A Structured Learning-Based Graph Matching Method For Tracking Dynamic Multiple Objects

Hongkai Xiong[1], *Senior Member, IEEE,* Dayu Zheng[1], Qingxiang Zhu[1], Botao Wang[1],
Yuan F. Zheng[2], *Fellow, IEEE,*
[1]Department of Electronic Engineering, Shanghai Jiao Tong University, Shanghai 200240, China
[2]Department of Electrical and Computer Engineering, The Ohio State University, Columbus, OH 43210, USA

***Abstract*—Detecting multiple targets and obtaining a record of trajectories of identical targets which interact mutually, infer countless applications in a large number of fields. It however presents a significant challenge to the technology of object tracking. This paper describes a novel structured learning-based graph matching approach to track a variable number of interacting objects in complicated environments. Different from previous approaches, the proposed method takes full advantage of neighboring relationships as the edge feature in structured graph, which performs better than using the node feature only. Therefore, a structured graph matching model is established, and the problem is regarded as structured node and edge matching between graphs generated from successive frames. In essence, it is formulated as the maximum weighted bipartite matching problem to be solved using the dynamic Hungarian algorithm, which is applicable of optimally solving the assignment problem in situations with changing edge costs or weights. In the proposed graph matching model, the parameters of the structured graph matching model are determined in a stochastic learning process. In order to improve the tracking performance, the bilateral tracking is also used. Finally, extensive experimental results on dynamic cell, football, and car sequences demonstrate that the new approach deals with complicated target interactions effectively.**

***Index Terms*—Multiple object tracking, structure feature, learning-based graph matching, dynamic environments, dynamic Hungarian algorithm**

I. Introduction

This work is concerned with the problem of multi-object tracking in dynamic environments, which is an active research field in recent years. It is significantly more challenging to track multiple targets than a single object, and even more difficult to obtain a record of trajectories of multiple identical targets, which however has many applications in reality. This paper hence deals with the tracking of multiple targets which have complicated interactions. We call those types of objects dynamic objects (targets) in this paper.

A. Problem Description

The difficulty of tracking multiple dynamic objects grows considerably with increasing density of objects, while frequent dynamic interactions between the objects make the problem even more challenging. Research on video object tracking can be categorized into two major classes: object representation and localization, e.g. mean-shift tracking [1], and data association, and filtering, e.g. particle filtering [2]. Mean-shift tracking finds local minima of a similarity measure between the color histograms or kernel density estimates of the model and target image, and could be considered as local search with low computational cost and little information requirement on motion and structure factors. Particle filtering solves the object localization problem by sequentially estimating the state of objects using a sequence of noisy measurements about the object states. Unfortunately, particle filtering and its variants fail to deal with the interaction of the objects in structured environments [5] [6], where objects of interest are constrained in certain area, e.g. street and football pitch. Consequently, there has been active research on learning based methods for analyzing and understanding behavior prediction in videos [3]. Through the observations, this paper is motivated to handle complicated interactions of targets in dynamic environments, which could involve the occurrences of entering, exiting, splitting, and touching of objects as shown in Fig. 1.

(a) Cells (b) Cars (c) Football match

Fig. 1: Illustrations of object tracking difficulties in different scenes where object entering, exiting, touching, and splitting could occur.

B. The Proposed Approach

In this paper, we propose a structured learning-based graph matching method for dynamic multiple object tracking. The chart of the proposed method is illustrated in Fig. 2, where the cell sequences are used for describing the structured learning-based graph matching method. The contributions of the proposed method are primarily in three aspects.

The first contribution of the paper is to include structure features for tracking a variable number of interacting objects in complicated dynamic environments. The proposed structure features involve neighboring relationships including the lengths and angles of the edges in the structured graph,

The work was supported in part by the NSFC, under grants No. 60632040, No. 60772099, and No. 60928003.

Hongkai Xiong, Dayu Zheng, Qingxiang Zhu and Botao Wang are with the Department of Electronic Engineering, Shanghai Jiao Tong University, Shanghai 200240, China, xionghongkai@sjtu.edu.cn, (Tel):86-21-34204515.

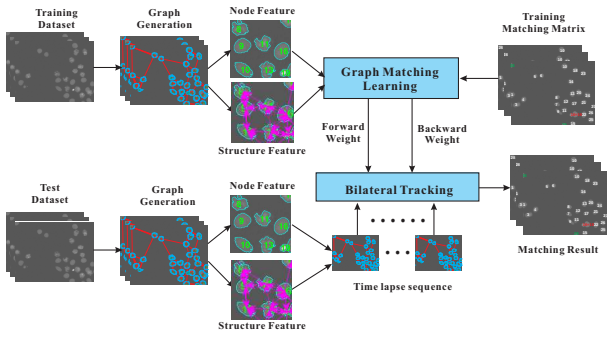


Fig. 2: The proposed multi-object tracking method

and represent nonlocal structure information of the whole graph. Most previous approaches [4] [7] using graph matching consider only the features of the objects, but not take into account the structure features of the graphs. As a result, they cannot obtain good performance for tracking multiple dynamic targets. The proposed method pays more attention to the structure relationships among multiple objects, and the structure information of the graphs is considered as the features of graphs, which can improve the performance of object tracking significantly, especially in complicated situations.

The second contribution is to establish a structured learning-based graph matching model. The new model replaces the generic graph matching cost with a novel structured graph matching cost to incorporate the structured factor. Unlike previous graph matching approaches [8] which only exploit the local geometrical features to derive a similarity measure, it makes full use of the whole graph topology to obtain mutual correspondences between components of the graphs. By doing so, we are able to generate an undirected graph whose nodes represent the targets to be tracked. The proposed structured graph matching problem maximizes the matching cost of subgraphs that consist of structured nodes and edges, where two graphs are considered isomorphic only if the correspondence between their nodes pairs up nodes with equal labels. In essence, it is formulated as the maximum weighted bipartite matching problem, which is to be solved by the dynamic Hungarian algorithm. Because it is applicable of optimally solving the assignment problem in situations with changing edge costs or weights, which are most common in the subsequent image pair of this problem. The method based on the dynamic Hungarian algorithm can effectively reduce computational complexity when the number of objects is large, and bilateral tracking is utilized in order to obtain even better performance.

The third and final contribution is to avoid the manual parameter selection process which is time-consuming. The parameters of the model can be acquired in a stochastic learning process. In previous works [1] [4], the graph matching model has to be chosen in terms of a large amount of experiments, and different models are manually established to achieve good performance on different objects according to their inherent characteristics, which we use the proposed structured graph matching model for an approximation. When the dimension of parameters is large, it is difficult to adjust them manually. Therefore, the learning step is proposed for the multi-object tracking problem. Consequently the structured graph matching model can be adaptive for tracking different kinds of objects, by adjusting its own parameters automatically.

The rest of the paper is organized as follows. Section II provides a brief review of the previous related work. Section III describes the graph model and the proposed structure features. In section IV, a structured learning based graph matching algorithm is presented for tracking multiple dynamic objects. Section V provides experimental results, and we conclude the paper in Section VI.

II. Related Work

To track dynamic objects in the image sequences and videos, the model of the object has to be established first. A number of techniques have been developed to address this problem including (1) movement analysis, (2) dynamic state theory, (3) kernel-based tracking, and (4) graph matching.

A. Movement Analysis

The methods based on moving analysis can effectively track moving objects from stationary cameras, e.g. background subtraction and optical flow method. Elgammal et al. [9] present a non-parametric background model and related subtraction approach, which can handle situations where the background of the scene is cluttered and not completely static but contains small motions such as tree branches and bushes. However, it is not a concise enough representation for the long term model of the scene by estimating the required sample size for each pixel in the scene depending on the variations of the pixel. Unal et al. [10] consider the addition of a prediction step to active contour-based visual tracking using an optical flow, and clarify the local computation along the boundaries of continuous active contours with appropriate regularizers. Nevertheless, the target objects are more complicated, which cannot be approximated by a polygon.

B. Dynamic State Theory

The methods based on dynamic state theory, e.g. particle and Kalman filtering, model object states and related observations in tracking. Jin et al. [11] propose an edge-based multi-object tracking framework which tracks multiple objects with occlusions using a variational particle filter. The method can avoid complicated object shape model assumptions in the optical flow method. An object is modeled by a mixture of a non-parametric contour model and a non-parametric edge model using kernel density estimation. However, the particle filtering method needs a large number of samples in the process of tracking, which is difficult to satisfy in some cases.

Chen et al. [15] introduce an HMM model for contour detection based on multiple visual cues in the spatial domain and improve it by joint probabilistic matching to reduce background clutters. Instead of assuming the one-to-one mapping between observations and targets in traditional multiple hypothesis trackers [12], Markov Chain Monte Carlo (MCMC) based sequential tracking methods [13] allow multiple temporal associations between observations and targets, and simulate the distribution of the association probability with a number of targets [14]. However, a prohibitively large number of

samples would be required to approximate the underlying density functions with desired accuracy.

It is further integrated with an unscented Kalman filter to exploit object dynamics in nonlinear systems for robust contour tracking [16]. However, being applied to a high-dimensional state space, a prohibitively large number of samples may be required to approximate the underlying density functions with desired accuracy, and Kalman filters become quite inefficient.

C. Kernel-based Tracking

The methods based on kernel-based tracking represent local object characteristics, e.g. mean shift and generalized kernel-based tracking [4] etc. Comaniciu et al. [1] proposed a new approach toward target representation and localization, with the central component for visual tracking of non-rigid objects. The feature histogram based target representations are regularized by spatial masking with an isotropic kernel. The method can overcome the shortcoming of particle filtering in terms of computational load, however, the performance cannot be improved when the feature dimension increases.

D. Graph Matching

The methods based on graph matching [17] solve the global optimization problem of target matching, for instance region matching and feature matching etc. Using abstractive representations for complex scenes, attributed graph matching problems could be formulated to find the close-to-optimum solution [18] where two graphs are considered isomorphic only if the correspondence between their node pairs up nodes with equal labels. Chen et al. [19] presented a tracking algorithm to address the interactions among objects, and to track them individually and confidently via a static camera. It is achieved by constructing an invariant bipartite graph to model the dynamics of the tracking process, of which the nodes are classified into objects and profiles. Pallavi et al. [20] proposed a graph-based approach for detecting and tracking multiple players in broadcast soccer videos. In the method, a directed weighted graph is constructed, where probable player candidates correspond to the nodes of the graph while each edge links candidates in a frame with the candidates in the next two consecutive frames. The method can gain good performance on multi-object tracking. However, applying it to tracking other kinds of objects, the tracking model has to be modified through a large number of experiments, which is inconvenient in applications, e.g. biological and medical imaging. In [8], a cell tracking approach exploited the local geometrical and topological features of cells to generate graphs, where a seed cell pair as a starting point between local regions in the graph is progressively moving outwards to obtain correspondences of neighboring cells. However, the correspondence of successive graphs is a local estimation, which cannot achieve good performance without the global topology information.

To avoid the time-consuming manual labeling of correspondences, active research on learning based methods for analyzing and understanding tracking model from videos has recently been pursued [21], including both supervised and unsupervised learning methods. By clustering both similarity and comparison confidence, Wang et al. [42] provided a comparison confidence measure to indicate the approximation between the measured image-based similarity and the physical similarity. Hospedales et al. [43] developed an approximation to online Bayesian inference which is in favor of dynamic scene understanding and behavior mining in video data. With the insight of learning-based graph matching, Caetano et al. [22] integrated the structural quadratic compatibilities on mutual association (labeled as 0 or 1) and local compatibilities on point pattern matching into the objective function to find the optimal assignment in a dynamic behavioral model. A method for learning the activity patterns from video was proposed in [23], which extracts a large set of object motion patterns from videos over extended periods of time. This method utilizes a codebook of activity patterns in terms of an online vector quantization on the whole set of acquired motion patterns. However, little structure information on motion prediction is provided, which is useful for validating the efficiency and improving the performance of object tracking.

III. Structure Feature

In this section, the concept of graph is defined for the proposed algorithm in section III-A. In order to make full use of neighboring relationships in the whole structured graph, the node and structure features are defined in section III-B.

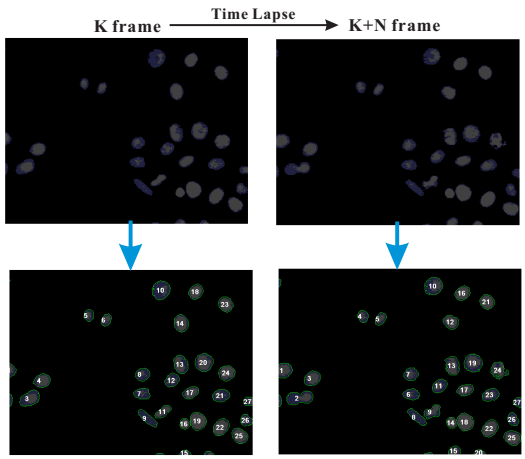


Fig. 3: Original indices of cells.

A. Graph Definition

In this sub-section, we first generate the conception of an undirected graph, in which the target objects are the nodes.

Definition 1 (Graph Definition). *In the sequence of images, graphs are generated from two consecutive images which are denoted by G and G', respectively. Each graph is a complete set of all the nodes and edges (i.e., $G := (V, E)$) residing in the target image. Each segmented object is defined as a node in the graph. As shown in Fig. 3, node set that generated from G with k elements can be expressed by:*

$$V = \{v_1, v_2, v_3, \cdots, v_k\} \tag{1}$$

Basic features for each separate object are recorded by a node, including the features of its spatial coordinate, color,

shape and so on. These features are utilized in nodes matching and parameters learning. The relationships between the structured nodes are described as the neighborhood, which represent the structure of the graph.

Definition 2 (Neighborhood Definition). *Let V denotes the set of all nodes of a graph. $\forall v_i \in V$, the neighborhood of v_i is defined as*

$$N(v_i) = \{v_i^1, v_i^2, v_i^3, \cdots, v_i^m\} \tag{2}$$

such that

$$\forall v' \in N(v_i), \forall v'' \in V - N(v_i), |v_i - v'| \leq |v_i - v''| \tag{3}$$

where m is the number of neighbors.

The neighborhood of a node is defined as its m nearest neighbors, where m is called the degree of the node. An edge will be constructed between every two spatially neighboring nodes, and the edge set inside graph G is denoted by E. Another possible criterion to define neighborhood is with respect to distance. However, it may lead to unbalanced neighborhood, namely, nodes in high density area would have much more neighbors than those in low density area. To avoid the unbalanced density occurrence, each node is endowed with the same degree to generate regular graphs. All the nodes generate either one connected graph or several connected graphs, which does not affect the deviation in the following sections. Fig. 4 shows a graph of the cell sequence, where the degree of nodes is set to 3, and the arrows are from neighboring objects to target objects.

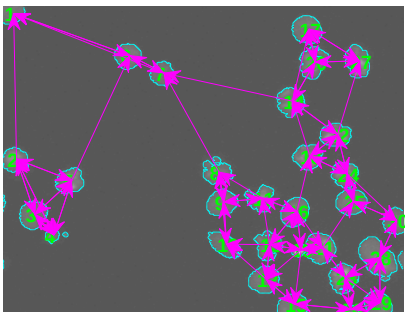


Fig. 4: Neighborhood and degrees of nodes in cell graph.

B. Structure Feature

In the proposed method, a node $v_i \in G_n$ has three kinds of features: spatial feature $Spat(v_i)$, gray level feature $Gray(v_i)$, and shape feature $Shape(v_i)$, which are defined as follows.

Definition 3 (Node Feature).

$$Spat(v_i) = (x_i, y_i) \tag{4}$$

where (x_i, y_i) is the coordinate of the node v_i.

$$Gray(v_i) = (g_{aver}(v_i),\ g_{max}(v_i),\ g_{min}(v_i),\ g_{var}(v_i)) \tag{5}$$

where $g_{aver}(\cdot)$, $g_{max}(\cdot)$, $g_{min}(\cdot)$ and $g_{var}(\cdot)$ fetch the average, maximum, minimum, and variance gray level of the objects, respectively.

$$Shape(v_i) = (s_{size}(v_i),\ s_{aver_r}(v_i),\ s_{var_r}(v_i),\ s_{comp}(v_i)) \tag{6}$$

where $s_{size}(\cdot)$, $s_{aver_r}(\cdot)$, $s_{var_r}(\cdot)$ and $s_{comp}(\cdot)$ are the area, average radius, variance radius and compactness of the objects.

To join the three kinds of node features together, the node features of objects are formulated as

$$f_n(v_i) = (Spat(v_i), Gray(v_i), Shape(v_i)) \tag{7}$$

where $f_n(\cdot)$ is the node features of the node v_i.

Since the exponential decay can achieve good performance, the node matching cost between the node pair $\{v_i,\ v_{i'}\}$, where $v_i \in G,\ v_{i'} \in G'$, is defined as

Definition 4 (Node Matching Cost).

$$F_c(v_i, v_{i'}) = \exp\left\{-|f_n(v_i) - f_n(v_{i'})|^2\right\} \tag{8}$$

To make use of the neighboring relationships in the structured graphs, the structures in the graph should be taken into account. The structure reflects the relationship between the object and its neighborhood, which is demonstrated in Fig. 5. Obviously, the probability of correct matching would be much higher when the object and its neighborhood can simultaneously satisfy. Hence, we define the edge feature in the proposed algorithm. As a related work, [22] utilized a coarse edge representation where 1 or 0 demonstrates whether the edge exists or not. It takes advantage of the nodes and the connection of them in the edge matching process; however, the attributes of edges are ignored, e.g. the length and angle. The structure features are derived from edge features which are the relationships between the adjacent nodes, including the cost and angle of the edges between the object and its neighborhood. In this way, the edge feature between node pair $\{v_i^\alpha, v_i\}$ to $\{v_{i'}^\beta, v_{i'}\}$ is calculated by the following definition:

Definition 5 (Edge Feature).

$$Dist(v_i, v_j) = \left((x_i - x_j)^2 + (y_i - y_j)^2\right)^{\frac{1}{2}} \tag{9}$$

$$Arg(v_i, v_j) = \arctan\frac{(y_j - y_i)}{(x_j - x_i)} \tag{10}$$

where (x_i, y_i) and (x_j, y_j) are the coordinates of the node v_i and node v_j, $Dist(v_i, v_j)$ is the distance between node v_i and v_j, and $Dist(v_i, v_j)$ is the angle of edge e_{ij}

Definition 6 (Edge Matching Cost).

$$\begin{aligned} & f_e(v_i^\alpha, v_i, v_{i'}^\beta, v_{i'}) \\ = \ & F_c(v_i^\alpha, v_{i'}^\beta) \times \exp\left\{-\frac{|Dist(v_i^\alpha v_i) - Dist(v_{i'}^\beta v_{i'})|}{dist}\right\} \\ & \times \exp\left\{-\frac{|Arg(v_i^\alpha v_i) - Arg(v_{i'}^\beta v_{i'})|}{\theta}\right\} \end{aligned} \tag{11}$$

where $v_i^\alpha \in N(v_i)$, $v_{i'}^\beta \in N(v_{i'})$, the function $f_e(\cdot)$ is the edge matching cost of the graph, including node features (spatial feature, gray level feature, and shape feature) and edge features ($Dist(\cdot)$ and $Arg(\cdot)$).

The edge features are kept relevant to not only the node features of the object's neighborhood but also the angles and lengths of two matching edges. It is noted that the angular

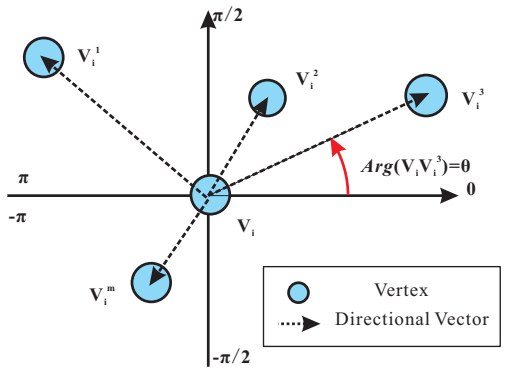


Fig. 5: The subgraph of the node and its neighborhood.

factor considers the degrees between two edges $e(v_i^{\alpha}v_i)$ and $e(v_{i'}^{\beta}v_{i'})$. Fig. 4 shows the edge features in cell tracking.

The proposed algorithm adopts both node and subgraph matching, where a subgraph consists of one node and its neighborhood as Fig. 5. For several edges in a subgraph, the total structure feature is defined as.

Definition 7 (Structure Matching Cost)**.**

$$F_e(v_i, v_{i'}) = \sum_{v_i^{\alpha_k} \in N(v_i), v_{i'}^{\beta_k} \in N(v_{i'})} f_e(v_i^{\alpha_k}, v_i, v_{i'}^{\beta_k}, v_{i'}) \quad (12)$$

where $v_i^{\alpha_k}$ *and* $v_{i'}^{\beta_k}$ *are the neighborhood of the object* v_i *and* $v_{i'}$*, respectively, i.e.* $v_i^{\alpha_k} \in N(v_i)$ *and* $v_{i'}^{\beta_k} \in N(v_{i'})$*,* $k = 1, 2, \cdots, m$*.*

To combine the node matching and the structure matching cost, we obtain the final matching cost on node v_i and node $v_{i'}$.

$$F(v_i, v_{i'}) = (F_c(v_i, v_{i'}), F_e(v_i, v_{i'})) \quad (13)$$

Once the features of the node and its neighbors are attained, the structured learning graph matching approach for multi-object tracking is presented in the next section.

IV. The Proposed Structured Learning-based Graph Matching Method

In order to track multiple dynamic objects, we consider a structured learning graph matching algorithm. Since graph matching models are variable for different targets, an adaptive graph matching model is required to ensure a robust response. For this purpose, the proposed algorithm is learning-based to adjust the coefficients by itself in a stochastic learning process and avoid manual interference. Moreover, the dynamic Hungarian algorithm is utilized to reduce computational complexity as well as bilateral tracking to improve the tracking performance in dynamic environments.

A. Structured Graph Matching Model

Initially, we define the structured graph matching model and denote the notations in the model. Given a pair of graphs G and G', v_i is denoted as the i^{th} attribute of the node and e_{ij} as the edge ij in graph G. In a standard graph, the edge attributes $e_{ij} \in \{0, 1\}$ are binary.

In a matching matrix y of the structured graph matching problem, $y_{ii'} \in \{0, 1\}$, $y_{ii'} = 1$ if node i of G matches node i' of G', and $y_{ii'} = 0$ otherwise. $c_{ii'}$ is defined as the coefficient of the compatibility function for linear assignment $(i \rightarrow i')$, and $d_{ii'jj'}$ is defined as the coefficient of the compatibility function for quadratic assignment $(ij \rightarrow i'j')$. The structured graph matching problem is formulated in a typical way, namely, the matching solution $\hat{y}$ is given by a quadratic assignment problem:

$$\begin{aligned} \hat{y} &= \arg\max_{y} \sum_{ii'} c_{ii'} y_{ii'} + \sum_{ii'jj'} d_{ii'jj'} y_{ii'} y_{jj'} \\ &\text{s.t.} \begin{cases} \sum_{i} y_{ii'} \leq 1, \text{for all } i' \\ \sum_{i'} y_{ii'} \leq 1, \text{for all } i \end{cases} \end{aligned} \quad (14)$$

When only node matching is considered, Eq. (14) of the structure graph matching model is simplified as a linear assignment. It is a quadratic assignment when one attempts to match edges. The quadratic assignment problem (QAP) is a NP-hard problem, where the coefficients of $c_{ii'}$ and $d_{ii'jj'}$ depend on the node feature $(v_i, v'_{i'})$ and the edge feature $(e_{ij}, e'_{i'j'})$, respectively. Most previous work neglects the coefficients of $d_{ii'jj'}$ to reduce the computational complexity, so that it is degraded into a linear assignment problem of $o(n^3)$. It is unable to achieve the optimal performance without the structure features of the graph. It is worth mentioning that edges exist only between the objects and their neighbors, which can be expressed as:

$$\begin{cases} d_{ii'jj'} \neq 0, \ v_j \in N(v_i) \text{ and } v_{j'} \in N(v_{i'}) \\ d_{ii'jj'} = 0, \ \text{otherwise} \end{cases} \quad (15)$$

Hence, the matching cost in Eq. (14) can be changed into

$$\begin{aligned} &\sum_{ii'} c_{ii'} y_{ii'} + \sum_{ii'jj'} d_{ii'jj'} y_{ii'} y_{jj'} \\ &= \sum_{ii'} y_{ii'} (c_{ii'} + \sum_{jj'} d_{ii'jj'} y_{jj'}) \\ &= \sum_{ii'} y_{ii'} (\underbrace{c_{ii'}}_{\text{node matching cost}} + \underbrace{\sum_{\substack{v_j \in N(v_i) \\ v_{j'} \in N(v_{i'})}} d_{ii'jj'} y_{jj'}}_{\text{structure matching cost}}) \end{aligned} \quad (16)$$

which involves both node and edge features.

Since $y_{jj'} \in \{0, 1\}$, $y_{jj'} = 1$ if node j of G matches node j' of G', and $y_{jj'} = 0$ otherwise. According to $v_j \in N(v_i)$ and $v_{j'} \in N(v_{i'})$, node v_j and v'_j are, respectively, in the subgraph of node v_i and $v_{i'}$. Therefore, the proposed structured graph matching model could make use of the edge feature of the neighborhood $N(v_i)$ instead of each neighborhood node

$$\sum_{ii'} c_{ii'} y_{ii'} + \sum_{ii'jj'} d_{ii'jj'} y_{ii'} y_{jj'} = \sum_{ii'} y_{ii'} (c_{ii'} + d'_{ii'}) \quad (17)$$

where $d'_{ii'}$ is the matching cost of the edge feature between the subgraphs of nodes i and i'.

B. Structured Learning-based Graph Matching

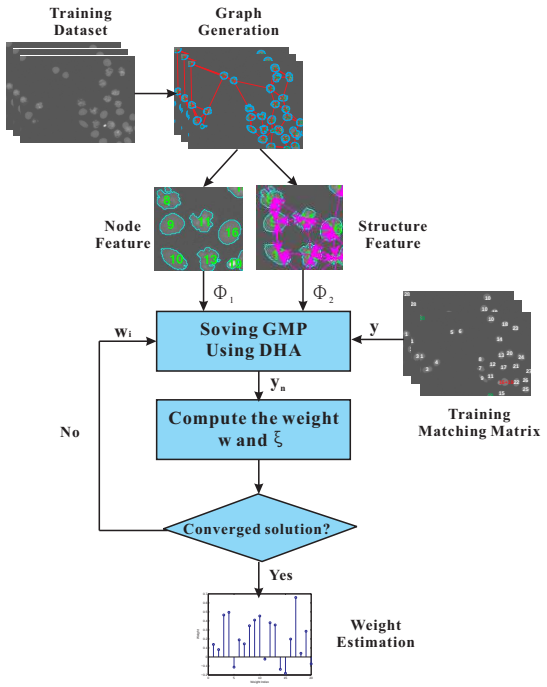


Fig. 6: A diagram of the structured learning-based graph matching algorithm.

The structured learning-based graph matching algorithm is demonstrated in Fig. 6. Training datasets are N observations x from an input set $\mathcal{X}$, and N corresponding labels from an output set $\mathcal{Y}$, which compose the structured training pairs of dataset $\{(x^1, y^1), (x^2, y^2), \ldots, (x^N, y^N)\}$. x^n is an observation of graphs G^n and G'^n, including the node feature and edge feature, and y^n is the matching matrix between G^n and G'^n, as defined in the previous section. $\mathcal{G}$ is the space of graph, $\mathcal{X} = \mathcal{G} \times \mathcal{G}$ is the space of graph pairs, and $\mathcal{Y}$ is the space of matching matrices. The learning task of the structured graph matching model would find a parameterized function of the graph matching model $g_\omega : \mathcal{G} \times \mathcal{G} \to \mathcal{Y}$, which minimizes the matching cost on the test dataset. Hence, we define a graph matching loss which consists of the empirical risk and a regularization term. The empirical risk is the average loss in the training dataset, and the regularization term is introduced to avoid learning overfitting.

$$\underbrace{\frac{1}{N}\sum_{n=1}^{N}\Delta(g_\omega(G^n, G'^n), y^n)}_{\text{empirical risk}} + \underbrace{\lambda\Omega(\omega)}_{\text{regularization term}} \tag{18}$$

where $\Delta(g_\omega(G^n, G'^n), y^n)$ is the loss incurred by the predictor g when predicting. The output $g_\omega(G^n, G'^n)$, as the prediction of the matching matrix y^n, is used instead of y^n. The term $\Omega(\omega)$ is a regularization function of ω and λ a parameter in the loss, which are used against overfitting in the training dataset. In $g_w(G, G')$, the parameter ω would be optimized over a loss function Δ in Eq. (23) and the regularization term $\Omega(\omega)$. To specify the function $g_\omega(G, G')$, we use the standard approach of discriminant functions. The discriminant function $f(G, G', y; \omega)$ is maximal for the case of $g_w(G, G')$, which is the optimal estimate for y (i.e., $g_\omega(G, G') = \arg\max f(G, G', y; \omega)$). In a typical way, we define $f(G, G', y; \omega)$ as linear functions, $f(G, G', y; \omega) = \langle \omega, \Phi(G, G', y)\rangle$, where $\Phi(G, G', y)$ is the discriminant function of object features (node and edge features). Correspondingly, the predictor $g_w(G, G')$ is formulated as:

$$g_\omega(G, G') = \arg\max_{y \in \mathcal{Y}} \langle \omega, \Phi(G, G', y)\rangle \tag{19}$$

Furthermore, the joint feature of graph pairs is required to be defined to contain the properties of both graphs and a matching matrix y between graphs. To achieve it, we can find the relationship between the learning scheme given by Eq. (19) and the structured graph matching model given by Eq. (17). The solution of the optimization problem is the estimate of function g, i.e., $y^\omega = g_\omega(G, G')$. The discriminant function in Eq. (19) is introduced into Eq. (17) to yield:

$$\langle \Phi(G, G', y), \omega\rangle = \sum_{ii'} y_{ii'}(c_{ii'} + d'_{ii'}) \tag{20}$$

The graphs and the parameters must be represented in the compatibility functions. As the way $f(G, G', y; \omega)$ is defined, we choose the coefficients of the compatibility functions as:

$$\begin{aligned} c_{ii'} &= \langle F_c(v_i, v'_{i'}), \omega_1\rangle \\ d'_{ii'} &= \langle F_e(v_i, v'_{i'}), \omega_2\rangle \end{aligned} \tag{21}$$

where $F_c(v_i, v'_{i'})$ represents the node matching cost of node pairs $(v_i, v'_{i'})$. $F_e(v_i, v'_{i'})$ is the edge matching cost in the subgraph pairs of the nodes v_i and $v'_{i'}$, which can be defined in details in the experiment. v_i and $v_{i'}$ are regarded as a potential candidate pair $\{v_i, v_{i'}\}$, $v_i \in G, v_{i'} \in G'$.

As an extreme case of Eq. (20), $c_{ii'}$ and $d_{ii'jj'}$ only relate with the features of node and edge of graphs by defining $\omega := [\omega_1\ \omega_2]$ where ω_1 and ω_2 are constants. In this way, we obtain the final form of $\Phi(G, G', y)$ from Eq. (20) and Eq. (21) as:

$$\Phi(G, G', y) = \left[\sum_{ii'} y_{ii'} F_c(v_i, v'_{i'}), \sum_{ii'} y_{ii'} F_e(v_i, v'_{i'})\right]. \tag{22}$$

In the loss function $\Delta(\hat{y}, y^n)$, $\hat{y}$ is the estimation of the matching matrix and y^n is the matching matrix of the n^{th} training set. Over the multi-object tracking, average error is defined as the fraction of mismatches between matrices $\hat{y}$ and y^n in Eq. (23), which is also called normalized Hamming loss:

$$\Delta(\hat{y}, y^n) = 1 - \frac{1}{\|y^n\|^2}\sum_{ii'} \hat{y}_{ii'} y^n_{ii'} \tag{23}$$

Finally, the regularization term $\Omega(\omega)$ is specified as $\frac{1}{2}\|\omega\|^2$.

To solve the problem, one approach to minimize (18) is to replace the empirical risk by a convex upper bound on the empirical risk. In our problem, the convex function $\frac{1}{N}\sum_n \epsilon_n$ is an upper bound for $\frac{1}{N}\sum_n \Delta(g_\omega(G^n, G'^n), y^n)$ with appropriate constraints. The optimization problem of structured learning-based graph matching becomes

$$\min_{\omega,\xi} \frac{1}{N}\sum_{n=1}^{N}\xi_n + \frac{\lambda}{2}\|\omega\|^2$$
$$\text{s.t. } \langle \omega, \Psi^n(y)\rangle \geq \Delta(y, y^n) - \xi_n, \forall n \text{ and } y \in \mathcal{Y}. \tag{24}$$

However, the constraints of Eq. (24) is of a large amount which is determined by the number of possible matching matrices $\|y\|$ times the number of training instances N. It is difficult to obtain the exact solution of the optimization problem. To overcome it, an optimization method known as column generation [24] is used to find the solution. To solve it, we might find the worst boundary and replace the formulation in Eq. (24) by an equivalent form, which has only single slack variable instead of n variables. Instead of directly solving Eq. (25), one could calculate the most violated constraint in Eq. (24) iteratively for the solution. The final optimization problem becomes:

$$\min_{\omega,\xi} \xi + \frac{\lambda}{2}\|\omega\|^2$$
$$\text{s.t. } \frac{1}{N}\sum_{n}\langle \omega, \Psi^n(y)\rangle \geq \frac{1}{N}\sum_{n}\Delta(y, y^n) - \xi_n, \tag{25}$$
$$\forall n \text{ and } y \in \mathcal{Y}$$

where $\Psi^n(y) = \Phi(G^n, G'^n, y^n) - \Phi(G^n, G'^n, y)$.

It is the tightest form of the constraint of Eq. (24), and the solution is obtained by:

$$\hat{y}_n = \arg\max_{y} \langle \omega, \Phi(G^n, G'^n, y)\rangle + \Delta(y, y^n) \tag{26}$$

In the next step, we introduce Eq. (26) into the problem of Eq. (25) and obtain

$$\frac{1}{N}\sum_{n}\Delta(\hat{y}_n, y^n) - \langle \omega, \Psi^n(\hat{y}_n)\rangle + \frac{\lambda}{2}\|\omega\|^2 \tag{27}$$

$$\xi = \Delta(\hat{y}_n, y^n) - \langle \omega, \Psi^n(\hat{y}_n)\rangle \tag{28}$$

whose gradient (with respect to ω) is

$$\lambda\omega - \frac{1}{N}\sum_{n}\Psi^n(\hat{y}_n) \tag{29}$$

Hence, we can thus obtain the final form with the loss function:

$$\langle \Phi(G, G', y), \omega\rangle + \Delta(y, y^n) = \sum_{ii'} y_{ii'}(c'_{ii'} + d'_{ii'}) + C \tag{30}$$

where C is a constant, $c'_{ii'} = \langle F_c(v_i, v'_{i'}), \omega_1\rangle + \frac{\|y^n_{ii'}\|}{\|y^n\|^2}$ and $d'_{ii'} = \langle F_e(v_i, v'_{i'}), \omega_2\rangle$.

It is equivalent to find the solution to the maximization of (30), which is a quadratic assignment problem. In the process of simplifying the structured learning-based graph matching model in terms of the constraints, the training and predicting optimization could be turned into linear assignment problems in a low complexity. Finally, we summarize the training process of the structured learning-based graph matching as shown in Algorithm 1, e.g. the learned weight ω for the cell tracking as shown in Fig. 7.

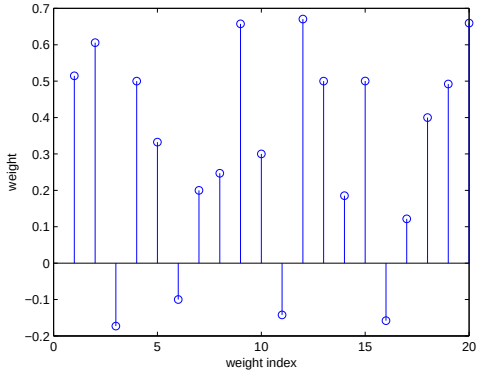


Fig. 7: The learned weight ω on cell tracking.

Define:
$\Psi^n(y) := \Phi(G^n, G'^n, y^n) - \Phi(G^n, G'^n, y)$
$H^n(y) := \langle \omega, \Phi(G^n, G'^n, y^n)\rangle + \Delta(y, y^n)$
Input:
training graph pairs $\{(G^n, G'^n)\}$,
training matching matrices $\{y^n\}$,
sample size N, tolerance ϵ
Initialize $i = 1$, $\omega = 0$
while $\xi < \epsilon$ **do**
  for $n = 1$ to N **do**
    $\hat{y}_n = \arg\max_{y\in(Y)} H^n(y)$
  end
  Compute $p_i = \lambda\omega - \frac{1}{N}\sum_n \Psi^n(\hat{y}_n)$
  Compute
  $q_i = \frac{1}{N}\sum_n \Delta(\hat{y}_n, y^n) - \langle \omega, \Psi^n(\hat{y}_n)\rangle + \frac{\lambda}{2}\|\omega\|^2$
  $\omega_{i+1} := \arg\min_\omega \frac{\lambda}{2}\|\omega\|^2 + \max_{j\leq i}\left(\langle p_j, \omega\rangle + q_j\right)$
  $i \leftarrow i + 1$
end

Algorithm 1: The learning process in the proposed algorithm

C. Dynamic Hungarian Algorithm

The structured learning-based graph matching problem is regarded as the maximum weighted bipartite matching problem, and we adopt a dynamic Hungarian algorithm [25] to solve it in situations with changing edge costs or weights. Initially, we describe the assignment problem of the basic Hungarian algorithm which is also called the Kuhn-Munkres algorithm. Given a graph pair $\{G, G'\}$ and a matrix F of the matching cost, it assigns dual variables α_i to each node v_i and dual variables $\beta_{i'}$ to each node $v_{i'}$, where $v_i \in G$ and $v_{i'} \in G'$. P is the set of edges on the selected augmenting path. The output y is the matching matrix of the assignment problem.

In the proposed model, we can get the matching results as

$$\hat{y} = \arg\max_{y} \sum_{ii'} y_{ii'}(c_{ii'} + d'_{ii'})$$
$$\text{s.t.} \begin{cases} \sum_{i} y_{ii'} \leq 1, \text{for all } i' \\ \sum_{i'} y_{ii'} \leq 1, \text{for all } i \end{cases} \tag{31}$$

Because there is a short instant between two consequent images in the sequence, it is possible to add or remove one or several nodes between the graphs of consequent image. The dynamic Hungarian algorithm can solve this problem with changing edge costs or weights effectively. It has a computational complexity of $o(kn^2)$, where k is the number of rows or columns of the cost matrix that have changed, instead of the traditional Hungarian algorithm's $o(n^3)$. Therefore, the dynamic Hungarian algorithm is more suitable for the graphs of large size. The adopted dynamic Hungarian algorithm is described as Algorithm 2, where $\{G, G'\}$ is defined as a graph pair and F is the matrix of the feature cost in Section III-B. F^* and y^* are, respectively, the feature cost matrix and the match solution of the previous assignment problem. Finally, the output y is the solution of the current assignment problem. y_{k-1} is the matching from the previous stage, and P is the set of edges on the selected augmenting path.

D. Bilateral Tracking

As in Fig. 8, bilateral tracking is considered in the proposed multi-object tracking algorithm. It involves both the forward and backward tracking to acquire the forward weight ω_f and backward weight ω_b, and the matching solution to obtain a smaller one from the forward and backward matching costs is selected.

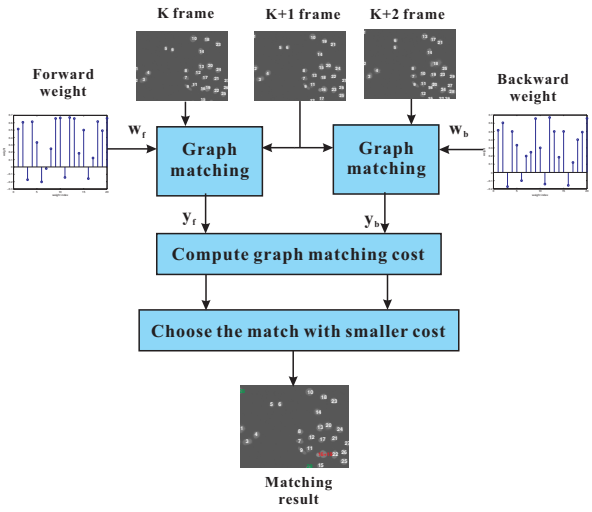


Fig. 8: The process of bilateral tracking

The bilateral tracking examples on cell images are shown in Fig. 9. It is evaluated between unilateral and bilateral tracking, where the error of cell tracking is labeled in red rectangular, the emerging cells are indexed in green numbers, and the errors of segmentation are indexed in red numbers. Fig. 9 describes the difference between the unilateral and bilateral tracking methods. In Fig. 9 (a), the cell indexed '16' cannot be tracked due to the missed segmentation in the previous frame where the two cells indexed '16' and '19' are overlapped. Blob indexed '16+19' actually consists of two cells. It can be distinguished from neither threshold based nor geometric method. However, the correct results can be attained by tracking from the next frame, because the backward tracking cost is much less than the forward one. In Fig. 9 (b), the cells indexed '16' and '19' can be easily tracked by the bilateral tracking. It can reduce the errors of object tracking, especially for disappeared and emerged objects.

Input: A bipartite graph, (G, G') and an matrix of feature cost F.
The feature cost matrix F^* and match solution y^* of the previous graph matching problem.
Output: A new matching matrix y of the new graph matching problem
1. Initialization:
if a row i^* of the cost matrix changed: **then**
  (a) Remove the edge $(v_{i^*}, mate(v_{i^*}))$ from the matching y^*
  (b) Assign $\alpha_{i^*} = min_{i'}(F_{i^*i'} - \beta_{i'})$
else
  if a column $i'^* X$ of the cost matrix changed: **then**
    (a) Remove the edge $(mate(v_{i'^*}), v_{i'^*})$ from the matching y^*
    (b) Assign $\beta_{i'^*} = min_i(F_{ii'^*} - \alpha_i)$
  end
end
2. Perform one iteration from the basic Hungarian algorithm
(a) Designate each exposed (unmatched) node in G as the root of a Hungarian tree.
(b) Grow the Hungarian trees rooted at the exposed nodes in the equality subgraph.
if an augmenting path is found **then**
  go to step (d).
else
  proceed to step (c).
end
(c) Modify the dual variables α and β as follows to add new edges to the equality subgraph. Then go to step (b) to continue the search for an augmenting path.
$\theta = \frac{1}{2} min_{i \in I^*, i' \notin I'^*}(F_{ii'} - \alpha_i - \beta_{i'})$
$\alpha_i \leftarrow \begin{cases} \alpha_i + \theta & i \in I^* \\ \alpha_i - \theta & i \notin I^* \end{cases}$
$\beta_{i'} \leftarrow \begin{cases} \beta_{i'} - \theta & i' \in I'^* \\ \beta_{i'} + \theta & i' \notin I'^* \end{cases}$
(d) Augment the current matching by flipping matched and unmatched edges along the selected augmenting path. The matching matrix at stage k,
$y_k = (y_{k-1} - P) \cup (P - y_{k-1})$
3. Output the resulting matching y.

Algorithm 2: The dynamic Hungarian algorithm in the graph matching problem

V. Experiments and Results

To validate the efficiency of the proposed method, we demonstrate extensive experimental results on medical cell sequences and standard multi-object sequences, e.g. football, car, and UBC hockey [44]. The experiments are performed in MATLAB (Version R2008a) on the computer with Intel Core2 Duo CPU E8400 @3.00GHz.

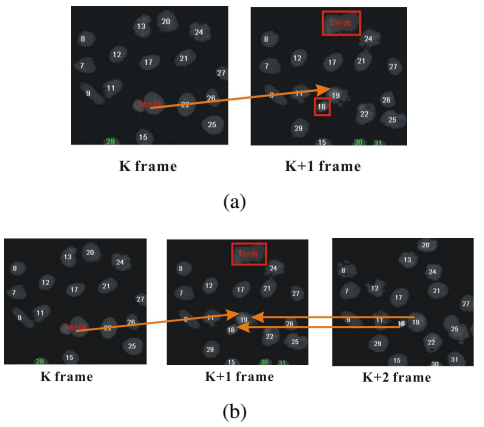


Fig. 9: The comparison between unilateral and bilateral tracking on cell images. (a) Unilateral tracking, (b) Bilateral tracking.

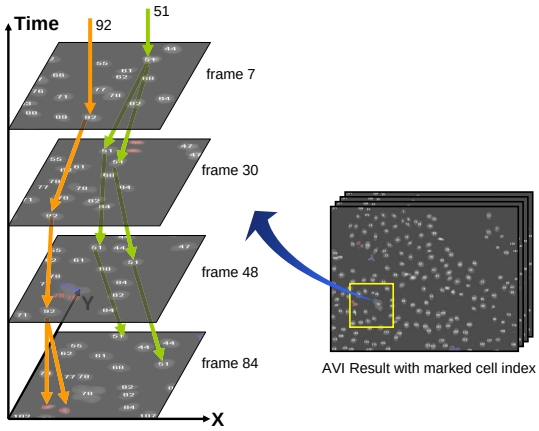


Fig. 10: Sampled image blocks from tracking results of cell sequences.

A. Medical Cell Sequence

In the first experiment, we consider the cell tracking problem [26] which is a challenging problem in biological and medical imaging. Initially, cell image sequences with distinct features are used to test efficiency of the proposed approach, which consist of 550 frames sampled from time-lapse fluorescence microscopy in the cell tracking experiment. We divide them into two parts, the training dataset of 50 frames and the test dataset of remaining frames. These image sources are all recorded in a spatial resolution of 617×512 pixels, and a temporal resolution of 3 minutes between two consecutive frames. Cells in the sequence are low in intensities but with much irregularity in shapes. Difficulty in processing the cell sequence lays in correct segmentations. Due to the irregular properties of shapes, usually more convoke points are found than expected, thus bringing the problems of over-segmentation. Sampled image blocks from the sequence are shown in Fig. 10. Cell indices are marked with different colors for discrimination of cells tracked correct, with error, and emerging in the sequence.

(a) Cell Segmentation

In most of cell tracking researches, accurate segmentation of original images is crucial for the subsequent tracking process.

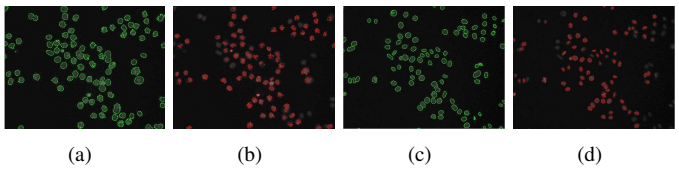


Fig. 11: Segmentation results comparison between the LBF level set and the Mumford-Shah functional based level set [28]. (a), (c) are the results of segmentation using the LBF level set, and (b), (d) are the results on the same test images using the Mumford-Shah functional based level set.

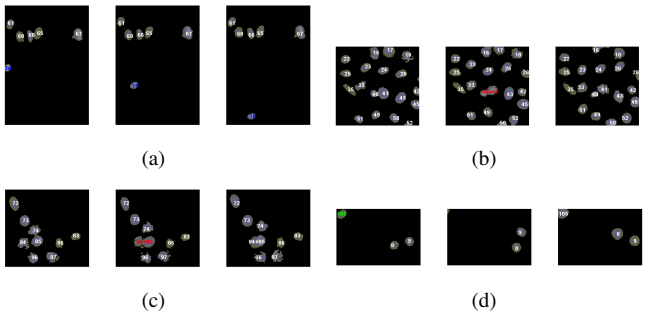


Fig. 12: Discriminations against isolated, touching, merging, disappearing cells etc. (a) Isolated Cells, (b) Touching Cells, (c) Merging Cells, (d) Boundary Cells.

Previous methods primarily utilize low-level information of image, such as pixel intensity and image gradient. Threshold based methods can obtain good results only if the images have the specific characteristics, both intensity homogeneity within the objects and good constant contrast between foreground and background. Its essence limits its application to images with touching or occlusion objects. Comparatively, the watershed algorithm shows better ability in segmentation of touching objects. However, the scheme may suffer from the over-segmentation problem. Many efforts have been made to overcome the shortcoming.

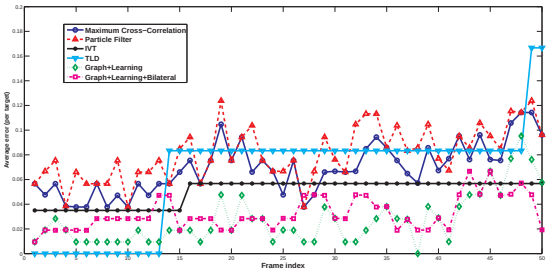
Other methods based on deformable models generate a more robust and accurate boundary. The contours of objects are driven by different kinds of forces and converge to the real boundary where the energy function of the contours reaches its minimum. In this way, the segmentation problem can be transformed into finding the optimal solution to the minimal defined energy. The level set methods indeed have many strong points, but the need of the costly re-initialization procedure and the incompetence in dealing with intensity in-homogeneity which often occurs in medical images induce bad performance in some specific segmentation cases. The level set evolution scheme without re-initialization using the local intensity information was proposed by Chunming Li et al. [27] [29] for intensity in-homogeneity images segmentation. Success of the local binary fitting (LBF) energy based active contour evolving algorithm has been demonstrated on segmentation of images.

Furthermore, it is a very challenging segmentation task after the denoising process for the wide dynamic range of image contrast. It is not an easy task to segment bright and dark cells simultaneously using only one algorithm. Because the bright cells are easy to handle, we focus on testing the proposed

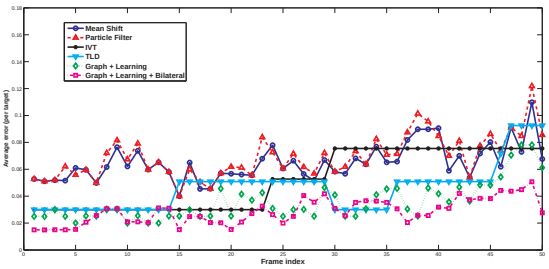
approach on the segmentation of dark cells. The segmentation results of the LBF based level set [27] and the Mumford-Shah functional based level set [28] methods are given in Fig. 11. In Fig. 11 (a) (c), a slice with both bright and dark cells is selected. Specifically, most cells of Fig. 11 (c) are very dark. The approach can segment all the cells with little error. In Fig. 11 (b) (d), we can find that the piecewise constant model can only find the bright ones and consider the dark ones as noise to be ignored.

(b) Cell Tracking

In this experiment, we manually index the cells in a number of frames and label the training matching matrices. In the cell sequences, we set the degree of nodes to 3. The node and edge features in the experiment refer to Section III-B.



(a)



(b)

Fig. 13: Statistical performance of six tracking algorithms in each frame. (a) Statistical performance on cell sequence 1, (b) Statistical performance on cell sequence 2

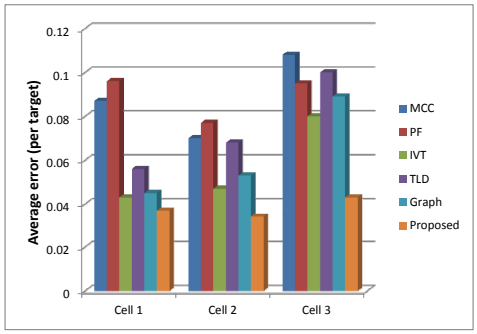


Fig. 14: Average performance of six tracking methods on cell sequences.

With the proposed algorithm, Fig. 12 demonstrates the discriminations against cell touching, merging, and disappearing etc. In Fig. 12 (a), for isolated cell indexed '78', it is reasonable to take the one that is nearest to its previous position in the next frame as a matched record, even though the distance might be large. It works well in tracing fast moving cells. Fig. 12 (b) shows touching cells, which are not necessarily merged because they separate later on. During the process that cells '40' and '41' approach to each other and separate once again, no frame indicates that they are close enough to be judged as fusion, even though we can predict their exact overlapping according to their moving directions. Different from conditions in Fig. 12 (b), the approaching cell pair shows sufficient evidence to have merged, for instance cells '84' and '85' in the second image of (c). Since the cells are merged and not separated later on, only one of the indexes would be retained. Fig. 12 (d) shows the location of boundary cells. Boundary cells are defined for those newly appeared or disappeared from image boundaries. When a boundary cell moves out of view, its node and edges go away with it. To identify it when it appears once again, its vanished location should be recorded. It is demonstrated by cell indexed '105' which disappears in the second image of (d) and returns later on.

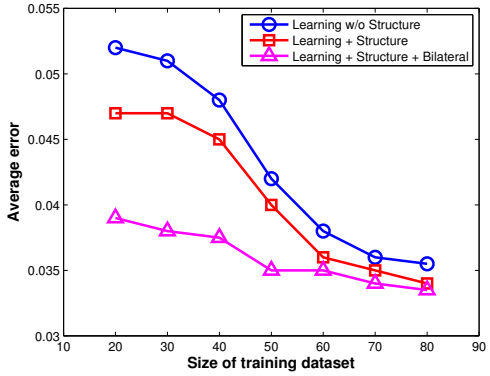
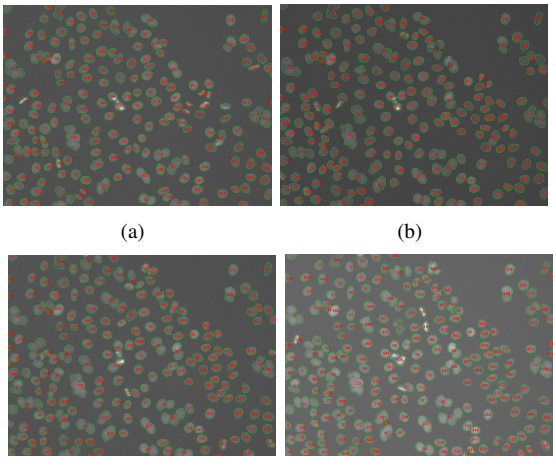


Fig. 15: Statistical performance of different learning-based graph matching methods converges with the increasing size of training dataset ($\lambda = 10$).

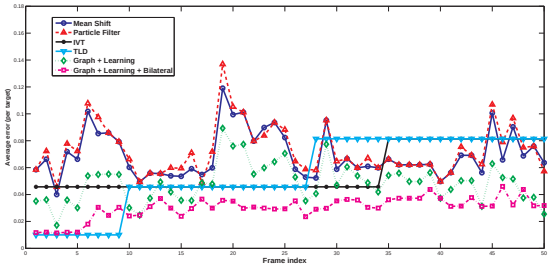


(a) (b) (c) (d)

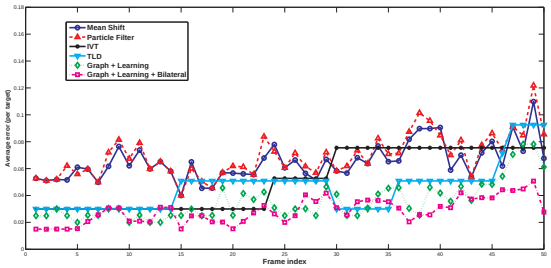
Fig. 16: The dense cell tracking results in different frames (dynamic environments).

Fig. 13 provides the average error Δ in Eq. (23) on each sequence. Besides the proposed method as well as the graph matching with learning (bilateral tracking), we also use the Maximum Cross-Correlation (MCC) [30], the particle filtering (PF) [31], Incremental Learning for Robust Visual Tracking (IVT) [39], and TLD (Tracks the object, Learns appearance and Detects) [40] for comparisons. As IVT and TLD are originally designed for single object tracking, we test them by individually tracking each object in the test sequence to ensure their reliable performance in evaluation. It can be seen that the proposed algorithm always achieves better performance. With the increase of the frame number, there is a smaller accumulation of tracking error by bilateral tracking than other methods. Fig. 14 illustrates the statistical performance of Maximum Cross-Correlation (MCC), Particle Filter (PF), IVT, TLD, and the proposed approach. In the three cell sequences, the third one has a larger density than the others. Obviously, the proposed method can favor the cells of large density.

Fig. 15 illustrates the average errors of cell tracking in different learning-based graph matching methods without structure features, with structure features, and with structure features plus bilateral tracking. Obviously, the structured learning-based graph matching with structure features obtains better performance on cell sequence while the one with learning and bilateral tracking obtains the best performance than the others, especially when the size of the training dataset is not large enough. As shown in Fig. 15, the average errors in the three methods converges as the density. Fig. 16 shows the tracking results on a dense cell sequence where the cell indexes are labeled in red.



(a)



(b)

Fig. 17: Statistical performance of six tracking algorithms in each frame. (a) Statistical performance on football sequence 1, (b) Statistical performance on football sequence 2.

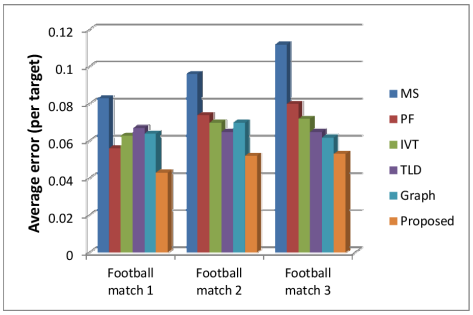
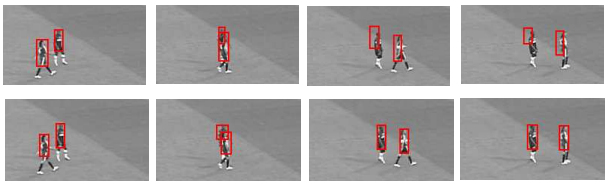


Fig. 18: Average performance of six tracking algorithms on football sequences.

B. Football Match Sequence

In this experiment, we use a football match sequence of 221 frames from 2010 UEFA Champion League final, Bayern Munich vs. Internazionale, whose resolution is 1280×720 pixels. We divide them into two parts, the training dataset of 50 frames and the test dataset of remaining frames. Similar to cell sequences, we also set the degree of nodes to 3 for the football match sequence. Besides the node and edge features in Section III-B, the popular scale-invariant feature transform (SIFT) [32] is extracted to train the detection and tracking models.



(a)

(b)

Fig. 19: Comparisons on football match sequences. (top) Tracking players using the particle filtering method. (bottom) Tracking players using the proposed method.

Fig. 17 provides the average error Δ in Eq. (23) on each sequence. Besides the proposed method as well as the graph matching with learning (bilateral tracking), we also use the Mean Shift [1], the Particle Filtering [31], Incremental Learning for Robust Visual Tracking (IVT) [39], and TLD (Tracks the object, Learns appearance and Detects) [40] for comparisons. The diagraph shows that the proposed approach always achieves better performance. With the increase of the frame number, there is a smaller accumulation of tracking error than other methods with bilateral tracking. Fig. 18 illustrates the statistical performance in Mean Shift (MS), Particle Filter (PF), IVT, TLD, and the proposed approach.

Fig. 19 demonstrates the comparison between the particle filtering (PF) algorithm and the proposed algorithm on the football match sequence. It can be seen that the proposed

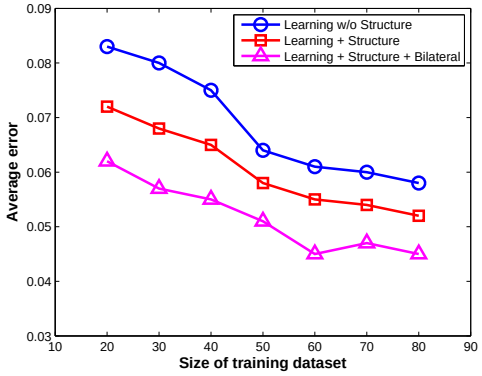
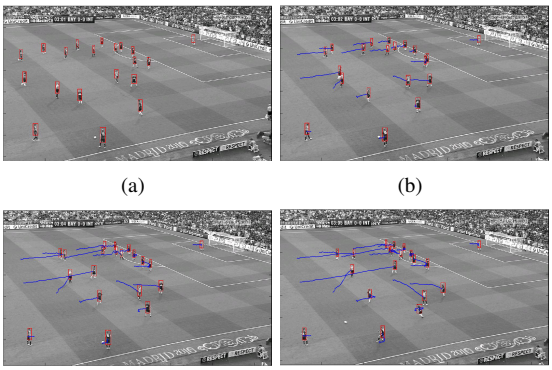


Fig. 20: Statistical performance of different learning algorithms converges with the increasing size of training dataset ($\lambda = 1$).



(a) (b) (c) (d)

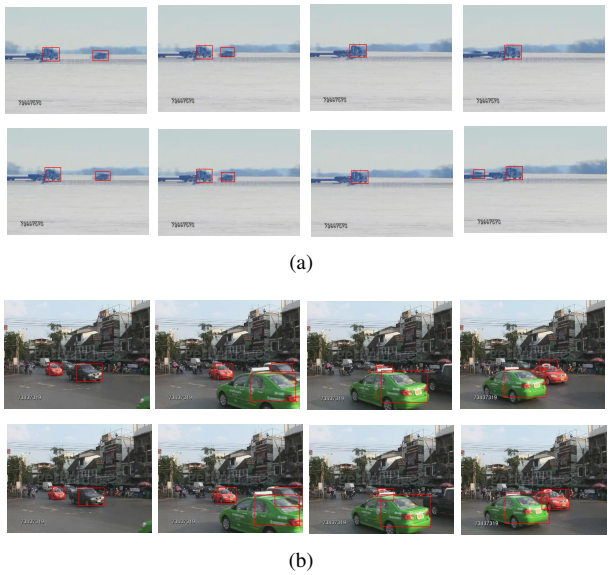
Fig. 21: The tracking results on football match sequence. Players are labeled in red rectangle. The blue lines are the traces of players' motion.

scheme achieves better performance, especially for partial occlusions. Furthermore, we compare the learning-based graph matching methods with and without bilateral tracking for different sizes of the training dataset. Fig. 20 illustrates that the average errors converge as the size increases, and the one with structure features and bilateral tracking always obtains a large gain. It is different from the cell sequence because the motions of cells are much more stochastic than players. Therefore, the bilateral tracking is much more effective for player sequence, especially when the training dataset is in large size.

The tracking results of frames 1, 76, 151 and 221, are shown in Fig. 21, where players are labeled in red rectangle and the blue lines are the traces of players' motion.

C. Car Tracking

In this experiment, we use the car sequences from the dataset of [33] whose spatial resolution is 480×360 pixels and the temporal resolution is 15 frames per second. Likewise, we divide them into two parts, the training dataset of 50 frames and the test dataset of remaining frames. Similarly, we still adopt both the node and edge features as well as the SIFT feature for tracking target cars. We manually label the cars in the training dataset, and then track the cars using the structured learning-based graph matching algorithm. Since the number of



(a)

(b)

Fig. 22: Comparisons on car sequences. (top) Tracking cars using the particle filtering approach. (bottom) Tracking cars using the proposed approach. (a) Full occlusion, (b) Partial occlusion.

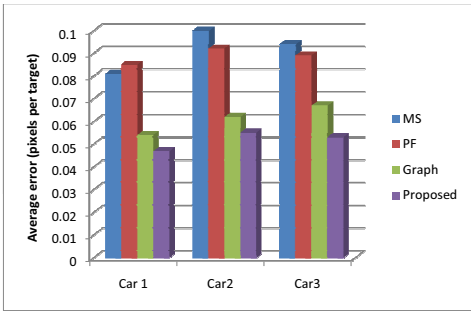


Fig. 23: Statistical performance on car sequences in four different methods.

cars in the sequence is not large and the relationship between adjacent cars is not as important as the previous experiments, the normalized Hamming loss function defined in Section IV-B may be too rough so that the tracking errors grow considerably. Instead of the normalized Hamming loss, we choose a more precise loss function which can penalize incorrect matches less if they are "close to" the correct match:

$$\Delta(G, G', m, m') = \frac{1}{|m|} \sum_i \left[\frac{d(G'_{m(i)}, G'_{m'(i)})}{\sigma} \right]^2 \qquad (32)$$

where (G, G') are the graph pair (G is the "query" graph and G' is the "target" graph), $m(i)$ is the index of the object in G' to which the ith point in G is mapped, and $m'(i)$ is the ground truth. d is simply the Euclidean distance between the object pair and is scaled by σ. Hence, the loss function defines how distant the matches are from the ground truth, and the loss is small with a decreasing distance between the matching and the correct results. The degree of nodes is set to 1 or 2 based on the number of target cars.

Fig. 22 demonstrates the comparison between the particle

filtering and the proposed algorithm on the car sequences. The full and partial occlusions always lead to tracking error. In Fig. 22 (a), the car in full occlusion is lost in the particle filtering method, while the proposed algorithm can track the car. In Fig. 22 (b), when the car is in partial occlusion, the proposed method also obtains a more accurate result than particle filtering. The average errors of the mean-shift algorithm (MS), particle filtering method (PF), the generic graph matching without learning and the proposed method, are shown in Fig. 23.

D. More Tracking examples

Finally, we evaluate the performance on the "standard sequence: UBC hockey [44]. In this experiment, we focus on evaluating the performance on objects of high speed and intense interactions. The degree of nodes is set to 3 for the UBC hockey sequence, and the performance of the proposed algorithm is shown in Fig. 24. Because the four tracked players would occasionally move out of view with the temporal movement, it could give rise to tracking results of partial players in the corresponding frames. The statistical tracking result is demonstrated in Fig. 25 and Fig. 26. Fig. 25 describes the average error Δ on UBC hockey sequence. Besides the proposed method as well as the learning-based graph matching with bilateral tracking, we also use Mean-Shift [1], Particle Filtering [31], Incremental Learning for Robust Visual Tracking [39], and TLD (Tracks the object, Learns appearance and Detects) [40] for comparisons. Fig. 26 illustrates the statistical performance within Mean Shift (MS), Particle Filtering (PF), Incremental Learning for Robust Visual Tracking (IVT), TLD, and the proposed approach.

(a) (b) (c) (d)

Fig. 24: The tracking results on UBC Hockey sequence, and players are labeled in red rectangle.

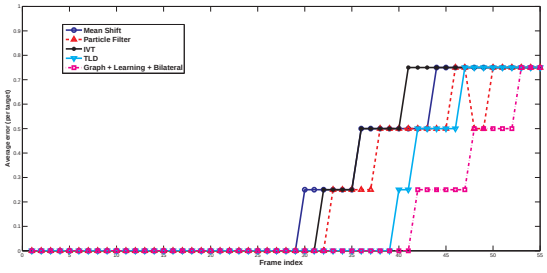


Fig. 25: Statistical performance of five tracking algorithms in each frame of UBC Hockey sequence.

VI. Conclusions and Discussion

A novel structured learning based graph matching method on multi-object tracking is proposed in this paper, which utilizes both the node and structure features in the graphs

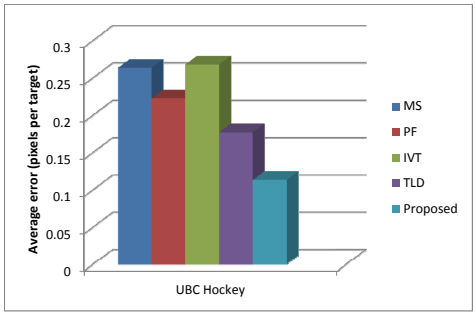


Fig. 26: Statistical performance of five tracking methods on UBC Hockey sequence.

instead of only the node feature. In the proposed method, the parameters of graph matching model are acquired in a learning phase, rather than determined by experience. Moreover, we also use the dynamic Hungarian algorithm to solve the optimization problem, which can reduce the computation complexity in the multi-object tracking problem, and bilateral tracking is also used in the method. There is an interesting issue in our experiments discussed in section V. We test the proposed method on three different types of objects, i.e., cells, football players, and cars. It is shown that the structured learning-based graph matching algorithm has better performance than existing tracking approaches, e.g. mean-shift and particle filtering. Specifically, the proposed scheme would be in favor the dynamic multi-object tracking with mutual interaction and identical appearance.

Occlusion is the most difficult obstacle for multi-object tracking. It can be seen that the proposed algorithm can solve some occurrences, but there still exist a kind of occlusion to result in errors, e.g. the football player tracking. Several variational methods [36] on pictorial structure model [34] claim to handle the occlusion in some scenes, e.g. people [35] [37] or animals [38]. However, it is not proved for more complicated applications. Therefore, our future work would focus on improving the structured learning-based graph matching model for multi-object tracking over occlusions.

References

[1] D. Comaniciu, V. Ramesh, and P. Meer, "Kernel based object tracking," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 25, no. 5, pp. 564-577, May 2003.

[2] C. Hue, J.-P. Le Cadre, and P. Perez, "Sequential monte carlo methods for multiple target tracking and data fusion," *IEEE Trans. Signal Processing*, vol. 50, no. 2, pp. 309-325, Feb. 2002.

[3] W. Hu, X. Xiao, Z. Fu, D. Xie, T. Tan, and S. Maybank, "A system for learning statistical motion patterns," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 28, no. 9, pp. 1450-1464, Sept. 2006.

[4] C. Shen, J. Kim and H. Wang, "Generalized kernel-based visual tracking," *IEEE Trans. Circuits and System for Video Technology*, vol. 20, no. 1, pp. 119-130, Jan. 2010.

[5] J. Zhu, Y. Lao, and Y. F. Zheng, "Object tracking in structured environments for video surveillance applications," *IEEE Trans. Circuits and System for Video Technology*, vol. 20, no. 2, pp. 223-235, Feb. 2010.

[6] N. Jacobs, M. Dixon, S. Satkin, and R. Pless, "Efficient tracking of many objects in structured environments," in *Proc. European Conference on Computer Vision Workshops*, pp. 1161-1168, 2009.

[7] G. Stamou, Nikolaidis, and I. Pitas, "Object tracking based on morphological elastic graph matching", in *Proc. IEEE International Conference on Image Processing*, Genoa, Italy, pp. 709-712, Sept. 2005.

[8] M. Liu, A. Roy-Chowdhury, and G. Reddy, "Automated Tracking of Stem Cell Lineages of Arabidopsis Shoot Apex Using Local Graph Matching," *The Plant Journal*, vol. 62, pp. 135-147, 2010.

[9] A. Elgammal, D. Harwood, and L. Davis, "Non-parametric Model for Background Subtraction," in *Proc. European Conference on Computer Vision*, Dublin, Ireland, pp. 751-767, June 2000.

[10] G. Unal, H. Krim, and A. Yezzi, " Fast incorporation of optical flow into active polygons," *IEEE Trans. Image Processing*, vol. 14, no. 6, pp. 745-759, June. 2005.

[11] Y. Jin and F. Mokhtarian, "Variational particle filter for multi-object tracking," in *Proc. International Conference on Computer Vision*, Rio de Janeiro, Brazil, pp. 1-8, Oct. 2007.

[12] D. Reid, "An algorithm for tracking multiple targets," *IEEE Trans. Automat. Contr.*, vol. 24, no. 6, pp. 843-854, Dec. 1979.

[13] Z. Khan, T. Balch, and F. Dellaert, "MCMC-based particle filtering for tracking a variable number of interacting targets," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 27, no. 11, pp. 1805-1819, Nov. 2005.

[14] K. Smith, D. Gatica-Perez, and J.-M. Odobez, "Using particles to track varying numbers of interacting people," in *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, San Diego, USA, pp. 962-969, Jun. 2005.

[15] Y. Chen, Y. Rui, and T. Huang, " Multicue HMM-UKF for real-time contour tracking," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 28, no.9, pp. 1525-1529, Sep. 2006.

[16] J. Ko, D. J. Klein, D. Fox, D. Haehnel, "GP-UKF: Unscented Kalman Filters with Gaussian Process Prediction and Observation Models," in *Proc. the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, San Diego, USA, pp. 1901-1907, Oct. 2007.

[17] T. Paixao, A. Graciano, R. Cesar, and R. Hirata, " A Backmapping Approach for Graph-Based Object Tracking," *Proc. Brazilian Symposium on Computer Graphics and Image Processing (SIBGRAPH)*, Campo Grande, Brazil, pp. 45-52, Oct. 2008.

[18] T. S. Caetano, T. Caelli, D. Schuurmans, and D. A. C. Barone, "Graphical models and point pattern matching," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 28, no. 10, pp. 1646-1663, Oct. 2006.

[19] H. Chen, H. Lin, and T. Liu, "Multi-object tracking using dynamical graph matching," in *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, Kauai, USA, pp. 210-217, Dec. 2001.

[20] V. Pallavi, J. Mukherjee, A. Majumdar, and S. Shamik, "Graph-Based Multiplayer Detection and Tracking in Broadcast Soccer Videos," *IEEE Trans. Multimedia*, vol. 2008, no. 5, pp. 794-805, Aug. 2008.

[21] M. Leordeanu and M. Hebert, "Unsupervised learning for graph matching," in *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, Florida, USA, pp. 864-871, June 2009.

[22] T.S. Caetano, J.J. McAuley, Li Cheng, Q.V. Le, and A.J. Smola, "Learning graph matching," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 31, no. 6, pp. 1048-1058, Jun. 2009.

[23] C. Stauffer and W. Grimson, "Learning patterns of activity using real-time tracking," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 22, no. 8, pp. 747-757, Aug. 2000.

[24] C. Papadimitriou and K. Steiglitz, "Combinatorial Optimization: Algorithms and Complexity," *Dover Publications*, New York, 1998.

[25] G. Mills-Tettey, A. Stentz, and M. Dias. "The dynamic hungarian algorithm for the assignment problem with changing costs," *Tech. Rep. CMU-RI-TR-07-27*, Robotics Institute, Pittsburgh, PA, July 2007.

[26] X. Chen, X. Zhou, and S. Wong, "Automated segmentation, classification, and tracking of cancer cell nuclei in time-lapse microscopy," *IEEE Trans. Biomedical Engineering*, vol. 53, no. 4, pp. 762-766, Apr. 2006.

[27] C. Li, C. Kao, J. Gore, and Z. Ding, "Implicit active contours driven by local binary fitting energy," in *Proc. IEEE Conf. Comput. Vis. Parttern Recognit.*, pp. 1-7, 2007.

[28] T. Chan and L. Vese, "Active contours without edges," *IEEE Trans. Image Process.*, vol. 10, no. 2, pp. 266-277, Feb. 2001.

[29] C. Li, C. Kao, J. Gore, and Z. Ding, "Minimization of regionscalable fitting energy for image segmentation," *IEEE Trans. Image Process.*, vol. 17, no. 10, pp. 1940-1949, Oct. 2008.

[30] B. Mantooth, Z. Donhauser, K. Kelly, and P. Weiss, "Cross-correlation image tracking for drift correction and adsorbate analysis," *Rev. Sci. Instrum.*, vol. 73, no. 2, pp. 313-317, Feb. 2002.

[31] M. Arulampalam, S. Maskll, N. Gordon, and T. Clapp, "A tutorial on particle filters for online nonlinear/non-gaussian bayesian tracking," *IEEE Trans. Signal Process.*, vol. 50, no. 2, pp. 174-188, Feb. 2002.

[32] D. Lowe, "Distinctive image features from scale-invariant keypoints," *Int. Journal Comput. Vis.*, vol. 60, no. 2, pp. 91-110, Feb. 2004.

[33] http://www.cs.ucf.edu/ arslan/vidmatching/index.htm

[34] P. Felzenszwalb and D. Huttenlocher, "Pictorial structures for object recognition," *International Journal of Computer Vision*, vol. 61, no. 1, pp. 55-79, Jan. 2005.

[35] H. Bhaskar, L. Mihaylova, and S. Maskell, "Human body parts tracking using pictorial structures and a genetic algorithm," in *Proc. IEEE Conf. Intell. Syst.*, vol. 2, pp. 102-106, 2008.

[36] R. Poppe, and M. Poel, "Body-Part Templates for Recovery of 2D Human Poses under Occlusion", in *Proc. of International Conference on Articulated Motion and Deformable Objects*, 2008.

[37] D. Ramanan, D. Forsyth, and A. Zisserman, "Tracking people by learning their appearance," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 29, no. 1, pp. 65-81, Jan. 2007.

[38] D. Ramanan, D. Forsyth, and K. Barnard, "Building models of animals from video," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 28, no. 8, pp. 1319-1334, Aug. 2006.

[39] David Ross, Jongwoo Lim, Ruei-Sung Lin, Ming-Hsuan Yang, "Incremental Learning for Robust Visual Tracking", *International Journal of Computer Vision*, vol. 77, no. 1-3, pp. 125-141, 2007.

[40] Z. Kalal, J. Matas, and K. Mikolajczyk, "P-N Learning: Bootstrapping Binary Classifiers by Structural Constraints," in *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, 2010.

[41] M. Isard, and A. Blake, "CONDENSATION - Conditional Density Propagation for Visual Tracking", *Intertanational Journal of Computer Vision*, vol. 29, no. 1, pp. 5-28, 1998.

[42] X. Wang, K. Tieu, and E. Grimson, "Learning semantic scene models by trajectory analysis", in *Proc. European Conference on Computer Vision*, pp. 1165-1172, 2006.

[43] T. Hospedales, S. Gong, and T. Xiang, "A Markov clustering topic model for mining behavior in video", in *Proc. International Conference Computer Vision*, pp. 110-123, 2009.

[44] K. Okuma, A. Taleghani, N. De Freitas, J. Little, and D. Lowe, "A Boosted Particle Filter: Multitarget Detection and Tracking", in *Proc. European Conference on Computer Vision*, pp. 28-39, 2004.