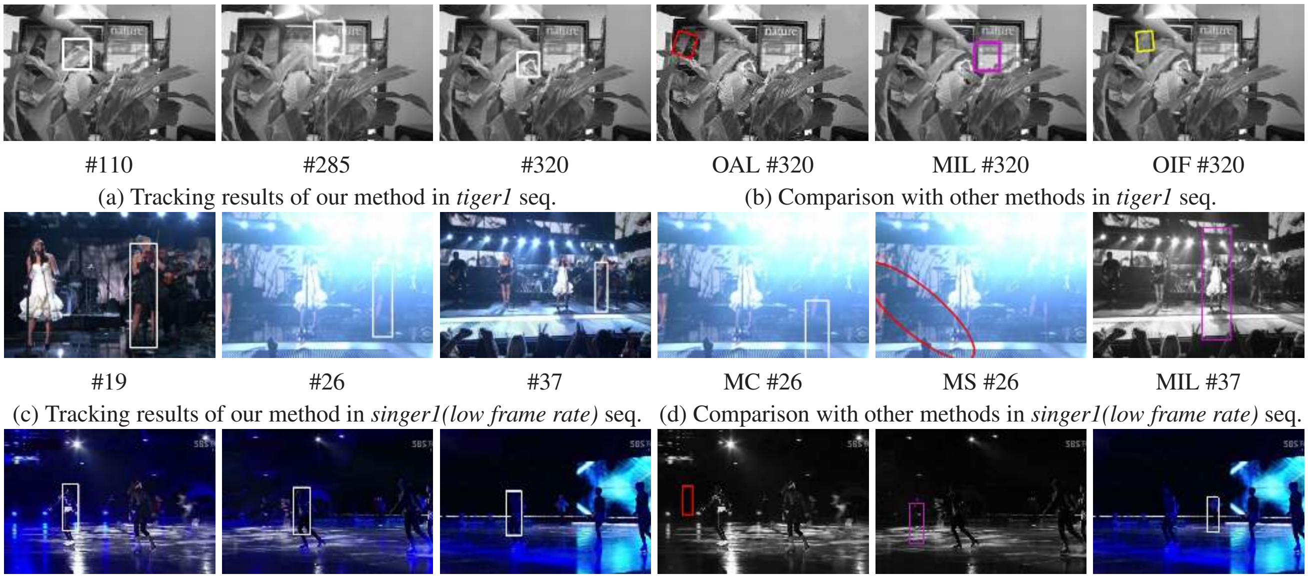
In this paper, we proposed an effective tracking algorithm with the visual tracking decomposition scheme. The algorithm efficiently addresses the tracking of an object whose motion and appearance change drastically and combinatorially. The experimental results demonstrated that the proposed method outperformed conventional tracking algorithms in severe tracking environments. Since our decomposition scheme is easy to extend by adding new features or trackers, the tracking results could be improved further.



#177 #158 #361 OAL #361 MIL #361 OIF #361

(a) Tracking results of our method in *football* seq. (b) Comparison with other methods in *football* seq.

Figure 8. Tracking results when there is **severe background clutter**.



#110 #285 #320 OAL #320 MIL #320 OIF #320

(a) Tracking results of our method in *tiger1* seq. (b) Comparison with other methods in *tiger1* seq.

#19 #26 #37 MC #26 MS #26 MIL #37

(c) Tracking results of our method in *singer1(low frame rate)* seq. (d) Comparison with other methods in *singer1(low frame rate)* seq.

#225 #235 #245 OAL #225 MIL #235 WLMC #245

(e) Tracking results of our method in *skating1(low frame rate)* seq. (f) Comparison with other methods in *skating1(low frame rate)* seq.

Figure 9. Tracking results when there are **abrupt motions** and **severe illumination changes**.

References

[1] B. Babenko, M. Yang, and S. Belongie. Visual tracking with online multiple instance learning. *CVPR*, 2009.

[2] D. Comaniciu, V. Ramesh, and P. Meer. Real-time tracking of non-rigid objects using mean shift. *CVPR*, 2000.

[3] J. Corander, M. Ekdahl, and T. Koski. Parallell interacting MCMC for learning of topologies of graphical models. *Data Min. Knowl. Discov.*, 17(3), 2008.

[4] A. d'Aspremont, L. El Ghaoui, M. Jordan, and G. Lanckriet. A direct formulation for sparse PCA using semidefinite programming. *SIAM Review*, 49(3), 2007.

[5] W. Du and J. Piater. A probabilistic approach to integrating multiple cues in visual tracking. *ECCV*, 2008.

[6] B. Han, S. Joo, and L. S. Davis. Probabilistic fusion tracking using mixture kernel-based bayesian filtering. *ICCV*, 2007.

[7] M. Isard and A. Blake. Icondensation: Unifying low-level and high-level tracking in a stochastic framework. *ECCV*, 1998.

[8] A. D. Jepson, D. J. Fleet, and T. F. E. Maraghi. Robust online appearance models for visual tracking. *PAMI*, 25(10):1296–1311, 2003.

[9] Z. Khan, T. Balch, and F. Dellaert. MCMC-based particle filtering for tracking a variable number of interacting targets. *PAMI*, 27(11):1805–1918, 2005.

[10] P. Kovesi. Image features from phase congruency. *A Journal of Computer Vision Research*, 1(3), 1999.

[11] J. Kwon and K. M. Lee. Tracking of abrupt motion using wang-landau monte carlo estimation. *ECCV*, 2008.

[12] J. Kwon, K. M. Lee, and F. C. Park. Visual tracking via geometric particle filtering on the affine group with optimal importance functions. *CVPR*, 2009.

[13] B. Leibe, K. Schindler, N. Cornelis, and L. Van Gool. Coupled object detection and tracking from static cameras and moving vehicles. *PAMI*, 30(10):1683–1698, 2008.

[14] Y. Li, H. Ai, T. Yamashita, S. Lao, and M. Kawade. Tracking in low frame rate video: A cascade particle filter with discriminative observers of different lifespans. *CVPR*, 2007.

[15] H. Ling and K. Okada. Diffusion distance for histogram comparison. *CVPR*, 2006.

[16] P. Perez, C. Hue, J. Vermaak, and M. Gangnet. Color-based probabilistic tracking. *ECCV*, 2002.

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

[18] B. Stenger, T. Woodley, and R. Cipolla. Learning to track with multiple observers. *CVPR*, 2009.

[19] A. Yilmaz, O. Javed, and M. Shah. Object tracking: A survey. *ACM Comput. Surv.*, 38(4), 2006.

[20] T. Zhao and R. Nevatia. Tracking multiple humans in crowded environment. *CVPR*, 2004.