Summer 2016

Detecting, segmenting and tracking bio-medical objects

Mingzhong Li

Department: Computer Science

DETECTING, SEGMENTING AND TRACKING BIO-MEDICAL OBJECTS

by

MINGZHONG LI

A DISSERTATION

Presented to the Graduate Faculty of the

MISSOURI UNIVERSITY OF SCIENCE AND TECHNOLOGY

In Partial Fulfillment of the Requirements for the Degree

DOCTOR OF PHILOSOPHY

in

COMPUTER SCIENCE

2016

Approved by

Dr. Zhaozheng Yin, Advisor
Dr. Wei Jiang
Dr. Simone Silvestri
Dr. Dan Lin
Dr. Ruwen Qin

ABSTRACT

Studying the behavior patterns of biomedical objects helps scientists understand the underlying mechanisms. With computer vision techniques, automated monitoring can be implemented for efficient and effective analysis in biomedical studies. Promising applications have been carried out in various research topics, including insect group monitoring, malignant cell detection and segmentation, human organ segmentation and nano-particle tracking.

In general, applications of computer vision techniques in monitoring biomedical objects include the following stages: detection, segmentation and tracking. Challenges in each stage will potentially lead to unsatisfactory results of automated monitoring. These challenges include different foreground-background contrast, fast motion blur, clutter, object overlap and etc. In this thesis, we investigate the challenges in each stage, and we propose novel solutions with computer vision methods to overcome these challenges and help automatically monitor biomedical objects with high accuracy in different cases.

ACKNOWLEDGMENTS

First I would express my greatest gratitude to my advisor Dr. Zhaozheng Yin for his meticulous guidance, patient advice and continuous encouragement throughout my entire Ph.D. career. His expertise, sincere and valuable guidance and encouragement always enlightens my path to success in Ph.D. studies. Without his help, I could never reach the accomplishment I have so far.

I am also grateful to all the professors in Missouri University of Science and Technology, especially my committee member Dr. Wei Jiang, Dr. Dan Lin, Dr. Simone Silvestri and Dr. Ruwen Qin. It is their help and support that led me into the field of Computer Science.

My sincere thankfulness also goes to my colleagues in Dr. Zhaozheng Yin's research team, including Wenchao Jiang, Yunxiang Mao and Haohan Li. They are my best friends in my Ph.D. life, my comrades in arms who accompanied me in the last 4 years of study.

Last but not the least, a deep sense of gratitude goes to my family for their endless support for my endeavors on the road to pursue my dreams.

TABLE OF CONTENTS

Page

LIST OF ILLUSTRATIONS

LIST OF TABLES

1. INTRODUCTION

1.1. PROBLEM OVERVIEW

Behavioral analysis of biomedical organisms can inform us about the molecular mechanisms and biochemical pathways. Specifically, researches as well as industrial applications have been taken advantages of studying biomedical objects such as insects, cells, nano-particles and etc. Among some of these studies, computer vision techniques are utilized to realize effective and efficient automated monitoring with visual data such as photos and videos. Medical equipments with computer vision technologies are already in real applications, and some are even under mass manufacturing.

To generate automated monitoring results of bio-medical objects, computer vision techniques are utilized in three stages of processing: detection, segmentation and tracking. The overview workflow of these three stages are illustrated in Fig.1.1.

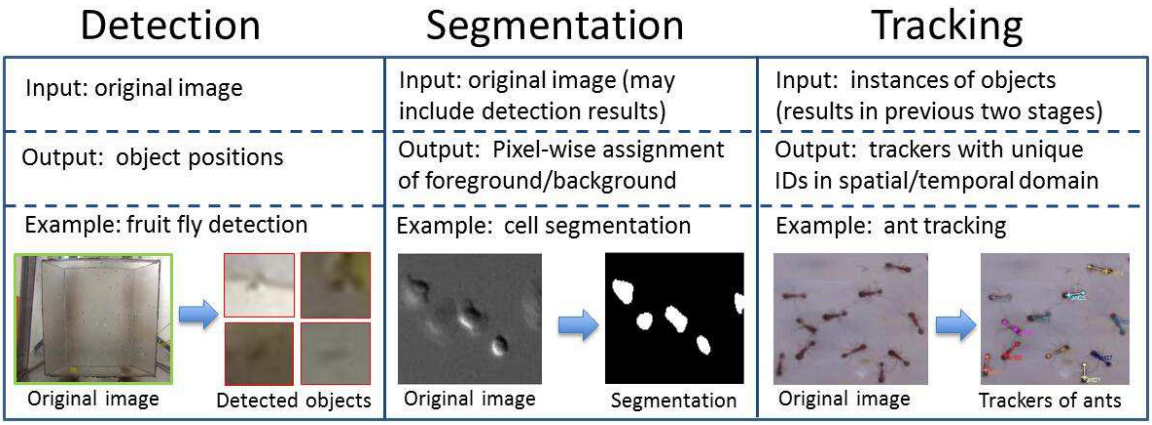

Figure 1.1. Overview on different stages of automated monitoring.

In the detection stage, we aim at locating the positions where an object class of interest is included given images or video frames. In bio-medical monitoring, this includes

tasks such as cell detection, insect (flies, ants, bees, etc.) detection and bio-particle detection. Next in the segmentation stage, precise pixel-wise classification between foreground and background is yielded to generate simplified and consistent representation of object class, which benefits further analysis and data association. Finally in the tracking stage, our goal is to associate different object instances in each frame with an appropriate ID number, in order to solve the "who is who" problem.

1.2. RELATED WORKS

In biological behavior monitoring, researches have been investigating detection, segmentation and tracking algorithms for decades. Different detection methods for insects have been proposed, such as ants in [2, 3, 4, 5], bees in [6] and flies in [7]. Model fitting and machine learning techniques are frequently used among these researches through which good detection results are achieved. Beside insect monitoring, computational algorithms have also been developed to analyze cell images automatically for detection and segmentation in microscopy images captured in high-throughput biological experiments, as discussed in [8] and [9]. In addition to common image processing algorithms ([10, 11]), and graph model algorithms ([12, 13]), recently some cell image analysis methods based on microscope optics models have been explored in [1, 14, 15, 16] and achieve highly reliable segmentation output.

Automated multi-object tracking algorithms are also investigated by numerous researchers, as summarized in [17]. Multi-Hypothesis Tracking (MHT) ([18]) and Joint Probabilistic Data Association Filters (JPDAF) ([19]) are two representative examples for multi-object tracking. To improve the tracking performance, tracklet-based association has been widely studied recently in numerous researches such as [20, 21] and [22]. First, short reliable tracklets are generated which are fragments of trajectories formed by confident grouping of detection responses, then the tracklets are connected by algorithms such as the

Hungarian algorithm ([23]), Linear Programing ([24]), Dynamic Programing ([25]) and network flow ([26]).

For biomedical object tracking specifically, research scientists have proposed various algorithms to overcome the challenges we discussed in [2, 3, 4, 5, 6, 27, 28] and [7]. Among these various approaches, particle filter is one of the most widely used methods for tracking insects like ants and flies ([2, 3, 6]). But a disappointing fact is, that none of these existing tracking approaches is capable of achieving 100 percent accuracy under complicated circumstances. The common reasons that lead to these imperfections include object overlapping and occlusion, clutter in high object density scenarios, fast motion, camouflaging, high appearance similarity between objects and etc. To overcome these tracking failure, researchers then have proposed methods such as video playback, part-based detection and fragment correction ([4]), gap filling and occlusion tunnels ([5]), but yet none of them is achieving 100 percent accurate tracking result due to the unpredictable huge and frequent challenges.

Recommender systems ([29, 30, 31, 32]) are capable of using historical data of a user to infer her/his preference on items and then predicting other items that the user might like. Websites such as Google.com, Amazon.com and Ebay.com have widely equipped their searching engines with specialized recommender systems to serve their customers. Particularly, content-based recommender systems analyze descriptions of items previously rated/bought by a user and build a model to predict the user's interests ([30, 31]). The key idea of content-based recommender system is to construct a proper user profile by collecting data representing the user preferences. By gathering descriptive data of different items as training data sets, user's profile parameters are estimated and refined iteratively via updating strategies. The learning process can be implemented through linear or non-linear regression approach, or other complicated regression schemes such as KNN, decision tree or SVM.

1.3. OUR PROPOSAL

In this thesis, we focus on the implementation of computer vision techniques in all three stages of detecting, segmenting and tracking biomedical objects. Challenges in applications such as fruit fly detection and tracking and microscopy cell image restoration and segmentation are discussed. Multiple novel algorithms are demonstrated and discussed in depth.

In Section 2, a new feature based detection approach to adaptively fit the inconsistent contrast is presented. Next in Section 3, a multi-modal restoration algorithm for segmentation and classification of cell microscopy images is proposed, followed by the discussion of a cell segmentation and classification method based on MMSER. Then in Section 4, we formulate a new cascaded data association algorithm for accurate object tracking, along with a tracking error correction system which helps debugging automated tracking data. Finally in Section 5, we draw conclusion to this thesis and discuss potential future works.

2. DETECTION OF BIOMEDICAL OBJECTS

2.1. INTRODUCTION

Detection of biomedical objects is the first stage of monitoring, which is essential to provide highly reliable targets for further segmentation and tracking. Due to reasons such as tiny size, inconsistent contrast and motion blur, detection of biomedical object can be very challenging. In most cases, the key to achieve a highly reliable detection result is to construct a classifier which has high adaptability to different scenes. In this section, we demonstrate our work by introducing a novel approach for detecting tiny object with small size, motion blur and inconsistent contrast. Monitoring the fruit flies through camera captured videos are discussed as an research example.

We have established a behavioral paradigm in which flies are housed in a 7in x 7in x 1.5in open field with water and food provided (Fig.2.1). Within the glass chamber, we diffuse and change the light to simulate the day/night transition and control the temperature and air pressure to simulate different weather conditions. Flies are free to walk, fly, and interact with other males and females. These behaviors rely on positions of the fly, and their inter-relationship with one another. But due to the reason as we stated, it is difficult to automatically detect the flies with all the challenges above, which motivates us to develop a novel detection approach for analysis of the fly behaviors.

There are three main challenges for our detection problem: (1) the contrast between the flies and their surrounding background is low at specific regions (Fig.2.1.1-2.1.3), making the automatic object detection hard; (2) the size of a fly (around 3×6 pixels in Fig.2.1) is small and the appearances of flies are very similar to each other. Therefore it is hard for us to extract rich feature descriptors on flies to build distinctive object models, making the appearance-based object tracking methods [17] unsuitable here; (3) the flies can fly as fast

as 1.7 meters/second [33], or 30 pixels/frame in videos captured by a 120fps video camera with the resolution of 480×848 pixels. The motion blur caused by fast-motion (Fig.2.1.4) makes the object detection hard.

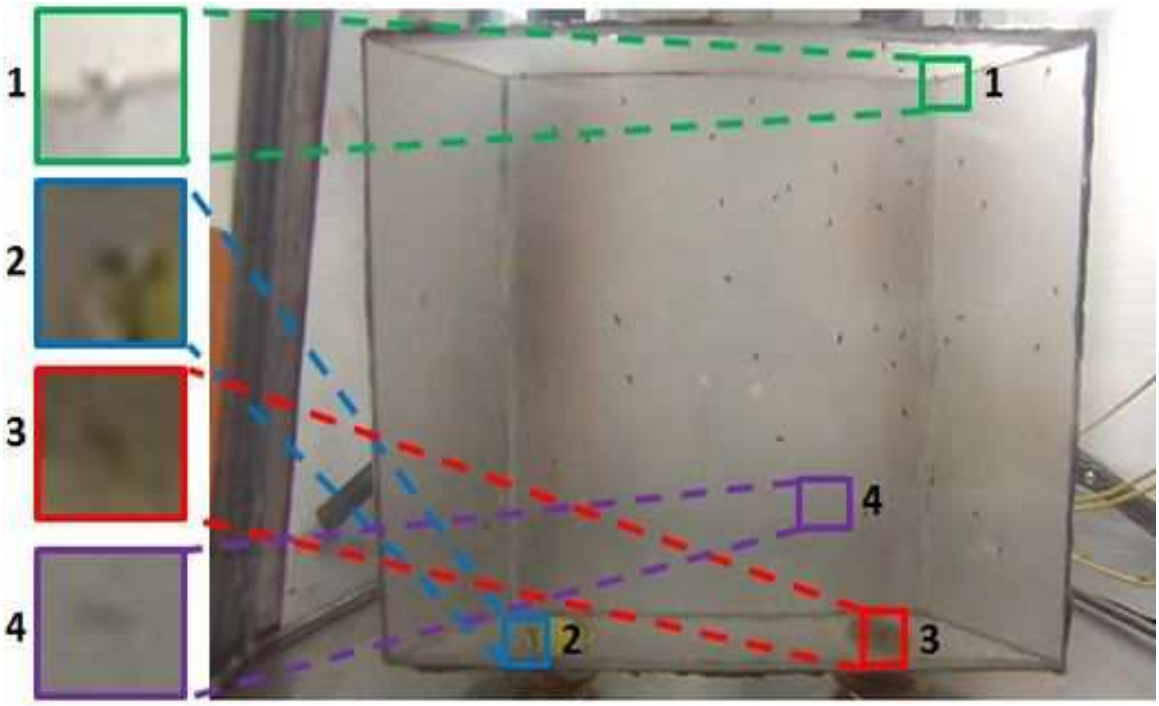

Figure 2.1. Images of flies in a chamber.

In this section, we propose to conquer the challenges with An Adaptive Local Binary Pattern (ALBP) feature, which is designed to classify pixels into objects and background, attacking the challenges of fly detection caused by low image contrast, lighting fluctuation and motion blur. Detail of our method is explain in the next sections of this section.

2.2. METHODOLOGY

In this section we will demonstrate our adaptive fly detection algorithm based on ALBP.

2.2.1. Experiment Set-up for Data Acquisition. Our experiment set-up for data for video data acquisition is illustrated in Fig.2.2. Fruit flies are kept in a glass chamber for long term monitoring. Food and water are supplied in small bowls on bottom of the chamber. Day/night transition and temperature/air pressure is controlled by a connected computer unit to simulate different weather conditions. A common kinetic camera (GoPro Hero 2) is utilized to capture normal video data without high resolution. Infrared light is equipped for night vision monitoring.

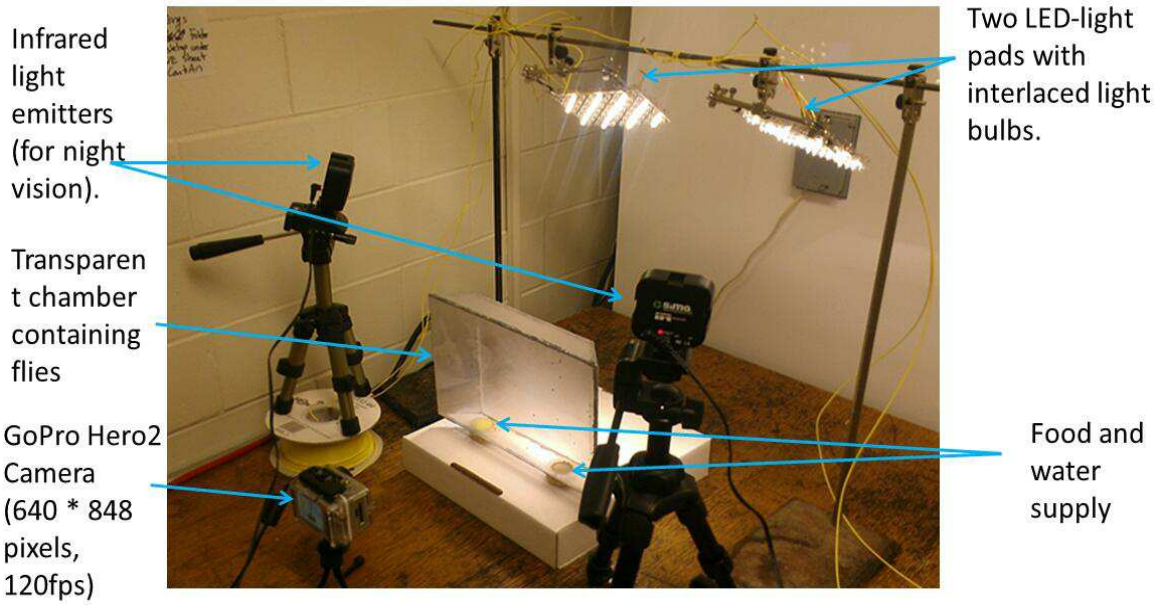

Figure 2.2. Experiment set-up.

2.2.2. Adaptive LBP Feature for Fly Detection. Considering the background variation caused by various reasons, it is not easy to build and update an accurate background model to detect flies by background subtraction. Simply thresholding the images in Fig.2.4(b) by the Otsu method [34] does not work either, as shown in Fig.2.4(c).

Observing the small contrast between flies and their surrounding background, we explore the Local Binary Pattern (LBP, [35]) feature that characterizes the local spatial structure of the image texture. The overview of our method is illustrated in Fig.2.3.

Given the center pixel I_c, a binary code is computed by comparing I_c with its neighboring pixels I_n:

$$LBP = \sum_{n=0}^{N} s(I_n - I_c)2^n \tag{2.1}$$

where $s(x)$ is a step function, i.e., $s(x) = 0$ if $x < 0$ and $s(x) = 1$ otherwise. N is the number of neighbors (e.g., $N = 8$ for a 3×3 neighborhood). Due to the fluctuation of intensity values, different flies in an image may exhibit different LBP features. We train and apply a Support Vector Machine (SVM) classifier on the LBP features to classify image pixels into flies and background. However, classification using the LBP feature does not generate good results. This is expectable since the LBP feature is easily affected by small fluctuation of pixel value changes, especially when the pixel value of flies varies while they are moving around locations with different light conditions.

To increase the robustness over intensity fluctuation, we introduce a threshold T into the step function in Eq.2.1, i.e, $s(x) = 0$ if $x < T$ and $s(x) = 1$ otherwise. When using the same T to get thresholded LBP for all image pixels, the classification still does not work well, due to inconsistent contrast between flies and background. Flies seem to be experts in camouflaging, which always create challenges for our detection tasks using traditional feature descriptors. Therefore, we propose an Adaptive LBP (ALBP) feature by adapting threshold T at different locations (x,y):

$$T(x,y) = \begin{cases} T_L, & \text{if } \mu(x,y) < \mu_L, \\ T_H, & \text{if } \mu(x,y) > \mu_H, \\ \frac{T_H - T_L}{\mu_H - \mu_L}\mu(x,y) + T_L, & \text{otherwise.} \end{cases} \tag{2.2}$$

where $\mu(x,y)$ computes the mean intensity value within a patch around (x,y) (e.g., the patch size is 11×11 in this paper). The parameters (μ_H, μ_L, T_H, T_L) in Eq.4.7 are learned by linear regression from a training set of fly pixels $\{I_i\}$ with their corresponding $\{\mu_i\}$. $\mu_H = \max_i \mu_i$ and $T_H = I_{i^*} - \mu_H$ where $i^* = \arg\max_i \mu_i$. Similarly, we define μ_L and T_L.

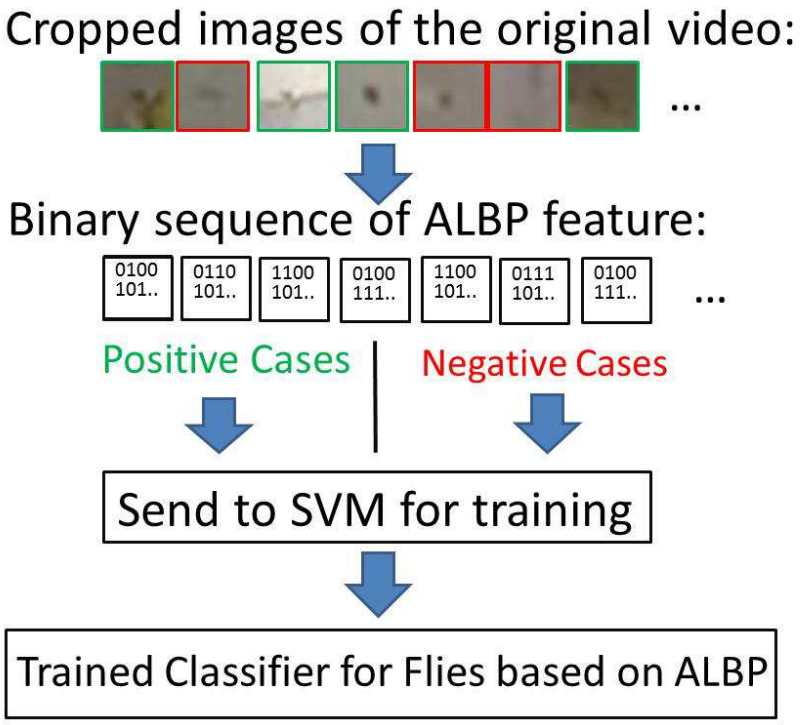

Figure 2.3. Overview of our ALBP detection method.

We train and apply a SVM classifier on the ALBP feature to classify pixels into flies and background, which achieves much more reliable detection results. For flies who camouflage themselves in background with similar pixel values, and those who create motion blur while moving fast, lower threshold value will be adopt in ALBP extraction. On the other hand, flies who expose themselves will obtain higher threshold values, in order to minimize the influence of noises.

In next section, we experimentally test our detection method, and compare it with related previous algorithms to support our methodology.

2.3. EXPERIMENTS

In Fig.2.4, we show the qualitative comparison between our method and other approaches, including the results by Otsu thresholding (Fig.2.4(c)), original LBP (Fig.2.4(d)) and constant thresholded LBP (Fig.2.4(e)). Finally, the detected flies using our method corresponding to Fig.2.4(a) are shown in Fig.2.4(g).

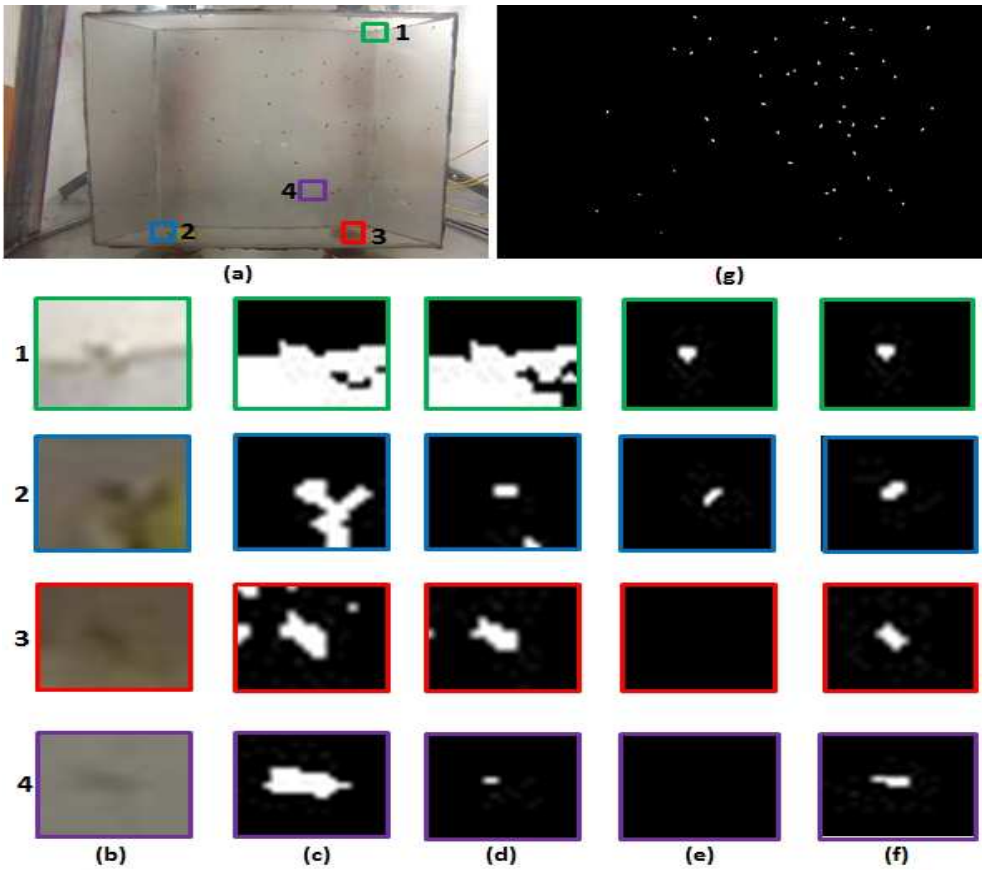

Figure 2.4. Fly detection. (a) Input image; (b) Zoom-in details of four subimages in (a); (c) Segmentation by Otsu thresholding (white and black denote fly and background pixels, respectively); (d) Classify each pixel into fly or background by its LBP feature; (e) Classify pixels by constant thresholded LBP feature; (f) Classify pixels by our adaptive LBP feature; (g) Detected fly objects by grouping nearby classified fly pixels.

We also present the quantitatively evaluation of our ALBP detection method in Table.2.1 and Table.2.2. It is obvious a much more reliable detection result is achieved by using our method, comparing to previous detection algorithms.

Table 2.1. Detection precision of different methods.

	# Frames	Otsu	LBP	TLBP	**ALBP**
Video1	21000	0.487	0.613	0.731	**0.937**
Video2	220000	0.531	0.574	0.746	**0.942**

Table 2.2. Detection recall of different methods.

	# Frames	Otsu	LBP	TLBP	**ALBP**
Video1	21000	0.911	0.839	0.669	**0.926**
Video2	220000	0.875	0.794	0.713	**0.944**

2.4. SUMMARY

We propose an Adaptive LBP feature to detect tiny flies with different image contrast and motion blur. With experimental comparison to traditional methods, the high performance of our approach shows its potential to enable automated detection of flies. In the following sections, we will discuss how to implement automated object tracking with the basis of the detection results.

3. SEGMENTATION OF BIOMEDICAL OBJECTS

In many biomedical applications, object segmentation is hard due to their vague boundaries and the noise and artifacts. Segmentation of cell regions in microscopy images is a typical case. The task is challenging due to the noise and artifacts introduces by the optics, as well as the blurring boundaries of the cells themselves. In this section, we focus on solving cell region segmentation problem.

First, we present a novel microscopy image restoration algorithm capable of co-restoring Phase Contrast and Differential Interference Contrast (DIC) microscopy images captured on the same cell dish simultaneously, which is different from previous methods which mostly rely on simple pixel-wise processing or restoration through single-modal microscopy. Cells with different phase retardation and DIC gradient signals are restored into a single image without the halo artifact from phase contrast or pseudo 3D shadow-casting effect from DIC. The co-restoration integrates the advantages of two imaging modalities and overcomes the drawbacks in single-modal image restoration. Evaluated on a datasets of five hundred pairs of phase contrast and DIC images, the restored microscopy images demonstrate their effectiveness to greatly facilitate the cell image analysis tasks such as cell segmentation and classification.

Second, considering that multi-modality microscopy images are not always available at all times, we propose a novel cell segmentation approach by extracting Multi-exposure Maximally Stable Extremal Regions (MMSER) in phase contrast microscopy images on the same cell dish. Instead of using co-related redundant image data in multiple modalities, we adopt images with various exposure times under same microscopy modality. Using our method, cell regions can be well identified by considering the maximally stable regions with response to different camera exposure times. Meanwhile, halo artifacts with

regard to cells at different stages are leveraged to identify cells' stages. The experimental results validate that high quality cell segmentation and cell stage classification can be achieved by our approach.

3.1. INTRODUCTION

Microscopy imaging techniques are critical for biologists to observe, record and analyze the behavior of specimens. Two well-known non-fluorescence microscopy imaging modalities based on light interferences are phase contrast microscopy ([36]) and differential interference contrast (DIC) microscopy (chapter 10 in [37]). As non-invasive techniques, phase contrast and DIC have been widely used to observe live cells without staining them.

The overview of different cell segmentation methods in previous works are demonstrated in Fig.3.1. With the large amount of microscopy image data captured in high-throughput biological experiments, computational algorithms have been developed to analyze cell images automatically ([8] and [9]). In addition to common image processing algorithms in [10, 11], recently some cell image analysis methods based on microscope optics models have been explored. Due to the specific image formation process, phase contrast microscopy images contain artifacts such as the halo surrounding cells (Fig.3.2(a)), and DIC microscopy images has the pseudo 3D shadow-cast effect (Fig.3.2(c)). The computational imaging model of phase contrast microscopy was derived in [14], based on which algorithms have been developed to restore artifact-free images for cell segmentation and detection ([1] and [15]). The computational imaging model of DIC microscopy was derived in [16] and corresponding preconditioning algorithms were developed to preprocess the DIC images to greatly facilitate the cell segmentation as discussed in [16] and [38].

The imaging system of phase contrast microscopy consists of a phase contrast microscope and a digital camera to record time-lapse microscopy images on cells, hence the microscopy images depend on both the optics and the camera setting such as its exposure

time. Recently, cell image analysis methods based on microscope optics models have been explored in [14, 38, 39]. One challenge of these methods is to segment cells at different stages [1]. For example, cells become thick in the culturing dish during mitotic and apoptotic stages, leading to different phase retardations in the phase contrast microscopy imaging compared to cells under the migration stage. Therefore, a dictionary of diffraction patterns has been derived to approximate various phase retardations [15, 40].

	Corresponding cell segmentation methods	Drawbacks
Optics features (front end)	Imaging model based restoration [Z. Yin, et al. MedIA 2012]	Cannot handle different cell types
Camera settings (rear end)	Cell-sensitive imaging [Z. Yin, et al., MedIA 2015]	Might neglect information in non-cell regions
Cell types (object features)	Machine learning based methods by learning cell features [O. Ronneberger, et al., MICCAI 2015]	Overlook the information given by optics and camera features, complicated learning structure, heavy training processes

Figure 3.1. Overview of Different Cell Segmentation Methods.

In addition to the front-end of the imaging pipeline (optics), a cell image segmentation approach based on the rear-end of the imaging pipeline (camera setting) was developed [41, 42]. Variously exposed phase contrast microscopy images on the same cell dish are used to restore cells' irradiance signals, while the irradiance signals from non-cell background regions are restored as zero. The image artifact such as halo around cells is restored as zero in [41], but this artifact is informative to classify cells at different stages.

Cell segmentation and classification methods based on machine learning techniques are also widely investigated in recent years [43, 44, 45, 46]. Although most of these works have achieved promising experimental results, there are still a few drawbacks of these methods, including overlooking the information given by optics and camera features, the demand for construction of very complicated learning structures as well as heavy workload on training processes.

In this section, we first present a multi-modality cell segmentation approach, considering the fact that there are still some challenges which are hard to be conquered by a single microscopy modality. For example, during mitosis (division) or apoptosis (death) events, cells appear brighter than their surrounding medium (a different phenomenon compared to halos around dark cells during their migration cell stages, as shown in Fig.3.2(a)). Since cells become thick during mitotic and apoptotic stages, mitotic and apoptotic cells have different phase retardation compared to migration cells. As a result, mitotic and apoptotic cells are not well restored by the phase contrast microscopy model suitable for migration cells (Fig.3.2(b)). But, on the other hand, the DIC image has strong gradient response corresponding to mitotic/apoptotic cells (Fig.3.2(c)), thus they are well restored by the DIC imaging model (Fig.3.2(d)). However, some flat cells during their migration stages in the DIC image have low gradient signal (regions with shallow optical path slopes produce small contrast and appear in the image at the same intensity level as the background, as shown in Fig.3.2(c)), which is very challenging for DIC image restoration (Fig.3.2(d)). But, those flat migration cells can be easily restored in phase contrast models (Fig.3.2(b)). More detailed comparison on the advantages and disadvantages of phase contrast and DIC microscopy can be found in

$http://www.microscopyu.com/tutorials/java/phasedicmorph/index.html$.

Observing that phase contrast and DIC imaging modalities are complementary to each other, we propose this novel multimodal microscopy image restoration approach via both phase contrast and DIC imaging, with the following contributions:

(1) We capture phase contrast and DIC microscopy images on the specimens simultaneously and develop a co-restoration algorithm such that the two image modalities can be restored into one single image without any artifact from either modality (halo in phase contrast or pseudo 3D relief shading in DIC);

(2) The co-restoration algorithm is adaptive to integrate the advantages of two imaging modalities and overcome the drawback in single-modal image restoration, so regions of mitotic/apoptotic cells rely more on DIC imaging and regions with shallow optical path slopes focus more on phase contrast imaging;

(3) The co-restored images from phase contrast and DIC imaging greatly facilitate cell image analysis tasks such as cell segmentation and cell classification.

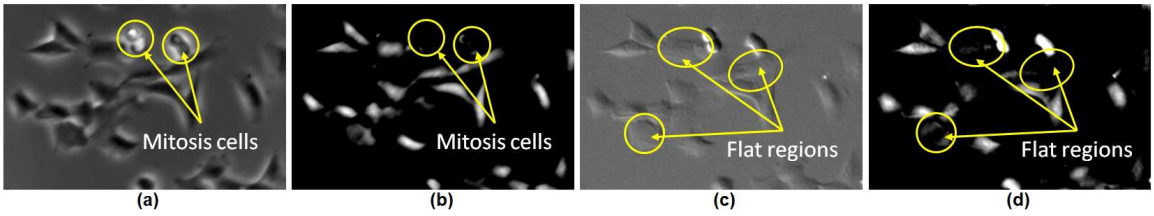

Figure 3.2. Challenges. (a) Phase contrast image; (b) Phase contrast image restoration; (c) DIC image; (d) DIC image restoration.

Secondly, considering that obtaining microscopy images with different modalities is not always feasible in all cases, we investigate the possibility of still achieving highly accurate segmentation results by creating redundant information in single microscopic modality. Therefore, next in this section following the discussion on multi-modal restoration, we present a novel cell segmentation approach by extracting Multi-exposure Maximally Stable Extremal Regions (MMSER) in variously exposed phase contrast microscopy images. Due to different exposure time length, irradiance signals have different responses to cell regions and artifacts. By extracting MMSER components over different intensity thresholds and exposure times, we are able to identify the most stable regions indicating cells, as well as those artifacts around them. Our contribution is twofold:

(1) First, we consider multi-exposed microscopy images to extract Multi-exposure Maximally Stable Extremal Regions (MMSER) to identify cells and their artifact regions;

(2) Second, we accurately classify cell and halo regions via a local Graph-cut algorithm, facilitating cell stage monitoring.

3.2. CELL SEGMENTATION USING MULTIPLE MODALITIES

In this section, we first give a brief overview of the problems of co-restoreing microscopy images as shown in Fig.3.3. We then discuss methodological details as well as experiments.

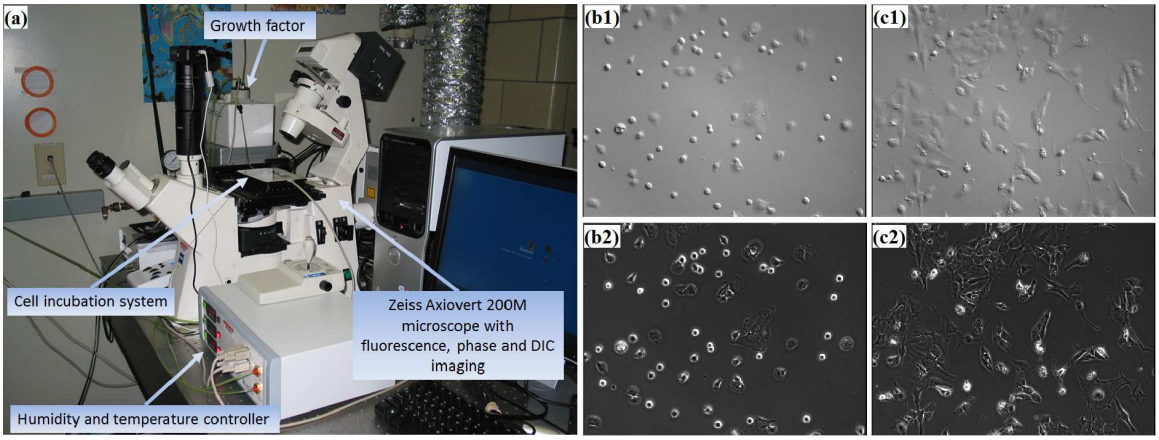

Figure 3.3. Zeiss Axiovert 200M microscope with both phase and DIC imaging.

3.2.1. Data Acquisition. Generating phase contrast and DIC images simultaneously is also not an issue on a common motorized microscope. Fig.3.3(a) is the microscope (Zeiss Axiovert 200M) we used for imaging live cells. The single instrument has multiple microscopy functions including phase contrast and DIC. Different optical components such as the phase plate and DIC analyzer are mounted on a turret. A servo motor in the microscope allows for different optical components to be moved out and into the optical pipeline without any human manipulation. The cells were cultured in an incubation system

placed on the top stage of the microscope which didn't move during the entire experiments. Therefore, no image registration was needed.

Switching optical components is very fast (less than 1 second) and the time-lapse images on living cells are taken every 5 minutes. The cell movement within 1 second is very tiny. Thus we can have phase contrast and DIC images "simultaneously" in an automated manner without human interaction, other than setting the time interval and hitting the start button to start image acquisition. For example, Fig.3.3 (b1) and (b2) show the first DIC and phase contrast images of an experiment, respectively, and Fig.3.3 (c1) and (c2) show the images after 37 hours (images are taken every 5 minutes).

3.2.2. Theoretical Foundation of Microscopy Image Restoration. Given cells in a petri dish, we capture their phase contrast microscopy image $\mathbf{g_p}$ and DIC microscopy image $\mathbf{g_d}$, simultaneously. Let $\mathbf{f_p}$ and $\mathbf{f_d}$ be the artifact-free phase contrast and DIC images, respectively, which are related to cell's physical properties such as the optical path length, we adopt the linear imaging models of phase contrast and DIC microscopy used in [14, 16]:

$$\mathbf{g_p} \approx \mathbf{H_p f_p} \tag{3.1}$$

$$\mathbf{g_d} \approx \mathbf{H_d f_d} \tag{3.2}$$

where all the images are represented by vectorized $N \times 1$ vectors with N pixels in an image. $\mathbf{H_p}$ and $\mathbf{H_d}$ are two $N \times N$ sparse matrices defined by the Point Spread Function (PSF) of phase contrast [14] and DIC [16], respectively.

Λ Rather than restoring $\mathbf{f_p}$ and $\mathbf{f_d}$ independently, we formulate the following constrained quadratic function to restore a single artifact-free image $\mathbf{f}$ from two microscopy modalities, which is related to cell's physical properties but without any artifacts from either phase contrast or DIC:

$$\mathbf{O}(\mathbf{f}) = \left\|\mathbf{W}_p(\mathbf{H}_p\mathbf{f} - \mathbf{g}_p)\right\|_2^2 + \left\|\mathbf{W}_d(\mathbf{H}_d\mathbf{f} - \mathbf{g}_d)\right\|_2^2 + \omega_s \mathbf{f}^T \mathbf{L}\mathbf{f} + \omega_r \left\|\Lambda\mathbf{f}\right\|_1 \tag{3.3}$$

where $\mathbf{W}_p$ and $\mathbf{W}_d$ are two $N \times N$ diagonal matrices. $diag(\mathbf{W_p}) = \{w_p(n)\}$ and $diag(\mathbf{W}_d) = \{w_d(n)\}$ where $w_p(n)$ and $w_d(n)$ are the weights for phase contrast and DIC restoration error cost of the nth pixel ($n \in [1,N]$), respectively. $\mathbf{W}_p + \mathbf{W}_d = \mathbf{I}$ where $\mathbf{I}$ is the identity matrix. $\mathbf{L}$ is the Laplacian matrix defining the local smoothness [12]. Λ is a diagonal matrix with $diag(\Lambda) = \{\lambda_n\}$ where $\lambda_n > 0$. ω_s and ω_r are the weights for smoothness and sparseness terms, respectively.

Algorithm 1 Algorithm 1: Solver for Nonnegative-constrained Quadratic Problem.

Initialization: $t = 1$, $\mathbf{f}^{(t)} = \mathbf{1}$ and $\Lambda = \mathbf{I}$ (i.e., $\lambda_n = 1$)

1: **Repeat:**
2: **Perform the following steps for all pixel n's:**
3:
4: $$\mathbf{f}_n^{(t+1)} \leftarrow \left[\frac{-(\mathbf{b}_n + \frac{\omega_r}{2}\lambda_n^{(t)}) + \sqrt{(\mathbf{b}_n + \frac{\omega_r}{2}\lambda_n^{(t)})^2 + 4(\mathbf{Q}^+\mathbf{f}^{(t)})_n(\mathbf{Q}^-\mathbf{f}^{(t)})_n}}{2(\mathbf{Q}^+\mathbf{f}^{(t)})_n} \right] \mathbf{f}_n^{(t)}$$
5: $$\lambda_n^{(t+1)} \leftarrow \frac{1}{\mathbf{f}_n^{(t+1)} + \varepsilon}$$
6: $t \leftarrow t + 1$
7: **Until the change on f between two iterations are smaller than a tolerance.**

Since the objective function in Eq.3.3 has a l_1 sparseness regularization, there is no closed-form solution on $\mathbf{f}$. We constrain the restored $\mathbf{f}$ to have nonnegative values and convert Eq.3.3 to a Nonnegative-constrained Quadratic Problem (NQP):

$$\mathbf{f}^* = \arg\min_{\mathbf{f}} \mathbf{f}^T\mathbf{Q}\mathbf{f} + 2(\mathbf{b} + \frac{\omega_r}{2} diag(\Lambda))^T\mathbf{f} + c, \; s.t. \; \mathbf{f} \geq 0 \tag{3.4}$$

where

$$\mathbf{Q} = \mathbf{H}_p^T\mathbf{W}_p^T\mathbf{W}_p\mathbf{H}_p + \mathbf{H}_d^T\mathbf{W}_d^T\mathbf{W}_d\mathbf{H}_d + \omega_s\mathbf{L} \tag{3.5}$$

$$\mathbf{b} = -\mathbf{H}_p^T\mathbf{W}_p^T\mathbf{W}_p\mathbf{g}_p - \mathbf{H}_d^T\mathbf{W}_d^T\mathbf{W}_d^T\mathbf{g}_d \tag{3.6}$$

$$c = \mathbf{g}_p^T\mathbf{W}_p^T\mathbf{W}_p\mathbf{g}_p - \mathbf{g}_d^T\mathbf{W}_d^T\mathbf{W}_d^T\mathbf{g}_d \tag{3.7}$$

Given $\mathbf{W_d}$ and $\mathbf{W_d}$, we solve the NQP problem in Eq.3.4 by the following algorithm using the non-negative multiplicative update ([47]) and re-weighting techniques ([48]) in Algorithm 1.

Here ε is a small constant to avoid divide-by-zero. $\mathbf{Q}^+$ and $\mathbf{Q}^-$ represent the positive and negative components of $\mathbf{Q}$:

$$\mathbf{Q}^+_{i,j} = \begin{cases} \mathbf{Q}_{i,j}, & \text{if } \mathbf{Q}_{i,j} > 0 \\ 0, & \text{otherwise} \end{cases} \quad \textit{and} \quad \mathbf{Q}^-_{i,j} = \begin{cases} \left|\mathbf{Q}_{i,j}\right|, & \text{if } \mathbf{Q}_{i,j} < 0 \\ 0, & \text{otherwise.} \end{cases} \tag{3.8}$$

3.2.3. Multimodal Microscopy Image Restoration Algorithm. When solving for the artifact-free image $\mathbf{f}$ in Eq.3.3 by Algorithm 1, we need two $N \times N$ diagonal matrices $\mathbf{W_p}$ and $\mathbf{W_d}$, defining the weights of phase contrast and DIC imaging modalities, respectively. Ideally, for each pixel we expect that the imaging modality which has better restoration performance on the pixel has larger weight on that pixel in the objective function. For example, we expect large $w_p(n)$'s on pixel regions with small slopes of optical path length, and large $w_d(n)$'s on pixel regions where mitosis and apoptosis occur. This reasoning leads to a chicken-or-egg problem: restoring $\mathbf{f}$ needs $\mathbf{W_p}$ and $\mathbf{W_d}$ but defining $\mathbf{W_p}$ and $\mathbf{W_d}$ needs the restoration $\mathbf{f}$.

To solve this dilemma, we can initialize the weights ($w_p(n)$ and $w_d(n)$, $w_p(n) + w_d(n) = 1$) randomly between 0 and 1. Then, we restore $\mathbf{f}$ using the weights on two imaging modalities. Based on restoration result, we update the weights by checking the corresponding restoration errors. The process of restoration and weight updating are iterated until convergence.

Based on the restoration $\mathbf{f}^{(t)}$ at iteration t, the restoration errors of phase contrast and DIC (denoted as $\mathbf{E}_p^{(t)}$ and $\mathbf{E}_d^{(t)}$, respectively) are calculated as:

$$\mathbf{E}_p^{(t)} = \left|(\mathbf{H}_p\mathbf{f}^{(t)} - \mathbf{g}_p)\right| \tag{3.9}$$

$$\mathbf{E}_d^{(t)} = \left| (\mathbf{H}_d \mathbf{f}^{(t)} - \mathbf{g}_d) \right| \tag{3.10}$$

where $|\cdot|$ computes the element-wise absolute value of a vector. $\mathbf{E}_p$ and $\mathbf{E}_d$ are two $N \times 1$ vectors with elements defining restoration errors at N pixel locations.

Then, the weighting matrices are updated as:

$$diag(\mathbf{W}_p^{(t+1)}) = diag(\mathbf{W}_p^{(t)}) + 1 - \mathbf{E}_p^{(t)} ./ (\mathbf{E}_p^{(t)} + \mathbf{E}_d^{(t)}) \tag{3.11}$$

$$diag(\mathbf{W}_d^{(t+1)}) = diag(\mathbf{W}_d^{(t)}) + 1 - \mathbf{E}_d^{(t)} ./ (\mathbf{E}_p^{(t)} + \mathbf{E}_d^{(t)}) \tag{3.12}$$

where './' computes the element-wise division between two vectors. After normalization such that $\mathbf{W}_p + \mathbf{W}_d = \mathbf{I}$, the weight matrices are updated as:

$$diag(\mathbf{W}_p^{(t+1)}) = (diag(\mathbf{W}_p^{(t)}) + \mathbf{E}_d^{(t)} ./ (\mathbf{E}_p^{(t)} + \mathbf{E}_d^{(t)}))/2 \tag{3.13}$$

$$diag(\mathbf{W}_d^{(t+1)}) = (diag(\mathbf{W}_d^{(t)}) + \mathbf{E}_p^{(t)} ./ (\mathbf{E}_p^{(t)} + \mathbf{E}_d^{(t)}))/2 \tag{3.14}$$

We summarize our iterative co-restoration in Algorithm 2 below.

Algorithm 2 Algorithm 2: Multimodal Microscopy Image Restoration.

Input: $t = 1$, $\mathbf{W}_p^{(1)}$ and $\mathbf{W}_d^{(1)}$

1: **Repeat:**
2: **Solve for $\mathbf{f}^{(t)}$ in Eq.3.4 using $\mathbf{W}_p^{(t)}$ and $\mathbf{W}_d^{(t)}$ by Algorithm1;**
3: **Calculate the restoration error vectors ($\mathbf{E}_p^{(t)}$ and $\mathbf{E}_d^{(t)}$) in Eq.3.9 and Eq.3.10;**
4: **Update $\mathbf{W}_p^{(t)}$ and $\mathbf{W}_d^{(t)}$ using Eq.3.13 and Eq.3.14;**
5: $t \leftarrow t + 1$**;**
6: **Until the change on f between two iterations is smaller than a tolerance.**

3.2.4. Cell Segmentation and Classification based on Co-restoration. Fig.3.4 shows the outline of our segmentation and classification procedure based on co-restoration

results. Fig.3.4(b) shows the restored images by three different approaches from Fig.3.4(a), where it is noticeable that the non-cell background region has uniform low pixel values, and the contrast between cells and background is high. By simply thresholding the restored images, segmentation results is obtained in Fig.3.4(c).

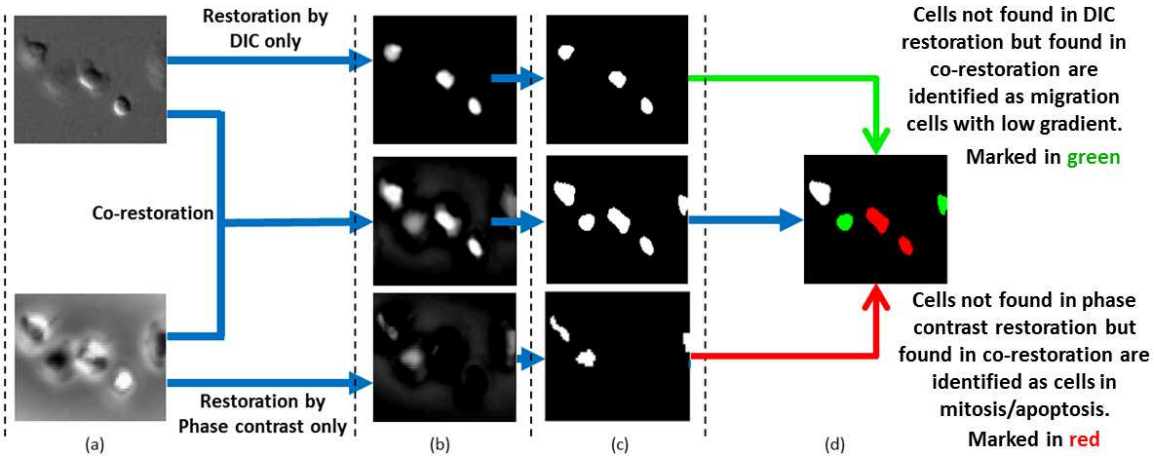

Figure 3.4. Cell Segmentation and classification. (a)Original images; (b)Restored images; (c) Segmented images by thresholding; (d)Cell classification.

Then, we classify cells into three classes: (1) mitosis/apoptosis cells (challenging for phase contrast microscopy); (2) flat migration cells (challenging for DIC microscopy); and (3) migration cells before mitosis/apoptosis. By comparing our co-restoration results with the single modality results in Fig.3.4(c), we define:(1) cells detected through co-restoration but not detected in phase contrast restoration are mitosis/apoptosis cells (shown in Fig.3.4(d) in red color); (2) cells detected through co-restoration but not detected in DIC restoration are flat migration cells (shown in Fig.3.4(d) in green color); (3) the rest cells detected through co-restoration are cells before mitosis/apoptosis.

3.2.5. Experiments. We collected microscopy images from both phase contrast microscopy and DIC microscopy on the same cell dish, and 500 pairs of microscopy images (1040×1388 resolution) with various cell densities were collected to validate our algorithm.

Qualitative Evaluation:

In Fig.3.5 we show the qualitative comparison between our co-restoration method and previous single-modality microscopy restoration methods. Fig.3.5(b)(c) show some examples of the phase contrast and DIC microscopy images, respectively. Restoration results of single-modality methods are shown in Fig.3.5(d)(e), where we can see some cells are not detected. Using our co-restoration approach, the challenging cases are handled well as shown in Fig.3.5(f). Cell classification is obtained by incorporating both co-restoration result and single-modality results, which is demonstrated in Fig.3.5(g).(Red: cells in mitosis or apoptosis; Green: cells in migration with flat gradients; White: cells before mitosis/apoptosis)

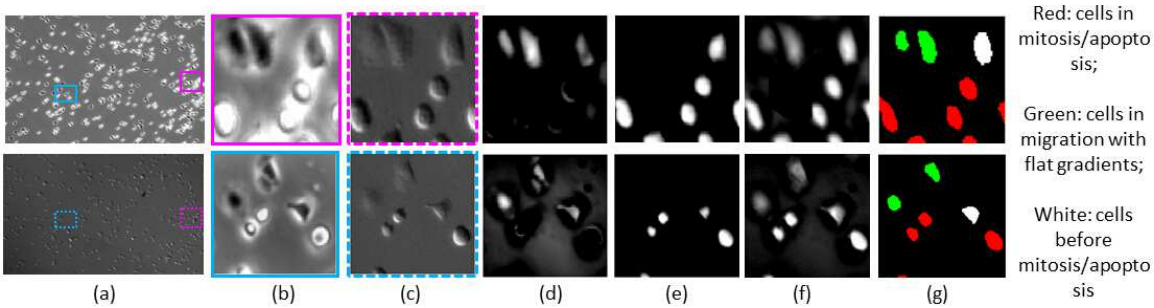

Figure 3.5. Comparison with different restoration approaches.

Quantitative Evaluation:

To evaluate the performance of our algorithm quantitatively, we manually labeled cell masks (include mitosis/apoptosis cells, cells before mitosis/apoptosis and flat migrating cells) in all microscopy images as ground truth. We define True Positive (**TP**) as cells segmented correctly, and False Positive (**FP**) as cells segmented mistakenly. Positive (**P**) and negative (**N**) samples are defined as cells and background, respectively. Precision and recall are calculated by: $Precision = \mathbf{TP}/(\mathbf{TP}+\mathbf{FP}), Recall = \mathbf{TP}/\mathbf{P}$, By adjusting different thresholds on the restoration images and comparing with ground truth, we obtain segmentation results with different *precision*s and *recall*s for 500 microscopy images, and

get an ROC curve. The results are shown in Fig.3.6 where our co-restoration outperforms single-modal restoration largely.

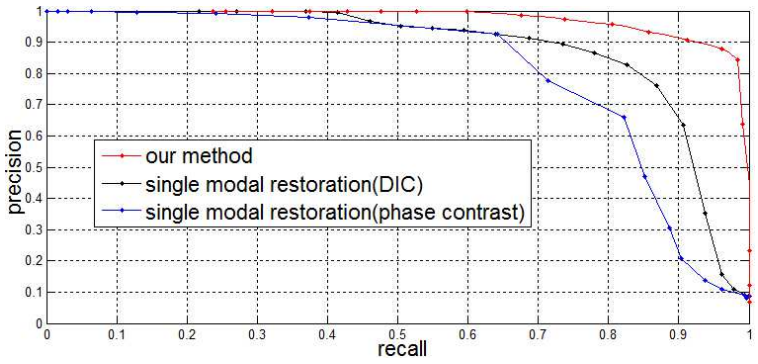

Figure 3.6. ROC curve of segmentation results by 3 approaches

We also evaluate the mitosis/apoptosis event detection accuracy (EDA), defined as $EDA = (|\mathbf{TP}_e| + |\mathbf{NE}| - |\mathbf{FP}_e|)/(|\mathbf{E}| + |\mathbf{NE}|)$, where True Positive of event detection ($\mathbf{TP}_e$) denotes the mitosis/apoptosis detected correctly, and False Positive of event detection ($\mathbf{FP}_e$) denotes the event detected mistakenly; $\mathbf{E}$ and $\mathbf{NE}$ define the mitotic/apoptotic cells and non-mitotic/apoptotic cells, respectively. By choosing the segmentation threshold at which outputs the best $\mathbf{F}$ score ($\mathbf{F} = 2 \cdot \frac{precision \cdot recall}{precision + recall}$), the EDA of our algorithm is 94.75%, which is also highly reliable considering that we are achieving very high segmentation result at the same time.

3.3. CELL SEGMENTATION USING STABLE EXTREMAL REGIONS IN MULTI-EXPOSURE MICROSCOPY IMAGES

In this section, we first give a brief overview of the multi-exposure cell segmentation problems we address in this work, as shown in Fig.3.7. We then explain the methodologies and discuss on the experiment results.

3.3.1. Overview of Methodology. Local region descriptors have been widely used for object segmentation, detection and identification. Among these methods, Mikolajczyk

and Schmid [49] revealed that the Maximally Stable Extremal Region (MSER) detector introduced by Matas et al. [50] performs very well on a wide range of experiments. MSERs denote a set of distinguished regions, which are defined by an extremal property of its intensity function in the region and on its outer boundary. In this section, we will first introduce our proposed methods of extracting Multi-exposure MSERs denoting cell and artifact regions in multi-exposure microscopy images. Then we will discuss how to classify these regions into cells and halos, for accurate cell segmentation and cell stage monitoring.

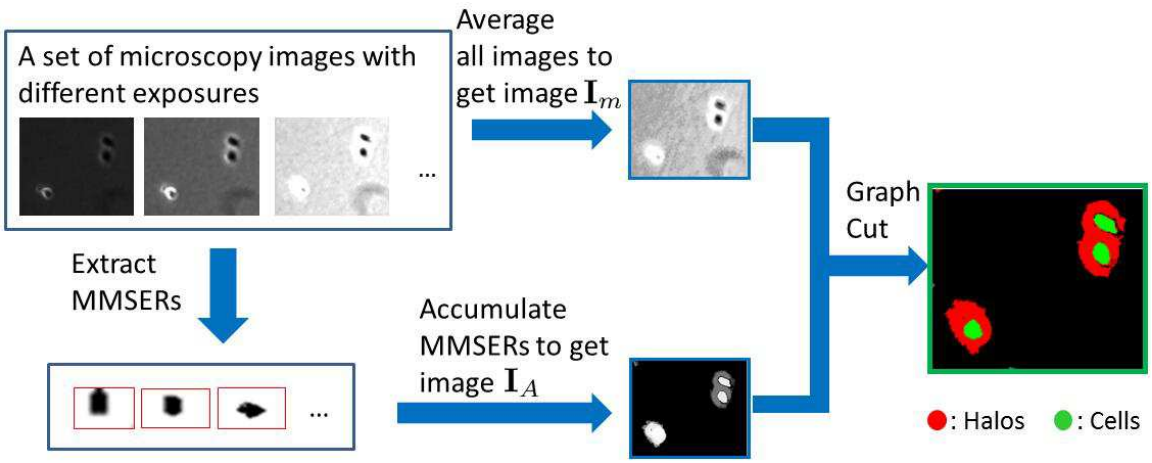

Figure 3.7. Overview of our system.

For our time-lapse microscopy image sequences, each set of multi-exposure images is taken every 5 minutes with a range of known exposure durations ([50, 100, 200, 250, 300, 350, 400, 500]ms, in total, about 2:15 seconds for capturing images per set. Due to the fact that cells are migrating very slowly in a dish, and the time taken for capturing each image set (2:15 seconds) is relatively very small compared to the time-lapse interval of 5 minutes, we can consider the irradiance signal for each cell is stable and there is no position change for each cell pixel within the time we capture each set of multiple exposure images. Thus, no further image registration procedure is needed. Zeiss Axiovision 4.7 microscope is used for microscopy image acquisition for our experiments.

The overview of our system is illustrated in Fig.3.7. Firstly, MMSER regions are extracted given the original image, which represent stable regions which does not easily change with the fluctuation of exposure times and thresholds. Then the MMSERs are accumulated to account for different regions representing cells and halos. Meanwhile, average image is also generated. Finally, by implementing a local graph-cut algorithm on both the MMSER accumulated image and the average image, we are able to obtain segmentation results on cells and halos. Further cell classification on different cell stages are implementable based on prior segmentation results.

3.3.2. Multi-exposure MSER Extraction. MSERs denote a set of distinguished regions that are detected in a gray scale image, which have relative stable cardinalities across different intensity thresholds. These regions are defined by an extremal property of the intensity function in the region R and on its outer boundary ∂R. For MMSERs we consider two types of extremal regions R which are defined by:

(1)$\forall p \in R, \forall q \in \partial R, I(p) > I(q)$ (maximum intensity region, denoting bright blobs)

(2)$\forall p \in R, \forall q \in \partial R, I(p) < I(q)$ (minimum intensity region, denoting dark blobs)

where $I(p)$ and $I(p)$ denote the pixel intensity at locations p and q, respectively.

These extremal regions are represented as connected components in binary images I_{exp}^{thr} which is obtained by:

$$I_{exp}^{thr}(p) = \begin{cases} 1, & \text{if } I_{exp}(p) > thr \\ 0, & \text{otherwise} \end{cases} \tag{3.15}$$

where I_{exp} is the image with exposure time exp, thr is the threshold and $thr \in [min(I_{exp}), max(I_{exp})]$. Given a set of multi-exposure phase contrast microscopy images $exp \in \{50,100,200,...,500\}$, each image can be used to produce a set of these connected components. Fig.3.8 illustrates an exemplary set of 3 input multi-exposure images, and some of the thresholded images which contain related connected components. Note that cells are mostly dark blobs, which

will be identified as minimum intensity region. We use $R_{exp}^{thr}(n)$ to denote the nth connected component region obtained from the *exp* exposure time with the threshold value *thr*.

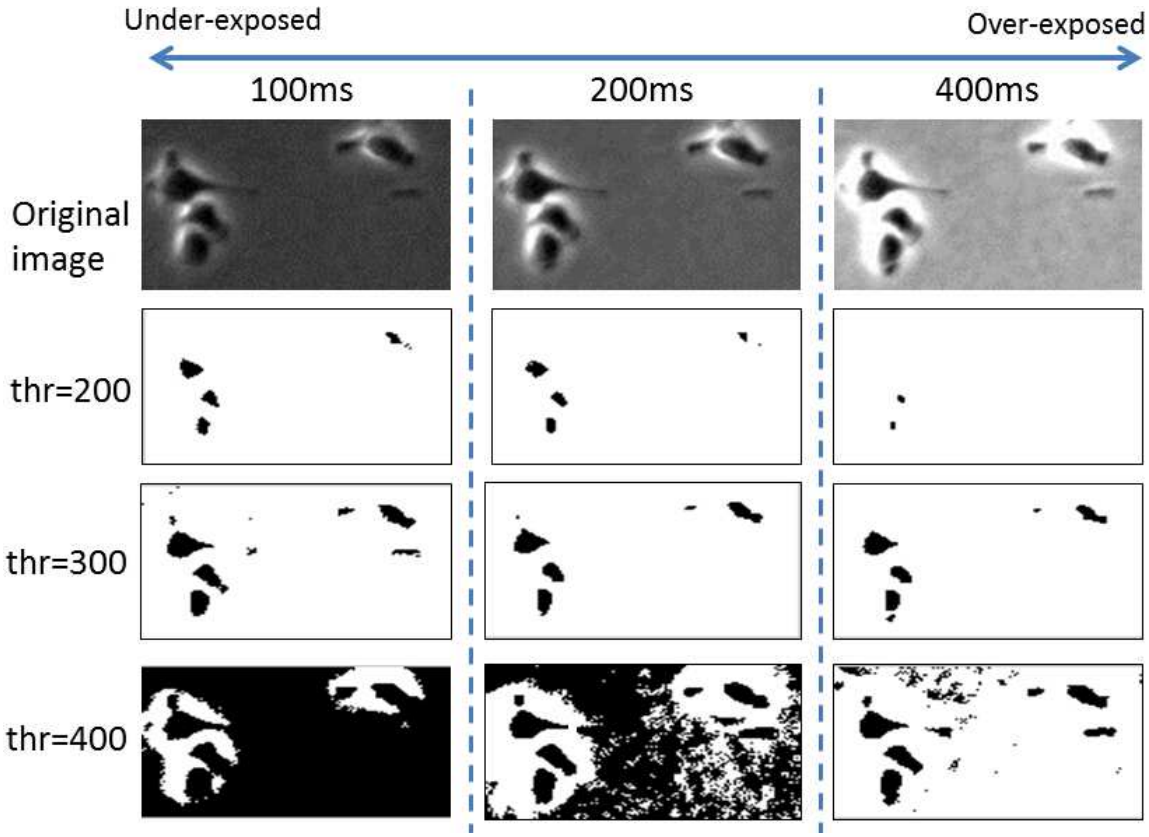

Figure 3.8. Multiple exposure images on the same cell dish (ms: millisecond) and binary images with different thresholds.

An inclusion relationship between a smaller region $R(m)$ and a larger region $R(n)$ ensures that $R(n)$ is a dilated region of $R(m)$:

$$R(m) \subset R(n) \Leftrightarrow \forall p \in R(m), p \in R(n). \tag{3.16}$$

Because we observe that for a connected component $R_{exp}^{thr}(m)$, we can either change the threshold *thr* by Δthr, or change the exposure time by Δexp, to increase/decrease its local region to obtained a dilated/eroded region $R_{exp\pm\Delta exp}^{thr_{\pm}\Delta thr}(n)$, as shown in Fig.3.9. Here Δthr denotes the step size of changing the intensity threshold value. Δthr denotes the step

size of changing the exposure time. In Fig.3.9 we use solid/dotted arrows from $R_{exp}^{thr}(m)$ to $R_{exp'}^{thr'}(n)$ to represent that $R_{exp'}^{thr'}(n)$ is an dilated/eroded region of $R_{exp}^{thr}(m)$.

For example in Fig.3.9, $R_{exp=200}^{thr=300}$ denotes a connected component region obtained from 200ms exposure time and $thr = 300$. By either decreasing the exposure time to 100ms, or using a higher threshold $thr = 400$, we can obtain connected components $R_{exp=100}^{thr=300}$ and $R_{exp=200}^{thr=400}$, respectively, which are dilated regions of $R_{exp=200}^{thr=300}$. Contrarily, by either increasing the exposure time to 400ms, or using a lower threshold $thr = 200$, we can obtain connected components $R_{exp=400}^{thr=300}$ and $R_{exp=200}^{thr=200}$, respectively, which are eroded regions of $R_{exp=200}^{thr=300}$. Thus, overall $R_{exp=200}^{thr=300}$ has four outward arrows, pointed to $R_{exp=100}^{thr=300}$, $R_{exp=200}^{thr=400}$, $R_{exp=400}^{thr=300}$ and $R_{exp=200}^{thr=200}$, respectively.

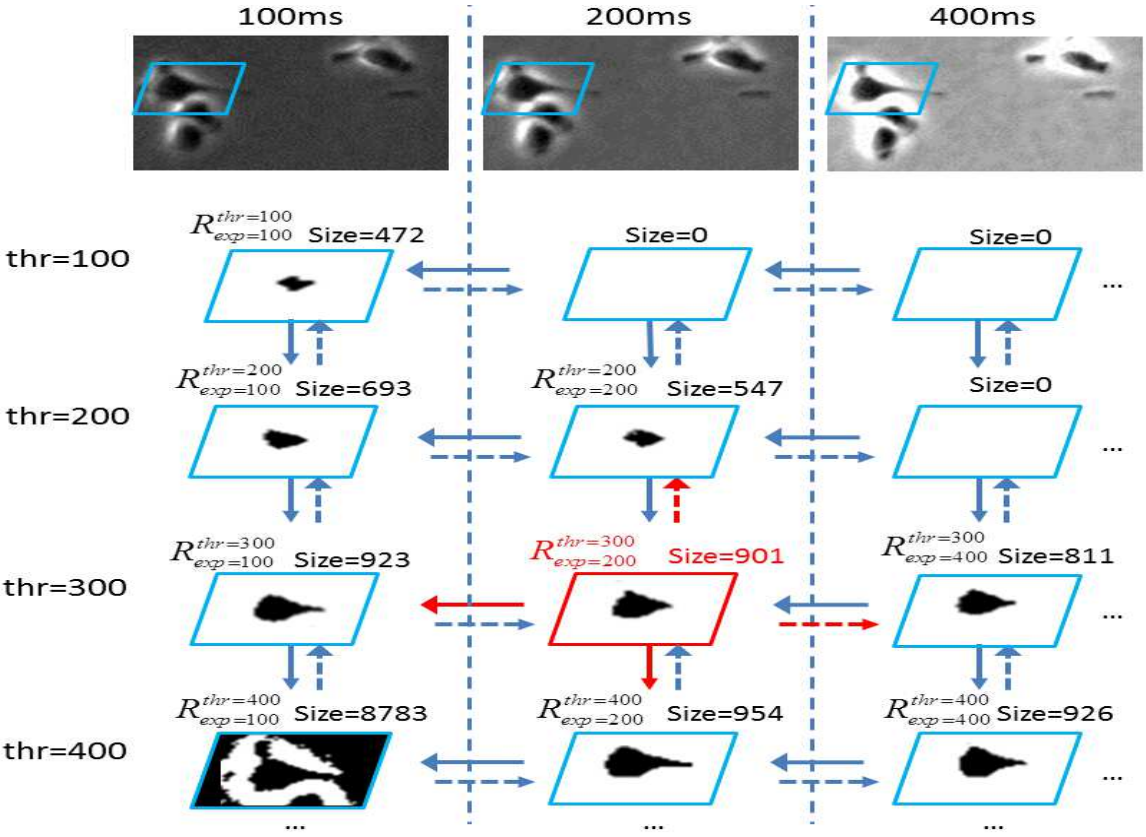

Figure 3.9. An Example of finding MMSER.

For each connected component $R(m)$ obtained from any exposure image, a variability value $\Psi(R(m))$ is calculated by:

$$\Psi(R(m)) = \frac{1}{|\mathfrak{N}(\mathfrak{m})|_c} \sum_{n \in \mathfrak{N}(\mathfrak{m})} \left| \frac{(|R(m)|_c - |R(n)|_c)}{|R(m)|_c} \right| \tag{3.17}$$

where $\mathfrak{N}(\mathfrak{m})$ denotes the set of all the connected components pointed by arrows from $R(m)$, and $|.|_c$ denotes the cardinality. Eq.3.17 reflects how much an extremal region will be affected by different thresholds or exposure times. Multi-exposure MSERs correspond to those connected components that have locally minimal variability values Ψ of the graph.

For example in Fig.3.9, $R^{thr=300}_{exp=200}$ has four outward arrows pointing to four connected components with size {954,923,811
,547}. So we can calculate $\Psi(R^{thr=300}_{exp=200}) = \frac{1}{4}[|954-901| + |923-901| + |811-901| + |547-901|]/901 \approx 0.144$. Similarly, we can calculate the Ψ of its four neighbors:$\Psi(R^{thr=200}_{exp=200}) \approx 0.729$, $\Psi(R^{thr=300}_{exp=100}) \approx 2.929$, $\Psi(R^{thr=400}_{exp=200}) \approx 2.764$, $\Psi(R^{thr=300}_{exp=400}) \approx 0.418$. Therefore, $R^{thr=300}_{exp=200}$ is an MMSER with local minimum Ψ, whose shape and region stay relatively stable and unaffected by different intensity thresholds and exposure times.

By applying the MMSER searching technique to a set of multi-exposure microscopy cell images on the same dish, we can extract a large set of MMSERs denoting cells, as well as their artifact (halos) regions by finding local minimum Ψ values.

3.3.3. Unsupervised Identification of Cell Regions. It is observed that in most exposures, cells appear to be darker than the background in phase contrast microscopy images, while halos appear to be brighter than the background. Therefore, by creating a new averaged image I_m (second row in Fig.3.10) via taking the mean of all exposure images, pixels at cell regions should have low intensities, and pixels at halo regions should have high intensities.

We also create a new accumulated image I_A (first row Fig.3.10), by counting the number of times each pixel appears in every MMSER extracted from a multi-exposure image set. For a specific pixel p, the intensity $I_A(p)$ represents the number of times pixel p appears in all MMSERs. Since cells have stable irradiance signal in all exposures compared to background and halos, cell regions should have steady output of MMSERs in all exposures, while halos usually have MMSERs in higher exposures only. Therefore normal cell regions should have higher pixel intensities in I_A, while in halo regions pixels should have lower intensities in I_A, as shown in Fig.3.11(b).

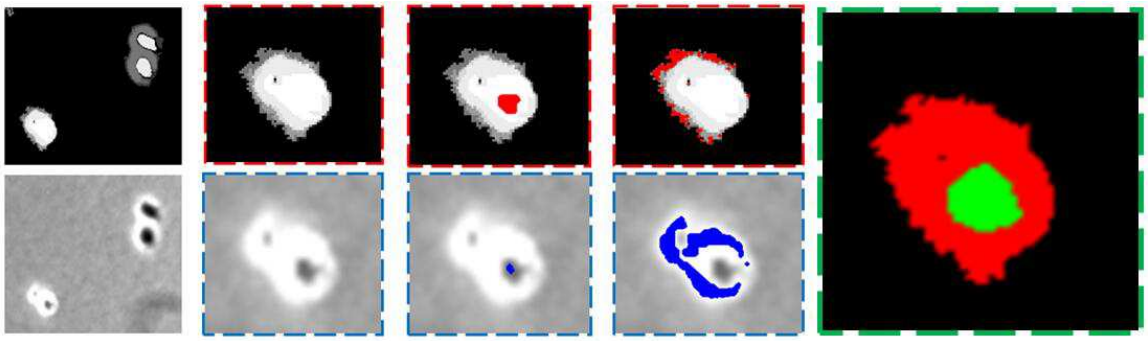

Figure 3.10. Examples of seeds selection for cells and halos. (a) Original averaged microscopy image I_m; (b)zoomed image of (a); (c) Seeds for cell regions; (d) Seeds for halo regions; (e) Cell-halo classification results by Graph-cut.

But for cells during mitotic/apoptotic stages, they usually become thicker and thus they have different phase retardations compared to migration cells. The halos of these cells will be much brighter and stable in all exposures. Therefore halos of these cells also have steady output of MMSERs in all exposures, which leads to high intensities in I_A. As shown in Fig.3.11(e), the mitosis cell has high intensities in both cell and halo regions in I_A.

Considering the observations above, we propose a local Graph-cut algorithm to implement the unsupervised cell-halo classification. The algorithm consists of 3 steps:

(1) A clustering procedure on I_A is undertaken for implementing the local Graph-cut inside each pixel cluster. Each non-zero connected components with their centroid

distances less than D are considered to be in the same cluster. In this thesis, D is set as 3 times the average diameter of cells for each data set. For example in Fig.3.11(a)(b), 4 clusters are found from I_A by clustering, and in Fig.3.11(d)(e), 2 clusters are obtained.

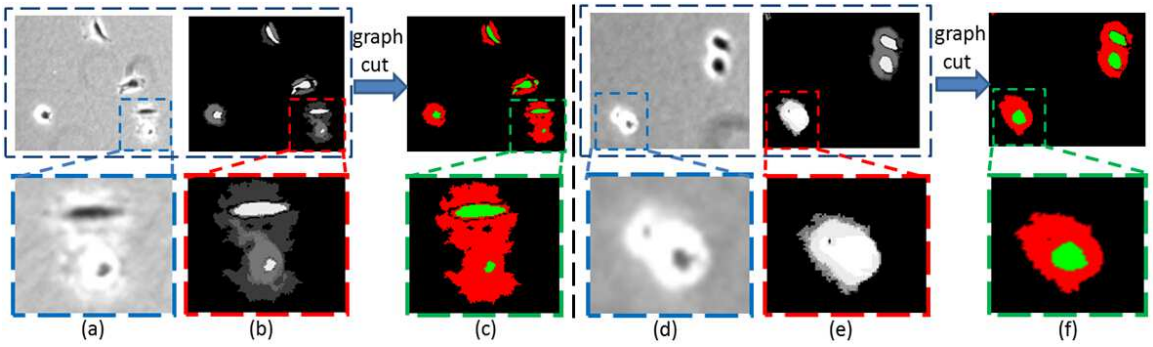

Figure 3.11. Examples of unsupervised classification between cells and halos. (a)(d) Original averaged microscopy image I_m; (b)(e) Acculumated MMSER image I_A for (a) and (d); (c)(f) Segmentation of cells and halos using Graph-cut.

(2) Inside each pixel cluster, we define **seeds for cells** as: (pixels with the highest intensity in I_A) $\cup$ (pixels with the lowest intensity in the averaged image I_m); Likewise we also define **seeds for halos** in each pixel cluster as: (pixels with the lowest intensity in I_A) $\cup$ (pixels with the highest intensity in I_m). Noted that in this step we only consider pixels inside each cluster that we found in step(1), so only cell and halo pixels are considered.

(3) Using the seeds, we apply the local Graph-cut inside each pixel cluster to identify cell pixels and halo pixels, with the energy function defined as:

$$E = \sum_{p \in V} E_p(x_p) + \sum_{(p,q) \in E} E_{p,q}(x_p, x_q) \tag{3.18}$$

where (V,E) defines an undirected graph of one cluster, whose nodes V correspond to pixels inside the cluster. E denotes the link set between neighboring nodes. $x_p \in \{0,1\}$ is the segmentation label of pixel p, where 0 and 1 correspond to the halos and the cells,

respectively. The energy function includes an unary cost of each node, and the pairwise cost between neighboring pixels.

The unary cost is defined as:

$$E_p(x_p) = (1 - x_p) * (-\ln P_h(p)) + x_p * (-\ln P_c(p)) \tag{3.19}$$

where $P_h(p)$ and $P_c(p)$ are the probability of pixel p being classified as halos and cells, respectively. The probabilities can be computed by fitting pixel p into Gaussian Mixture Models of halos and cells, which are built by the seeds of these two classes using their pixels' intensities in I_A and I_m.

The pairwise cost is defined as:

$$\begin{aligned} E_{p,q}(x_p, x_q) = \exp[-((&\frac{I_m(p) - I_m(q)}{max(I_m)})^2 \\ &+ (\frac{I_A(p) - I_A(q)}{max(I_A)})^2)/\sigma^2] \end{aligned} \tag{3.20}$$

where σ is the boundary sharpness parameter which controls the smoothness of pairwise term. The pairwise cost considers the smoothness in both the average intensity image I_m and the accumulated MMSER image I_A.

By implementing our Graph-cut algorithm in each cluster, we can distinguish between cells and halos as shown in Fig.3.11(c)(f). Halos are presented in red color, and cells are shown in green. We can notice that cell regions are well identified. Meanwhile, the size of surrounding halos can provide us with information about what stage a specific cell is currently in, since cells in mitosis/apoptotis create brighter and larger halos than regular migrating cells. In our next section, we will experimentally test our algorithm on microscopy cell segmentation tasks. Meanwhile, we also yield a basic criterion for evaluating cells' stage by halo inferring.

3.3.4. Experiments. We collected phase contrast microscopy images on 4 different cell dishes with low and high cell densities, and each set has 8 different exposure durations ([50 100 200 250 300 350 400 500]ms). In Fig.3.12 we show the qualitative comparison between our multi-exposure MSER cell segmentation approach with other methods. In Table.3.1 we show the quantitative results.

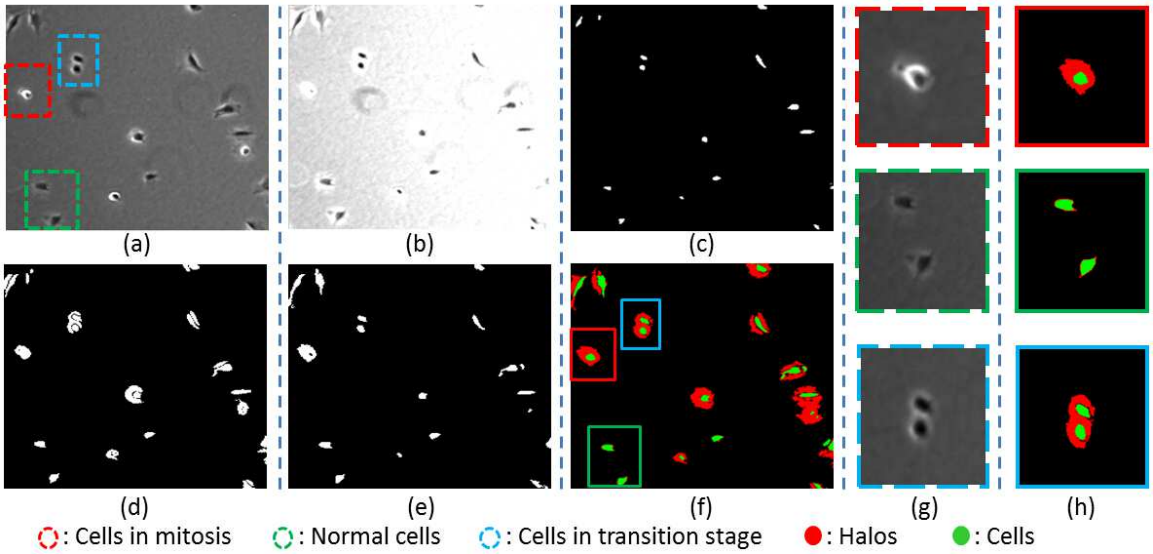

Figure 3.12. The Comparison of different cell segmentation methods. (a) Original image (200ms); (b) Original image (400ms); (c) Segmentation result by [1]; (d) MSER segmentation from (a); (e) MSER segmentation from (b); (f) Segmentation by our method; (g) Zoom-in of three types of cells from (a); (h) Segmentation result of (g) by our method.

Qualitative Evaluation:

Fig.3.12(d) and Fig.3.12(e) are the segmentation results from single exposure MSER segmentation with 200ms (Fig.3.12(a)) and 400ms (Fig.3.12(b)), respectively. Obvious mistakes can be easily noticed on halos and mitosis cell regions using single exposure methods. In Fig.3.12(c) we show the segmentation result obtained by the phase contrast restoration method introduced in [1], which encountered similar segmentation problem for mitotsis cells, exemplified by the cell in the red rectangle in Fig.3.12(a).

In Fig.3.12(f) we show the result by using our method. We can see that not only cells in all types are segmented accurately, but also the halos are identified which can inform us of the cell's current stages. As shown in Fig.3.12(g), we exemplarily pick three types of cells with different halo artifacts, which can be easily identified by our method shown in Fig.3.12(h). Considering that mitosis cells usually have large halos, and normal migrating cells have very small halos, we use the area ratio between the cell region and its surrounding halo as the criterion to decide what stage a specific cell is currently in.

Table 3.1. Cell segmentation accuracy of different methods.

SACC	dish1	dish2	dish3	dish4
Our method	0.996	0.994	0.971	0.947
Single-image (200ms) MSER segmentation	0.741	0.665	0.628	0.631
Optic based restoration [1]	0.974	0.974	0.956	0.849
Cell-sensitive segmentation [41]	0.993	0.994	0.975	0.918

Quantitative Evaluation:

We obtained ground truth cell masks (no halos considered) by multiple annotators who manually label cell masks in all microscopy images with exposure time 200ms. To reduce the inter-person variability, the intersection of their annotations is used as the ground truth for testing. We choose the Segmentation ACCuracy (SACC) to evaluate the performance of different methods, which is defined as: $SACC = (|TP| + |N_s| - |FP|)/(|N_s| + |P_s|)$ where P_s and N_s denote cell and background pixels, respectively. True positive (TP) denotes cell pixels segmented correctly and false positive(FP) denotes cell pixels segmented mistakenly. Table 1 compares the performance of different segmentation methods on 4 cell microscopy image sequences. The results show that our Multi-exposure MSER segmentation method achieves highly reliable results compared to other single-exposure methods and optics-based segmentation approaches.

We also experimentally evaluate the detection accuracy of mitosis cells and normal cells on 4 cell dishes. The Detection ACCuracy(*DACC*) is defined similar to segmentation accuracy, whose objects are mitosis cells instead of pixels. We classify cells with cell-halo ratio larger than 6.1 as those in mitosis stage, and cells with ratio less than 1.4 as normal cells (these optimal thresholds are chosen by cross-validation). Cells with ratio between 6.1 and 1.4 are classified as cells in the transition stage. In our experiments, the average DACCs of mitosis cells and normal cells are 0.979 and 0.966, respectively.

3.4. SUMMARY

In this section, we first introduce a novel cell segmentation approach by extracting Multi-exposure MSERs for local cell-halo classification. A set of variously exposed phase contrast microscopy images on the same cell dish are obtained to estimate different irradiance signals from cells and halos, which are later used for accurate cell segmentation and cell stage inference. The experimental results validate the reliability of our approach in high-accuracy cell segmentation and the capability in monitoring cell stages.

We also introduce a newly proposed novel cell image co-restoration approach by considering Differential Interference Contrast(DIC) and Phase Contrast microscopy images captured on the same cell dish. The challenges in restoring single modular microscopy images is overcome in our algorithm, by leveraging different weighting parameters to cells with different phase retardation and DIC gradient signals. The experimental results show that our approach achieve high quality cell segmentation and accurate event detection.

4. TRACKING OF BIOMEDICAL OBJECTS

4.1. INTRODUCTION

In biomedical applications which consecutive frames of images are provided, multi-object tracking is necessary for better quantitative behavior analysis of biomedical objects, after obtaining detection and segmentation results. In these vision-aided biomedical applications with their goal of uncovering hidden patterns of a complex biological process, high quality visual tracking algorithms are required to accurately track bio-specimens over a long period. However, in biomedical object analysis, it is common to face problems such as clutter, heavy occlusion, trajectory overlap, high similarity between objects and etc., which make our multi-object tracking much more challenging than usual cases.

To deal with these problems, we propose a new cascaded data association framework for multi-object tracking. At first, only data with high confident scores are associated into small tracklets in spatial-temporal domain, which is ensured to achieve high accuracy. Then a iterative global assignment task with constraints is yielded to further associated these short tracklets into longer trajectories denoting object trackers. At each iteration, spatial/temporal gating regions are increased at a fixed rate to link trajectories gradually. Finally all the trackers with complete trajectories is generated. On top of our cascaded data association approach, we also experimentally test the performance on multiple datasets, which validates the good performance of our algorithm.

Although our cascaded tracking approach is able to achieve very good experimental results, just like any other tracking methods, it is still extremely difficult to get 100 percent perfect tracking performance without error, due to numerous challenges in biomedical image data. But meanwhile, biological discovery and health diagnosis usually require high-quality tracking results for solid analysis and dependable medical treatment. Since

full accuracy by automated tracking is not reachable so far, to pursue solid scientific discovery and error-free health diagnosis, biologists and doctors are willing to exchange a small amount of their human effort to double-check the automated tracking results manually. Hence, it is worthy to consider how to incorporate least human efforts to better debug (verify and correct) the automated tracking results that is nearly perfect, which leads us to the following three problems:

(1) Guidance problem: Human labor is expensive so we cannot afford to manually check every object's trajectory frame by frame. *How to find out which tracking data are error-prone and have more influence on others, that are worth to be checked by human?*

(2) Collaboration problem: Checking tracking data on specimens captured over months with thousands of frames is too tedious for a single person. *How can we integrate the crowdsourcing to check the data collectively?*

(3) Propagation problem: A tracking error found by a human may have many analogous errors in the tracking data and the error can also affect its nearby data. *How can we propagate the costly human correction to other unchecked data and automatically correct similar errors so human burden is alleviated and the convergence to the best tracking accuracy is accelerated?*

Considering these problems, we design a recommender system framework for tracking error debugging(verifying and correcting) to replace tedious manually debugging process.

In this section, we first introduce our cascaded data association approach in Section 4.2. Tracklets with the highest confidence are first generated, after which longer tracklets are gradually associated with increasing gating region. Feature vectors will be recorded during this cascaded process. Then in Section 4.3, we investigate how to debug automated object tracking results with humans in the loop. A novel iterative recommender system with correction propagation is proposed to assist multiple human annotators to debug tracking

results in an effective, collaborative and efficient way. Tracking data that are highly erroneous are recommended to annotators based on their propagation influence and debugging histories. Different annotators debug the tracking data independently and their debugging results are collected for joint correction propagation. Since an error found by an annotator may have many analogous errors in the tracking data and the errors can also affect their nearby data, we propose a correction propagation scheme to propagate corrections from all human annotators to unchecked data which efficiently reduces human efforts and accelerates the convergence to perfect tracking accuracy in Section 4.4. Our proposed approach is evaluated on three challenging biological datasets. The quantitative evaluation and comparison validate that the recommender system with correction propagation is effective and efficient to help humans debug tracking results generated by automated tracking algorithms, as discussed in Section 4.5.

4.2. TRACKLET-BASED OBJECT TRACKING

In this section we focus on the object tracking method based on tracklet association. Experiments are shown after the discussion on methodology details.

4.2.1. Overview of Cascaded Data Association in Object Tracking. In this section, we demonstrate our cascaded data association method with fine-to-coarse gating region control based on which a feature vector generation method will be introduced in the following section.

We proposed a cascaded data association approach with “tracking-by-detection” strategy to deal with broken tracklets caused by fast motion or occlusion , which consisting of four steps, as illustrated in Fig.4.1: (1) The original image sequence with N frames(N is very large) is divided into multiple subsequences (Fig.4.1(a)). Each subsequence consists

of n images(n is relatively small to N, and efficient to process). There are 2 main reasons that we apply this divide-and-conquer technique here: (a) reduce a large-scale CPU-burning optimization problem into many small-scale solvable optimization problems, and (b) enable the fast parallel processing in each subsequence. For example, a video of one week monitoring may result in billions of tracklets (short trajectories) and make it unsolvable on any normal computers or workstations to optimally link all the tracklets globally with billions of variables. (2) Detected flies between consecutive frames are matched into tracklets. Due to the camouflage, occlusion and fast motion, some flies are not detected occasionally, resulting in broken tracklets in the spatiotemporal domain (Fig.4.1(b)). (3) The short tracklets within each subsequence are linked into long trajectories by iteratively solving a Linear Assignment Problem (LAP) with fine-to-coarse gating region control in the spatio-temporal domain (Fig.4.1(c)). (4) The long trajectories of all subsequences are sequentially connected by solving a series of LAPs to form the complete trajectories of all flies (Fig.3(d)).

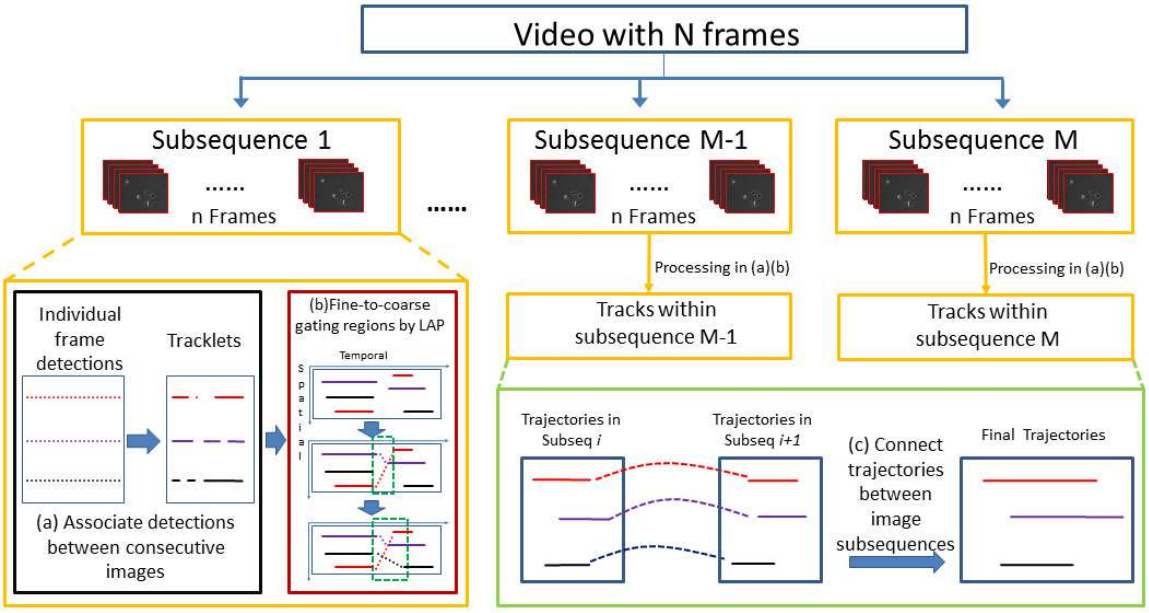

Figure 4.1. Cascaded Data Association with Fine-to-coarse Gating Region Control.

4.2.2. Tracklet Generation. Assuming that we already have detected objects in each individual frames. In this section, we leverage the object detection results from previous approaches such as segmentation based on restored images [1, 41] or detection by adaptive local binary patterns which is discussed in previous section. Then, detected objects between consecutive frames are matched into tracklets incrementally. Every detected object candidate in a frame is represented as a node with corresponding features such as color distribution, gradient histogram, object shape, location, etc. We denote $F_i^t = [\mathbf{f}_1(n_i^t), ..., \mathbf{f}_K(n_i^t)]$ as the vector of K features of node i in frame t. The dissimilarity cost between a pair of nodes in two consecutive frames is computed as:

$$c(n_i^t, n_j^{t-1}) = \begin{cases} \frac{1}{K}\sum_{k=1}^{K} \frac{\left\|\mathbf{f}_k(n_i^t) - \mathbf{f}_k(n_j^{t-1})\right\|}{\Delta_k}, \\ \qquad \text{if} \left\|\mathbf{f}_k(n_i^t) - \mathbf{f}_k(n_j^{t-1})\right\| \leq \Delta_k, \forall k \in [1, K] \\ \infty, \text{otherwise} \end{cases} \tag{4.1}$$

where $\|\cdot\|$ is the L_2 norm and Δ_k is the normalization factor of the kth feature. For example, when $\mathbf{f}_k$ is the location feature, Δ_k controls the spatial gating region (i.e., the size of local neighborhood to search a node's correspondence between consecutive frames).

To match objects between consecutive frames and construct short tracklets, we use the Hungarian Algorithm [23] with its cost function defined as:

$$c(F_i^t, F_j^{t-1}) = \|L(F_i^t) - L(F_j^{t-1})\| + \|L(F_i^t) - \hat{L}(F_j^{t|t-1})\| \tag{4.2}$$

where function $L(x)$ retrieves the location element from the feature x. $\|\cdot\|$ is the L_2 norm. $\hat{L}(F_j^{t|t-1})$ is the predicted location (with linear prediction model) of the jth object in frame t based on its previous trajectories.

After sequentially performing the matching on every pair of consecutive frames, we generate many tracklets within a subsequence (Fig.4.1(a)).

Note that, (1) we use small gating regions in the frame-by-frame assignment, which generates tracklets with less errors but also causes short broken tracklets when objects move beyond the gating regions; (2) the Hungarian algorithm solves the 1-to-1 bipartite assignment problem but it can not solve the 2-to-1 or 1-to-2 assignment problem when there exists object merging or division, which causes broken or wrong connections among tracklets; (3) it is usually difficult to have perfect detection results for every frame, hence false positives and miss detections will cause broken or wrong connections among the tracklets (Fig.4.1(a)). In the following two subsection, we describe how to gradually link the short tracklets into longer object trajectories.

4.2.3. Fine-to-Coarse Association of Tracklets within Each Subsequence. We denote the ith tracklet $\mathbf{T_i}$ by its node representation, $\mathbf{T}_i = \{n_i^{s_i}, n_i^{s_i+1}, ..., n_i^{e_i}\}$. $n_i^{s_i}$ and $n_i^{e_i}$ represent the nodes in the start frame s_i and the end frame e_i, respectively. We define five types of hypotheses on tracklets:

(1) Migration (1-to-1 case): the head of a tracklet is associated with a tail of another tracklet. The cost of a Migration Hypothesis is:

$$c(\mathbf{T}_i \rightarrow \mathbf{T}_j) = \frac{1}{m+1} \sum_{k=1}^{m+1} \frac{\left\| \mathbf{f}_k^+(n_i^{e_i}) - \mathbf{f}_k^+(n_j^{s_j}) \right\|}{\Delta_k^+} \quad \text{when} \forall k \in [1, m+1], \mathbf{f}_k^+(n_i^{e_i}) - \mathbf{f}_k^+(n_j^{s_j}) \leq \Delta_k^+ \tag{4.3}$$

where $\mathbf{f}^+(n_i^{e_i}) = [\mathbf{f}(n_i^{e_i}), \theta(n_i^{e_i-\delta})]$, $\mathbf{f}^+(n_j^{s_j}) = [\mathbf{f}(n_j^{s_j}), \theta(n_j^{s_j+\delta})]$ are the feature vectors for end and start node of a tracklet respectively. $\theta(\cdot)$ is is the orientation of the tracklet considering δ frames of motion. Δ_k^+ is the extended normalization vector with the same length as $\mathbf{f}^+(\cdot)$.

(2) Division (1-to-2 case): the tail of a tracklet is associated with heads of two tracklets. The cost of a Dividing Hypothesis is listed in Eq.4.4, where the first two terms on the right

of Eq.4.4 are the costs of associating a parent node with children nodes. The third term represents the cost of hypothesizing two nodes are siblings. Similarly, the hypothesis cost can be generated into (1-to-n) and (n-to-1) cases.

Constraint between two head tracklets are imposed because we assume that two children have to be close after mitosis or other dividing activities.

$$\begin{aligned} c(\mathbf{T}_i \rightarrow (\mathbf{T}_{j_1}, \mathbf{T}_{j_2})) = & c(\mathbf{T}_i \rightarrow \mathbf{T}_{j_1}) \\ & + c(\mathbf{T}_i \rightarrow \mathbf{T}_{j_2}) + c(\mathbf{n}_{j_1}^{s_{j_1}}, \mathbf{n}_{j_2}^{s_{j_2}}) \end{aligned} \tag{4.4}$$

(3) Combining (2to1 case): the tail of two tracklet is associated with heads of a tracklets. The cost of a Dividing Hypothesis is:

$$\begin{aligned} c((\mathbf{T}_{i_1}, \mathbf{T}_{i_2}) \rightarrow \mathbf{T}_{j_2}) = & c(\mathbf{T}_{i_1} \rightarrow \mathbf{T}_j) \\ & + c(\mathbf{T}_{i_2} \rightarrow \mathbf{T}_j) + c(\mathbf{n}_{i_1}^{e_{i_1}}, \mathbf{n}_{i_2}^{s_{i_2}}) \end{aligned} \tag{4.5}$$

Constraint between two tail tracklets are imposed similar to 1to2 case.

(4) Disappearing (1to0 case): the tail of a tracklet is not linked to any other tracklets. The cost of a Disappearing Hypothesis is:

$$c(\mathbf{T}_i \rightarrow \phi) = \begin{cases} \frac{\mathbf{d}^{(t)}(n_i^{e_i}, e)}{\Delta t}, & \text{if } \mathbf{d}^{(t)}(n_i^{e_i}, e) \leq \Delta t, \\ \frac{\mathbf{d}^{(s)}(n_i^{e_i})}{\Delta s}, & \text{if } \mathbf{d}^{(s)}(n_i^{e_i}) \leq \Delta s \\ \eta, & \text{otherwise.} \end{cases} \tag{4.6}$$

$\mathbf{d}^{(t)}(n_i^{e_i}, e)$ denotes the temporal distance from the ending node of $\mathbf{T}_i$ to the last frame. $\mathbf{d}^{(s)}(n_i^{e_i})$ denotes the spatial distance from the ending node of $\mathbf{T}_i$ to the image boundary. η is a large constant. Δs and Δs represent the gating regions for spatial and temporal domain.

(5)Appearing (0to1 case): similar to 1to0 case, we define the 0to1 case which represents the tracklets which serve as heads of each trajectories:

$$c(\phi \rightarrow \mathbf{T}_i) = \begin{cases} \frac{\mathbf{d}^{(t)}(n_i^{s_i},s)}{\Delta t}, & \text{if } \mathbf{d}^{(t)}(n_i^{s_i},s) \leq \Delta t, \\ \frac{\mathbf{d}^{(s)}(n_i^{s_i})}{\Delta s}, & \text{if } \mathbf{d}^{(s)}(n_i^{s_i}) \leq \Delta s \\ \eta, & \text{otherwise.} \end{cases} \tag{4.7}$$

$\mathbf{d}^{(t)}(n_i^{s_i},s)$ denotes the temporal distance from the starting node of $\mathbf{T}_i$ to the first frame. $\mathbf{d}^{(s)}(n_i^{s_i})$ denotes the spatial distance from the starting node of $\mathbf{T}_i$ to the image boundary. η is a large constant.

The association relationship is determined by solving the LAP problem:

$$\arg\min_{\mathbf{a}} \mathbf{c}^T\mathbf{a}, \;\; s.t. \; \mathbf{Q}^T\mathbf{a} = \mathbf{1} \tag{4.8}$$

Where $\mathbf{Q}$ is a constraint matrix of size M by $2N_x$. M is the number of hypothesis, N_x is the number of tracklets. $\mathbf{Q}(i,j) = 1$ only if jth tracklet is considered in the ith hypothesis. The constraint $\mathbf{Q}^T\mathbf{a} = \mathbf{1}$ makes sure that one tracklet is only associated once on their head or tail. a is a M by 1 vector, where x_k indicates that the kth hypothesis is selected to be true in the optimization solution.

Algorithm 3 Algorithm 3: Iterative Tracklet Association.

Initialization: $k \leftarrow 0$, $\Delta_t^{(0)} = \Delta_{t_{start}}, \Delta_s^{(0)} = \Delta_{s_{start}}$;

Repeat:
    Solve the LAP problem in Eq.4.8;;
    $k \leftarrow k+1$**;;**
        $\Delta_t^{(k)} = q * \Delta_t^{(k-1)}$**;**
        $\Delta_s^{(k)} = q * \Delta_s^{(k-1)}$**;**
Until no change happens to the association.

Then we propose a fine-to-coarse algorithm as below to increase the gating regions gradually and iteratively link tracklets within a subsequence, as shown in (Fig.4.1(b)):
where q controls the increasing rate of gating region. $\Delta_{t_{start}}$ and $\Delta_{t_{start}}$ are two constant values which serve as the starting thresholds for Δ_t and Δ_s. The iterative algorithm solves the most confident associations first and then gradually solve the less-confident ones, which helps to handle the fast motion and occlusion (by increasing spatial or temporal threshold first).

Finally the tracklets in each subsequence is combined together, on which cascaded association is performed again to form final trajectories, as shown in (Fig.4.1(c)).

4.2.4. Cascaded Association with Feature Vector Recording. Following our previous work, we adopt the "tracking-by-detection" strategy to track multi-objects to serve our work in this paper. But the data association methods are not limited to Hungarian algorithm or linear programming. Note that the detected objects in individual frames can be treated as special tracklets whose trajectory lengths are one. Therefore, the whole tracking problem is converted into a tracklet association problem.

In the initial frame-by-frame association, we use small gating region $\beta = [\Delta_t,\ \Delta_s]$, which generates tracklets with less errors but also causes short broken tracklets when objects move beyond the gating regions. Initially the distance gating region (Δ_t) is set as 10% of the average object moving displacement between two frames and the temporal gating region (Δ_s) is set as one. Then, the gating region β is increased gradually and initial short tracklets are linked into longer and longer ones, as shown in Fig.4.2(a) which has a similar mechanism as Fig.4.1(b). The gating region is increased by 30% between two consecutive stages in our experiments.

In the initial frame-by-frame association, we use small gating region $\beta = [\Delta_t,\ \Delta_s]$, which generates tracklets with less errors but also causes short broken tracklets when objects move beyond the gating regions. Initially the distance gating region (Δ_s) is set as 10% of the average object moving displacement between two frames and the temporal gating

region (Δ_t) is set as one. Then, the gating region β is increased gradually and initial short tracklets are linked into longer and longer ones, as shown in Fig.4.2(a) which has a similar mechanism as Fig.4.1(c). The gating region is increased by 30% between two consecutive stages in our experiments.

Association-based features are recorded during each stage of the fine-to-coarse incremental association, only for those nodes who are involved in the current stage association. Features need to be recorded are listed in Table.4.1. Given node n_k^t of tracklet $\mathbf{T}_k$ in frame t, it has features related to both object detection ($\mathbf{f}(n_k^t)$) and tracklet association. We summarize the features in Table 1. $|\cdot|$ denotes the cardinality of a set and $\delta(\mathbf{T_k}, \mathbf{T_j})$ is an indicator function ($\delta(\mathbf{T_k}, \mathbf{T_j}) = 1$ when $\mathbf{T_j}$'s head is within the gating region of the tail of $\mathbf{T_k}$; $\delta(\mathbf{T_k}, \mathbf{T_j}) = 0$, otherwise).

Table 4.1. Features for nodes in tracklets.

$\mathbf{f}(n_k^t)$	features obtained in object detection (e.g., location, color, shape, etc.)
$c_s(n_k^t)$	the cost of the hypothesis involving n_k^t
$c_g(n_k^t)$	number of times the gating-region has been increased on node n_k^t
$l(n_k^t)$	length of the shortest tracklet among n_k^t's gating region
$c_t(n_k^t)$	# of possible tracklets tailing n_k^t $\left\|\left\{\mathbf{T_j} : \delta(\mathbf{T_k}, \mathbf{T_j}) \neq 0, j \neq k\right\}\right\|$
$c_h(n_k^t)$	# of possible tracklets heading n_k^t $\|\{\mathbf{T_i} : \delta(\mathbf{T_i}, \mathbf{T_k}) \neq 0, i \neq k\}\|$
$d(n_k^t)$	degree of n_k^t $d(n_k^t)$=$c_t(n_k^t) + c_h(n_k^t)$

We use one example to explain how to compute the association-based features. In Fig.4.2(b)(c), $c_s(a)$ and $c_s(w)$ are the linking cost of $a \rightarrow w$. $c_s(b)$, $c_s(x)$ and $c_s(y)$ are the linking cost of $b \rightarrow (x,y)$. Since β has been increased only once from the 1^{st} stage to the 2^{nd} stage, $c_g(\cdot)$ for all nodes are 1. $l(b)$ is the length of the shortest tracklet within b's gating region, so for node b it is the length of the tracklet with starting node x, which is 2.

$c_t(\cdot)$ and $c_h(\cdot)$ compute the degrees of corresponding nodes in the association hypothesis graph (Fig.4.2(c)), thus $c_t(a) = 1$, $c_t(b) = 2$, $c_h(w) = 1$; $c_h(x) = 1$, and $c_h(y) = 1$.

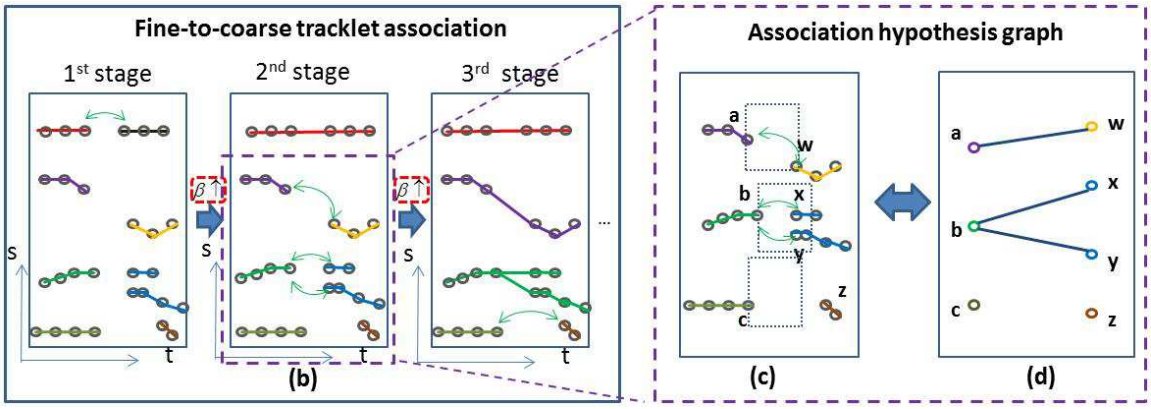

Figure 4.2. Multi-object tracking.

$c_s(n_k^t)$ stores the latest association cost involving the node. The higher $c_s(n_k^t)$ is, the more unreliable the association happened on node n_k^t is. The motivation of considering $c_g(n_k^t)$ as one of the features of node n_k^t is that we want to evaluate at which stage node n_k^t is associated to the longest possible trajectory at last. If the stage counter is high, which means the gating region has been increased many times, the association on node n_k^t is more likely to be a mistake. Very short tracklets near node n_k^t are highly possible to be false positives and associating them with node n_k^t is prone to cause errors, thus we consider $l(n_k^t)$ as one feature. When there are more association possibilities around node n_k^t, the association on node n_k^t may be more erroneous, hence we consider $c_t(n_k^t)$ and $c_h(n_k^t)$ as features.

After iteratively running the tracklet linking and feature vector recording until no changes occur in association, the tracking data as well as the feature vectors are forwarded to the recommender system to further improve the tracking accuracy, as discussed in next section.

4.3. RECOMMENDER SYSTEM

When the tracking algorithms meet the bottleneck, human effort is the only way to ensure reliability in tracking data. In this section we will explain how our recommender system can guide human to correct tracking result and maximized the correction influence to other tracking data.

4.3.1. Overview. Fig.4.3 shows the overview of our recommender system with the following steps:

(1) Initialization: all nodes in the large node pool is divided into 4 subsets by k-means clustering based on their feature vectors. Each user selects a same amount of positive (nodes with tracking errors) and negative (nodes with tracking errors) nodes from each subsets independently. All other nodes unselected by any user are transferred to the uncertain node pool. The consistence of annotations will be checked (discussed in Section 3.3 later). Note that this manual initialization step only needs to be done once.

(2) User Profile Learning: Each user's profile parameters are learned from the positive and negative node sets.

(3) Score Computing: The recommendation on every node in the uncertain node pool by every user is computed by the user's profile and node's feature vector.

(4) Recommendation: Top-ranked recommendations regarding recommendation scores and node's degree are sent to users for verification and correction.

(5) Correction Propagation: The corrections made by human are automatically propagated to other uncertain nodes and their feature vectors are updated accordingly.

(6) Data Relocation: The nodes in uncertain node pool after correction propagation are either relocated to positive/negative sets of users for updating users' profiles or still maintain in the uncertain pool if not affected by the correction propagation.

The process iterates until the uncertain node pool is empty.

Note that in the initialization step, we yield a k-means clustering before initial human selection of nodes for training. We choose 4 subset to better represent all the nodes in various tracking conditions and feature classes, though it does not necessarily represent 4 types of nodes. The motivation is to avoid representation bias in training, or simply focusing on a few types of tracking errors and ignore the others.

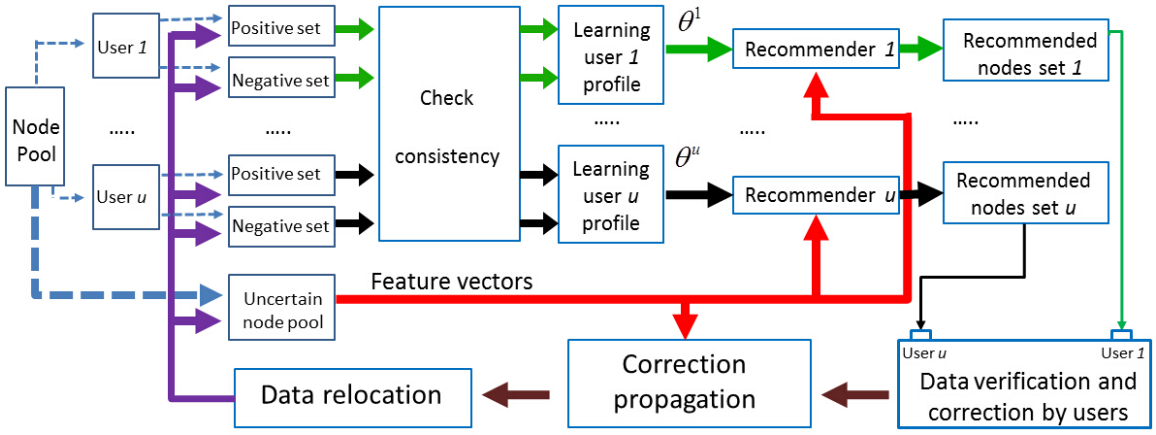

Figure 4.3. Workflow of our recommender system with correction propagation.

4.3.2. Learning User's Preference and Recommendation. We formulate the following least square problem to learn the profile parameters of annotator, $\theta^{(u)}$, who can verify automated tracking results and correct corresponding errors

$$\theta^{(u)} = \arg\min_{\theta} \sum_{i} \lambda_i \left(\theta^{(u)^T} \psi\left(\mathbf{F}^{(i,u)}\right) - \mathbf{y}^{(i,u)} \right)^2 + \|\theta^{(u)}\| \tag{4.9}$$

where $\mathbf{F}^{(i,u)}$ is the feature vector (Table 1) of node i in user u's positive/negative training node sets. $\psi(\cdot)$ is the kernel function (we use linear kernel in this paper). $\theta^{(u)}$ actually defines the maximum-margin hyperplane that classifies $\mathbf{F}^{(i,u)}$'s according to their class labels $\mathbf{y}^{(i,u)}$ provided by annotator u. The label of each node represents if the node is correctly tracked or not. λ_i is the weight for each sample. Initially, the weights are equal. During the

subsequent recommendations, the weights for inconsistent labeled samples will be doubled at each iteration.

After learning user u's profile, recommendation on any node j in the uncertain node pool to user u is computed by

$$\theta^{(u)^T} \psi\left(\mathbf{F}^{(j,u)}\right) \tag{4.10}$$

A small set of nodes (e.g., 40 nodes) from the uncertain node pool which have high suspicious scores (i.e., high probability of tracking errors) added to a **CheckList**, as candidates to be sent to user u for verification and correction at each stage.

The profiles of different users are learnt independently hence different sets of nodes are recommended for different users to verify and correct, but their correction nodes are collected together for efficient correction propagation (Section 4). The motivation to use personalized model in this recommender system is because different annotators may check the automated tracking results from different perspectives and they may focus on different portions of the huge set of tracking results. The personalized recommendation can provide annotators tasks which they are likely to do well on. The personalized model may introduce a bias (e.g., an annotator sees only one type of error or she choose one type of task intentionally based on her personal preference), but the crowdsourcing with different preferences (e.g., Amazon Mechanical Turk) should be able to complement each other.

4.3.3. Solving Duplicated or Inconsistent Annotations. Our system is capable of multi-agent interaction in annotations. As a matter of fact, duplicated, inconsistent annotations, even malicious human annotations are possible from the crowdsourcing.

If the duplicated annotations are consistent, this may waste human efforts since multi-annotators may have the same personalized models (in real world we believe it is very rare to have two persons with exactly the same model). To mitigate the duplication issue, the original tracking data are divided into non-overlap sub-sections for initialization,

which avoids the duplicated labeling from the beginning. In the subsequent iterations of learning annotators' preferences, if two annotators happen to have the same training sets, we will randomly swap some samples in their training sets with others recommended from the uncertain node pool.

When an inconsistent node label is found compared to its labels given by other human annotators (i.e., erroneous annotations), all annotators will be required to annotate this particular node again. Consensus is made by the majority labeling from all annotators.

4.4. CORRECTION PROPAGATION

Since all the nodes interact with their neighbors, the correction on some nodes can help correct their neighboring nodes. Therefore, we propose a correction propagation approach to spread the correction information around corrected nodes in the uncertain node pool, making the debugging more efficient.

Algorithm 4 Algorithm 4: Propagation Set Detection.

Input: Association Hypothesis Graph **G**, correction set **R**;
Output: node set $\mathbf{V}_{Propagation}$;
Initialization: queue $\mathbf{Q} \leftarrow \mathbf{R}$; $\mathbf{V}_{Propagation} \leftarrow \mathbf{R}$;

1: **Repeat:**
2: $t \leftarrow \mathbf{Q}.dequeue$**;//get the first element of the queue**
3: **for all edges of node t in G,** e**;**
4: $v \leftarrow \mathbf{G}.adjacentVertex(t,e)$;
5: **if** $v \notin \mathbf{V}_{Propagation}$**, then add** v **to** $\mathbf{V}_{Propagation}$**, enqueue** v **into Q;**
6: **end if;**
7: **end for;**
8: **Until Q is empty**
9: **Return** $\mathbf{V}_{Propagation}$**;**

First, we develop a graph-based Propagation Set Detection (PSD) algorithm. In the association hypothesis graph G, each vertex is a node in the node pool and an edge exists between two nodes if and only if there is an association hypothesis involving both of the

nodes in the data-association step. Given a set of nodes **R** corrected by multi-annotators, we detect the propagation set by Algorithm 4 below:

Algorithm 5 Algorithm 5: Iterative Recommender System with Correction Propagation.

Input: node set **V** of all nodes and their features;
Initialization: **Uncertain Node Pool=V**; **CheckList=**$\emptyset$;
Positive Set$\leftarrow$ Pick μ nodes with errors in tracking data;
Negative Set $\leftarrow$ Pick μ nodes without errors in tracking data;

Repeat:
  Update RecommenderProfiles using Eq.3;
  Compute the scores of nodes in the Uncertain Node Pool by Eq.4.10;
  for all node v $\in$ **V**
    if node score of v $> \omega$**;**
      add v to CheckList;
    else if node score of v $< \omega/2$**;**
      add v to Negative Set;
    end if;
  end for;
  sort CheckList by each nodes' degree $d(n_k^t)$ **in hypothesis graph;**
  for the top μ **nodes on CheckList;**
    recommend for human check;
    if node has tracking error;
      add the nodes' feature vectors into Positive Set, correct their associations, add their corrected feature vector into Negative Set;
    else add the nodes' feature vectors into Negative Set;
    end if;
  end for;
  Find $\mathbf{V}_{\textbf{Propagation}}$ **using Algorithm 1;**
  Implement data-association algorithm within $\mathbf{V}_{\textbf{Propagation}}$ **where human corrections are added as additional hard constraints;**
  Uncertain Node Pool $\leftarrow$ **Uncertain Node Pool- Positive Set** $\cap$ **Negative Set** $\cap$ $\mathbf{V}_{\textbf{Propagation}}$**;**
until Uncertain Node Pool is empty.

By implementing this PSD algorithm, we find all the nodes influenced by the current corrections in automated tracking data. We denote the affected nodes and corrected nodes as set $\mathbf{V}_{Propagation}$. Using the corrections as hard constraints, we perform the tracklet association problem (e.g., linear programing) on $\mathbf{V}_{Propagation}$, which updates the tracklet

association and corresponding node features, i.e., automatically propagates correction information to nodes close to the corrected nodes. While the recommender system runs iteratively, this correction propagation performs like the Butterfly Effect and sweeps gradually over the entire node pool.

After correction propagation, data relocation is performed before we move to the next iteration:

(1) nodes with scores higher than ω are sorted based on their nodes' degree in Hypothesis Graph. Top μ nodes in the sorted list(nodes with highest degree) are recommended to an annotator for double-checking (we set $\mu = 40$ and $\omega = 0.5$ in our experiments). After human verification and correction, recommended nodes are separated into two subsets: Positive Set (nodes with errors) and Negative Set (nodes without errors);

(2) nodes rated by our recommender system with very low scores are assigned to Negative Set;

(3) all other nodes, if they are not in the propagation set $\mathbf{V}_{Propagation}$, remain in Uncertain Node Pool.

We summarize our iterative recommender system with correction propagation in Algorithm 5.

4.5. EXPERIMENTAL EVALUATION

In this section, we first test our cascaded data-association tracking algorithm on two datasets. Then we experiment the efficiency and effectiveness of our tracking error correction scheme on 3 different datasets in different biomedical scenarios. Qualitative examples as well as quantitative evaluations are shown to validate the performance.

4.5.1. Metrics for Tracking Evaluation. Three well-known metrics are adopted to evaluate the tracking performance: (1) Tracker Purity (TP, [51]), the ratio of frames that a tracklet correctly follows the ground-truth to the total number of frames that the tracklet

has; (2) Target Effectiveness (TE or Object Purity, [51]), the ratio of frames that the object is correctly tracked to the total number of frames in which the object exists; (3) Multiple Object Tracking Accuracy (MOTA, [52]) that considers the number of missed detections, false positives and ID switches.

4.5.2. Datasets for Experiments on our Tracking Approach. Two videos of fruit fly monitoring were captured using a GoPro Hero2 camera with resolution of 480×848 pixels at 120 fps. Dataset 1 has 21000 frames of 52 flies and dataset 2 has 220000 frames of 82 flies.

4.5.3. Quantitative Evaluation for our Tracking Approach. We show the quantitative evaluation of our approach on the two datasets in Table.4.2. We observe that in general our approach tracks the fast-moving flies very well. Very few tracking errors remain, mostly due to the following reasons:(1) flies are camouflaged around the joint boundaries of two glass surfaces; (2) flies move extremely fast exceeding the Δ_s. The related video demos can be accessed at http://www.mst.edu/~yinz/Projects/FlyTracking/.

Table 4.2. Quantitative evaluation of our approach.

	# Frames	# flies	$\bar{TP}$	$\bar{TE}$	MOTA
Video1	21000	52	0.983	0.931	0.973
Video2	220000	82	0.966	0.928	0.984

4.5.4. Datasets for Experiments on Tracking Error Correction. Some tracking errors are unsolvable by automated tracking algorithms only. We therefore test the performance of our tracking error correction approach on three different biomedical image sequences whose specifications are summarized in Table.4.3. Fig.4.4 shows some sample images. In datasets 1 and 2 downloaded from
$http://www.celltracking.ri.cmu.edu/downloads.html$, the main challenges are the frequent occurring of cell merging and division (causing many tracklets in a small local neighborhood), false positives in detection (causing distractions and wrong associations)

and miss detections (causing broken tracklets). In dataset 3 acquired under the same condition in Section 2, the main challenges are fast motion blurring, object camouflaging and false positive distractions due to low image contrast.

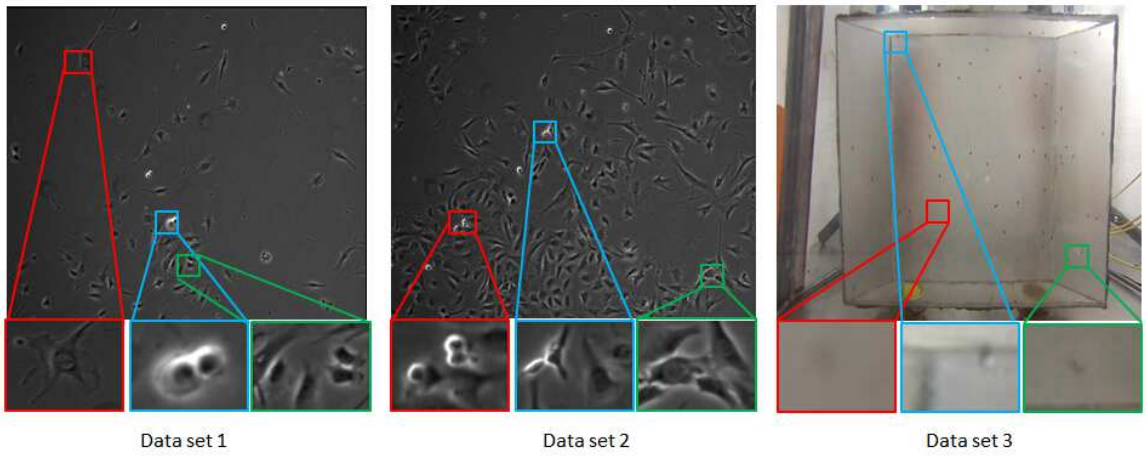

Figure 4.4. Examples of 3 datasets.

4.5.5. Quantitative Evaluation for Tracking Error Correction. According to different options on μ which is the number of maximum number of nodes be recommended for human review, performance may vary. Low value of μ will increase the precision rate, but more unrated nodes will be returned to the Node Pool, and consequently increase the number of iterations human interpolation will be used. Here in our experiment we choose $\mu = max(10, min(20, \lceil N/10 \rceil))$ where N is the number of nodes in Node Pool.

Table 4.3. Specifications of datasets.

	#Frames	Obj type	#Obj/frame	Image size
Set1	400	Stem Cells	20-100	1392×1040
Set2	380	Stem Cells	100-400	1392×1040
Set3	10000	Fruit Flies	52	848×480

In this thesis, we do not evaluate the performance of different object detection and tracking algorithms. Instead, we assume no object tracking algorithm can achieve perfect performance in challenging scenarios. Our focus is to investigate how to debug object

tracking results with human in the loop. The results are illustrated in Fig.4.5, 4.6, 4.7, 4.8, 4.9, 4.10.

Efficiency and Effectiveness:

To evaluate how well our recommender system and correction propagation can assist human annotators on debugging automated object tracking results, we use two metrics:

(1) *efficiency*: how fast will the number of uncertain nodes reduce so less human effort is needed to verify and correct nodes?

(2) *effectiveness*: what percentage of false nodes (nodes with tracking errors) is detected by the recommender system (i.e., how effective can our system guide human annotators towards false nodes)?

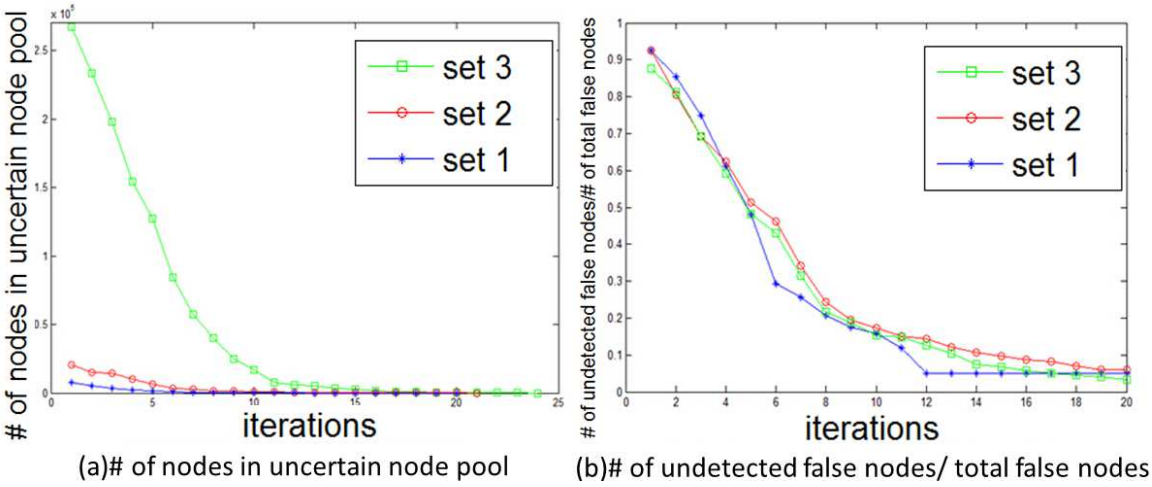

Figure 4.5. Efficiency and effectiveness of our iterative recommender system. (a) Number of nodes in the uncertain node pool; (b): # of undetected false nodes/# of total false nodes.

Fig.4.5(a) shows the number of nodes remaining in the uncertain node pool at different iterations for the 3 datasets. We observed that it decreases drastically when using our recommender system and correction propagation to assist human annotators, which proves that our system is efficient with less human effort to debug object tracking results. Fig.4.5(b) shows "*# of undetected false nodes/# of total false nodes*" at different iterations

on the 3 datasets. It is noticeable that the percentage curves drop quickly, and within 20 iterations the number of undetected false nodes falls below 5 percent out of the total false nodes, which shows that our recommender system can effectively guide human annotators to find questionable tracking results for correction.

Performance when Class-imbalance Occurs:

Our recommender system is tested on two cases of class-imbalance datasets: one with low percentage of false nodes and the other one with high percentage of false nodes. Experiments are implemented on dataset 3. Based on the ground truth of dataset 3, we randomly modified some tracking results to obtain a low-percentage error data and let an independent annotator to debug the tracking data. We apply a single-stage Hungarian algorithm with a large gating region (i.e., without fine-to-coarse gating region control) on dataset 3 to obtain a high-percentage error data.

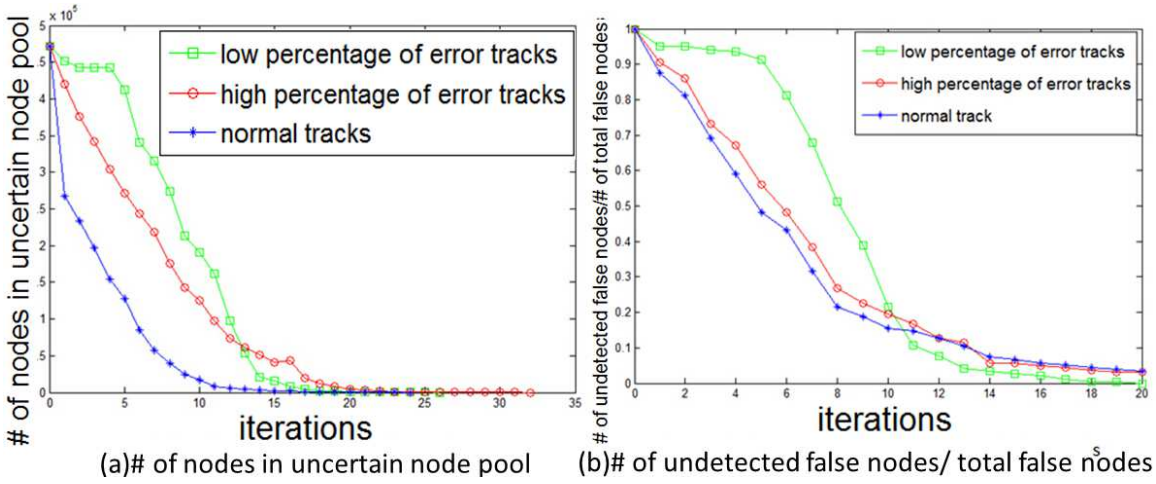

Figure 4.6. System performance in class-imbalance cases. (a) Number of nodes in the uncertain node pool; (b): # of undetected false nodes/# of total false nodes.

The performance of our system on these two imbalanced datasets is shown in Fig.4.6. When there is a very low percentage of false nodes (bad tracks) in the tracking results, the convergence will be slow initially when the under-represented false nodes are hard to find. Once the rare class is identified, the convergence rate accelerates drastically.

In Fig.4.6, the convergence on the tracking data with high percentage of false nodes is faster than that with low percentage of false nodes, because false nodes are easier to identify in the tracking data with high percentage of false nodes and our correction propagation can accelerate the correction.

Parameter Sensitivity:

In Algorithm 2, μ is the value to control the maximum number of nodes to be recommended to an annotator. A smaller μ makes sure human annotators will not be annoyed by heavy duty works, which meets one of our motivations to relieve human labor efforts. In our experiments μ is set as 20. In Algorithm 2, ω is the threshold to decide if a node is recommended or not. Fig.4.7 shows the experimental results of different ω values. High ω value guarantees that only the most suspicious nodes are sent for human labeling, but the convergence rate will be slower. On the other hand, lower ω value will accelerate the converging rate. But as a result, more human labor will be placed on every iteration, especially in the initial iterations when the training sample sets are not representative enough for each class of nodes.

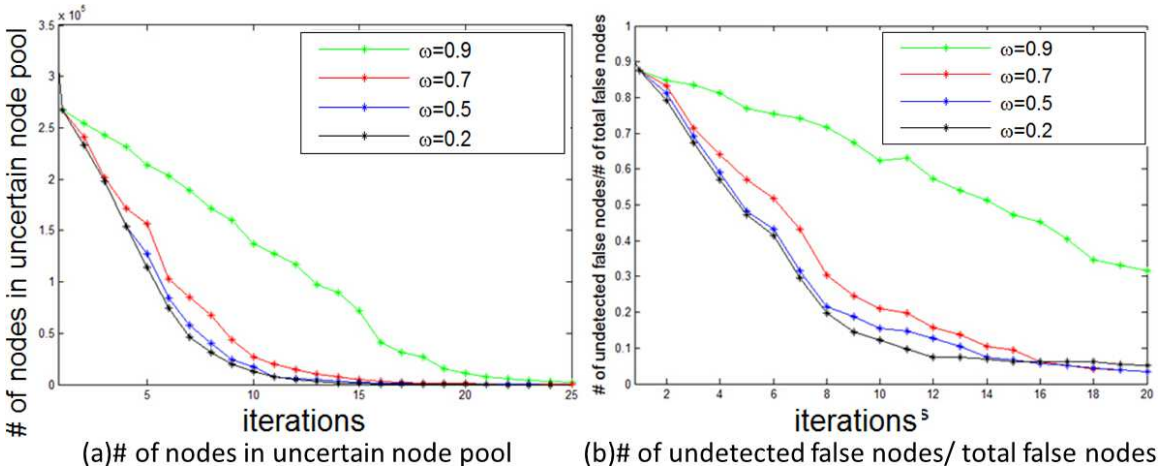

Figure 4.7. Experiments on different ω values.

Collaborative Debugging:

Our system is suitable to incorporate multi-agent interactions into collaborative annotations. We test the system using multi-annotators from our lab. As shown in Fig.4.8, multi-annotators will lower the individual workload and accelerate the overall converging speed.

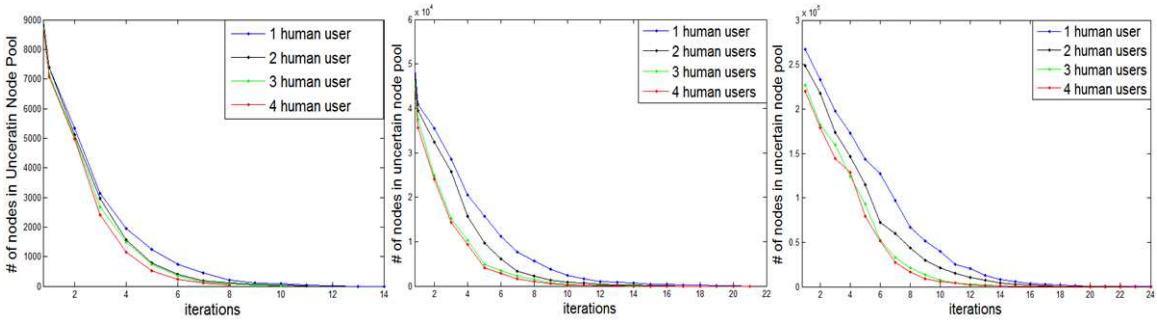

Figure 4.8. Uncertain node pool shrinking rate with multi-annotators.

4.5.6. Quantitative Comparison for Tracking Error Correction. In order to demonstrate the effect of our recommender system and correction propagation, we compare the following four approaches:

(1) Random selection without propagation: Nodes in uncertain node pool are randomly selected by humans to verify and correct. Human corrections are not propagated to neighboring nodes. This approach basically trains a classifier based on a human's annotations and predict if a new node is good or bad.

(2) Random selection with propagation: Nodes in uncertain node pool are randomly selected by humans to verify and correct. Human corrections are propagated to neighboring nodes.

(3) Recommendation without propagation: Nodes in uncertain node pool are recommended by our recommender system for humans to verify and correct. Human corrections are not propagated to neighboring nodes.

(4) Recommendation with propagation: Nodes in uncertain node pool are recommended by our recommender system for humans to verify and correct. Human corrections are propagated to neighboring nodes.

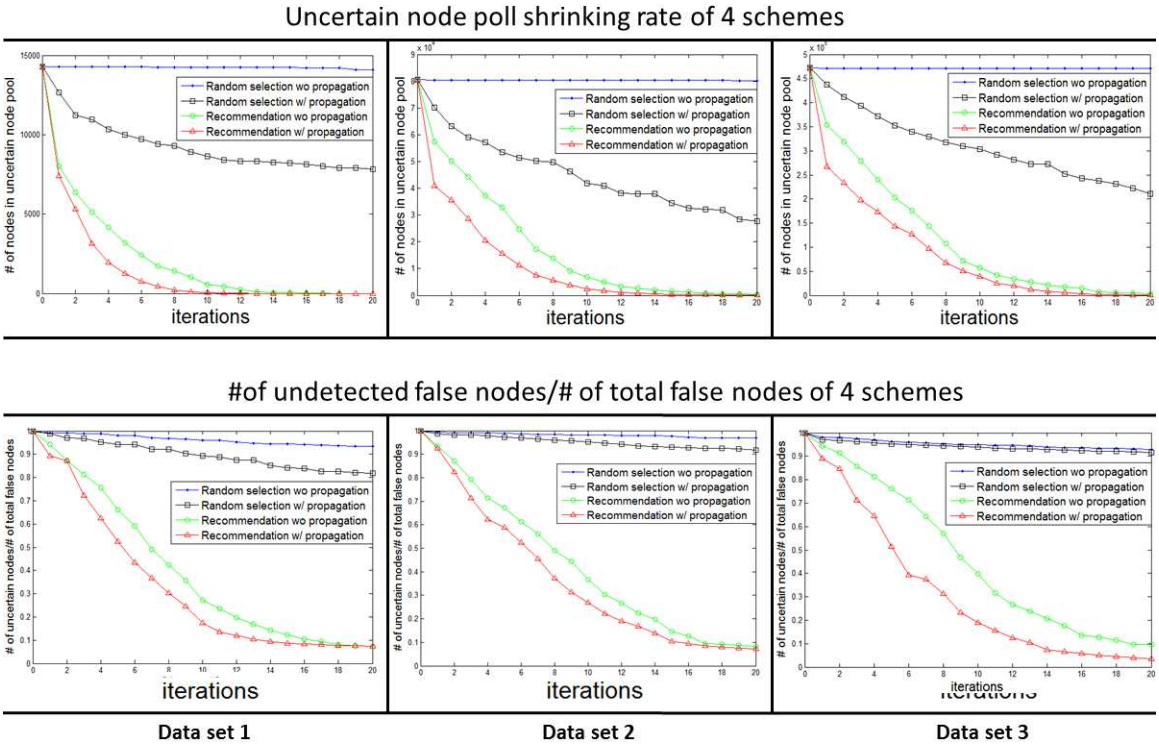

Figure 4.9. Uncertain node pool shrinking rates of 4 different approaches on 3 datasets.

As shown in Fig.4.9, **Random selection without propagation** (i.e., a classifier trained using human's annotations) is very inefficient to debug the object tracking results since the human randomly verifies uncertain nodes without any guidance and no further propagation is applied to human correction in each iteration. With correction propagation applied to the random selection, **Random selection with propagation** decreases the number of uncertain nodes faster, but the human annotators still have no clue on what nodes should be checked. **Recommendation without propagation** adds recommendation to humans for tracking results debugging, which reduces the number of uncertain node further

faster. Finally, when **Recommendation with propagation** is applied together, the uncertain node pool shrinks the fastest, which means that it takes much less time and human labeling costs.

From the top row of Fig.4.9, we observed that both our recommender system and correction propagation can help reduce the size of uncertain node pool. In the bottom row of Fig.4.9, our recommender system with correction propagation performs beyond the other 3 approaches in helping humans detect nodes with tracking error. Another observation from Fig.4.9 is that a classifiers trained from random selections has much lower effectiveness than our recommender system since it has no guidance on which nodes should be verified and corrected.

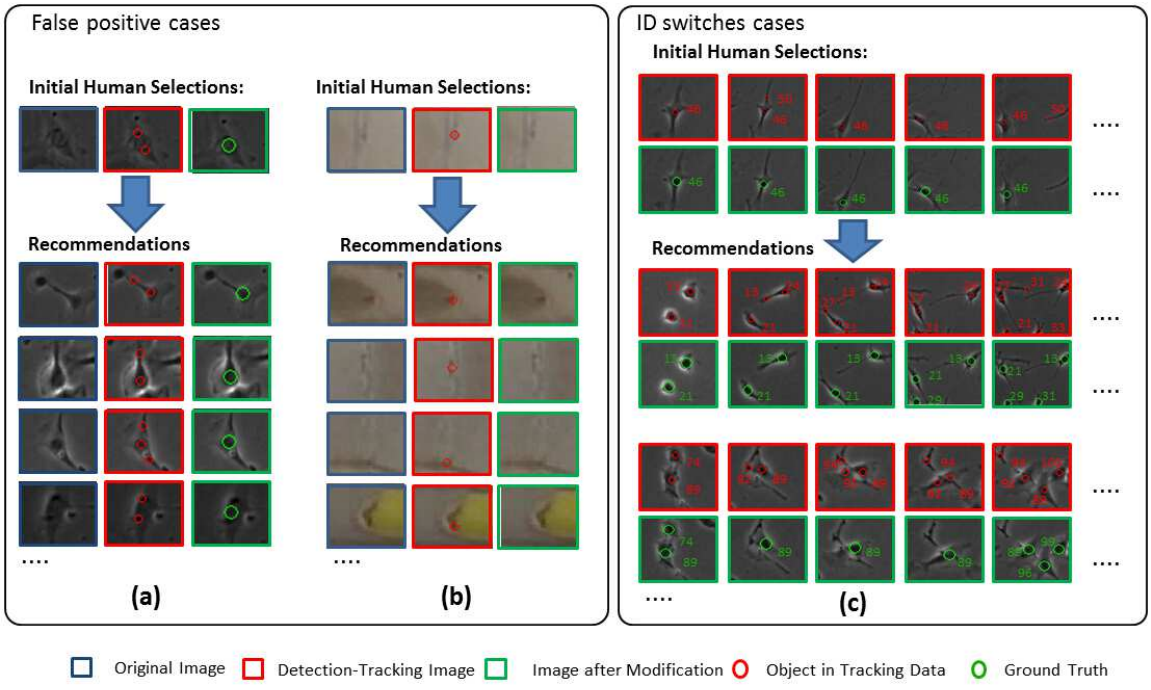

Figure 4.10. Examples of recommended nodes for human verification and correction based on initial human selection.

4.5.7. Qualitative Examples for Tracking Error Correction. In Fig.4.10 we present some examples to show how our recommender system helps annotators find similar false tracking data. In Fig.4.10(a), a cell is detected as multiple fragments and similar cases are

found by the recommender system. Fig.4.10(b) shows the scenario of background noises generating false positives and Fig.4.10(c) shows the wrong ID associations due to nearby distractors. In our recommender system, all of these tracking errors can be represented by their nodes' feature vectors. Initially, human selects some false nodes and makes corresponding corrections, then the recommender system search similar false nodes in the uncertain node pool and let human to verify and correct them.

4.6. SUMMARY

In this section, we first introduce a cascaded data association approach with fine-to-coarse gating region control for multi-object tracking in biomedical object monitoring. During each stage, feature vectors are recorded for later evaluation on the reliability of corresponding tracking data. Then we demonstrate a newly proposed novel iterative recommender system with correction propagation to help humans debug (verify and correct) tracking results generated by automated object tracking algorithms. At each iteration, human annotators only need to debug a sparse set of nodes recommended by the recommender system. Different collaborators debug the tracking data independently and their debugging results are collected together. Since every correction made on one node will affect its neighboring nodes, we propagate all the corrections to related nodes, which ensures the tracking consistency and speeds up the debugging process. After correction propagation, each human annotator's profile parameters are updated based on the new debugged nodes. The process is iterated until the uncertain node pool is empty. Experimental performance on our cascaded tracking algorithm is first validated on two fruit fly monitoring videos. Then discussion regarding the effectiveness and efficiency of our recommender system on three sets of biomedical image sequences is presented. The experimental results show that our recommender system with correction propagation can effectively guide human annotators to debug tracking data in an efficient and collaborative way.

5. CONCLUSION AND FUTURE WORKS

5.1. CONCLUSION

Biomedical object monitoring is difficult using computer vision techniques, due to a wide range of various challenges in detection, segmentation and tracking stages. In this paper, we investigate these problems and propose novel approaches in multiple biomedical research works to achieve highly reliable experimental results. Firstly, in the object detection stage, we propose a novel Adaptive LBP feature to overcome inconsistent contrast and fast motion problem for tiny objects. Secondly, for accurate object segmentation in microscopy images, we propose a multi-modal microscopy restoration method, based on which further cell segmentation and classification is easily implementable, followed by discussion on a Multi-exposure Maximum Stable Extremal Region (MMSER) based approach for accurate cell segmentation and classification in cases where only one microscopic modality if available. Experiments show that these techniques outperform other state-of-the-art microscopy image segmentation methods. At last, to generate highly reliable biomedical object tracking results, we first introduce a cascaded data association method with fine-to-coarse gating region control. Then we present a semi-automatic tracking error correction recommender framework with human-in-the-loop to replace exhaustive manual annotation. Experimental results on various metrics show that our method is capable of drastically saving human effort in correcting automated tracking data.

5.2. FUTURE WORKS

The research works in this paper serve as a solid foundation for future investigation of computer vision techniques on biomedical researches as well as civil and industrial applications. Considering that challenges such as overlapping, low contrast and appearance

inconsistency in biomedical objects remain big problems, we aim to solve these issues with better detection methods. In addition, the high computation cost in data association and iterative re-weighting is time-consuming and resource-demanding. Thus optimization is needed in respect of algorithmic efficiency. Our future work will also include possible transformation as well as better propagation of the influence from limited corrected tracking errors to unchecked data, to help obtaining error-free tracking data more efficiently and effectively.

BIBLIOGRAPHY

[1] Zhaozheng Yin, Takeo Kanade, and Mei Chen. Understanding the phase contrast optics to restore artifact-free microscopy images for segmentation. *Medical Image Analysis*, 16(5):1047–1062, 2012.

[2] Zia Khan, Tucker Balch, and Frank Dellaert. Mcmc data association and sparse factorization updating for real time multitarget tracking with merged and multiple measurements. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 28(12):1960–1972, 2006.

[3] Mary Fletcher, Anna Dornhaus, and Min C Shin. Multiple ant tracking with global foreground maximization and variable target proposal distribution. In *Applications of Computer Vision (WACV), IEEE Workshop on*, pages 570–576. IEEE, 2011.

[4] Hoan Nguyen, Thomas Fasciano, Daniel Charbonneau, Anna Dornhaus, and Min C Shin. Data association based ant tracking with interactive error correction. In *Applications of Computer Vision (WACV), IEEE Winter Conference on*, pages 941–946. IEEE, 2014.

[5] Thomas Fasciano, Anna Dornhaus, and Min C Shin. Ant tracking with occlusion tunnels. In *Applications of Computer Vision (WACV), IEEE Winter Conference on*, pages 947–952. IEEE, 2014.

[6] Protik Maitra, Stan Schneider, and Min C Shin. Robust bee tracking with adaptive appearance template and geometry-constrained resampling. In *Applications of Computer Vision (WACV), IEEE Winter Conference on*, pages 1–6. IEEE, 2009.

[7] Yi Deng, Philip Coen, Mingzhai Sun, and Joshua W Shaevitz. Efficient multiple object tracking using mutually repulsive active membranes. *PloS one*, 8(6):e65769, 2013.

[8] Erik Meijering. Cell segmentation: 50 years down the road [life sciences]. *Signal Processing Magazine, IEEE*, 29(5):140–145, 2012.

[9] Jens Rittscher. Characterization of biological processes through automated image analysis. *Annual Review of Biomedical Engineering*, 12:315–344, 2010.

[10] Lei Qu, Fuhui Long, and Hanchuan Peng. 3-d registration of biological images and models: registration of microscopic images and its uses in segmentation and annotation. *Signal Processing Magazine, IEEE*, 32(1):70–77, 2015.

[11] Jinwei Xu, Jiankun Hu, and Xiuping Jia. A multistaged automatic restoration of noisy microscopy cell images. *Biomedical and Health Informatics, IEEE Journal of*, 19(1):367–376, 2015.

[12] Leo Grady. Random walks for image segmentation. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 28(11):1768–1783, 2006.

[13] Yousef Al-Kofahi, Wiem Lassoued, William Lee, and Badrinath Roysam. Improved automatic detection and segmentation of cell nuclei in histopathology images. *Biomedical Engineering, IEEE Transactions on*, 57(4):841–852, 2010.

[14] Zhaozheng Yin, Kang Li, Takeo Kanade, and Mei Chen. Understanding the optics to aid microscopy image segmentation. In *Medical Image Computing and Computer-Assisted Intervention–MICCAI*, pages 209–217. Springer, 2010.

[15] Hang Su, Zhaozheng Yin, Seungil Huh, and Takeo Kanade. Cell segmentation in phase contrast microscopy images via semi-supervised classification over optics-related features. *Medical Image Analysis*, 17(7):746–765, 2013.

[16] Kang Li and Takeo Kanade. Nonnegative mixed-norm preconditioning for microscopy image segmentation. In *Information Processing in Medical Imaging*, pages 362–373. Springer, 2009.

[17] Alper Yilmaz, Omar Javed, and Mubarak Shah. Object tracking: A survey. *Acm Computing Surveys (CSUR)*, 38(4):13, 2006.

[18] Donald B Reid. An algorithm for tracking multiple targets. *Automatic Control, IEEE Transactions on*, 24(6):843–854, 1979.

[19] Thomas E Fortmann, Yaakov Bar-Shalom, and Molly Scheffe. Sonar tracking of multiple targets using joint probabilistic data association. *Oceanic Engineering, IEEE Journal of*, 8(3):173–184, 1983.

[20] Asad A Butt and Robert T Collins. Multiple target tracking using frame triplets. *Computer Vision–ACCV, Asian Conference on*, pages 163–176, 2013.

[21] Robert T Collins and Peter Carr. Hybrid stochastic/deterministic optimization for tracking sports players and pedestrians. In *Computer Vision–ECCV, Europoean Conference on*, pages 298–313. Springer, 2014.

[22] Chang Huang, Bo Wu, and Ramakant Nevatia. Robust object tracking by hierarchical association of detection responses. In *Computer Vision–ECCV, European Conference on*, pages 788–801. Springer, 2008.

[23] Harold W Kuhn. The hungarian method for the assignment problem. *Naval Research Logistics Quarterly*, 2(1-2):83–97, 1955.

[24] George B Dantzig, Alex Orden, Philip Wolfe, et al. The generalized simplex method for minimizing a linear form under linear inequality restraints. *Pacific Journal of Mathematics*, 5(2):183–195, 1955.

[25] Stéphane Bonneau, Maxime Dahan, and Laurent D Cohen. Single quantum dot tracking based on perceptual grouping using minimal paths in a spatiotemporal volume. *Image Processing, IEEE Transactions on*, 14(9):1384–1395, 2005.

[26] Li Zhang, Yuan Li, and Ramakant Nevatia. Global data association for multi-object tracking using network flows. In *Computer Vision and Pattern Recognition–CVPR. IEEE Conference on*, pages 1–8. IEEE, 2008.

[27] Kang Li, Eric D Miller, Mei Chen, Takeo Kanade, Lee E Weiss, and Phil G Campbell. Cell population tracking and lineage construction with spatiotemporal context. *Medical Image Analysis*, 12(5):546–566, 2008.

[28] Ryoma Bise, Kang Li, Sungeun Eom, and Takeo Kanade. Reliably tracking partially overlapping neural stem cells in dic microscopy image sequences. In *MICCAI Workshop on OPTIMHisE*, volume 5, 2009.

[29] Shari Trewin. Knowledge-based recommender systems. *Encyclopedia of Library and Information Science*, 17(32):180, 2000.

[30] Pasquale Lops, Marco De Gemmis, and Giovanni Semeraro. Content-based recommender systems: State of the art and trends. In *Recommender Systems Handbook*, pages 73–105. Springer, 2011.

[31] Deuk Hee Park, Hyea Kyeong Kim, Il Young Choi, and Jae Kyeong Kim. A literature review and classification of recommender systems research. *Expert Systems with Applications*, 39(11):10059–10072, 2012.

[32] Paul Resnick and Hal R Varian. Recommender systems. *Communications of the ACM*, 40(3):56–58, 1997.

[33] James H Marden, Melisande R Wolf, and Kenneth E Weber. Aerial performance of drosophila melanogaster from populations selected for upwind flight ability. *The Journal of Experimental Biology*, 200(21):2747–2755, 1997.

[34] Nobuyuki Otsu. A threshold selection method from gray-level histograms. *Automatica*, 11(285-296):23–27, 1975.

[35] Timo Ojala, Matti Pietikainen, and Topi Maenpaa. Multiresolution gray-scale and rotation invariant texture classification with local binary patterns. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 24(7):971–987, 2002.

[36] Frits Zernike. How i discovered phase contrast. *Science*, 121(3141):345–349, 1955.

[37] Douglas B Murphy. *Fundamentals of light microscopy and electronic imaging*. John Wiley & Sons, 2002.

[38] Zhaozheng Yin, Takeo Kanade, et al. Restoring dic microscopy images from multiple shear directions. In *Information Processing in Medical Imaging*, pages 384–397. Springer, 2011.

[39] Zhaozheng Yin and Takeo Kanade. Restoring artifact-free microscopy image sequences. pages 909–913, 2011.

[40] Hang Su, Zhaozheng Yin, Takeo Kanade, and Takeo Huh. Restoring artifact-free microscopy image sequences. pages 615–622, 2012.

[41] Zhaozheng Yin, Hang Su, Elmer Ker, Mingzhong Li, and Haohan Li. Cell-sensitive microscopy imaging for cell image segmentation. pages 41–48, 2014.

[42] Zhaozheng Yin, Hang Su, Elmer Ker, Mingzhong Li, and Haohan Li. Cell-sensitive phase contrast microscopy imaging by multiple exposures. *Medical Image Analysis*, 25(1):111–121, 2015.

[43] Yan Xu, Jun-Yan Zhu, Eric Chang, and Zhuowen Tu. Multiple clustered instance learning for histopathology cancer image classification, segmentation and clustering. In *Computer Vision and Pattern Recognition (CVPR), IEEE Conference on*, pages 964–971. IEEE, 2012.

[44] Meiguang Jin, Lakshmi Narasimhan Govindarajan, and Li Cheng. A random-forest random field approach for cellular image segmentation. In *Biomedical Imaging–ISBI, IEEE International Symposium on*, pages 1251–1254. IEEE, 2014.

[45] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: convolutional networks for biomedical image segmentation. In *Medical Image Computing and Computer-Assisted Intervention–MICCAI*, pages 234–241. Springer, 2015.

[46] Xundong Wu, Yong Wu, and Enrico Stefani. Multi-scale deep neural network microscopic image segmentation. *Biophysical Journal*, 108(2):473a, 2015.

[47] Fei Sha, Yuanqing Lin, Lawrence K Saul, and Daniel D Lee. Multiplicative updates for nonnegative quadratic programming. *Neural Computation*, 19(8):2004–2031, 2007.

[48] Emmanuel J Candes, Michael B Wakin, and Stephen P Boyd. Enhancing sparsity by reweighted $l1$ minimization. *Journal of Fourier analysis and applications*, 14(5-6):877–905, 2008.

[49] Krystian Mikolajczyk and Cordelia Schmid. A performance evaluation of local descriptors. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 27(10):1615–1630, 2005.

[50] Jiri Matas, Ondrej Chum, Martin Urban, and Tomĺćs Pajdla. Robust wide-baseline stereo from maximally stable extremal regions. *Image and Vision Computing*, 22(10):761–767, 2004.

[51] Kevin Smith, Daniel Gatica-Perez, Jean-Marc Odobez, and Sileye Ba. Evaluating multi-object tracking. In *Computer Vision and Pattern Recognition-Workshops, CVPR Workshops. IEEE Computer Society Conference on*, pages 36–36. IEEE, 2005.

[52] Rangachar Kasturi, Dmitry Goldgof, Padmanabhan Soundararajan, Vasant Manohar, John Garofolo, Rachel Bowers, Matthew Boonstra, Valentina Korzhova, and Jing Zhang. Framework for performance evaluation of face, text, and vehicle detection and tracking in video: Data, metrics, and protocol. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 31(2):319–336, 2009.

VITA

Mingzhong Li was born in Chongqing, China. In July 2011, he received his B.S. in Electronic and Information Engineering from the South China University of Technology, Guangzhou, China. Then, he joined the Electrical Engineering Department of Missouri University of Science and Technology (formerly the University of Missouri-Rolla) as a master student in August,2011. After one year he started his PhD study in the Computer Science Department at the same school, supervised by Dr.Zhaozheng Yin. He received his Ph.D. degree in Computer Science from Missouri University of Science and Technology in July 2016. His primary research interests were computer vision, bio-medical imaging and machine learning.