Computer Science Department
School of Sciences and Technologies
University of Crete

Visual Object Tracking and Segmentation in a Closed Loop

Master's Thesis

Konstantinos E. Papoutsakis

October 2010
Heraklion, Greece

Τμήμα Επιστήμης Υπολογιστών
Σχολή Θετικών και Τεχνολογικών Επιστημών
Πανεπιστήμιο Κρήτης

Οπτική Παρακολούθηση και Τμηματοποίηση Αντικειμένου σε Κλειστό Βρόχο

Μεταπτυχιακή Εργασία

Κωνσταντίνος Ε. Παπουτσάκης

Οκτώβριος 2010
Ηράκλειο, Ελλάδα

Αφιερωμένο

στους γονείς, τον αδερφό και τη Χρύσα μου

Abstract

The vision-based tracking and the segmentation of an object of interest in an image sequence are two challenging, tightly coupled computer vision problems. By solving the segmentation problem, a solution to the tracking problem can be obtained, while tracking may provide important input to segmentation. The coupling between these two problems is an actively researched topic because, besides its theoretic interest, it may lead to robust solutions in a number of important applications including object localization and recognition, vision-based automated surveillance, activity recognition, human-computer/robot interaction, etc.

In this work we propose a new method for integrated tracking and segmentation of a single non-rigid object in a monocular video, captured by a possibly moving camera. It is assumed that a binary mask is available for the initial frame of an image sequence, fully or partially indicating the previously unseen object of interest that is to be segmented and tracked throughout that image sequence. A closed-loop interaction between Expectation Maximization (EM) color-based tracking and Random Walker-based image segmentation is proposed. The tracking algorithm represents the position and the area of the object in the form of an ellipse in each frame of the image sequence. At each frame, a finely segmented object mask is available from the segmentation performed at the previous frame. The spatial position and variance of the object mask are utilized to initialize the ellipse of the tracking algorithm for the current frame. Through EM iterations performed by the tracking method, a new ellipse is computed, estimating the new position and variance of the object in the current frame. The initial and the evolved ellipses are used to estimate a 2D affine transformation that propagates the segmented object shape of the previous frame to the current frame. A shape band is then defined indicating a region of uncertainty where the true object boundaries lie. In the following, pixel-wise spatial and color image cues are fused using Bayesian inference to guide object segmentation. A finely segmented object mask of the target object is finally computed in the current frame using the Random Walker-based segmentation methodology, closing the loop between tracking and segmentation.

The proposed method efficiently tracks and segments previously unseen objects requiring no off-line training or prior knowledge regarding the object of interest and its scene context. As confirmed by both the qualitative and quantitative experimental evaluation carried out on a variety of image sequences, the proposed methodology results

in reduced tracking drifts and in fine object segmentation. Additionally, it operates effectively for previously unseen objects of varying appearance and shape that perform complex motions under varying illumination conditions.

Περίληψη

Η οπτική παρακολούθηση και η τμηματοποίηση ενός αντικειμένου σε μια ακολουθία εικόνων αποτελούν σημαντικά προβλήματα της υπολογιστικής όρασης που σχετίζονται στενά μεταξύ τους. 'Ενα αντικείμενο το οποίο έχει τμηματοποιηθεί μπορεί εύκολα να παρακολουθηθεί. Ταυτόχρονα, η παρακολούθηση του αντικειμένου παρέχει σημαντική πληροφορία για την τμηματοποίησή του. Η σύνδεση μεταξύ των δύο αυτών προβλημάτων αποτελεί μία ενεργή ερευνητική περιοχή καθώς πέρα από το θεωρητικό της ενδιαφέρον, μπορεί να οδηγήσει σε εύρωστες λύσεις σε μεγάλο αριθμό σημαντικών εφαρμογών όπως η αναγνώριση και η εκτίμηση θέσης αντικειμένων, η αναγνώριση δραστηριοτήτων από βίντεο, η οπτική επόπτευση χώρων, η αλληλεπίδραση ανθρώπου με υπολογιστή ή ρομποτικό σύστημα κ.α.

Στην εργασία αυτή περιγράφεται μια νέα μέθοδος συνδυασμένης οπτικής παρακολούθησης και τμηματοποίησης ενός αντικειμένου σε ακολουθία εικόνων που έχουν ληφθεί από μια ενδεχομένως κινούμενη βιντεοκάμερα. Θεωρείται πως η μοναδική γνώση για το προς παρακολούθηση αντικείμενο είναι μια δυαδική εικόνα-μάσκα που παρέχει μια περιγραφή του περιγράμματός του στην πρώτη εικόνα της ακολουθίας. Προτείνεται μια μεθοδολογία βασισμένη στην αλληλεπίδραση μεταξύ ενός αλγορίθμου Μεγιστοποίησης Προσδοκίας (Expectation Maximization - EM) για την παρακολούθηση αντικειμένου με βάση την χρωματική πληροφορία και μιας μεθόδου για τμηματοποίηση βασισμένη στη θεωρία των Τυχαίων Περιπάτων σε γράφους (Random Walks). Το αποτέλεσμα της τμηματοποίησης της εικόνας την προηγούμενη χρονική στιγμή οδηγεί στον ορισμό μίας έλλειψης που περιγράφει την θέση και την έκταση του αντικειμένου. Ο αλγόριθμος παρακολούθησης αρχικοποιείται με αυτή την έλλειψη και με την εφαρμογή μιας επαναληπτικής διαδικασίας Μεγιστοποίησης Προσδοκίας (EM) παράγει μια νέα έλλειψη που αποτελεί πρόβλεψη για τη θέση και έκταση του αντικειμένου την παρούσα χρονική στιγμή. Με βάση τις δύο αυτές ελλείψεις υπολογίζεται ένας δισδιάστατος αφινικός μετασχηματισμός που επιτρέπει την πρόβλεψη του σχήματος του αντικειμένου στην τρέχουσα εικόνα. Γύρω απο αυτή την πρόβλεψη σχήματος, ορίζεται μια περιοχή αβεβαιότητας εντός της οποίας μπορεί να προσδιοριστεί το ακριβές περίγραμμα του αντικειμένου για την τρέχουσα χρονική στιγμή. Αυτό επιτυγχάνεται με την εφαρμογή του αλγορίθμου τμηματοποίησης στην περιοχή αβεβαιότητας, που βασίζεται στην Μπεϋζιανή σύνθεση χαρακτηριστικών, όπως η θέση και το χρώμα κάθε σημείου της εικόνας στην περιοχή αβεβαιότητας.

Η προτεινόμενη μέθοδος συνδυάζει την παρακολούθηση και την τμηματοποίηση ενός αντικειμένου χωρίς να απαιτεί εκπαίδευση ή προηγούμενη γνώση για το αντικείμενο ενδιαφέροντος και το περιεχόμενο της σκηνής στην οποία αυτό εμπεριέχεται. 'Οπως επιβεβαιώνεται από την ποιοτική και την ποσοτική πειραματική αξιολόγησή της, η προτεινόμενη

μεθοδολογία μειώνει το σφάλμα παρακολούθησης ενός αντικειμένου και βελτιώνει την ακρίβεια της τμηματοποίησής του. Επιπρόσθετα, η μεθοδολογία λειτουργεί αποτελεσματικά για αντικείμενα των οποίων το σχήμα και η εμφάνιση μεταβάλλεται σημαντικά κατά τη διάρκεια της πιθανόν πολύπλοκης κίνησής τους σε συνθήκες μεταβαλλόμενου φωτισμού.

Ευχαριστίες

Θα ήθελα να εκφράσω τις θερμές ευχαριστίες μου σε ανθρώπους που ήταν και είναι δίπλα μου, στηρίζοντας έμμεσα ή άμεσα την προσπάθεια μου κατα τη διάρκεια των μεταπτυχιακών σπουδών και της εκπόνησης της εργασίας αυτής.

Πρώτα απο όλα, ευχαριστώ ιδιαίτερα τους γονείς μου, Μανώλη και Μαρία, και τον αδερφό μου Γιώργο, για την αγάπη και τη στήριξη που μου προσφέρουν με κάθε τρόπο σε κάθε μου επιλογή και προσπάθεια. Πάντα μου παρέχουν την απαραίτητη εμπιστοσύνη και τις συνθήκες για να προχωρήσω προς τους στόχους μου στη ζωή.

Οφείλω ένα μεγάλο ευχαριστώ στη Χρύσα μου, καθώς όλα αυτά τα χρόνια είναι πάντα δίπλα μου, σε όμορφες ή δύσκολες στιγμές. Η εκτίμηση και τα συναισθήματά της προς εμένα είναι τόσο σημαντικά και πολύτιμα, όσο και η ίδια σαν άνθρωπος.

Ένα πολύ μεγάλο ευχαριστώ όφείλω στον κ. Καθηγητή Αντώνη Αργυρό, επόπτη αυτής της εργασίας και των μεταπτυχιακών σπουδών μου. Η καλή του διάθεση, ο χρόνος που διέθεσε σε μένα και η υποστήριξη του με κάθε τρόπο, αποτέλεσαν σημαντικό κίνητρο και δύναμη για δουλειά. Ελπίζω οτι θα συνεχίσουμε την καλή συνεργασία μας στο μέλλον, και θα φανώ αντάξιος της εμπιστοσύνης του προς εμένα.

Θέλω να ευχαριστήσω τον κ. Καθηγητή Πάνο Τραχανιά και τον Ξενοφώντα Ζαμπούλη για την ευχάριστη και εποικοδομητική συνεργασία μας κατα τη διάρκεια των σπουδών μου στα πλαίσια του ΙΤΕ και του Πανεπιστημίου. Τους ευχαριστώ πολύ για την συμμετοχή τους στην Εξεταστική Επιτροπή της μεταπτυχιακής μου εργασίας. Επίσης, ευχαριστώ τον κ. Δημήτρη Τσακίρη για την καλή συνεργασία μας στα πλαίσια του εργαστηρίου και του μαθήματος του.

Ένα θερμό ευχαριστώ θέλω να απευθύνω σε συμφοιτητές, μέλη και φίλους απο το Εργαστήριο Υπολογιστικής Όρασης και Ρομποτικής του ΙΤΕ για την καλή διάθεση προς εμένα, την παρέα και τις συζητήσεις για οποιοδήποτε θέμα κατα τη διάρκεια των δύο ετών που είχα την τύχη να αποτελώ μέλος του εργαστηρίου. Ιδιαίτερα ευχαριστώ τους Ιάσονα, Νίκο, Κώστα, Μάρκο, Θωμά, Πασχάλη, Damien και Χάρη, εκτός των παραπάνω λόγων και για την συμβολή τους στην εργασία αυτή μέσα απο εποικοδομητικές συζητήσεις και ιδέες σε προβλήματα μου. Επίσης, ευχαριστώ τους Γιώργο, Μανώλη, Μιχάλη, Νίκο, Μανώλη, Μιχάλη, Μαρία, Μελίνα.

Τέλος, ευχαριστώ συγγενείς και κοντινούς φίλους μου, οι οποίοι στηρίζουν και εκτιμούν την προσπάθεια μου.

Contents

List of Figures

List of Tables

List of Algorithms

Chapter 1

Introduction

The vision-based tracking and the segmentation of an object of interest in an image sequence are two challenging problems in computer vision. Each of them has its own importance and challenges. The two problems are highly interrelated and can be considered as “chicken-and-egg” problems. By solving the segmentation problem, a solution to the tracking problem can be obtained, while tracking may provide important input to segmentation. Moreover, the robustness and the accuracy of the tracking object representation affects the quality of the information provided to the segmentation throughout an image sequence. The coupling between these two main problems in computer vision is an actively researched topic because of the large number of important applications including but not limited to automated surveillance, visual attention, video analysis, object pose estimation and recognition, activity recognition, human-computer interaction, robot navigation,etc.

In this work, a new methodology is proposed considering the efficient combination of tracking and explicit fine segmentation towards an online, robust 2D object tracking and segmentation framework.

Both qualitative and quantitative experimental evaluation had been carried out in order to assess the tracking and segmentation performance of the proposed method. A large variety of image sequences is selected as test datasets, containing previously unseen objects of varying appearance and shape, performing in presence of challenging environmental conditions. The proposed method is compared to the stand-alone EM-shift color-based tracking method [77] utilized in the proposed framework in order to assess the improvement on the localization accuracy and the prevention of tracking drifts. Moreover, a quantitative assessment is carried out based on ground truth tracking and segmentation data on two image sequences. The Recall, Precision and F-score statistic metrics are calculated to validate the segmentation performance of the proposed framework. An overview of the proposed method is presented in [54].

This report is organized as follows. In Chapter 2, an introduction to the algorithmic tools of EM-shift object tracking [77] and Random Walker based image segmentation [30], that have been utilized in the proposed framework, is provided. Chapter 3 describes each part of the proposed methodology in detail. In Chapter 4, the qualitative and the quantitative evaluation of the proposed method is presented. Finally, in Chapter 5, a discussion regarding the effectiveness and the contributions of the proposed methodology is accorded, whereas future work is discussed in order to extent its capabilities and to eliminate the reported weaknesses.

This chapter is organized as follows. An extended introduction to the problem of visual object tracking is provided in Section 1.1. A short categorization of the visual object tracking methods is provided in Section 1.1.1, whereas Section 1.1.2 specializes to methods that perform joint tracking and segmentation.

1.1 Visual Object Tracking

Visual object tracking can be defined as the problem of estimating the trajectory of a moving object throughout the frames of a video. Additionally, depending on the tracking domain and the application, a tracker can also provide object-centric information, such as orientation, size of its area, accurate or coarse representation of the its shape, etc.

Object tracking is a challenging problem due to the:

- loss of information caused by projection of the 3D world on a 2D image,
- complex and/or abrupt object motion,
- nonrigid and/or articulated object shape,
- complex appearance of the object,
- dynamic changes of the object's appearance and shape,
- scene context/background clutter,
- object occlusions,
- real-time processing requirements.

Being one of the most important and actively researched fields of computer vision, visual tracking is the main research topic in numerous publications trying to deal with a variety of applications concerning tracking of a single or multiple objects of interest throughout an image sequence. Human tracking, vehicle tracking, face tracking, hand tracking are only some of the main applications of visual object tracking. Most of the existing tracking methods are trying to impose constraints (i.e rigidity of object shape, smooth object motion etc.) or use prior information (i.e appearance models, shape prior information) regarding any of the aforementioned challenges towards an efficient solution of the tracking problem. There is a number of crucial decisions that need to be made throughout the development of any tracking algorithm regarding its functionality. These decisions can also be considered as keypoints, regarding its tracking performance and applicability, including:

- object shape and appearance representation
- image feature selection
- object detection
- object propagation/prediction

Each of the first three issues is briefly outlined in the following paragraphs, whereas a short categorization of the visual object tracking methods arises based on the last issue and is described in Section 1.1.1.

Object Representation

In a tracking scenario, an object can be defined as anything that is of interest for further analysis. An object can be represented by its shape and its appearance. Object shape representations commonly employed for tracking include: the object centroid point, points of interest in object area, primitive geometric shapes, object silhouette and contour, articulated shapes and skeletal models, as illustrated in Fig.1.1.

There are multiple ways to represent the appearance features of objects. There are many tracking approaches [19], where the shape representation of the target object is combined with its appearance representation. Some common appearance representations in the context of object tracking are:

- **Probability densities of object appearances**, which are divided in:
  - Parametric (Gaussian [76], mixture of Gaussians [56])

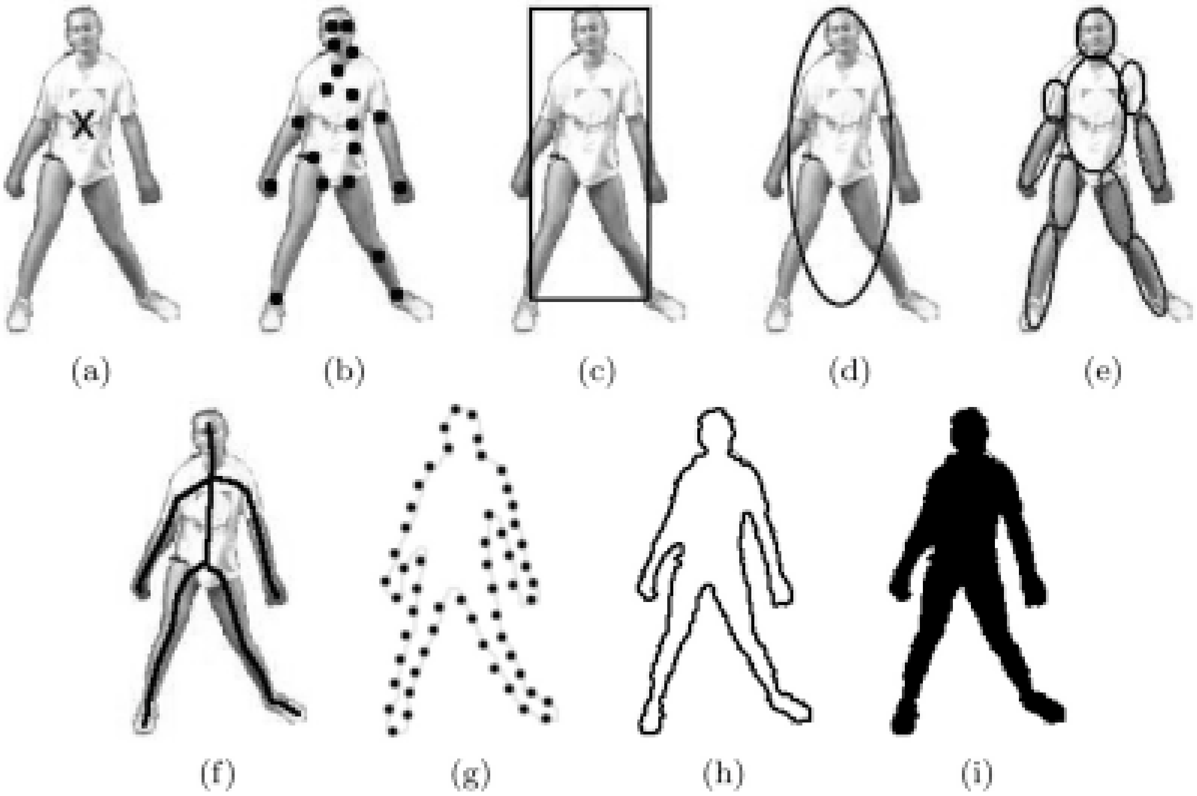


Figure 1.1: Object shape representations. (a) Object centroid, (b) multiple points of interest, (c) rectangle bounding box, (d) ellipsoid region of interest, (e) part-based (articulated) multiple shape patches, (f) object skeleton, (g) selected control points on object contour, (h) object contour, (i) object silhouette (Figure originally appeared in [72]).

  - Non-parametric (Parzen windows [25], histograms [18])

  The probability densities of object appearance features (color,texture) can be computed from the image regions specified by the shape models (interior region of an ellipse, a contour or a bounding box).

- **Shape templates** using geometric shapes or silhouettes [26], carrying both spatial and appearance information. Templates, however, only encode the object appearance generated from a limited number of views. Thus, they are only suitable for tracking objects whose poses do not vary considerably during the course of tracking.

- **Active appearance models** by simultaneously modeling the object shape and appearance [19]. Active appearance models are based on shape landmarks for which a model is computed capturing single or multiple appearance image features. They require a training phase where both the shape and its associated appearance is learned from a set of samples.

- **Multi-view appearance models** encoding different views of the appearance and the shape of the object usually by generating a subspace from given multiple views of the object using Principal Component Analysis [10], Independent Component Analysis, trained Bayesian Networks[57], trained support vector machines [3], etc.

There is a strong relationship between the chosen object representations and the tracking algorithms, according to the application domain. For tracking objects with complex shapes, for example humans, a contour or a silhouette-based representation is appropriate [9], whereas an ellipsoid region is commonly utilized as a shape representation in combination with a color histogram in to track non-rigid objects [18].

Image Feature Selection

Image feature selection is crucial in object tracking. The image features should be selected so that the objects of interest can be easily distinguished in the feature space. Feature selection is closely related to the object appearance representation. For example, a color-based histogram of the object area encodes the appearance of the object combining the image feature of color within the object area with a non-parametric appearance representation. Common primary visual features are the *color, edges, spatial pixel coordinates, optical flow* and *texture* image cues. In the majority of the proposed methods, visual features are chosen manually by the user depending on the application domain. However, there are many recently proposed techniques [14, 70] enabling automatic feature selection based on application-driven criteria that facilitate the discrimination between the object and the rest of the scene in the feature space. Moreover, combinations of image features are widely utilized to improve tracking performance.

Object Detection

The object detection mechanism is an indispensable part of every tracking method that is applied to every video frame. There are numerous stand-alone object detection methods that can be exploited in an integrated visual tracking framework.

Object detection can be based on information of a single frame or on temporal information provided by the outcome of the tracking process in the previous frame. Here, the most popular methods in the context of object tracking are briefly outlined.

- **Interest points detectors** have been long used in object detection towards visual tracking, motion estimation and stereo. Point detectors are used to find interest points in images which have an expressive texture in their respective localities. Desirable features of an interest point is its invariance to changes in illumination and camera viewpoint. Commonly used interest point detectors are Harris [33], SIFT [44], KLT [45] and SURF [7], which exhibit significant invariances towards illumination changes, camera viewpoints, object scaling and rotation etc. A concise review followed by a thorough benchmarking of various interest point detectors is presented by Mikolajczyk and Schmid in [49, 48].

- **Background subtraction** is a well-known and widely used method for detecting moving objects in an image sequence captured by a stationary camera. Object detection can be achieved by building a pixel-wise representation of the scene (background model) and then finding deviations between the model and the next frame. Any significant change in an image region from the background model signifies a detected moving object, indicated by a binary mask. Usually, a connected components algorithm is applied to obtain connected regions corresponding to objects. Efficient background subtraction is performed by several methods, using multi-modal statistical models to describe per-pixel background color. The method in [67] utilizes a mixture of Gaussians to model the pixel color. A pixel in the current frame is checked against the background model by comparing it with every Gaussian in the model until a matching Gaussian is found. Each pixel is classified based on whether the matched distribution represents the background process. Another efficient approach incorporates region-based (spatial) scene information instead of only using color-based information, using non-parametric kernel density estimation to model the per-pixel background [25].

- **Segmentation** is also widely utilized to perform object detection in visual tracking tasks. Image segmentation is an important research field in computer vision, including a variety of methods to perform efficient image partitioning. Each segmentation algorithm addresses two main problems, the definition of criteria for a good partition and the algorithmic method for performing efficient partitioning [65]. Some of the most widely-used methods in image segmentation are the Mean-Shift clustering [15], Graph-Cuts [11], Active Contours[61, 55, 19, 9], Random Walks[31], Bayesian classification etc.

- Finally, **supervised learning** can be used for object detection by learning different views of the objects of interest from a set of example views. Common techniques of supervised learning are Support Vector Machines [3, 53], Adaptive Boosting [70], Neural Networks [63], Decision Trees etc.

1.1.1 A Short Categorization of Visual Object Tracking Methods

The aim of an object tracking algorithm is to establish correspondences between the object instances among video frames and to generate the trajectory of the object's position over time [72]. Object tracking is mainly dependent on the selected shape representation, appearance representation and the object detection mechanisms in order to establish the correspondences of the detected object instances among video frames. The object detection and tracking can be performed separately or jointly. In both cases, the objects are represented using the shape and/or appearance models described in Section 1.1. In the first case, possible object regions in every frame are obtained by means of an object detection algorithm, and then the tracker corresponds objects across frames. In the latter case, the object region and correspondence is jointly estimated by iteratively updating object location and region information obtained from previous frames. The selected shape and appearance representations control the type of motion or deformation and the appearance changes that the tracked object can undergo, respectively. The suitability of a particular tracking algorithm depends on object appearances, object shapes, number of objects, object and camera motions, and illumination conditions.

A short categorization of the state-of-art object tracking methods is provided by the recent and thorough review in [72].

The categorization of object tracking methods concerns three main categories namely, point tracking, kernel tracking and shape tracking and it is graphically illustrated in Fig. 1.2.

- **Point Tracking**. Objects are represented by points. Correspondences between objects in consecutive frames can be established based on deterministic or probabilistic methods [40, 5]. One of the most popular among deterministic method is the Hungarian algorithm [41]. Greedy search methods also belong to this category. The most popular subcategory consists of the probabilistic methods including Kalman filters [6] based and particle filters [36] based algorithms, HMMs [58], as well as the Multiple Hypothesis Tracking (MHT) [59] and Joint Probability Data Association Filter (JPDAF) techniques [20].

- **Kernel Tracking**. In kernel tracking methods [18, 68, 38, 37] a simple geometric

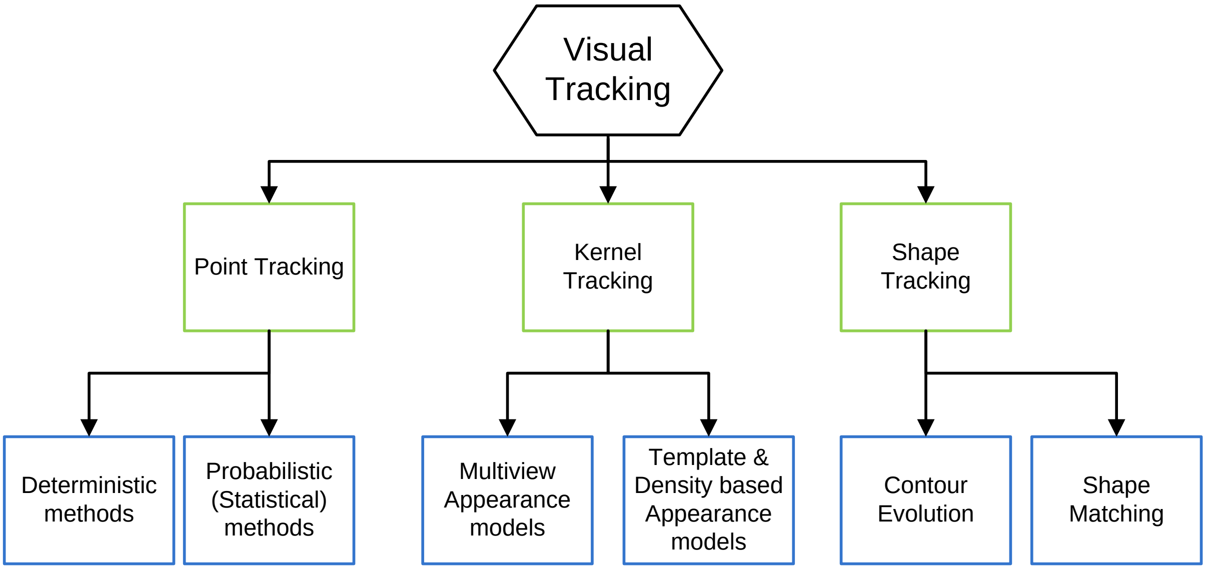


Figure 1.2: Categorization of visual tracking methods

shape is utilized to represent the region of the object of interest. Based on this, a parametric model for the object motion from frame to frame is computed. Thus, kernel-based methods provide a coarse representation of the object shape. Based on the utilized appearance representation, these tracking methods can be divided in two subcategories, including the template matching and density-based appearance models and the multi-view appearance-based models. The most popular and widely-used method of the first subcategory is the mean-shift object tracking algorithm [18, 15]. Moreover, Jepson et al. [38] proposed a novel method that tracks an object as a three-component mixture, consisting of the stable appearance features, transient features and a noise process. The object shape is represented by an ellipse, whereas an online version of the popular Expectation Maximization algorithm is used to learn the parameters of the three-component mixture.

Tracking methods based on the multi-view appearance models requires off-line learning of multiple views of an object. They mostly use Principal Component Analysis (PCA) to generate subspace-based representations as appearance models. They are able to track an object the appearance of which may undergo considerable changes over time [62]. Other methods use Support Vector Machines to classify on-line test views of the tracked object between positive and negative examples.

- **Shape Tracking**. The goal of the shape tracking methods is to track complex (non-rigid or articulated) shapes, providing an accurate shape description of the whole object area that evolves from frame to frame. They are able to capture potential deformations and transformations of the object shape and to provide an accurate object mask in each frame. There are two main subcategories, concerning the contour tracking approaches and the shape matching approaches. On the one hand, contour tracking methods iteratively evolve the initial object contour to capture the contour of the shape instance of the object in the current frame. Various energy minimization techniques (variational approaches) have been used to develop efficient contour tracking algorithms, such as level sets [55, 73, 9, 8], utilizing static image cues, optical flow information and region statistics. Moreover, state space models (Kalman filtering, Particle filtering) have been used to develop contour tracking methods [69],[36]. On the other hand, shape matching [66, 35] is closely related to tracking by template matching. An object silhouette supported by its associated appearance model is searched to capture the shape instance of the tracked object in the current frame. The appearance model as well as the object silhouette instance may have been incrementally updated exploiting the tracking result of the previous frame, thus handling appearance changes and shape deformations of the object from frame to frame.

1.1.2 Visual Object Tracking by Segmentation

This section provides a literature review on visual object tracking methods that explicitly or implicitly provide an accurate shape representation, enabling combined tracking and segmentation of the object area throughout an image sequence. The references on the research publications presented in this section are grouped toward the top-level categorization of tracking methods provided in Section 1.1.1, concerning the three main categories, namely the point, kernel and shape based tracking methods.

Shape-based tracking methods [69, 36, 55, 73, 9] provide an accurate representation of the tracked object, therefore they capture the entire object shape in each video frame, inherently providing combined tracking and segmentation. In [69], the object state is defined by the dynamics of the control points, which are modeled in terms of a spring model. This model moves the control points based on defined spring stiffness parameters. The new state (spring parameters) of the contour is predicted using the Kalman filter. The correction step uses the image observations which are defined in terms of the image gradients. The method presented in [36] defines the object state in terms of spline shape and affine motion parameters. The measurements consist of image edges computed in the normal direction to the contour. During the testing phase, the current state variables are estimated through particle filtering based on the edge observations along normal lines at the control points on the contour.

The direct minimization methods evolves the contour by minimizing the contour energy using direct minimization techniques (i.e gradient descent), variational methods (i.e level-sets) or heuristic approaches [61]. In [55], a variational framework is introduced for detecting and tracking multiple moving objects in image sequences using the front propagation theory and the level-set methodology. The motion detection boundaries are determined using a probabilistic edge detection on analysis of the inter-frame difference. The tracking boundaries are determined by performing edge detection on the input image. Then, a partial differential equation (PDE) is defined, as an objective function, to transform the detection and tracking into a geodesic computation problem. The equation is implemented using the level-set theory, whereas the obtained function is minimized using gradient descent.

A contour-based nonrigid object tracking method is proposed in [73]. The method is able to perform robust object tracking in the presence of occlusions in video acquired from moving cameras. Along with color and texture models generated for the object and the background regions, the method maintains an shape prior, which is generated on-line, for recovering occluded object parts. The energy functional, evolving the contour from frame to frame, is derived using a Bayesian framework and is evaluated within a band area around the estimated object contour. The energy function is minimized using gradient descent.

More recently, in [9], a probabilistic, level-set framework for robust visual tracking is introduced. The method handles the tracking problem using a bag-of-pixels representation, in terms of pixel-wise posteriors, as opposed to a product over pixel-wise likelihoods. On-line appearance learning provides continual refinement of both the object and background appearance models. The object shape is based on a level-set representation of its contour that is propagated by performing a rigid registration between frames. The proposed method is able to track previously unseen objects from a moving camera in real-time.

Point-tracking algorithms, either deterministic or statistics-based, can also combine tracking and object segmentation using multiple image cues. The two following tracking methods can be considered as a hybrid of point and shape-based tracking methods. In [60], figure/ground segmentation operates sequentially in each frame by utilizing both static image cues and temporal coherence cues. The method generates an appearance model of brightness (or color) and a spatial model propagating figure/ground masks through low-level region correspondence. A super-pixel-based Conditional Random Field (CRF) linearly combines cues and loopy belief propagation is used to estimate marginal posteriors of figure versus background, thus providing an accurate segmented object mask throughout an image sequence. A similar but more elaborate work is presented in [75]. This work provides a shape constrained figure-ground segmentation in a CRF graph model and proposes a new method to embed global shape probability and region-based probability of object boundary into graph link terms. Simulated annealing and local voting align the on-line obtained deformable shape template with the image to yield a global shape probability map. Moreover, multiple low-level image cues are fused to provide a region-based probability of the object boundary map. The obtained global shape probability is combined with the region-based probability of object boundary map and the pixel-level intensity gradient to determine each link cost in the graph formulation. The CRF energy is minimized by min-cut, followed by Random Walker-based segmentation on the uncertain boundary region to get a soft segmentation result. This method is able to handle partial occlusions of the object. The method described in [1], is mainly based on the efficient method proposed in [51], presenting a probabilistic framework that jointly considers both tracking and fine segmentation of multiple objects in videos captured by a stationary camera. The proposed method jointly formulates the pixel color and location in a Maximum a Posteriori (MAP) estimator to perform pixel-wise classification toward the target objects list and the background image. A Probabilistic PCA method (PPCA) is utilized to construct and on-line update a robust appearance model for each target object throughout the image sequence. Another multiple object tracking approach is introduced in [5] supporting hand and face tracking in videos captured by a possibly moving camera. A pixel-wise representation is utilized. The location and the speed of each object is modeled as a discrete time, linear dynamical system which is tracked using Kalman filtering. The spatial distribution of the pixels of each tracked object is passed on from frame to frame by propagating a set of pixel hypotheses, estimated by the Kalman filter.

The majority of the kernel-based tracking algorithms provide a coarse representation of each tracked object based on a bounding box or an ellipsoid region. The research work presented in [74], introduces a kernel-based tracking method that enables combined tracking and fine segmentation of non-rigid foreground objects in videos captured by a possibly moving camera. The foreground and background objects are modeled using spatial-color

Gaussian mixture models (SCGMM). These two models jointly capture the shape and the appearance, in terms of pixel-wise colors, of the foreground objects in the scene. Combining the two SCGMMs into a generative model of the whole image, the maximization of the joint data likelihood is computed using a constrained Expectation-Maximization (EM) algorithm [21]. The segmentation of the foreground objects is finally computed using the Graph-Cut algorithm, which minimizes a Markov Random Field (MRF) energy function modeled by the information encoded by the SCGMM models. Moreover, in [27] a novel method is presented for illumination invariant kernel tracking that is based on computing an illumination-invariant optical flow field in conjunction with a graph cuts formulation.

Another kernel-based method concerning foreground/background modeling, thus tracking foreground objects, is presented in [25]. A nonparametric kernel density estimation technique is presented, as a tool for constructing statistical representations for the scene background and foreground regions in video surveillance (stationary videos). A background modeling and background subtraction technique is also introduced. The statistical representations of the foreground regions (moving objects) support their tracking and occlusion reasoning throughout an image sequence. In [40] a Maximum a Posteriori (MAP) probabilistic framework for segmentation is presented , using multiple cues, such as spatial location, color and motion. A weighting scheme is introduced to weight pixel-wise color and motion terms, based on a confidence measure of each feature. The correct modeling of the spatial pdf imposes temporal and color consistency among the resulting image segments in consecutive frames. The segmentation and tracking of a specific object in the scene, could be a post-product of this work.

Finally, one of the most popular and efficient kernel-based tracking method of non-rigid objects is the mean-shift algorithm [18], which is not mentioned in this section because of its coarse representation of the tracking object area with an ellipse. One of the main drawbacks of the original work of mean-shift tracking is the lack of scale adaptation of the tracking kernel towards the object shape changes throughout an image sequence, which gradually diminishes the tracking performance. Numerous research works have been published trying to deal with the scale adaptation of the tracking kernel in order to get a more refined object representation. An extension of the original mean-shift tracking method is presented in [17], enabling a variable bandwidth of the mean shift search window. Another extension is presented in [13] exploiting the Lindeberg theory [42]. It refers to the feature scale selection based on local maxima of differential scale-space filters, providing a solution to the problem of selecting kernel scale for mean-shift blob tracking. A new formulation of the original mean-shift object tracking method is presented in [77], simultaneously estimating the position and the covariance matrix of the tracking kernel that describes the shape of tracking object based on a color-histogram and an EM-like procedure for scale selection. More recently, the proposed method in [71] presents an object tracking method based on the asymmetric kernel mean shift, in which the scale and orientation of the kernel adaptively change depending on the observations at each iteration. The afore-mentioned extensions of the original mean-shift method produce a better object representation than the original method capturing the shape/scale changes to some extend. However, this result is still characterized as a coarse representation of the tracked object area.

Despite the many important research efforts devoted to the problem, the development of algorithms for tracking objects in unconstrained videos constitutes an open research problem. Moving cameras, appearance and shape variability of the tracked objects, varying illumination conditions and clutter backgrounds constitute some of the challenges that a robust tracking algorithm needs to cope with. To this end, in this work we consider a novel framework that explicitly combines tracking and segmentation of previously unseen objects in monocular videos captured by a possibly moving camera. No strong constraints are imposed regarding the appearance and the texture of the target object or the rigidity of its shape. All of the above may dynamically vary over time under challenging illumination conditions and changing background appearance. The basic aim of this work is to preclude tracking failures by enhancing its target localization performance through explicit fine object segmentation that is appropriately integrated with tracking in a closed-loop algorithmic scheme. A kernel-based object tracking algorithm [77], a natural extension of the popular mean-shift tracker [16, 18], is efficiently combined with Random Walker-based image segmentation [29, 30]. Explicit segmentation of the target region of interest in an image sequence enables reliable tracking and reduces drifting by exploiting static image cues and temporal coherence. The final goal of the proposed methodology is to simultaneously enhance the performance of the kernel-based tracking and provide a fine segmentation result of the tracked object, as illustrated in Fig.1.3.

The key benefits of the proposed method are (i) the close-loop interaction between tracking and segmentation (ii) enhanced tracking performance under challenging conditions (iii) fine object segmentation (iv) the capability to track objects regardless of camera motion (v) increased tolerance to extensive changes of object's appearance and shape and, (vi) continual refinement of both the object and the background appearance

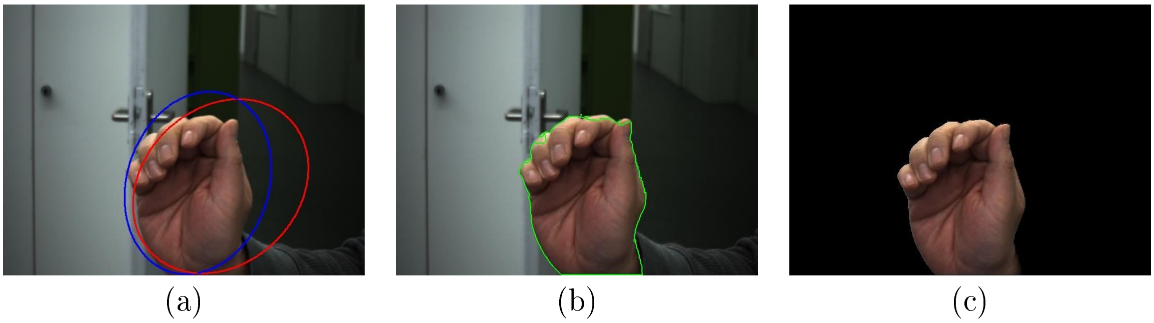


Figure 1.3: (a) Two ellipses representing a human hand while tracking. Red ellipse corresponds to the performance of the stand-alone EM-shift object tracking algorithm [77], whereas the blue ellipse corresponds to the tracking result obtained by the proposed methodology providing enhanced localization performance. (b) Precise object shape representation provided by the Random Walker-based segmentation procedure [31] of the proposed framework following the object tracking. (c) The desired result of the proposed methodology. A finely segmented object mask.

models. Last but not least, the proposed scheme can easily be extended to incorporate more image cues.

Chapter 2

Algorithmic Tools

The purpose of this chapter is to provide an introduction to previously presented algorithms that have been utilized in the proposed methodology, concerning EM-shift kernel-based object tracking [77] and Random Walker-based image segmentation [31], respectively.

2.1 Kernel-based Object Tracking

In this section, an kernel-based object tracking algorithm is presented, that is further utilized in the proposed joint tracking and segmentation methodology. The EM-shift algorithm for color-histogram based object tracking, that has previously appeared in [77], is a natural extension of the popular mean-shift tracking method [18, 16]. There are two main advantages regarding this previously introduced method. Firstly, its robustness in tracking performance keeps up with the performance of the popular Mean-Shift object tracking method. Secondly, the EM-shift algoritm simultaneously estimates the position of the local mode and the covariance matrix that describes the approximate shape of the local mode, thus adapting the position and the scale of the tracking kernel. Both object tracking methods can be seen as special versions of closely-related robust statistics procedures [47, 34] toward the extreme outlier model, described in [77]. Both methods perform kernel-based tracking and rely on image color. In the following, a brief description and derivation of the color-histogram based tracking algorithm is provided.

Given an image I_t of size $m \times n$, the data set of the N independent samples-pixels are denoted by $X = \left\{\vec{x}_1, \ldots, \vec{x}_N\right\}$, where x_i denotes the location (spatial coordinates) of pixel i. Moreover, a Gaussian probability density function $p\left(\vec{x}\right) = N\left(\vec{x}; \vec{\theta}, \Sigma\right)$ is considered, as a generative model to efficiently represent the data samples. The $\vec{\theta}$ and

Σ parameters correspond to the position mean vector and the covariance matrix of a Gaussian distribution, respectively, which approximates the shape of the tracked object. A projection of the Gaussian distribution on the image plane consists of an ellipsoid region that represents the object shape while tracking. The spatial covariance Σ is based on the second order moment that approximates the shape of the object:

$$\Sigma = \sum_{x_i \in X_o} \left(\vec{x_i} - \vec{\theta_0}\right) \left(\vec{x_i} - \vec{\theta_0}\right)^T , \tag{2.1}$$

where $X_o \subseteq X$ is the subset of pixels that belong to the object area.

The appearance of the ellipsoid region is modeled by an M-bins color histogram. Let $b(\vec{x_i}) : R^2 \rightarrow 1, \ldots, M$ be the function that assigns a color value of the pixel at location $\vec{x_i}$ to its bin. The color histogram model of the object consists of the values of the M bins of the histogram $\vec{o} = [o_1, \ldots, o_M]^T$. The value of the m-th bin is calculated by:

$$o_m = \sum_{i=1}^{N_{v_0}} N\left(\vec{x_i}; \vec{\theta_0}, \Sigma_0\right) \delta[b\left(\vec{x_i}\right) - m], \tag{2.2}$$

where δ is the Kronecker delta function. The effect of the utilized Gaussian kernel N is to rely more on the pixels in the center of the object and to assign smaller weights to the less reliable pixels near the borders of the object. Moreover, the pixels from a finite neighborhood N_{Σ_0} of the kernel N are used to populate the color histogram, whereas the pixels further than 2.5-sigma are disregarded.

The goal of object detection in each frame based on its appearance model can be achieved by computing a Maximum Likelihood (ML) estimation of the Gaussian pdf $p(\vec{x}_i)$ that maximizes the likelihood function $\prod_{i=1}^{N} p(\vec{x}_i)$. Based on the notion of the extreme outlier model and the Taylor expansion, a new pixel-wise weighted objective function to be maximized is derived:

$$f\left(\vec{\theta}, \Sigma\right) = \sum_{i=1}^{N} \omega_i N\left(\vec{x_i}; \vec{\theta}, \Sigma\right). \tag{2.3}$$

The pixel-wise weight factors ω_i of the objective function are estimated iteratively by computing the Bhattacharrya coefficient based similarity measure between the color histograms of the target and the candidate regions of a new frame, where the object is to be detected/tracked. Let $N_v \subseteq N$ be the subset of pixels that belong to a candidate subregion of the new image frame I_{t+1}, with position $\vec{\theta_c}$ and covariance Σ_c, where the target object may be localized. Using Eq. (2.2) to model the color information of pixels in N_v, an appearance model of the candidate region is generated, denoted as $r_m(\vec{\theta_c}, \Sigma_c)$. The goal is now to compare these two color histograms by using the Bhattacharrya coefficient ρ, as a measure of similarity between two histograms:

$$\rho[\vec{r}\left(\vec{\theta_c}, \Sigma_c\right), \vec{o}] = \sum_{m=1}^{M} \sqrt{r_m\left(\vec{\theta_c}, \Sigma_c\right)}\sqrt{o_m}. \tag{2.4}$$

The first Taylor approximation of the current estimate $\vec{r}(\vec{\theta_c}, \Sigma_c)$ is given by:

$$\vec{r}(\vec{\theta_c}, \Sigma_c) \approx c_1 + c_2 \sum_{i=1}^{N_v} \omega_i N\left(\vec{x_i}; \vec{\theta}, \Sigma\right), \tag{2.5}$$

where the c_1 and c_2 are constant factors. Since the last term of Eq. (2.5) has the same form as the object function in Eq. (2.3), an EM-shift algorithm can be utilized to search for the local maximum of the current similarity function Eq. (2.5), as will be described below. In other words, an EM-like procedure will search for the candidate image subregions, where the candidate appearance model maximizes the similarity with the target appearance model. The ω_i values in Eq. (2.5) are computed as:

$$\omega_i = \sum_{m=1}^{M} \sqrt{\frac{o_m}{\vec{r}\left(\vec{\theta_c}, \Sigma_c\right)}}\delta[b\left(\vec{x_i}\right) - m]. \tag{2.6}$$

The key point of the described method is the multiplication of the estimated density function (2.5) by $|\Sigma|^\gamma$. The objective function to be maximized now is called 'γ-normalized':

$$f_\gamma\left(\vec{\theta}, \Sigma\right) = |\Sigma|^{\gamma/2} f\left(\vec{\theta}, \Sigma\right). \tag{2.7}$$

Note that $\gamma \in (0, 1)$. The 'γ-normalization' introduces an informative prior for Σ to regularize the solution and get non-biased estimates. An interesting connection of this technique is with some image filtering algorithms. For example, in [42], γ-normalized image convolution was studied for selecting the scale of the filtering operator.

To bring up again the main computational core of the described tracking method, parameter $\vec{\theta}$ and Σ for which the maximum value of Eq. (2.7) is achieved. Based on the Jensen's inequality of the 'γ-normalized' density function, we get:

$$log f_\gamma(\vec{\theta}, \Sigma) \geq G(\vec{\theta}, \Sigma^{\gamma/2}, q_1, \ldots, q_N) = \sum_{i=1}^{N} log|\Sigma^{\gamma/2}| \left(\frac{\omega_i N\left(\vec{x_i}; \vec{\theta}, \Sigma\right)}{q_i}\right)^{q_i}, \tag{2.8}$$

where q_i-s are non-negative arbitrary constants and $\sum_{i=1}^{N} q_i = 1$. Jensen's inequality relates the value of a convex function of an integral to the integral of the convex function.

Given its generality, the inequality appears in many forms depending on the context, especially in probability theory, generalizing the statement that the secant line of a convex function lies above the graph of the function (see [64] for details).

The idea is to involve the q_i-s parameters that are contained in (2.8) and the desired spatial parameters of the tracking kernel $\vec{\theta}$ and Σ from the 'γ-normalized' objective function (2.7) in an iterative Expectation Maximization procedure to obtain the desired Maximum Likelihood solution. At the same time, the obtained parameters $\vec{\theta}$ and Σ will provide an accurate representation of the object area in the new image, position and covariance respectively, enabling the automatic scale selection of the tracking region, through the estimated covariance parameter Σ.

The EM algorithm is performed in the following E and M steps that are repeated until convergence. Denote by $\vec{\theta^{(k)}}$ and $\Sigma^{(k)}$ the estimates of the parameters at iteration k.

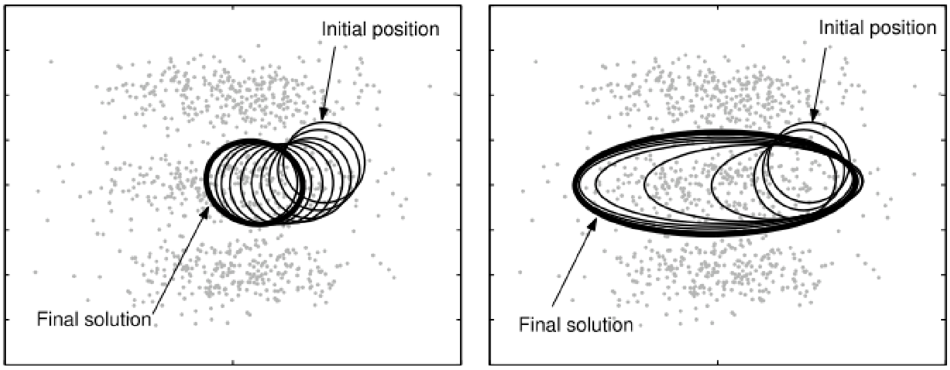
- **E step:** Find q_i-s to maximize G in (2.8) while keeping $\vec{\theta^{(k)}}$ and $\Sigma^{(k)}$ fixed, by using:
$$q_i = \frac{\omega_i N\left(\vec{x_i};\vec{\theta},\Sigma\right)}{\sum_{i=1}^{N}\omega_i N\left(\vec{x_i};\vec{\theta},\Sigma\right)}. \tag{2.9}$$
- **M step:** Maximize G in (2.8) with respect to $\vec{\theta^{(k)}}$ and $\Sigma^{(k)}$ while keeping q_i-s constant. To achieve this, the part of G that depends on the parameters need to be minimized. This part is $g(\vec{\theta}) = \sum_{i=1}^{N} q_i log|\Sigma|^{\gamma/2} N\left(\vec{x};\vec{\theta},\Sigma\right)$.
From $\frac{\partial}{\partial\theta}g(\vec{\theta},\Sigma) = 0$, the position and the covariance parameters are updated by:
$$\vec{\theta^{k+1}} = \sum_{i=1}^{N} q_i\vec{x_i}, \tag{2.10}$$
$$\vec{\Sigma^{k+1}} = \beta_{track}\sum_{i=1}^{N} q_i\left(\vec{x_i}-\vec{\theta^{(k)}}\right)\left(\vec{x_i}-\vec{\theta^{(k)}}\right)^T, \tag{2.11}$$

where $\beta_{track} = 1/(1-\gamma)$. An outline of the EM-shift color histogram based tracking method follows is provided in Algorithm.1.

In Fig. 2.1 an example is shown to illustrating the performance of the presented EM-shift object tracking algorithm. The simulated data consists of 600 samples generated using a mixture of three Gaussian distributions. The three modes are clearly visible (horizontally aligned). The evolution of the tracking kernel, computed through mean-shift iterations is illustrated in Fig. 2.1(a). In Fig. 2.1(b), the kernels computed during

Algorithm 1 EM-shift color-based object tracking algorithm

```
Input: Image $I_{t+1}$, object model $\vec{o}_{m,t}$, its initial location $\vec{\theta}^{0}_{t+1}$ and shape covariance $\Sigma^{0}_{t+1}$
    for frame $I_{t+1}$.

    1. Set k=0 (iterations)

    2. Compute the candidate color histogram $r_m\left(\vec{\theta}^{(k)}_{t+1}, \Sigma^{(k)}_{t+1}\right)$ of the current region
       defined by $\vec{\theta}^{k}_{t+1}$ and $\Sigma^{(k)}_{t+1}$.

    3. Calculate weights $\omega_i$ using (2.6).

    4. Perform E step of EM algorithm. Compute $q_i$-s using (2.9).

    5. Perform M step of EM algorithm.

        • Compute new position estimate $\vec{\theta}^{(k+1)}_{t+1}$ using (2.10).

        • Compute new covariance estimate $\Sigma^{(k+1)}_{t+1}$ using (2.11).

    6. If no new pixels are included in the new elliptical region defined by the new
       estimates $\vec{\theta}^{(k+1)}_{t+1}$, $\Sigma^{(k+1)}_{t+1}$ stop.
       Otherwise, set k=k+1 and go to 1.

Output: An ellipse that contains the tracked object in frame $I_{t+1}$, defined by $\vec{\theta}^{(k+1)}_{t+1}$,
    $\Sigma^{(k+1)}_{t+1}$.
```



a) mean-shift iterations

b) EM-shift iterations

Figure 2.1: Qualitative performance of (a)mean-shift and (b) EM-shift tracking algorithms on synthetic data representing a mixture of 3 Gaussians. Note the scale adaptation of the Gaussian kernel to covariance of the middle local mode of the mixture, achieved by the EM-shift algorithm, as opposed to the one of the Mean-shift algorithm where no scale adaptation is performed.(Figure originally appeared in [77]).

the iterations of the EM-shift algorithm with $\gamma = 1/2$ ($\beta_{track} = 2$) are illustrated. The algorithm simultaneously estimates both the position of the local mode and the covariance matrix that describes the shape of the mode.

As described in [77], $\beta_{track} = 2$ is appropriate in case of a Gaussian distribution. If some other distribution is approximated by a Gaussian some other value for β_{track} might be needed in order to avoid biased solutions.

2.2 Random Walks for Image Segmentation

The aim of image segmentation is the partition of the image pixels into a set of regions, which are visually distinct and uniform with respect to some property, such as gray level, texture, color, etc. Another natural bottom-up view of segmentation is the grouping of image sub-regions, or pixels, attempting to determine visually distinct and uniform regions from image parts that naturally "belong together", based on a given property/criterion.

Both partitioning and grouping can be considered as categories of the clustering problem, often referred in the literature as divisive and agglomerative clustering, respectively. The general intuitive goal of clustering is to divide the data points into several groups such that points in the same group are similar and points in different groups are dissimilar to each other.

A large number of segmentation methods have been proposed in the literature and a review or a taxonomy of the methods is beyond the scope of this report. A brief introduction to the segmentation based on the graph partitioning approach of spectral clustering will be firstly presented, enabling a smooth transition to the description of the Random Walker-based image segmentation method, which is the main point of the current section.

2.2.1 Mathematical Background

Spectral clustering goes back to Donath and Hoffman in 1973 [23], who first suggested to compute graph partitions based on eigenvectors of the adjacency matrix of an available dataset.

Let a set of data points $x_1, \ldots, x_n$ and some notion of similarity $s_{ij} \geq 0$ between all pairs of data points x_i and x_j. Consider $G = (V, E)$ to be an *undirected weighted graph* with *vertices (nodes)* $u \in V$ and *edges* $e \in E \subseteq V \times V$, with $n = |V|$ and $m = |E|$, where $| \cdot |$ denotes cardinality. Each vertex v_i in this graph represents a data point x_i. An edge e, spanning two vertices, u_i and u_j, is denoted as e_{ij}, weighted with a non negative value denoted as w_{ij}. The weighted *adjacency matrix* of the graph G is the $W = (w_{ij})_{i,j=1,\ldots,n}$. The *degree of a vertex* $u_i \in V$ is defined as

$$d_i = \sum_{e_{ij}} w_{(e_{ij})}, \qquad \forall e_{ij} \in \mathbf{E}. \tag{2.12}$$

The *degree matrix* D is defined as the diagonal matrix with the degrees $d_1, \ldots, d_n$ on the diagonal.

Define the $m \times n$ *edge-vertex incidence matrix* as

$$A_{e_{ij}u_k} = \begin{cases} +1 & \text{if } i = k \\ -1 & \text{if } j = k \\ 0 & \text{otherwise,} \end{cases} \tag{2.13}$$

for every vertex u_k and edge e_{ij}. The notation $A_{e_{ij}u_k}$ is used to indicate that the rows of A are indexed by edge e_{ij} and the columns by node u_k. Moreover, define the $m \times m$ *constitutive matrix*, C, as the diagonal matrix with the weights w_{ij} of each edge e_{ij} along the diagonal.

For any two subsets of vertices $A, B \subset V$, the *weight matrix* W is defined as

$$W(A, B) = \sum_{i \in A, j \in B} w_{ij}. \tag{2.14}$$

There are several popular constructions to transform a given set $x_1 \ldots, x_n$ of data points with pairwise similarities s_{ij} or pairwise distances d_{ij} into a graph. When constructing a weighted graph the goal is to model the local neighborhood relationships between the data points. Thus, the most common used types of graphs, concerning the neighboring connectivity are:

- ϵ **- neighborhood graph**: All vertices whose pairwise distances are smaller than e are connected.
- **k - nearest neighbor graphs**: Any vertex v_i is connected with vertex v_j if the latter is among the k-nearest neighbors of v_i.
- **Fully connected graph**: connect all vertices with positive weights with each other based on the evaluation of the defined similarity function over the corresponding data points.

The main tools for spectral clustering are graph Laplacian matrices. There exists a whole field dedicated to the study of those matrices, called spectral graph theory [46], however there is no unique convention which matrix exactly is called "graph Laplacian" [22]. As an operator, A may be interpreted as a combinatorial gradient operator and A^T as a combinatorial divergence [12]. The isotropic combinatorial Laplacian is the composition of the combinatorial divergence operator with the combinatorial gradient operator, $L = A^T A$. There are numerous approaches to the construction of a graph Laplacian matrix of a graph-based representation of a given dataset. The unnormalized graph Laplacian is defined as

$$L = D - W. \tag{2.15}$$

The unnormalized Laplacian matrix is symmetric and positive-definite. The smallest eigenvalue of L is 0, the corresponding eigenvector is the constant one vector. Moreover, L has n non-negative, real-valued eigenvalues $\lambda_i \geq 0$.

There are two matrices called normalized graph Laplacians, defined as

$$L_{sym} = I - D^{-1/2} W D^{-1/2}, \tag{2.16}$$

$$L_{rw} = I - D^{-1} W. \tag{2.17}$$

L_{sym} and L_{rw} have n non-negative, real-valued eigenvalues $\lambda_i \geq 0$. The *Laplacian operator matrix* can be defined as $L = A^T A$. As a matrix, the Laplacian may be derived directly from knowledge of V and E as:

$$L_{u_i u_j} = \begin{cases} d_i & \text{if } i = j \\ -w_{e_{ij}} & \text{if } e_{ij} \in E \\ 0 & \text{otherwise,} \end{cases} \tag{2.18}$$

The notation $L_{u_i u_j}$ is used to indicate that the matrix L is being indexed by vertices v_i and v_j. To give an intuitive implementation of a supervised spectral clustering, two general algorithmic templates are presented below, based on the unnormalized and the normalized Laplacian, respectively.

In all spectral clustering algorithms, the main idea is to change the representation of the abstract data points x_i to points y_i in $\Re^k$. It is due to the properties of the graph Laplacians that this change of representation is useful, so that clusters can be trivially detected in the new representation. The wide variety of spectral clustering algorithms is up to the number of choices concerning the type of similarity graph, the weighting function and the type of the Laplacian matrix that will be chosen to obtain the resulting clusters.

Graph-Cuts and Random Walks are two special cases of spectral clustering toward the graph-based partitioning problem. A random walk on a given similarity graph is a stochastic process which randomly jumps from vertex to vertex, according to [43]. A spectral clustering algorithm based on random walks can be interpreted as trying to find a partition of the graph such that a random walk that begins from a given cluster, stays long within that cluster and seldom jumps between clusters.

```
Algorithm 2 Unnormalized spectral clustering
Input: Similarity matrix $S \in \Re^{n \times n}$, number of $k$ clusters to construct
    • Construct a similarity graph based on the desired graph connectivity scheme. Generate W, the weighted adjacency matrix, based on the chosen weighting function.
    • Compute the unnormalized Laplacian L of (2.15).
    • Compute the first k eigenvectors $u_1, \ldots, u_k$ of L.
    • Let $U \in \Re^{n \times k}$ be the matrix containing the vectors $u_1, \ldots, u_k$ as columns.
    • Consider the vector $y_i \in \Re^k \; fori = 1, \ldots, n$ to be the $i - th$ row of $U$.
    • Cluster the points $y_i$ in $\Re^k$ with the $k$-means algorithm into clusters $C_1, \ldots, C_k$.
Output: Clusters $A_1, \ldots, A_k$ with $A_i = j | y_j \in C_i$.
```

```
Algorithm 3 Normalized spectral clustering according to [65]
Input: Similarity matrix $S \in \Re^{n \times n}$, number of $k$ clusters to construct
    • Construct a similarity graph based on the desired graph connectivity scheme. Generate W, the weighted adjacency matrix, based on the chosen weighting function.
    • Compute the unnormalized Laplacian L of (2.15).
    • Compute the first k generalized eigenvectors $u_1, \ldots, u_k$ of L of the generalized eigenproblem $Lu = \lambda Du$.
    • Let $U \in \Re^{n \times k}$ be the matrix containing the vectors $u_1, \ldots, u_k$ as columns.
    • Consider the vector $y_i \in \Re^k \; fori = 1, \ldots, n$ to be the $i - th$ row of $U$.
    • Cluster the points $y_i$ in $\Re^k$ with the $k$-means algorithm into clusters $C_1, \ldots, C_k$.
Output: Clusters $A_1, \ldots, A_k$ with $A_i = j | y_j \in C_i$.
```

The transition probability of a random walker jumping in one step from vertex v_i to vertex v_j is proportional to the edge weight w_{ij} and is given by $p_{ij} = w_{ij}/d_i$. The transition matrix $P = (p_{ij})_{i,j=1,...,n}$ of the random walk is thus defined by $P = D^{-1}W$.

If the graph is connected and non-bipartite, then the random walker always possesses a unique stationary distribution $\pi = (\pi_1, \ldots, \pi_n)$, where $\pi_i = d_i/vol(V)$ and $vol(V) = \Sigma_{i \in V} d_i$. It stands that λ is an eigenvalue of L_{rw} with eigenvector u if and only if $1 - \lambda$ is an eigenvalue of P with eigenvector u. It is well known that many properties of a graph can be expressed in terms of the corresponding random walk transition matrix P, see [43] for an overview. Therefore, the largest eigenvectors of P and the smallest eigenvectors of L_{rw} can be used to describe cluster properties of the graph, thus to further develop a spectral clustering algorithm based on random walks on a graph by utilizing the Laplacian L_{rw}.

An application of a graph partitioning algorithm based on Random Walks to the image segmentation problem [30, 29] will be presented in Section 2.2.2.

2.2.2 Random Walker-based Image Segmentation

In [31, 30] a novel approach to the K-way image segmentation problem is presented, based on the formulation of Random Walks on a graph-based representation of an image. Given user-defined seeds (each seed is a single or a set of image pixels) indicating regions of the image belonging to K objects, consider that each seed specifies a location with a user-defined label. The introduced algorithm labels an unseeded pixel by answering the question: *Given a random walker starting at this location, what is the probability that it first reaches each of the K seed points?*

By performing the algorithmic computation, a K-tuple vector is assigned to each unseeded pixel that specifies the probability that a random walker starting from that pixel will first reach each of the K seed points (soft segmentation-the values in each tuple sum up to unity). A final segmentation may be derived from these K-tuples by selecting for each pixel the most probable seed destination for its random walker. By biasing the random walker to avoid crossing sharp intensity gradients, a quality segmentation is obtained that respects object boundaries (including weak boundaries), as opposed to the popular graph-cut algorithm that is guaranteed to give the minimum-cut between two groups of labeled nodes. The above statement is validated by the segmentation of the synthetic-image-example illustrated in Fig. 2.2.

This calculation can be performed without the actual simulation of a random walk, which is infeasible for segmentation problems. To obtain the K-tuple vector of probabilities for each unseeded graph vertices, a sparse, symmetric positive-definite system of $K-1$ linear equations must be solved. An analytic mathematical formulation of the problem is presented in [30].

The advantage of formulating the problem on a graph is that purely combinatorial operators may be used that require no discretization and therefore incur no discretization errors or ambiguities. It has been previously established [39, 24] that the probability a random walker first reaches a seed point exactly equals the solution to the Dirichlet problem [39] with boundary conditions at the locations of the seed points and the seed point in question fixed to unity while the others are set to zero.

To begin with the review of the main algorithmic parts, an image should be treated as a purely discrete object, thus the undirected similarity graph $G = (V, E)$, as defined in Section 2.2.1. Each vertex of the graph now represents an image pixel, whereas an undirected edge between any two vertices represents the interaction between the corresponding image pixels or a set of pixels within a local neighborhood. Each edge is assigned a real-valued weight corresponding to the likelihood that a random walker will cross that edge (e.g., a weight of zero means that the walker may not move along that edge). The likelihood value is computed based on the weighting function W that evaluates a single or multiple combined properties of the interacting pixels.

Let $n = |V|$ the number of image pixels and $m = |E|$ the number of edges that connect interacting vertices (pixels) in the constructed graph. Given the seeds, the set of graph vertices V is divided into two disjoint subsets, the set of labeled (marked) vertices V_m and the set of unlabeled (unmarked) vertices V_U, such that $V_m \cup V_U = V$. The goal of the K-way graph-based segmentation is to label each *free vertex* $u_i \in V_U$ with a label from the set $G = \{g^1, \ldots, g^k\}$. The marked vertices are assigned with a label $y_i \in G$.

The random walker approach to this problem is to assign to each *free vertex* $u_i \in V_U$, *the probability* x^s that a random walker starting from that vertex first reaches a marked vertex $v_j \in V_m$ assigned to label g^s (set $x_j^s = 1$), as opposed to reaching a vertex $v_j \in V_m$ with label $g^{q \neq s}$ (set ti $s_j = 0$), obtaining a soft segmentation. The solution to this problem is given by the minimization of the following energy equation:

$$E_{spatial} = x^{sT} L x^s, \tag{2.19}$$

where x^s is a real-valued $n \times 1$ vector. L represents the combinatorial Laplacian matrix of size $n \times n$ defined in Eq. (2.18) of Section 2.2.1. By partitioning the Laplacian matrix into labeled L_x and free blocks B, we obtain:

$$L = \begin{bmatrix} L_M & B \\ B^T & L_U \end{bmatrix} \tag{2.20}$$

and defining the indicator vector f^s of size $|V_M| \times 1$, as

$$f_j^s = \begin{cases} 1 & \text{if } y_j = g^s \\ 0 & \text{if } y_j \neq g^s, \end{cases} \tag{2.21}$$

the minimization of the energy in (2.19) with respect to x_U^s can be obtained by solving the following sparse, symmetric, positive-definite system of K linear equations

$$L_U x_U^s = -Bf^s. \tag{2.22}$$

For the resulting probabilities, it holds that $\sum_s x_i^s = 1, \forall i$. Therefore, each graph vertex is soft-assigned to each of the K labels. The final segmentation is completed by assigning each free vertex to the label for which it has the highest probability, i.e., $y_i = max_i(x_i^s)$. The above derivation reveal a property of the Random Walker-based algorithm: In the absence of labeled points (i.e., $V_M = \oslash$), the probabilities are undefined. Therefore, this algorithm is presented as a strictly semi-automated segmentation algorithm.

An extension of this algorithm is introduced in [29], presenting a new mechanism that enables the incorporation of label priors into the above framework and resulting in a segmentation algorithm that need not have any user interaction or an explicit determination of seeds. Given a set of real-valued vertex-wise priors λ_i^s that represent the probability density that a feature (i.e pixel color intensity) at vertex u_i belongs to the distribution of label g^s, the diagonal square matrix Λ^s is defined having the values of λ^s on the diagonal. A new functional is considered based on the label prior values. The so called aspatial functional may be combined into a single functional weighted by the free parameter γ:

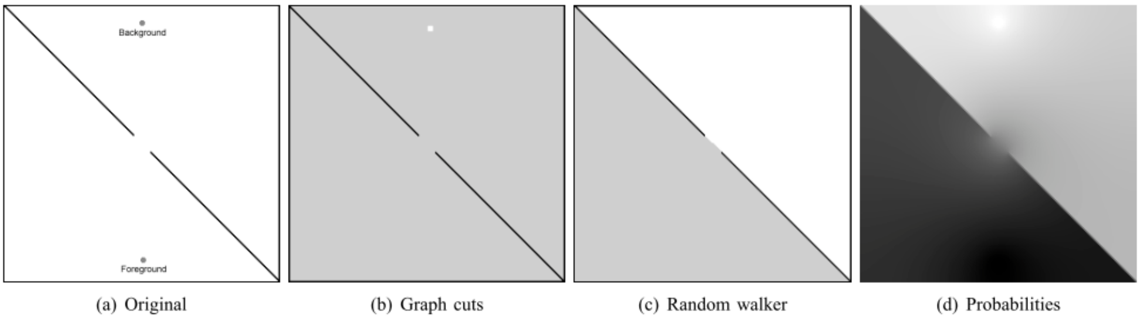
$$E_{total}^s = E_{spatial}^s + \gamma E_{aspatial}^s. \tag{2.23}$$

The minimum energy of (2.23) is obtained by solving the following modified system with respect to x^s

$$\left(L + \gamma \sum_{r=1}^{k} \Lambda^r \right) x^s = \gamma \lambda^s. \tag{2.24}$$

The modified system of linear equations is guaranteed to be positive definite (and therefore nonsingular), since L is positive semi-definite and the diagonal matrices (see Λ) are strictly positive definite. If desired, the seeds may also be incorporated by solving a new system

$$\left(L_U + \gamma \sum_{r=1}^{k} \Lambda^r \right) x_U^s = \gamma \lambda_U^s - Bf^s. \tag{2.25}$$



(a) Original (b) Graph cuts (c) Random walker (d) Probabilities

Figure 2.2: Comparison of Random Walker-based algorithm to graph cuts for a weak boundary using seeds. (a) Original synthetic image. (b) Graph cuts solution. (c) Random walker-based solution. (d) Pixel-wise probabilities computed by the Random Walker-based algorithm (Figure originally appeared in [30]).

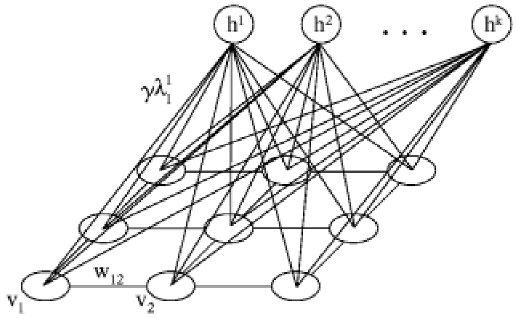


Figure 2.3: Graph construction of Random Walker-based graph partitioning technique using label priors (Figure originally appeared in [29]).

The introduced parameter γ controls the weighting of the prior values over the observations encoded by the edge weights of the graph and the seeds in case they are considered. Thus, the tuning of parameter γ is crucial to the behavior and the efficiency of the random walks based image segmentation technique presented. A new graph representation is required to enforce the incorporation of the prior values to the problem. The graph representation illustrated in Fig. 2.3, as well as the development of the algorithm bears a close resemblance to the construction of the graph cuts problem with the inclusion of vertex-wise priors. In the terminology of graph-cuts the weights w_{ij} of an edge e_{ji} between vertices u_i and u_j corresponds to the *N-links* (or pairwise energy potentials) and the weights $\gamma\lambda_i^s$ to the *T-links* (or unary energy potentials).

To summarize, three different closely related algorithms may be obtained based on the development in this subsection and the provided equation regarding the systems of linear equations that simulate the random walks of a similarity graph. A first algorithm is based only on the defined seed points, utilizing the Eq. (2.22). In that case, the user interaction or a automatic mechanism that will determine the seed points is necessary, based on specified image cues (i.e color, texture etc) depending on the application. A second algorithm is based only on label prior values. In that case, a mechanism that will provide the prior values is necessary, based also on specified image cues depending on the application and Eq. (2.24) is utilized. The third case refers to a combined algorithm, where both seeds and priors are combined in Eq. (2.25). A single algorithmic template of the three aforementioned algorithms is provided in Algorithm 4.

It is important to note that the main computational hurdle regarding the described random walks based image segmentation is the numerical solution of each of the large, sparse, symmetric positive-definite systems of linear equations i.e Eq. (2.22). Iterative methods, such as preconditioned conjugate gradient, exhibit a more acceptable memory consumption, as well as easy parallelization, as opposed to the direct methods (e.g LU decomposition), see [28] for an excellent treatment on matrix computations.

Algorithm 4 Random Walker based image segmentation algorithm [29, 30]

```
Input: Image I, weighting function W, optional:Seeds
    1. Construct an undirected graph to model pixels $I_{ij}$. Decide on their connectivity.
    2. Use the weighting function W to generate edge weights $w_{ij}$, between any two
       connected vertices of the graph, $u_i$ and $u_j$. Function W may model the distance
       of color intensities between connected vertices.
    3. Construct the graph Laplacian L.
    4. Use the seed vertices to generate vector $f$ from (2.21), if available.
    5. OPTIONAL: Compute prior values $\lambda_i^s$ for each vertex $u_i$ for all potential labels
       $g^s$ based on application-specific image cues.
    6. Use an efficient numerical method to solve the appropriate system of linear equa-
       tions ((2.22),(2.24),(2.25)).
Output: A probability vector $x_U^s$ for each unlabeled graph vertex(pixel) to belong to
    each of the potential labels $g^s$ (class regions).
```

Figure 2.4, illustrates the qualitative segmentation performance of the potential Random Walker-based algorithms against a simple density estimation based segmentation.

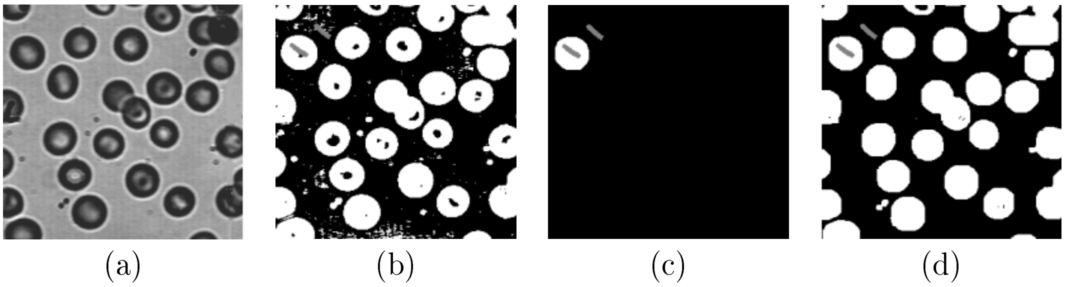


Figure 2.4: (a) input image of cells. (b) Density estimation-based segmentation result. (c) Random Walker-based segmentation result by using seeds only. (d) Random Walker-based segmentation result using seeds and priors (Figure originally appeared in [29]).

The simple density estimation of the two groups (cells, background) finds pieces of the cells and background, but ultimately yields a fractured segmentation that lacks spatial cohesion. Applying the Random Walker-based algorithm yields a correct segmentation of the cell within which the seeds were placed, but incorrectly identifies the other cells. In that case, additional seed points are required within each cell that is to be segmented, which leads to an undesired situation. The extended Random Walker-based segmentation formulations based on equations (2.24), (2.25) efficiently combine the intensity profiling and long-range aspects of the density estimation approach with the spatial cohesion of the Random Walker-based algorithm in a principled way that produces the correct result, despite variability of the intensity values present in the image, according to [29]. Moreover, their novelty is to extend the success of the basic Random Walker approach (Eq. (2.22)) by employing image priors to find disconnected pieces of an object and to remove the necessity of user interaction, which set these algorithmic procedures suitable and highly efficient to be used in the proposed framework of joint tracking and segmentation.

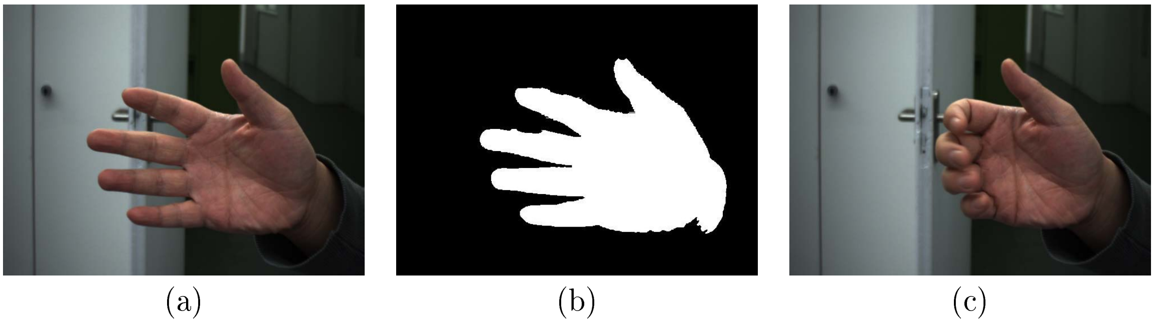
Chapter 3

Methodology

3.1 Method Preface

For each input video frame, the proposed framework encompasses a number of algorithmic steps, tightly interconnected in a closed-loop, which is illustrated schematically in Fig.3.2. To further ease understanding, Fig. 3.3 provides sample intermediate results of the most important algorithmic steps.

The method assumes that at a certain moment t in time, a new image frame I_t becomes available and that a fine object segmentation mask M_{t-1} is available, as a result of the previous time step $t-1$ (see Fig. 3.1). For time $t = 0$, M_{t-1} should be provided for initialization purposes. Essentially, M_{t-1} is a binary image, where foreground object pixels have a value of 1 and background pixels that of 0. The goal of the method is to produce the current object segmentation mask M_t. Towards this end, the spatial mean and covariance matrix of the foreground region of M_{t-1} is computed, thus defin-


(a)

(b)

(c)

Figure 3.1: (a) Previous image frame at time t-1. (b) Segmented object mask M_{t-1} of frame I_{t-1}. (c) New image frame I_t at current time t, where the object is to be tracked and segmented.

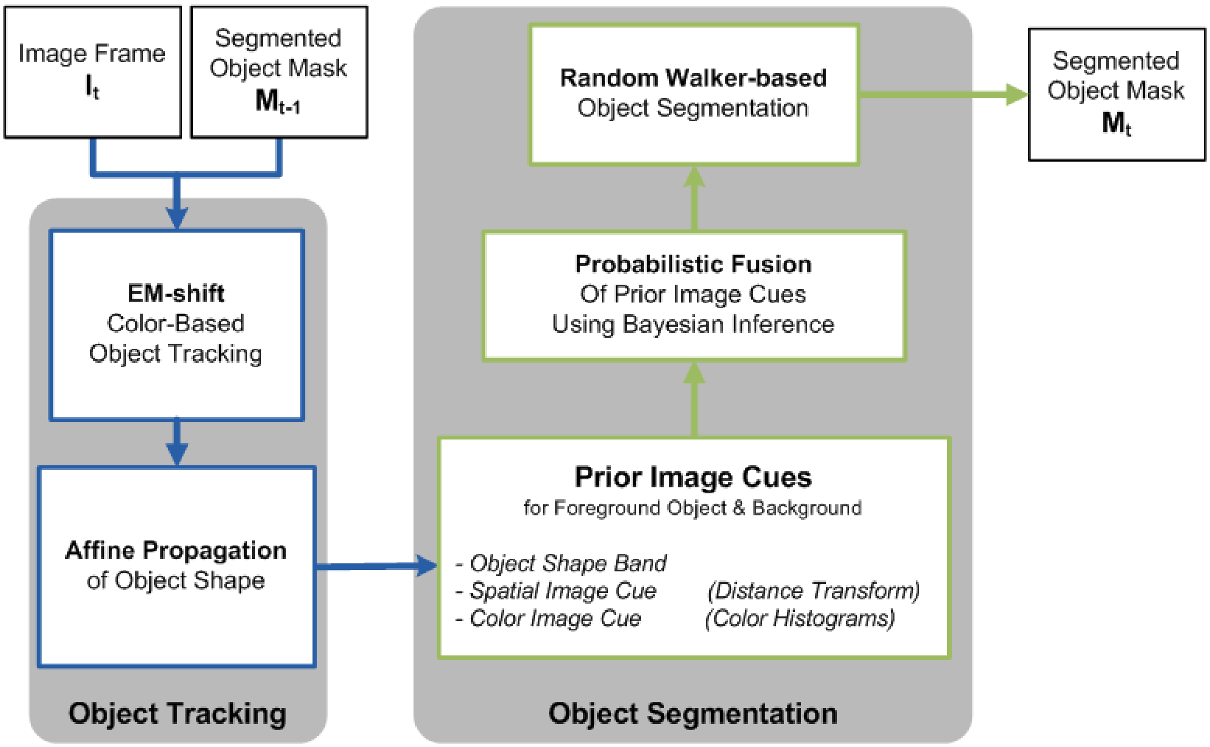


Figure 3.2: Outline of the proposed method.

ing a spatial Gaussian distribution, practically an ellipsoid region on image plane that coarsely representing the location and the shape of the object at time $t-1$. Additionally, a color-histogram-based appearance model of the segmented object (i.e., the one corresponding to the foreground of M_{t-1}) is computed using a Gaussian weighting kernel function. The iterative (EM-shift) tracking algorithm in [77] is initialized based on the computed Gaussian distribution (ellipsoid) and the object appearance model. The tracking thus performed, results in a prediction of the position and covariance of the ellipsoid representing the tracked object. Based on the transformation parameters of the ellipsoid between $t-1$ and t, a 2D spatial affine transformation of the foreground object mask M_{t-1} is performed. The propagated object mask $M_t^{'}$ indicates the predicted position and shape of the object in the new frame I_t. The Hausdorff distance [50] between the contour points of M_{t-1} and $M_t^{'}$ masks is then computed and a *shape band* [4] around the $M_t^{'}$ contour points is determined, denoted as B_t. The width of B_t is equal to the computed Hausdorff distance of the two contour point sets. This is performed to guarantee that the shape band contains the actual contour pixels of the tracked object in the new frame. Additionally, the pixel-wise Distance Transform likelihoods for the object and background areas are computed together with the pixel-wise color likelihoods based on region-specific color histograms. Pixel-wise Bayesian inference is applied to fuse spatial

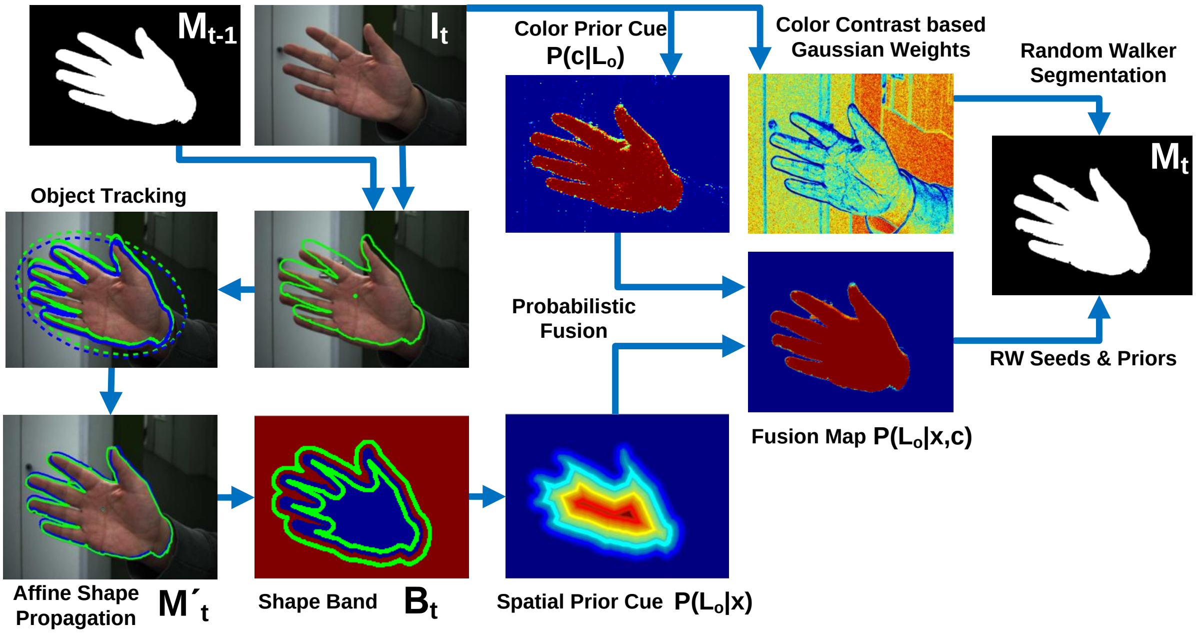


Figure 3.3: Sample intermediate results of the proposed tracking and segmentation algorithm. To avoid clutter, results related to the processing of the scene background are omitted.

and color image cues, in order to compute two probability distributions for the object and the background regions, respectively. Given the estimated pdfs for each region, a Random Walker-based segmentation algorithm is finally employed to obtain M_t in I_t.

The proposed methodology is divided in two distinct parts, one for object tracking and one for object segmentation. The following two sections of this chapter are dedicated to these parts.

3.2 Visual Object Tracking

This section presents the visual object tracking part of the proposed methodology (see the bottom-left part of Fig. 3.2). It is further divided in two subsections describing the functionality of the EM-shift color-based tracking algorithm and the affine propagation of the prior object shape.

3.2.1 EM-shift Color Based Object Tracking

The tracking method presented in [77], is closely related to the widely-used and robust mean-shift tracking method [18, 16]. More specifically, this algorithm coarsely represents the objects' shape by a 2D ellipsoid region, modeled by its center $\vec{\theta}$ that is a mean vector of spatial coordinates on the image plane and the covariance matrix V that approximates the shape of the tracked object, as can be seen in Fig. 3.4.

The spatial covariance Σ is based on the second order moment of the spatial coordinates of the pixels $\vec{x_i} \in Xo$, which are assigned to the object O of spatial mean $\vec{\theta}$ and can be computed as follows:

$$\Sigma = \sum_{x_i \in X_o} \left(\vec{x_i} - \vec{\theta}\right)\left(\vec{x_i} - \vec{\theta}\right)^T \tag{3.1}$$

Thus, a Gaussian probability density function $p\left(\overrightarrow{x}\right) = N\left(\overrightarrow{x}; \overrightarrow{\theta}, \Sigma\right)$ is utilized, as a generative model, to represent the image data samples of the tracked object area. The covariance of the Gaussian kernel is the crucial parameter towards the scale adaptation of the tracking region to the size/shape changes of the tracked object, that is presented by this algorithmic extension. A Maximum Likelihood (ML) estimation for the mean vector $\vec{\theta}$ and the covariance V is a solution toward the tracking task localizing the tracked object in a new frame based on its color appearance. A detailed derivation of the tracking algorithm is presented in Section 2.1.

The tracking task is performed based on color information only, thus the appearance model of the tracked object is represented by an M-bins color histogram of the image pixels under the 2D ellipsoid region corresponding to $\vec{\theta}$ and Σ, is computed using a Gaussian weighting kernel function.

Given M_{t-1} and I_{t-1}, $\vec{\theta}_{t-1}$, Σ_{t-1} the target appearance model o_m of the tracked object can easily be computed by Eq. (2.2), utilizing the color information of the pixels in I_{t-1} which are indicated to belong to the object area according to M_{t-1} at time $t-1$ of an image sequence. Given a new image frame I_t, where the tracked object is to be localized, the tracking algorithm evolves the initial ellipsoid region of previously computed covariance Σ_{t-1} and position $\vec{\theta}_{t-1}$ based on the Expectation-Maximization iterative procedure described in Algorithm 1, in order to determine the image area in I_t that best matches the appearance model $o_{m,t-1}$ of the tracked object in terms of a Bhattacharrya coefficient-based color similarity measure.

This gives rise to the parameters $\vec{\theta}_t$ and Σ_t that represent the predicted object position and covariance in I_t. The updated position indicates the localization of the object in the new frame, whereas the evolved covariance of the ellipsoid region indicates the scale adaptation of the tracking kernel towards the object shape/size. The latter is one of the main contributions of the utilized tracking method and is crucial regarding the efficiency of the proposed methodology, as will become more clear later in this chapter.

Figure 3.4 shows representative examples regarding the evolution of the tracking Gaussian kernel, computed by the iterative EM procedure of the Algorithm 1. Notice the adaptation of the Gaussian kernel covariance between the initial and the final estimation regarding the object shape and size changes.

Finally, Fig. 3.5 illustrates the output of the described EM-shift color-based tracking procedure described in this section, consisting of a new Gaussian kernel estimation tracking the object in the new frame I_t, represented by the position $\vec{\theta}_t$ and covariance Σ_t estimations/predictions.

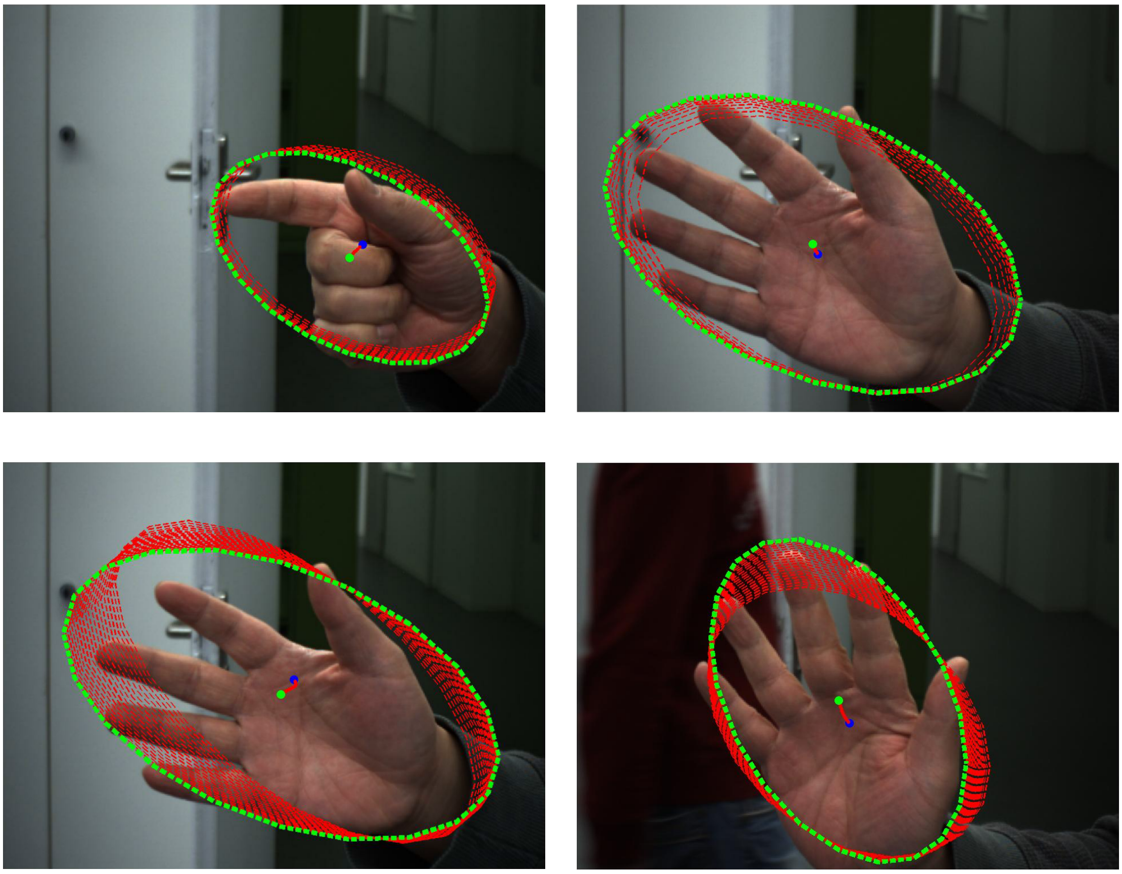


Figure 3.4: Representative examples of the Gaussian kernel evolution during the EM-shift tracking procedure. Red-dotted ellipses in each image correspond to intermediate estimations of the Gaussian kernel parameters (position and covariance), one for each EM iteration performed by the EM-shift tracking procedure. The green-dotted ellipse in each image represents the final estimation on the parameters of the Gaussian kernel, after EM convergence is achieved, giving rise to new estimated parameters regarding the location and spatial covariance of the Gaussian kernel, that is tracking the object in the new frame.

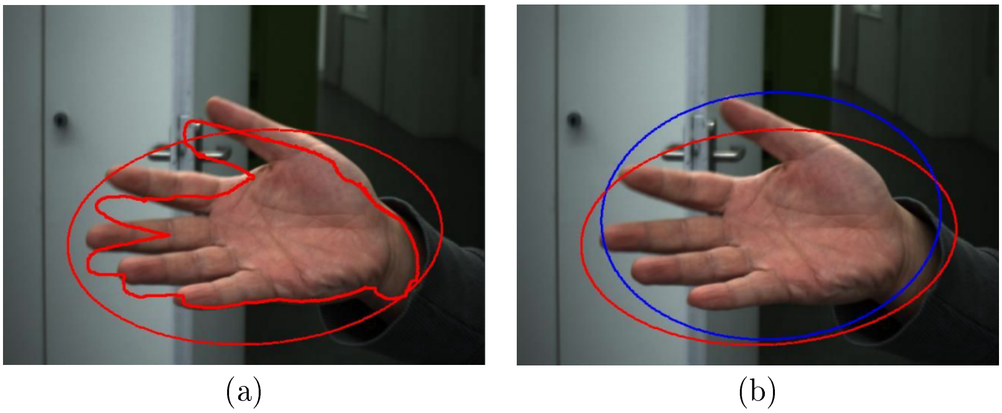

(a)
(b)

Figure 3.5: In (a) the Gaussian kernel of position $\vec{\theta}_{t-1}$ and covariance Σ_{t-1}, as well as the prior object shape mask M_{t-1} from previous I_{t-1} are illustrated, both superimposed on image frame I_t and colorized in red. Image in (b) illustrates the previous Gaussian kernel in red and the newly estimated Gaussian kernel in blue. The latter is computed by the tracking method for I_t, providing an updated position $\vec{\theta}_t$ and covariance Σ_t) of the tracked object.

3.2.2 Affine Propagation of Object Shape

The EM-shift tracking algorithm presented above assumes that the shape of an object can be accurately represented as an ellipse. In the general case, this is a quite limiting assumption. In the cases where this assumption does not hold, the objects' appearance model is forced to include background pixels, causing tracking to drift. The goal of this work is to prevent tracking drifts by integrating tracking with fine object segmentation.

To accomplish that, the contour C_{t-1} of the object mask in M_{t-1} is propagated to the current frame I_t based on the transformation suggested by the parameters $\vec{\theta_{t-1}}$, $\vec{\theta}_t$, Σ_{t-1} and Σ_t. A 2D spatial, affine transformation is defined between the corresponding ellipses. Exploiting the obtained Σ_{t-1} and Σ_t covariance matrices, a linear 2×2 affine transformation matrix A_t can be computed based on $\Sigma^{1/2}$. It is known that a covariance matrix is a square, symmetric and positive semidefinite matrix. The square root of the matrix Σ can be calculated by diagonalization as:

$$\Sigma^{1/2} = Q\Lambda^{1/2}Q^{-1}, \tag{3.2}$$

where Q is the square 2×2 matrix whose i^{th} column is the eigenvector q_i of Σ and $\Lambda^{1/2}$ is the diagonal matrix whose diagonal elements are the square values of the corresponding eigenvalues. Since Σ is a covariance matrix, the inverse of its Q matrix is equal to the transposed matrix Q^T, therefore $\Sigma^{1/2} = Q\Lambda^{1/2}Q^T$. Accordingly, we compute the transformation matrix A_t by:

$$A_t = Q_t \Lambda_t^{1/2} \Lambda_{t-1}^{-1/2} Q_{t-1}^T. \tag{3.3}$$

Finally, C_t' is derived from C_t based on the following transformation

$$C_t' = A_t(C_t - \vec{\theta}_{t-1}) + \vec{\theta}_t. \tag{3.4}$$

The result indicates a propagated contour C_t', practically a propagated object mask M_t' that serves as a prediction of the position and the shape of the tracked object in the new frame I_t, It attains temporal coherence of the implicitly tracked object contour, between consecutive object movements and appearance changes. Finally, Fig. 3.7 illustrates the procedure of object shape propagation, based on the estimated position and covariance parameters, computed by the EM-shift method closing the object tracking part of the proposed framework.

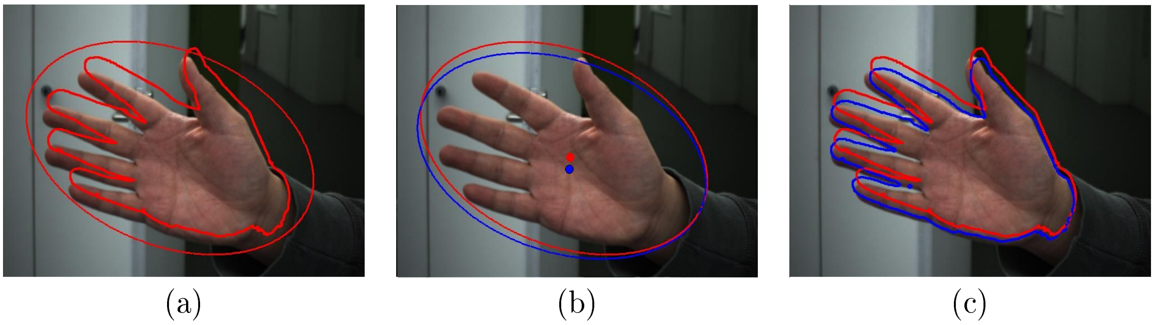


Figure 3.6: Affine propagation of the prior object shape. Image (a) illustrates the prior object contour C_{t-1} and the covariance estimation Σ_{t-1} for the previous frame I_{t-1} superimposed on the current frame I_t, in red color. In (b), the tracking task is initialized by the parameters $\vec{\theta}_{t-1}$ (red dot) and Σ_{t-1} (red ellipse) superimposed in the current frame I_t, and after convergence it results the updated parameters $\vec{\theta}_t$ (blue dot) and Σ_t (blue ellipse) representing the object's current position and shape/scale. These parameters suggest an affine transformation of the prior object contour C_{t-1} (red contour) to the C_t' (blue contour) (see Eq. (3.4)), which are illustrated in (c), approximating the current real object contour C_t.

3.3 Visual Object Segmentation

This section presents the segmentation part of the proposed methodology (see the right part of Fig. 3.2). First, the idea of creating a shape band area, based the affine-propagated

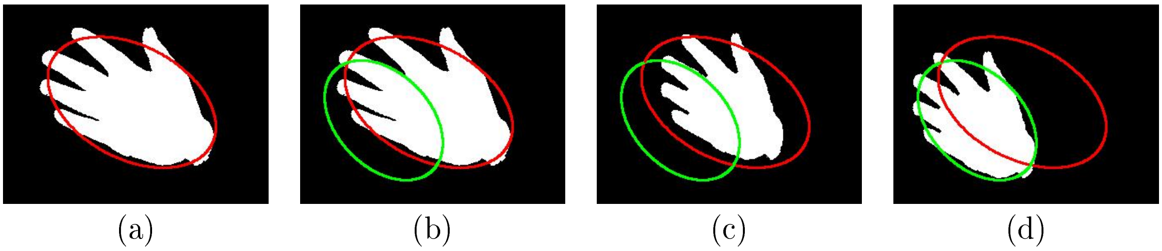

(a) (b) (c) (d)

Figure 3.7: Affine propagation of the prior object shape. Image (a) illustrates the prior object mask M_{t-1} and the covariance estimation Σ_{t-1} (ellipse in red color) for the previous frame I_{t-1}. In (b), the tracking task is initialized by the parameters $\vec{\theta}_{t-1}$ and Σ_{t-1} (red ellipse) and after convergence it results the updated parameters $\vec{\theta}_t$ and Σ_t (green ellipse) representing the object's new position and shape/scale. The affine transformation computed by Eq. (3.4) is illustrated in two parts. In (c) the linear transformation part of the Eq. (3.4) is applied to the object mask M_{t-1}, whereas in (d) the translation part of the affine transformation is applied to result the object mask M_t.

prior object shape is described. The following two subsections outline the extraction of pixel-wise spatial and color static image cues. Then, the probabilistic fusion of these cues using Bayesian Inference is presented. The probabilistic fusion results pixel-wise posterior values for the segmentation classes, which are finally utilized to guide the Random Walker-based segmentation method, resulting in the desired fine foreground object mask, as will be described in the last section.

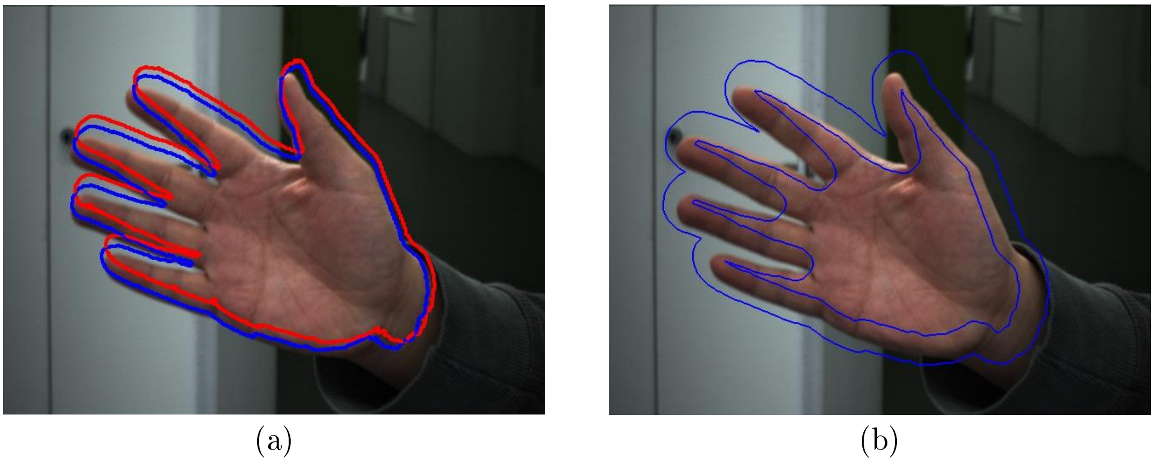
3.3.1 Object Shape Band

The propagated object contour $C_t^{'}$ approximates the actual but unknown object boundaries, noted as contour C_t, in the current frame I_t. Thus, a direct segmentation based on the $C_t^{'}$ will not provide an accurate mask of the tracked object. However, it is assumed that the actual object boundaries can be precisely localized around the predicted object contour $C_t^{'}$. To this end, the object shape band B_t is determined. Our notion of shape band is similar to the ones used in [4, 75]. B_t can be regarded as an area of uncertainty, where the true object contour may be detected in image I_t. An illustration of the shape-band area is provided in Fig. 3.8. The width of B_t is determined by the Euclidean 2D Hausdorff distance [50] between the contours C_{t-1} and $C_t^{'}$, that is given by:

$$d_H(C_t', C_{t-1}) = max\{sup_{x \in C_t'} inf_{y \in C_{t-1}} d(x,y), sup_{y \in C_{t-1}} inf_{x \in C_t'} d(x,y)\} \qquad (3.5)$$

where sup represents the *supermum* and inf represents the *infimum*. Given a subset S of a partially ordered set T, the supremum (sup) of S, if it exists, is the least element of T that is greater than or equal to each element of S. The infimum of a subset S of some set T is the greatest element (not necessarily in the subset) that is less than or equal to all elements of the subset S. Thus, the Hausdorff distance calculates the greatest of all the minimum distances of each point of each of the point sets to each point of the other set.

The width of B_t is limited by the spatial properties of C_t', in order to retain a non-compact annotated area. In other words, that the shape band area should never cover the entire inner object area. The automatically identified area can be seen as a symmetrically dilated object contour, defining an area of uncertainty, a search area in other words, where the object boundaries may be localized in the new frame I_t. The usability of the object shape band will become more clear in the following subsection.


(a)

(b)

Figure 3.8: (a) Given the previous object contour C_{t-1} (red outline) and the propagated object contour C_t' (blue outline), the Hausdorff distance between them determine the width of the shape band area. (b) The shape band is created symmetrically around the C_t' in the right image, defining an local area of uncertainty, where the true object boundaries may be detected.

3.3.2 Spatial Prior Image Cue

The first of the image cues that is computed to discriminate between the foreground object and the background classes refers to the pixel-wise spatial cue based on the known propagated object contour C_t'. The Euclidean 2D Distance Transform is used to compute the probability of a pixel $\vec{x_i}$ in image I_t to belong to either the object L_o or the background L_b region/class, based on its 2D location $\vec{x_i} = (x, y)$ on the image plane. As a first step, the shape band B_t of the propagated object contour C'_t is considered and its inner and outer contours are extracted. The Distance Transform is then computed starting from the outer contour of B_t towards the inner part of the object. The probability $P(L_o|x_i)$ of a pixel to belong to the object given its image location is set proportional to its normalized distance from the outer contour of the shape band. For pixels that lie outside the outer contour of B_t, it holds that $P(L_o|x_i) = \epsilon$, where ϵ is a small constant.

Similarly for the background, the Euclidean Distance Transform measure starting from the inner contour of B_t towards the exterior part of the object is computed. The probability $P(L_b|x_i)$ of a pixel to belong to the background given its image location is set proportional to its normalized distance from the inner contour of the shape band. For pixels that lie inside the inner contour of B_t, it holds that $P(L_b|x_i) = \epsilon$. Both probability maps are illustrated in Fig. 3.9.

3.3.3 Color Prior Image Cue

The second of the image cues that is computed to discriminate between the foreground object and the background classes is color, represented by a histogram that is updated after each segmentation step.

Based on the segmentation mask M_{t-1} of the image frame I_{t-1} that is available from the previous segmentation step, a partition of image pixels Ω into sets Ω_o and Ω_b is defined, indicating the object and background image pixels, respectively. The appearance model of the tracked object is represented by a color histogram defined as H_o computed on the Ω_o set of pixels. The normalized value in a histogram bin c encodes the conditional probability $P(c|L_o)$. Similarly, the appearance model of the background region is represented by the color histogram H_b, computed over pixels in Ω_b and encoding the conditional probability $P(c|L_b)$. Fig. 3.10 illustrates some examples of foreground color probability maps, representing the pixel-wise values for $P(c|L_o)$.

3.3.4 Probabilistic Fusion of Prior Image Cues

Image segmentation can be considered as a pixel-wise classification problem for a number of classes/labels. Neither the color, nor the spatial image cue individually provide an accurate representation of the desired foreground object class. Our goal is to efficiently combine the computed pixel-wise spatial and color probabilities to generate the posterior probability distribution for each of the classes L_o and L_b, which will be further utilized to guide the Random Walker-based image segmentation.

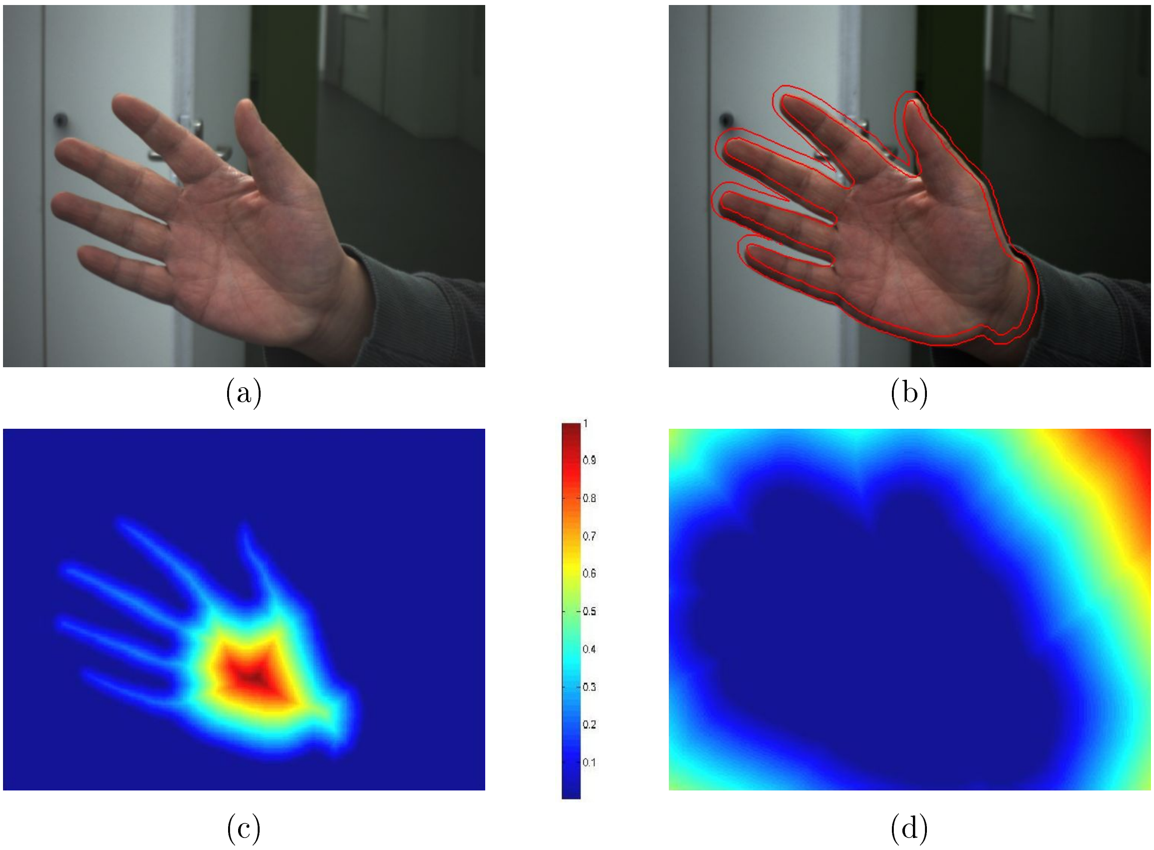


Figure 3.9: The Distance Transform based spatial cue computed for both foreground object and background classes. (a) The input image frame I_t and (b) the shape band computed. (c) The map indicating the probability of each pixel to belong to the foreground object $P(L_o|x_i)$, based on the normalized Distance Transform metric which starts from the outer contour of the shape band to the inner area of the object. (d) The map indicating the probability of each pixel to belong to the background $P(L_b|x_i)$ based on the normalized Distance Transform metric which starts from the inner contour of the shape band to the outer of the object.

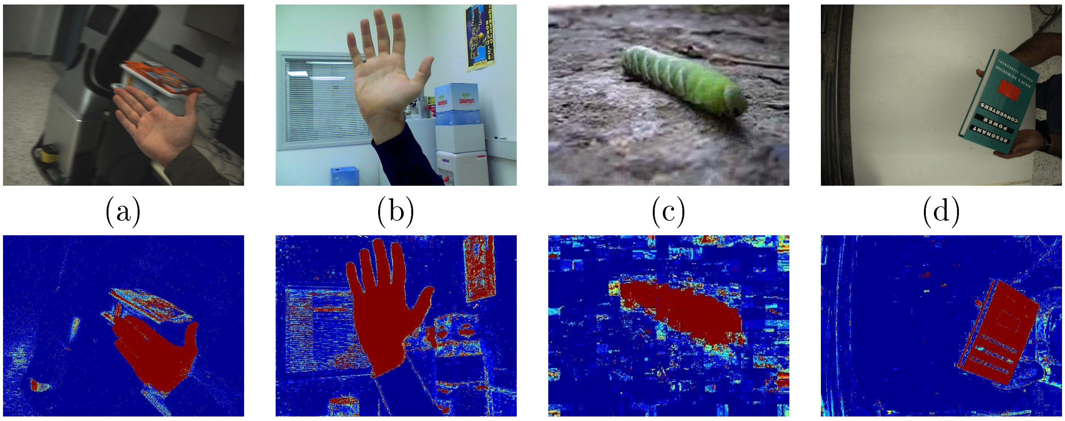

(a) (b) (c) (d)

Figure 3.10: Examples of input images and obtained foreground maps, modeling the pixel-wise probability $P(c|L_o)$ of the pixel's color to belong to the foreground object based regarding the updated foreground color histogram H_o. Notice the ambiguities that arise based on the color cue based on the natural foreground/background boundaries of the real images. (a) & (b) A human hand in action in a cluttered background. (c) A green caterpillar in an image of low resolution. (d) A green book of complex texture in a homogeneous background.

Using Bayesian inference, we formulate a probabilistic framework to fuse the available prior image cues, based on the pixel color and position information, as described earlier. Considering the pixel color c as the evidence and conditioning on pixel position x_i in image frame I_t, the posterior probability distribution for class L_l is given by

$$P(L_l \mid c, x_i) = \frac{P(c \mid L_l, x_i) P(L_l \mid x_i)}{\sum_{l=0}^{N} P(c \mid L_l, x_i) P(L_l \mid x_i)}, \tag{3.6}$$

where $N = 2$ in our case. The probability distribution $P(c \mid L_l, x_i)$ encodes the conditional probability of color c taking the pixel class L_l as the evidence and conditioning on its location x_i. We assume that knowing the pixel position x_i, does not affect our belief about its color c. Thus, the probability of color c is only conditioned on the prior knowledge of its class L_l following that $P(c \mid L_l, x_i) = P(c \mid L_l)$. Given this, Eq. (3.6) transforms to

$$P(L_l \mid c, x_i) = \frac{P(c \mid L_l) P(L_l \mid x_i)}{\sum_{l=0}^{N} P(c \mid L_l) P(L_l \mid x_i)}. \tag{3.7}$$

The conditional color probability $P(c \mid L_l)$ for the class L_l is obtained by the color histogram H_l, as described in Section 3.3.3. The conditional spatial probability $P(L_l \mid x_i)$ is obtained by the Distance-Transform measure calculation, as described in Section 3.3.2. Figure 3.11 illustrates the fusion procedure of the proposed probabilistic framework and highlights the advantages of combining two static class-specific image cues. Notice the results in Figs. 3.11(e) and 3.11(f). In the hand sequence, the background patches with similar color with the hand presented in (a) have been suppressed in (e) under the influence of the certainty provided by the foreground spatial cue in the corresponding image areas. In the book sequence the patches within the book area of similar color with the background illustrated in (b) have been filled in (f) for the same reason.

There are many cases in object tracking where the foreground object consists of colors that also appear to be dominant in the background, see for example Fig. 3.10(a,b). In such cases, the probabilistic fusion of the color cue with the spatial cue may not avert the probability $P(L_o \mid c, x_i)$ to be higher than the corresponding $P(L_b \mid c, x_i)$ for the foreground class for a pixel x_i. Such information will lead the following algorithmic step of the Random Walker-based segmentation to produce background seeds or priors to the foreground object region and vice versa. Thus, to prevent such behavior, the prior object shape information is exploited. Given the computed shape band of the propagated object contour C_t' for the current frame I_t, the image region of the interior of the inner contour of the shape band is assumed to belong to the foreground object. This conservative assumption enables us to discard posterior probabilities $P(L_l \mid c, x_i)$ of pixels within that region for which it holds that $P(L_o \mid c, x_i) \leq P(L_b \mid c, x_i)$ setting both to 0.5. Moreover, the posterior probabilities of pixels within the region of the image that is outside of the outer contour of the shape band for which it holds that $P(L_o \mid c, x_i) \geq P(L_b \mid c, x_i)$ are discarded and set equal to 0.5. By setting the posterior probabilities of pixels that exhibit such behavior to 0.5, we let their labeling to be “decided” by their neighboring pixels. This simple technique will prevent inexistent foreground object to be generated by the segmentation procedure in the background and background regions to be created within the true foreground object area, because of similar color appearance. To conclude, a more elaborate technique would be more effective in case of more complex appearance of the foreground object, where a large region within it will contain similar colors with those of the background.

3.3.5 Random Walker Based Object Segmentation

The resulting posterior distribution $P(L_l \mid c, x_i)$ for each of the two labels L_o and L_b (segmentation classes) on pixels x_i guides the Random Walker-based image segmentation towards an explicit and fine segmentation of the tracked object in I_t.

In order to represent the image structure by random walker biases, we map the edge weights to positive weighting scores computed by the Gaussian weighting function on the normalized Euclidean distance of the color intensities between two adjacent pixels, thus

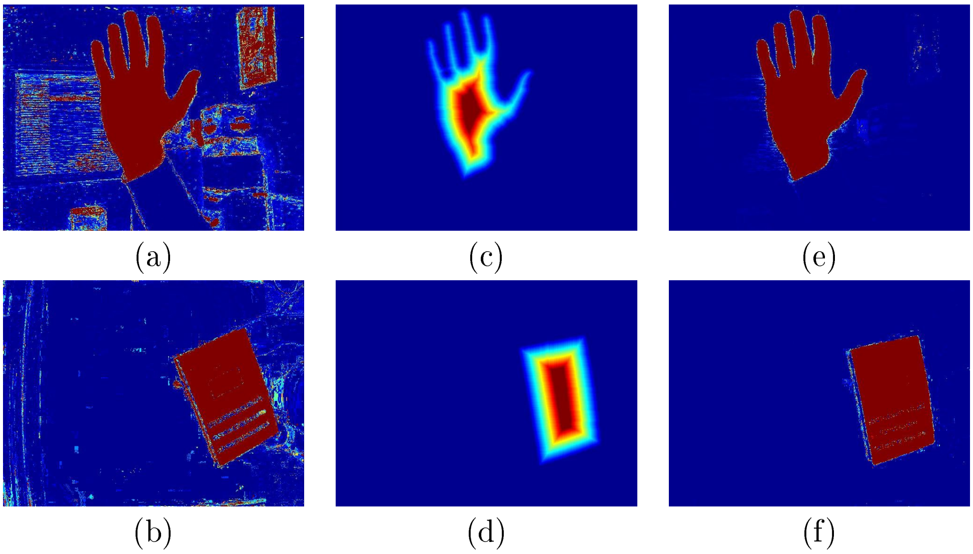


Figure 3.11: Images in (a),(b) represent the pixel-wise color probabilities $P(c|L_o)$. Images in (c),(d) refer to the pixel-wise spatial probabilities $P(L_o|x_i)$. Finally, performing the probabilistic fusion of the two image cues, images in (e),(f) represent the fusion maps, which efficiently indicates the foreground object after the posterior probabilities computed by Eq. (3.7).

the image brightness. The Gaussian weighting function is

$$w_{i,j} = e^{-\frac{\beta}{\rho}(\|c_i - c_j\|)^2} + \epsilon, \tag{3.8}$$

where c_i stands for the vector containing the color channel values of pixel/node i, ϵ is a small constant (i.e $\epsilon = 10^{-6}$) and ρ is a normalizing scalar $\rho = \max(\|c_i - c_j\|), \forall i, j \in E$.

The parameter β is user-defined and modulates the spatial random walker biases, in terms of image color contrast (brightness). Figure 3.12 illustrates the Gaussian weighting function in Eq. (3.8) for different values of β. Moreover, Fig. 3.13 illustrates the graph Laplacian matrix, as a color contrast map. For different values of β in the weighting function of Eq. (3.8), the Laplacian matrix of size equal to the number of image pixels per dimension is populated. For a fixed β, we compute the column-wise summation and obtain a measure of brightness for each image pixel based on the edge weights of all its direct connected neighbors. For example, a pixel in a 4-connected graph, which models the image, with very low color contrast against all its direct connected pixels will obtain a mean value of 1 in the visualized color-contrast map, indicating that it is very likely for a random walker to cross any of its edges (weak bias).

Lower values of β, implies that even a high value of color residual, which indicates high color contrast (Euclidean distance of color vectors), thus strong edge, will be weighted with high probability of being crossed by a random walker, meaning relaxed random walker biases. See for example the function graph for $\beta = 10$ and the corresponding image from Fig. 3.13. Red color for an image pixel stands for high probability (near to 1) of a random walker to cross each of its adjacent edges. Blue color stands for low probability of a random walker to cross each of the adjacent edges of that pixel. Higher values of the β parameter implies for a strict scheme of weighting. Only edges that connect pixels of very low contrast will be weighted with a high probability to be crossed by a random walker (see Fig. 3.13(b-c)).

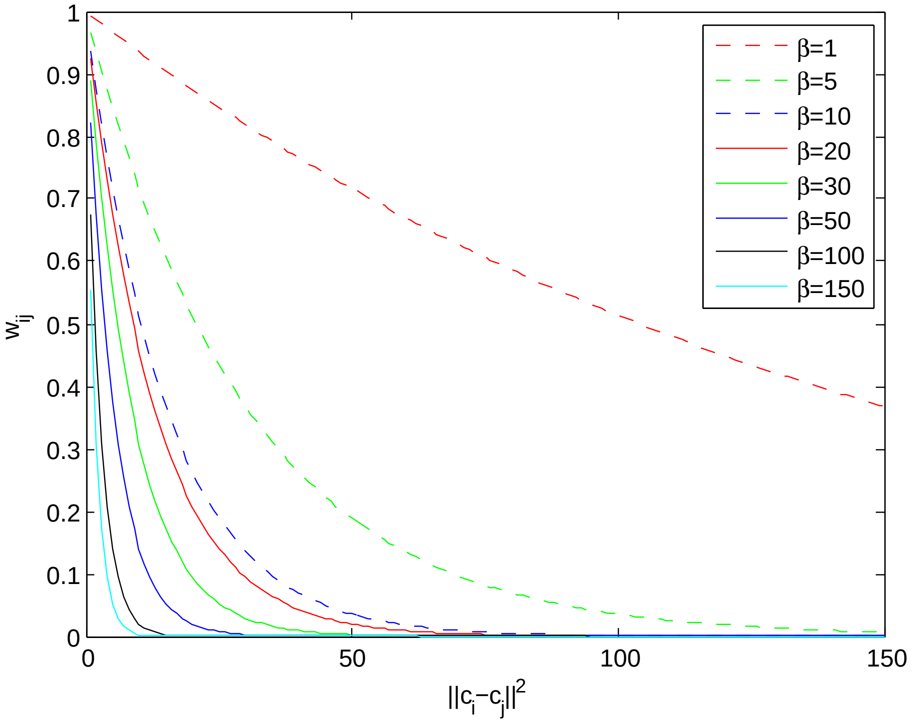
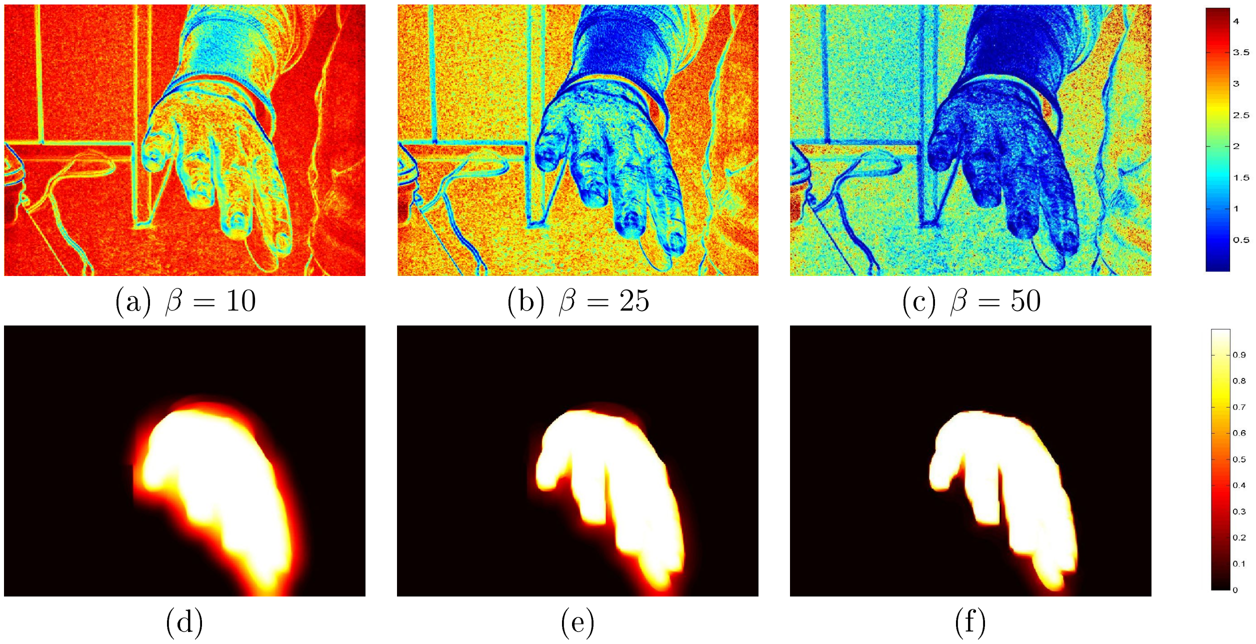


Figure 3.12: Illustration of the Gaussian weighting function, defined by Eq. (3.8) and utilized to measure color contrast between neighboring pixels and populate the combinatorial Laplacian graph of the image using Eq. (2.18). Multiple graphs of the function are superimposed in the figure for different values of β parameter.

Many good sources exist on the solution to large, sparse, symmetric, linear systems of equations [28]. A direct method, such as LU decomposition with partial pivoting is a fair option, although it is impractical for large systems because of high memory and computational requirements. The standard alternative to the class of direct solvers for large, sparse systems is the class of iterative solvers [32]. These solvers have the advantages of a small memory requirement and the ability to represent the matrix-vector

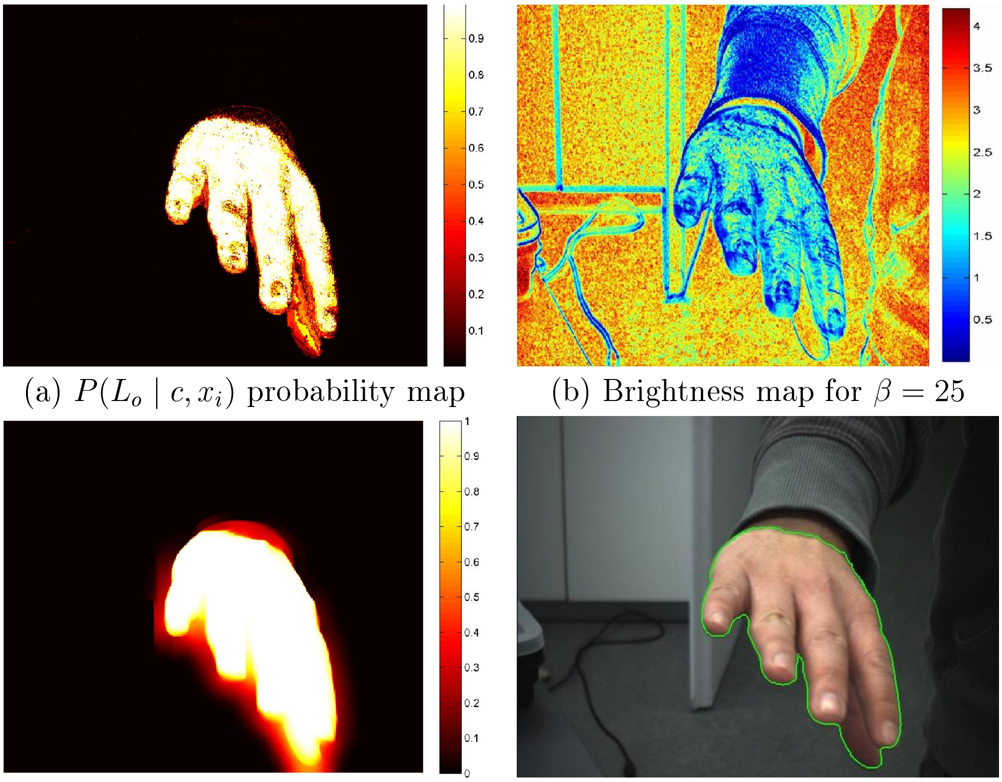


(a) $\beta = 10$ (b) $\beta = 25$ (c) $\beta = 50$

(d) (e) (f)

Figure 3.13: Visualization of Laplacian matrix for different values of β parameter. In (a) a low value of β equal to 10 is used in Eq. (3.8) to populate the Laplacian matrix. The resulting color contrast map (image brightness) indicates weak random walker biases even for natural strong edges in the image. In (b) β is set equal to 25. Random walker biases are now stronger and only edges of low color contrast are valued with higher Gaussian weighting values. In (c), β is set to 50. The random walker biases are now too strong, thus only edge with very low color contrast are valued with higher weights. Images in (d-f) illustrates the resulting probabilities for the foreground object label corresponding to the (a-c) Laplacian matrices.

multiplication as a function. Solving the linear system of equations, the obtained posterior probability distribution $P(L_l \mid c, x_i)$ computed over the pixels x_i of the current image I_t suggest the probability of the pixels to be assigned to the label L_l. Therefore, we consider the pixels of highest posterior probability values for the label L_l as pre-labeled/seeds nodes of that label in the formulated graph (see Fig. 3.14 for an intuitive example).



(a) $P(L_o \mid c, x_i)$ probability map

(b) Brightness map for $\beta = 25$

(c)x_U^s: Foreground probabilities

(d) Segmentation outline

Figure 3.14: (a) Fusion probability map is illustrated. Pixels of probability higher than 0.9 are selected to act as seeds in the Random Walker-based segmentation formulation. (b) The color contrast map computed by the Laplacian matrix of the graph with $\beta = 25$. (c) The soft segmentation result for the object label, x_U^o, obtained by solving the system in Eq. (2.22) and illustrated as a probability image/map. (d) The segmented foreground object indicating the outline of the binary mask which is computed by considering the pixels of highest posterior probability values between the labels L_o and L_b.

To further comprehend the influence of β regarding the resulting probabilities x_U^o, we compute the solution of Eq. (2.22) of Random Walker formulation for each of the Laplacian matrices visualized in (a-c) of Fig. 3.13. The resulting real-valued probability maps in (d-f) for the foreground object label gradually vary regarding the accuracy of the soft segmentation they provide. For the lowest value of β, the (a) contrast map is obtained, where the Random Walker biases are more relaxed, meaning that the natural image edges are easier to be crossed by a Random Walker. The resulting probabilities illustrated in (d) indicate high uncertainty regarding the object boundaries. For higher values of β, as in (b) and (c), more accurate probability maps are obtained regarding the object boundaries and the quality of the object segmentation. However, using a higher value of β, a high number of seed points within the foreground object area should be feasible in order to get an accurate real-valued, therefore an accurate binary segmentation.

An alternative formulation of the Random Walker-based image segmentation method is presented in [29]. This method incorporates non-parametric probability models, that is prior beliefs on label assignments. In [29], the sparse linear systems of equations that is to be solved to obtain a real-valued density-based multi-label image segmentation are also presented. The two modalities of this alternative formulation suggest for using only prior knowledge on the belief of a graph node toward each of the potential labels, or using prior knowledge in conjunction with pre-labeled/seed graph nodes, also presented in Section 2.2 of Chapter 2.

The prior probabilities $P(L_l \mid c, x_i)$ obtained using the probabilistic framework for the fusion of color and spatial image cues for both labels are illustrated as an image in Fig. (3.14)(a). In case of using only seeds solving the Eq. (2.22), user-defined thresholding was applied on these probabilities in order to get the most probable points belonging to the foreground object label (i.e up to 0.9) the whole information provided by these probabilities will be utilized with no thresholding, denoted as λ^s, in order to solve the following modified linear system of Eq. (2.24).

Regarding the second modified formulation, the seeds may also be incorporated in conjunction with the prior values. The modified system of equation to be solved is provided in Eq. (2.25).

The γ scalar weighting parameter is introduced in these formulations, controlling the degree of authority of the prior belief values towards the belief information obtained by the random walks per potential segmentation label. This extended formulation of using both seeds and prior beliefs on graph nodes is compatible with our approach considering the obtained posterior probability distributions $P(L_l \mid c, x_i)$ for the two segmentation labels. Considering the formulation utilizing both seeds and prior information, a low value of the γ parameter will set the system to behave like the seeds only formulation, whereas a high value of it will bias the results towards the input information, meaning that random walks and the contrast map will have minor contribution to the final real-valued segmentation result.

Regardless of the utilized formulation, the primary output of the algorithm consists of K probability maps, that is a soft image segmentation per label. By assigning each pixel to the label for which the greatest probability is calculated, a K-way segmentation is obtained. This process gives rise to object mask M_t for image frame I_t.

Chapter 4

Results

Experimental results, implementation issues and discussion on the effectiveness of the proposed methodology are presented in this chapter. The proposed method was extensively tested to simultaneously track and segment an object of interest on a variety of image sequences under challenging conditions. A description of each of these sequences is provided in the first section of this chapter. The second section is dedicated to some implementation issues regarding the proposed method. The third section is dedicated to a qualitative assessment of the proposed method, regarding the tracking and the segmentation results individually. Finally, a quantitative assessment is provided in the last section.

4.1 Test Image Sequences

A variety of test image sequences is chosen, illustrating a single object to validate the performance and the efficiency of the proposed joint tracking and segmentation framework. The represented objects in these image sequences go through persistent and extensive changes regarding their appearance, shape and pose. Additionally, these sequences differ with respect to the camera motion and the surrounding illumination changes, affecting the appearance of the tracked objects. Figure 4.1 provides a single frame for each of the 15 test image sequences.

In the first three sequences, illustrated in Fig. 4.1(a-c), a human hand undergoes complex articulations in a simple static background. The varying illumination conditions significantly affect its skin color tone, thus the object also undergoes noticeable appearance changes throughout each image sequence consisting of $340, 420$ and 630 frames respectively, of size 640×480 pixels each.

The sequence represented by the frame in Fig. 4.1(d) also contains a human hand acting in a static but rather complex background. Moreover, the surrounding illumination conditions vary over time. The sequence consists of 100 frames of resolution 640 × 480 pixels.

In each of the sequences shown in Fig. 4.1(e-f), a textured book is illustrated undergoing significant changes regarding its pose and shape, whereas light reflections on its glossy surface significantly affect its appearance over time. The image sequence of (e) consists of 420 frames and that of (f) consists of 360 frames. The size of each frame is 640 × 480 pixels.

The image sequences represented in Fig. 4.1(g-i) illustrate human faces. The goal here is to track the face skin color despite of the non-uniform colored face area. The human head in (g) undergoes abrupt scale changes and significant variations of the lighting conditions in a static background. The length of this sequence is 470 frames of resolution equal to 320 × 240 pixels. The video of the human head, illustrated in (h), is of lower quality and resolution and presents the same challenges, as the previous ones, but in a changing background. Its length is 380 frames, each one of size 174 × 144 pixels. The sequence illustrated in (i) goes through extended pose variations in front of a static but rather complex background. This image sequence consists of 400 frames of size 640 × 480 pixels.

Following, the image sequences of Fig. 4.1(j-l) also represent human hands. In the challenging sequences of (j) and (k), the articulations of a human hand are observed by a moving camera in the context of a continuously varying cluttered background. Moreover, the illumination conditions undergo extensive variations throughout the video. The image sequence represented in (l) include a static complex background. All these sequences consist of 550 frames each. The size of each frame is 640 × 480 pixels.

The last three image sequences shown in Fig. 4.1(m-o) are of low quality. The sequence in (m) is captured by a moving camera, illustrating the body deformations of a moving green caterpillar. The number of frames is 280 of size 320 × 240 pixels. The sequence in (n) illustrates a polar bear moving in a low-contrast background. The sequence consists of 160 frames of 320 × 240 pixels each. Finally, the low resolution sequence depicted in (o) has been acquired by a medical endoscope. A target white colored object is moving within a low-contrast background performing large position displacements between consecutive frames. It consists of 15 frames of size 256 × 256 pixels.

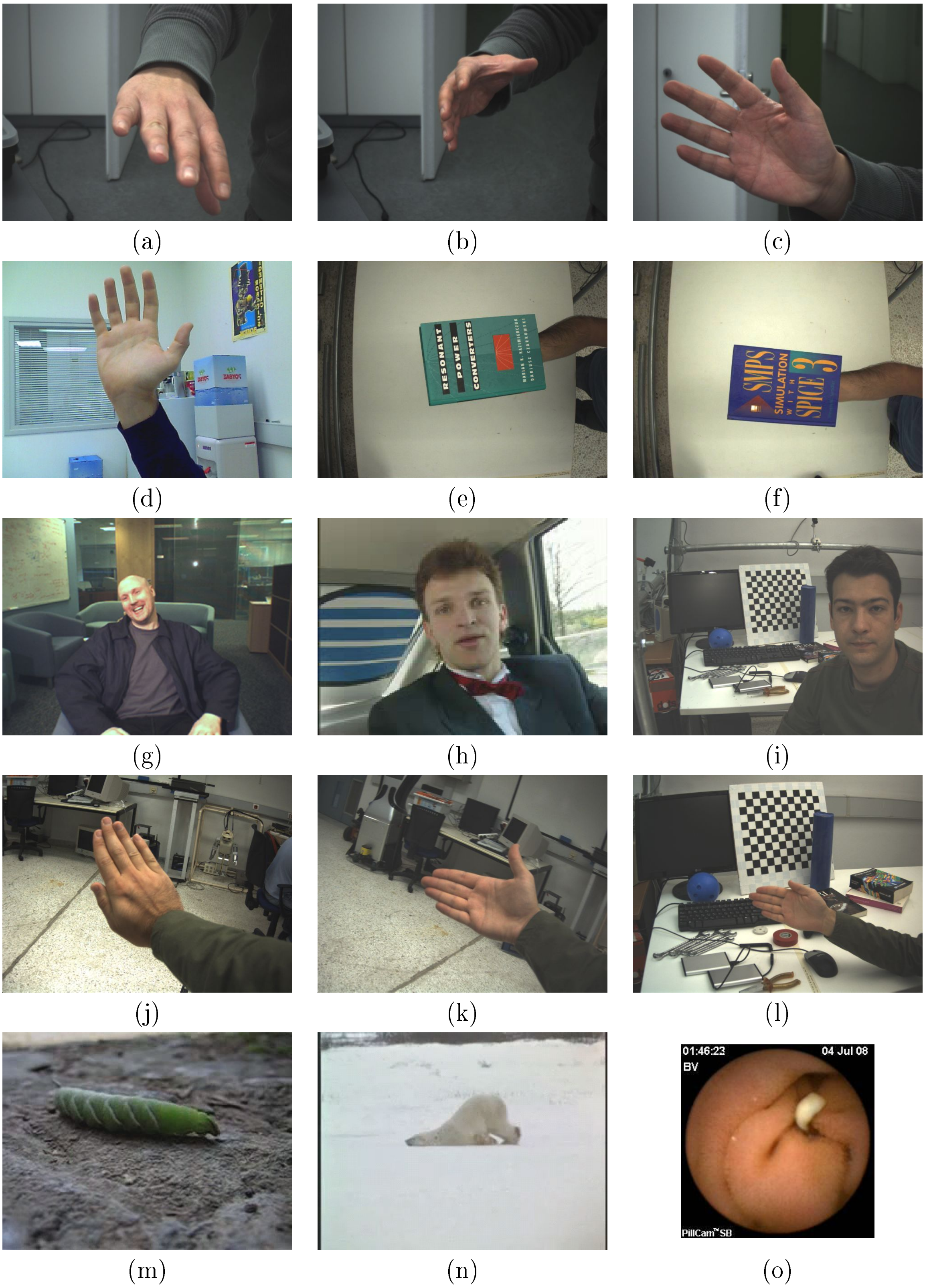

(a) (b) (c) (d) (e) (f) (g) (h) (i) (j) (k) (l) (m) (n) (o)

Figure 4.1: Single frames representing the 15 test image sequences used to validate the performance of the proposed joint tracking and segmentation framework.

4.2 Implementation Issues

The foreground object and background appearance models used to capture the appearance of the two regions consist of two three-dimensional histograms of 32 bins per dimension, based on RGB colorspace.

The original parameter configuration of the EM-shift color tracking algorithm is preserved, as described in [77], throughout the whole experimental evaluation carried out for the current work. The appearance model of the tracked object utilized in the EM-shift color tracking algorithm is based on the RGB color space. It is a three-dimensional color histogram of 8 bins per dimension. The convergence criterion of the EM procedure of the tracking method is a combination of maximum number of iterations, set to 20, and a stopping threshold value that refers to the number of new pixels added to the ellipsoid tracking region between consecutive iterations with respect to the image size. The threshold is set to 5% of the total number of pixels. The mean value of EM iterations throughout the test image sequences was 10. The crucial parameter β_{track} of the tracking method, in order to work properly for a Gaussian kernel is set to 1.2, as described in [77].

The Random Walker segmentation method involves three variant formulations to obtain the probabilities of each pixel to belong to each of the segmentation labels, as described in Section 3.3. The three formulations refer to the usage of seeds (pre-labeled graph nodes), prior values (probabilities/beliefs on label assignments for some graph nodes), or a combination of them. The edge weights of the graph are computed by Eq. (3.8), where the parameter β controls the scale of the color contrast (brightness) between adjacent graph nodes (pixels). The pixel-wise posterior values are computed using Bayesian Inference as described in Section 3.3 and are exploited to guide the segmentation. Each pixel x_i with posterior value $P(L_l \mid x_i)$ greater or equal to 0.9 is considered as a seed pixel for the label L_l. Any other pixel with posterior value $P(L_l \mid x_i)$ less than 0.9 is considered as a prior value for label L_l. In the case of prior values, the γ parameter is introduced to adjust the degree of authority of the prior beliefs towards the definite label-assignments expressed by the seed pixels of the image. In the experiments carried out toward the qualitative assessment of the proposed method presented in the following section, the β parameter was selected within the interval of $[10-50]$, whereas the γ ranges within $[0.05 - 0.5]$.

The reported experiments were generated based on a Matlab implementation, running on a PC equipped with an Intel i7 CPU and 4 GB of RAM memory. The Random Walker-based image segmentation method developed is based on the Graph Analysis

Toolbox of Matlab [30] and is available online [1]. The EM-shift tracking method performs in real-time on a conventional PC of 1Gz. The computational bottleneck of the proposed method is the solution of the large system of the sparse linear equations of the Random Walker formulation regarding the image segmentation part of the proposed method. The runtime performance of the current unoptimized Matlab implementation varies between 4 to 6 seconds per frame for 640 × 480 images on a PC with the aforementioned setup. However, a near real-time runtime performance is feasible by optimizing both the EM-shift part of the tracking method and the solution of the large sparse linear system of equations of the Random Walker-based image segmentation method.

4.3 Qualitative Assessment

A two-phase qualitative assessment has been carried out in order to validate the individual tracking and segmentation performance of the proposed method. First, we compare the proposed joint tracking and segmentation method with the stand alone EM-shift color tracking method, that is originally presented in [77] and utilized in our proposed framework, in order to present the effectiveness and the key role of the fine segmentation part of the method towards a more robust and drift-free tracking performance. In the following, a qualitative assessment of the proposed method with the state-of-art skin color detection and tracking algorithm, that is presented in [2], is carried out. The tracking and the detection results obtained by the skin color tracker are qualitatively compared with the corresponding result of the proposed method, in two test image sequences, representing human hands in action. Video containing the qualitative results are available online[2].

Proposed Method Vs. Stand-alone EM-shift Object Tracking

First, we compare the proposed joint tracking and segmentation method with the stand-alone EM-shift color tracking method presented in [77] and utilized in our proposed framework. The parameters of this algorithm were kept identical in the stand-alone run and in the run within the proposed framework. It is important to note that the stand-alone tracking method is initialized with the appearance model extracted in the first frame of the sequence. Moreover, its appearance model is not updated over time, because in the challenging sequences we used as the basis of our experimental evaluation,

[1] http://cns.bu.edu/∼lgrady/software.html

[2] http://www.ics.forth.gr/∼argyros/research/trackingsegmentation.html

updating the appearance model based on the results of tracking, soon causes tracking drifts and total loss of the tracked object.

Figure 4.2 illustrates representative snapshots of the tracking results (i.e., five frames for each of the eight sequences). Each frame shown in Fig. 4.2 is annotated with the results of the proposed algorithm and the results of the stand-alone tracking method.

Figure 4.3 illustrates representative snapshots of the tracking results on the rest of the test image sequences. Each frame shown in Fig. 4.3 is annotated with the tracking only results of the proposed algorithm and the results of the stand-alone tracking method.

Finally, Fig. 4.4 illustrates the segmentation results computed by the proposed method on the same test image sequences of Fig. 4.3.

The performance of the stand-alone EM-shift tracking method drifts out in cases where the appearance and shape of the object undergoes extensive changes, whereas the proposed method provides stable tracking and adaptation of the tracking kernel size to the shape changes exploiting the information provided by the incorporated segmentation procedure.

Proposed Method Vs. State-of-art Skin-Color Detection & Tracking

The efficient skin color detection and tracking method presented in [2] provides near-optimal results for the image sequences presenting human hands and/or head in action. In brief, the skin color tracking method adopts a non-parametric model of skin color. Skin-colored objects are detected with a Bayesian classifier that is bootstrapped with a small set of training data. By using on-line adaptation of skin-color probabilities the classifier is able to cope with considerable illumination changes. Moreover, the tracking over time is achieved by a novel technique that can handle multiple objects simultaneously, which may move in complex trajectories, occlude each other in the field of view of a possibly moving camera and vary in number over time.

We compare the tracking results of the proposed method to those of the skin color detection and tracking method in two image sequences to obtain a qualitative evaluation on the tracking performance. The object representation is based on a tracking ellipse in both methods. The first test image sequence illustrates a human hand performing complex articulations in a simple static background. It is the one represented by the single frame Fig. 4.1(a). Figure 4.5 presents results on selected frames of that sequence. In the left column, the tracking ellipse computed by the proposed method is superimposed to each frame. The tracking ellipses computed by the skin color tracking method are superimposed to each frame in the right column.

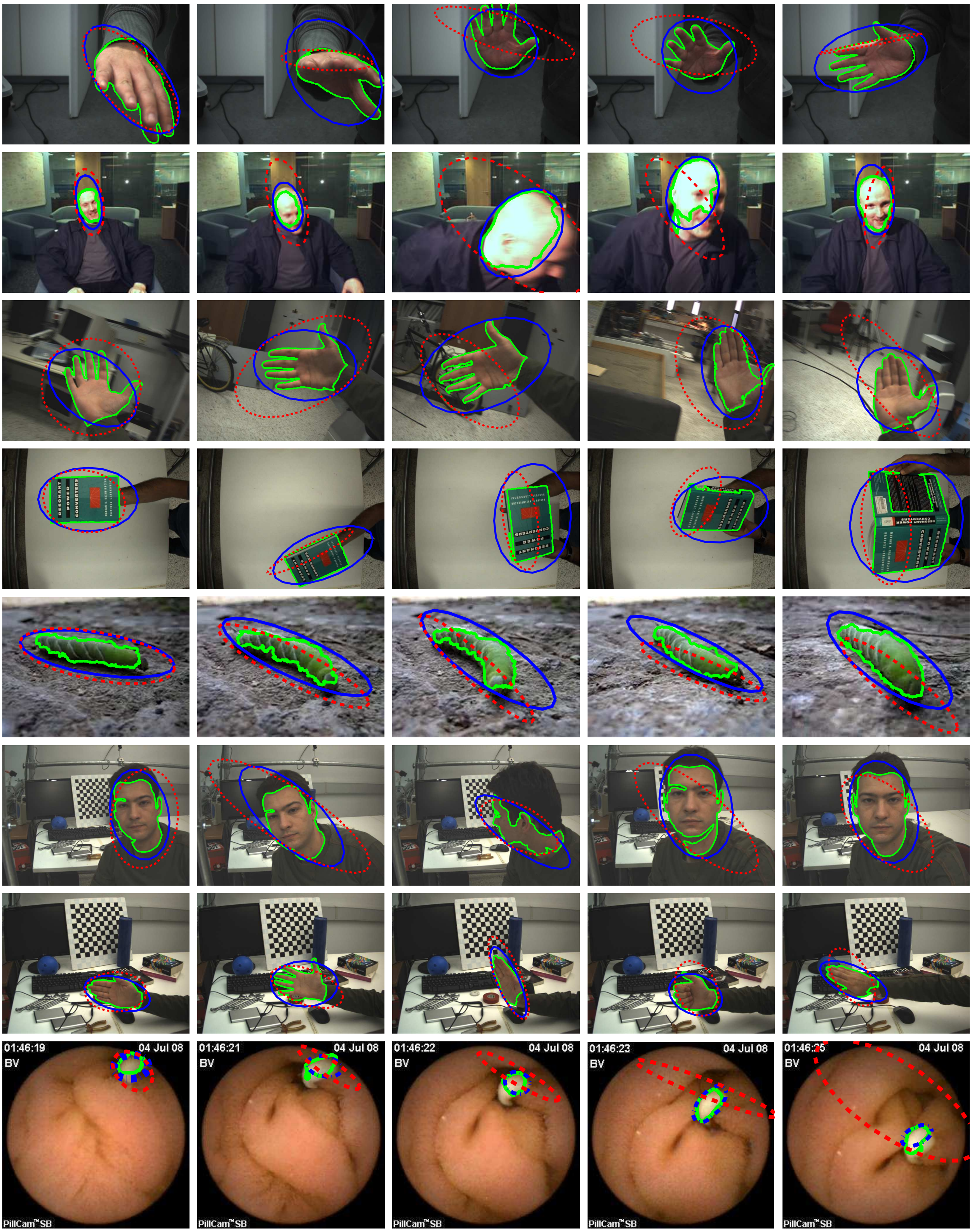


Figure 4.2: Experimental results and qualitative comparison between the proposed framework providing tracking and segmentation results (blue solid ellipse and green solid object contour, respectively) and the tracking algorithm of [77] (red dotted ellipse).

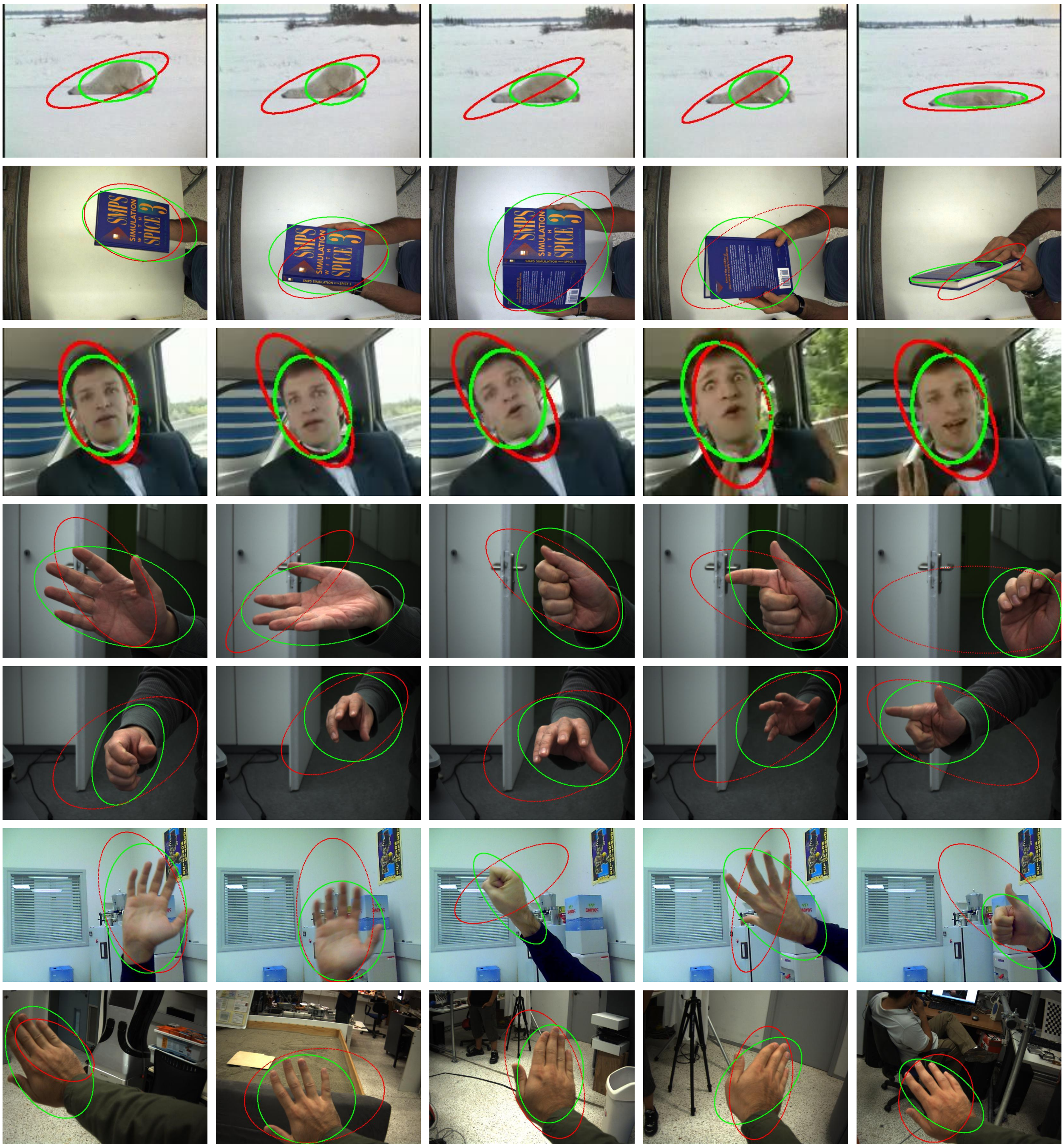


Figure 4.3: Experimental results and qualitative comparison between the proposed framework providing tracking only results (green solid ellipse) and the tracking algorithm of [77] (red solid ellipse).

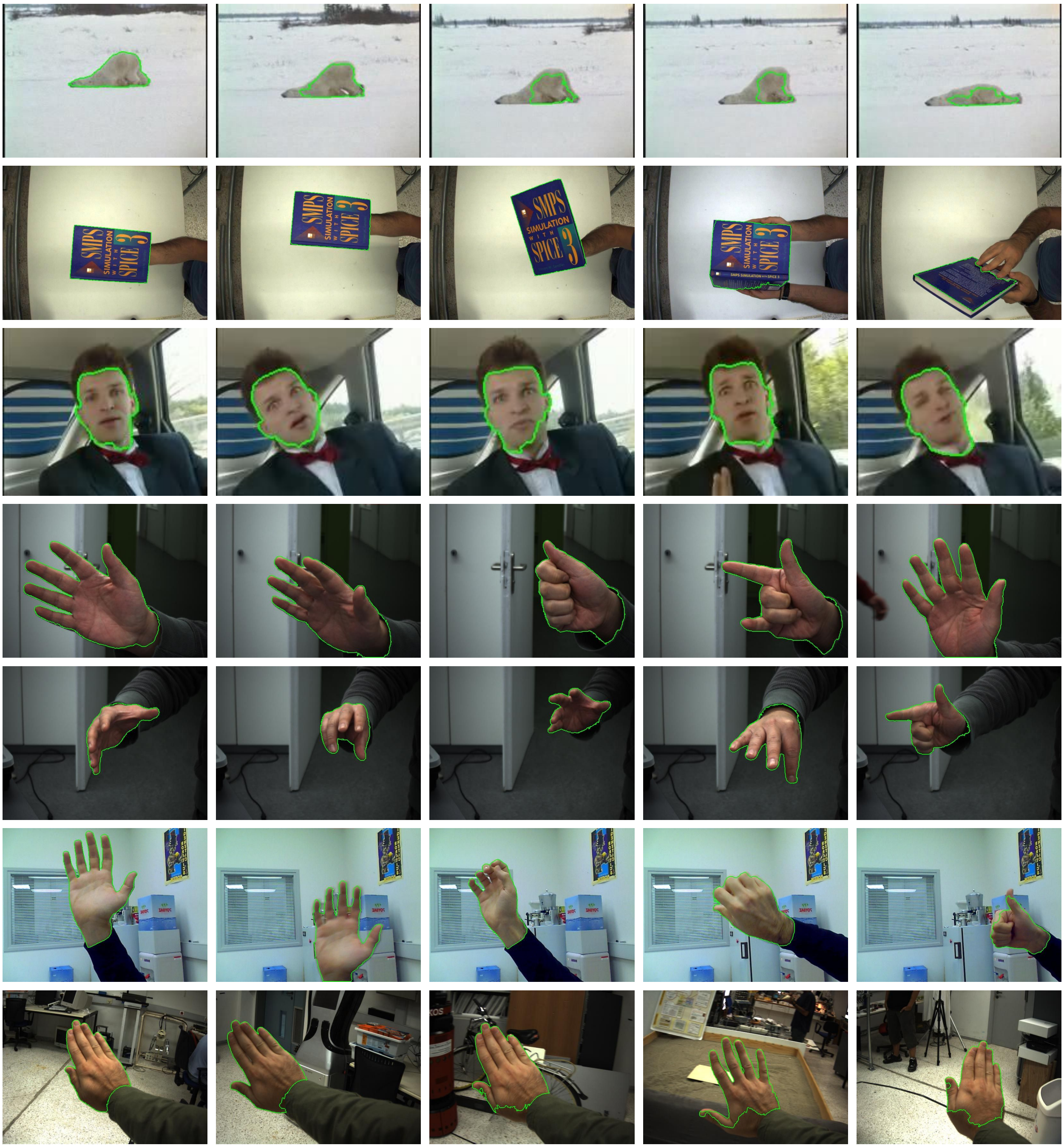


Figure 4.4: Experimental results on the segmentation performance of the the proposed framework providing the object outline in snapshots of the test sequences (green solid object contour).

The second image sequence that is utilized to obtain qualitative results is similar to the one presented in Fig. 4.1(c). The image sequence consists of 860 frames of size 640×480 pixels each. Figure 4.6 illustrates results on selected frames of that test sequence, organized in the same way as described above.

4.4 Quantitative Assessment

In this section, a quantitative assessment of the individual tracking and segmentation performance of the proposed method is presented. Ground truth data have been obtained for two test image sequences, representing human hands in action. The ground truth data consist of binary masks indicating the full area of the tracked hand throughout each of the image sequences. Moreover, an ellipse that includes the segmented hand is computed for each frame. The ground truth object masks have been obtained by visual inspection on the results of the the state-of-art skin color detection and tracking algorithm presented in [2].

The first test image sequence, represents a human hand performing articulations in a simple background. It is the one represented by the frame in Fig. 4.1(a) and is denoted as Hand-1. The second test image sequence is similar to the one presented in Fig. 4.1(c). This image sequence consists of 860 frames of size 640 × 480 pixels and is denoted as Hand-2.

A two phase analysis of the performance of the proposed methodology is carried out, regarding the tracking and the segmentation results, individually.

Quantitative Assessment on Tracking

A quantitative assessment regarding the tracking performance of the proposed method is provided for the Hand-1 and Hand-2 test image sequences. The performance of the EM-shift color tracking method [77] that is utilized in the proposed framework rely on a set of options regarding:

- the β_{track} parameter that controls the adaptation of the tracking Gaussian kernel
- the colorspace (or a subspace) that is utilized to represent color information
- the number of bins of the color histogram (appearance model)
- the convergence criterion of the EM algorithm (fixed number of EM iterations, minimum error of convergence or a combination of them)

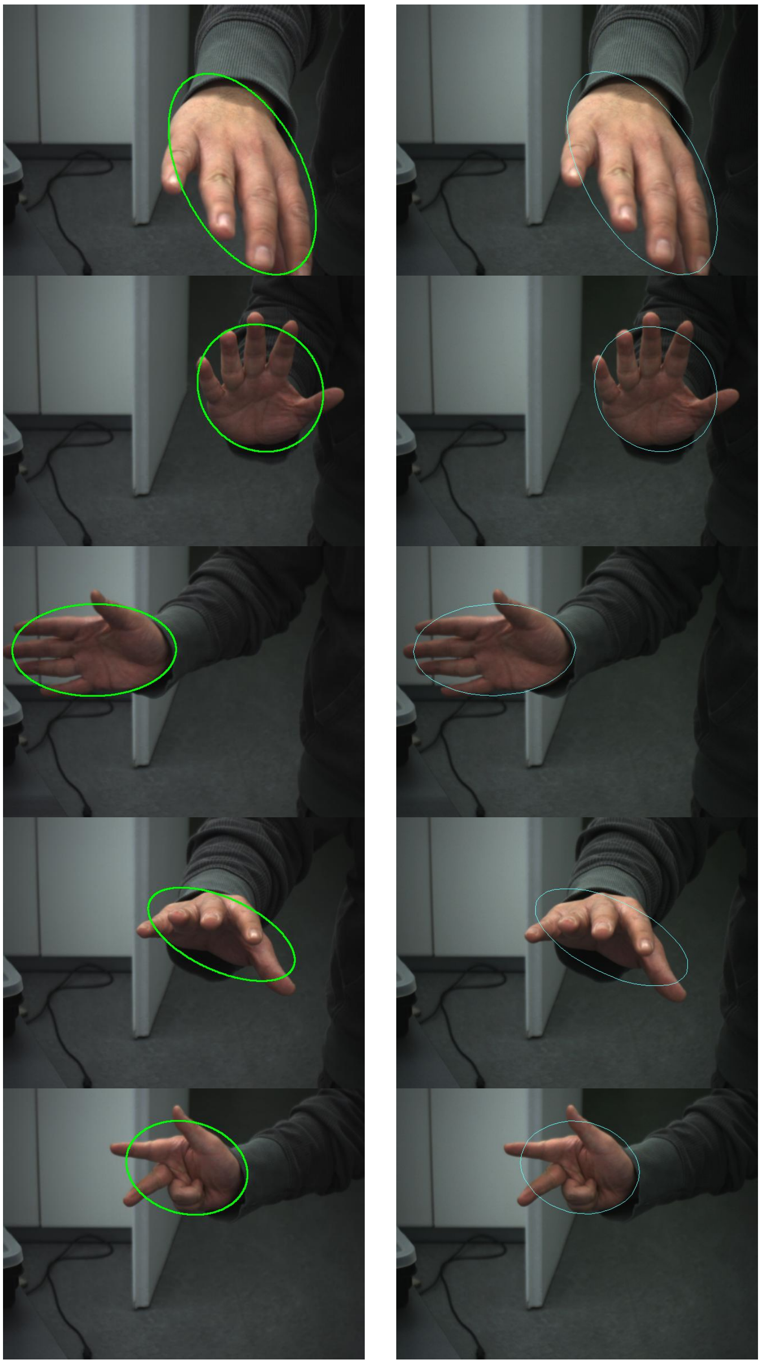


Figure 4.5: Selected frames from the image sequence of Fig. 4.1(a), representing a human hand in action are provided in this figure. Each row shows the same frame, whereas the left column illustrates the tracking green ellipse computed by the proposed frame and the right column illustrates the corresponding results, in cyan color, computed by the state-of-art skin color tracking method [2].

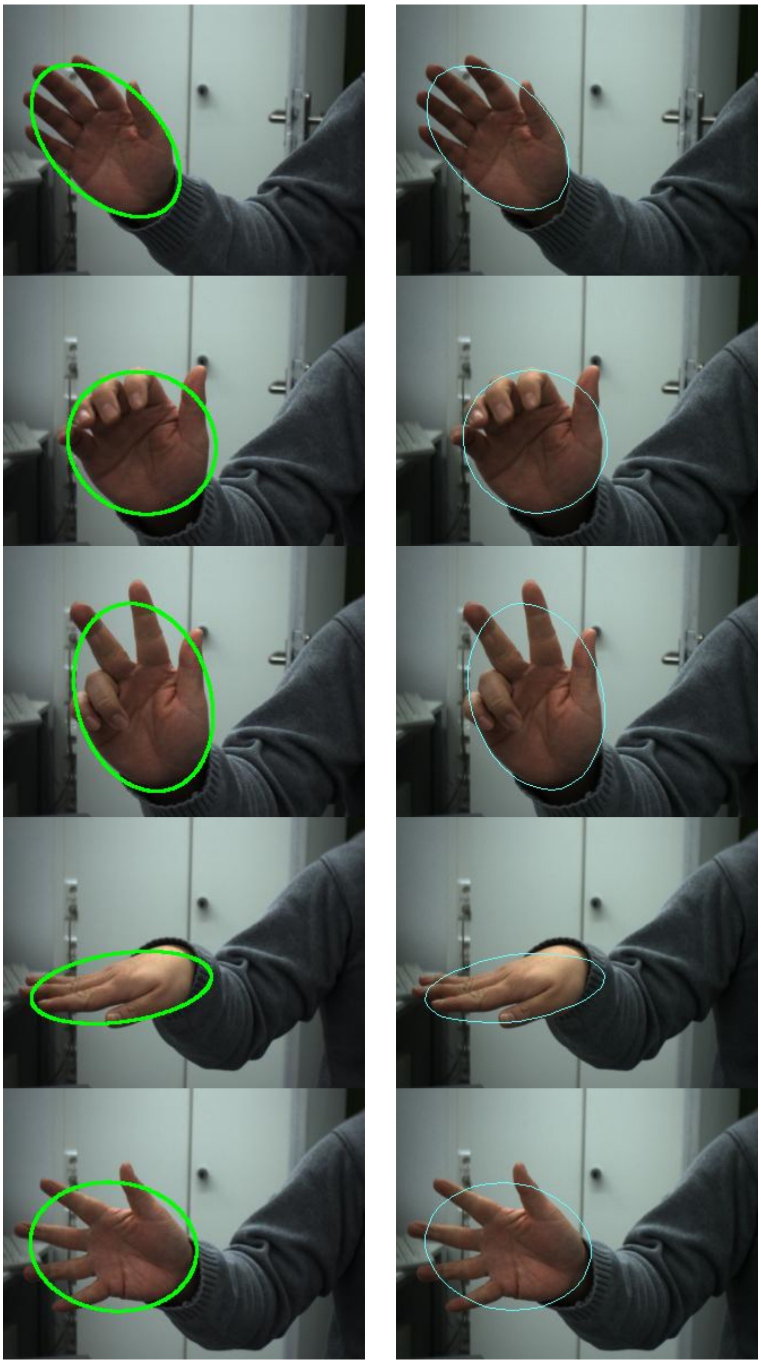


Figure 4.6: Selected frames from an image sequence representing a human hand in action are provided in this figure. Each row shows the same frame, whereas the left column illustrates the tracking green ellipse computed by the proposed frame and the right column illustrates the corresponding results, in cyan color, computed by the state-of-art skin color tracking method [2].

An efficient configuration of the β_{track} parameter depends on prior knowledge of the underlying distribution of the color that is to be tracked and the level of the noise that is present in the image sequence, according to the authors of the method in [77]. The parameter β_{track} practically controls the iso-contour of the Gaussian parametric kernel that will be considered to represent the foreground object and to build its appearance model.

An important decision regards the colorspace that is chosen to represent the color information of an image. For example, skin colored objects are efficiently represented in HSV or YCbCr colorspace. Discarding the V (value) or Y (luminance) component that stands for illumination-brightness, the skin-color objects representation is inherently more robust to illumination changes. Moreover, the choice for the number of histogram bins between $8, 16$ or 32 is to be determined according to the image content, adding only a insignificant influence to the performance of the tracking method.

Last but not least, the convergence criterion of the EM procedure of the tracking method is a crucial option towards its performance. A fixed number of EM iterations or a stopping threshold value can be determined to defined the convergence of the EM procedure. A combination of a maximum number of EM iterations and a stopping threshold value is the best setup for efficient tracking regardless the image content, the velocity of the moving object and the frame rate of the video that is processed.

Based on the described parameter configuration of the tracking part of the proposed method, the tracking performance is evaluated based on the selected test image sequences. Given the resulting binary object mask, produced by the proposed method for each frame of a test image sequence, the area of the bounding box (the number of pixels within it) containing the object mask is calculated, indicating its scale with respect to the total image area.

Figure 4.7 graphically illustrates the measurements of the resulting bounding box area throughout each of the test image sequences, approximating the true bounding box area. However, this statistic metric is provided to partially assess the tracking performance of the proposed method. To this end, two additional measures are computed based on the resulting bounding box object representation. The first measure regards the overlapping area between the bounding box produced by the proposed method and the true bounding box for each frame. The second measure refers to the Euclidean distance in pixels between the centers of these bounding boxes for each frame. More specifically, the ratio of the true bounding box area to the resulting bounding box area, indicates a measure of accuracy towards the tracking performance of the proposed method. Graphs (a) and

(b) of Fig. 4.8 illustrate the results for the two test image sequences, showing the high tracking performance of the proposed method. The overlapping area ratio approximates the unity throughout each of the test image sequences. Figure 4.9 illustrates the measurements regarding the Euclidean distance in pixels between the centers of the bounding boxes for each of the test image sequences.

Moreover, the algebraic differences in pixels per dimension of the bounding boxes are calculated. The algebraic difference for each dimension of the bounding boxes is normalized with respect to the corresponding dimension of the image. Figure 4.10 graphically illustrates the results for the two test sequences, Hand-1 in (a) and Hand-2 in (b). These results provide an additional confirmation of the high tracking performance of the proposed method in both test image sequences.

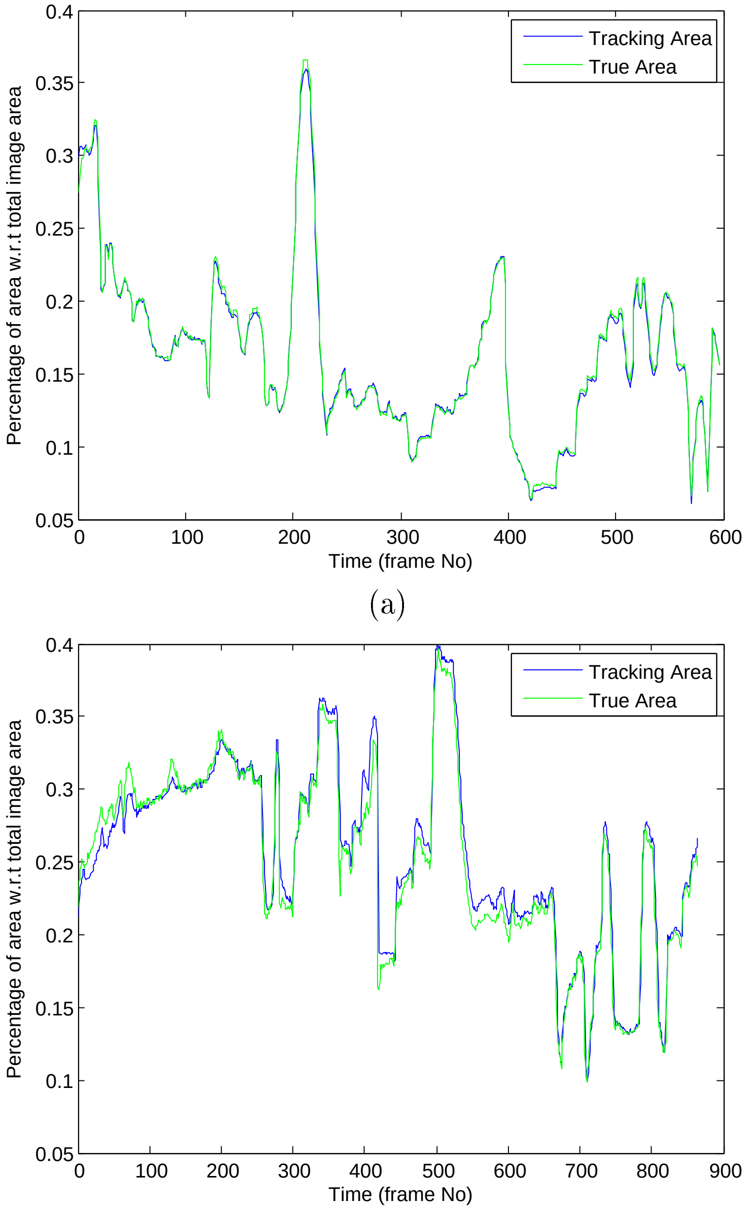
Quantitative Assessment on Segmentation

The second part of the quantitative evaluation of the proposed method regards the object segmentation performance, in each of the Hand-1 and Hand-1 test image sequences.

Based on the derivation of the Random Walker-based image segmentation method and its integration within the proposed framework presented in Section 2.2.2 and in Section 3.3.5, respectively, there are three customizable parts that control the segmentation performance affecting the performance of the overall framework.

The first of these parts regards the graph construction options, that is the graph connectivity policy and the weighting function that is chosen to weight the graph edges, practically to populate the Laplacian matrix (Eq. (2.18)). The connectivity policy of the graph controls the sparsity of the Laplacian matrix affecting the Random walker segmentation performance. In our case, the connectivity of the graph is set to 4-closest-neighbors of each graph node, that is an image pixel. Regarding the weighting function, the ubiquitous Gaussian function of Eq. (3.8) is utilized, that despite its simplicity, serves efficiently in mapping nodal intensities between the image pixels to connecting weights of the undirected graph representation. Moreover, a single parameter is introduced to the system by using the Gaussian function, that is β parameter, keeping the tuning of the procedure less complex. The parameter β controls the variance of the Gaussian function, thus the severity of the random walks biases on the graph. See Fig. 3.12 and Fig. 3.13 in Section 3.3.5 for more details. A more elaborate function could easily be introduced to the proposed method providing a different type of mapping of the pixel intensities or any other cue or combination of cues to the connecting weights of the Laplacian graph.

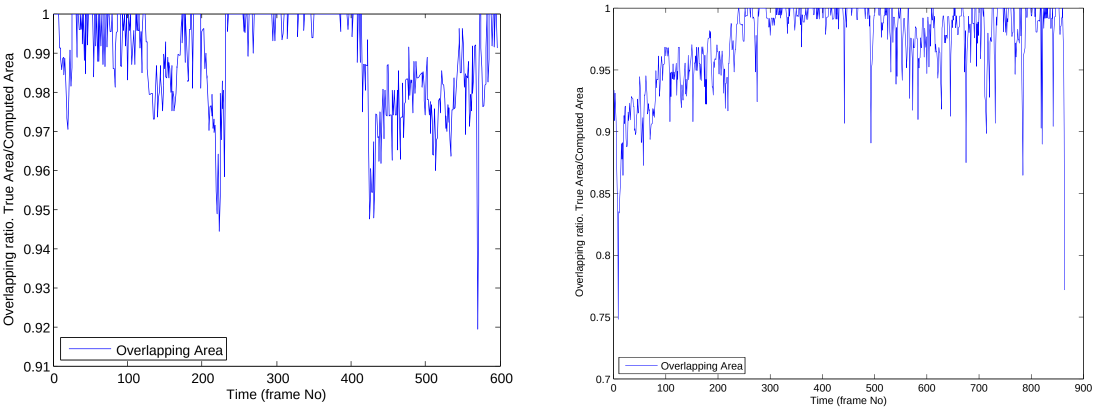
The second part of customizable options regards the choice between the three variants



(a)

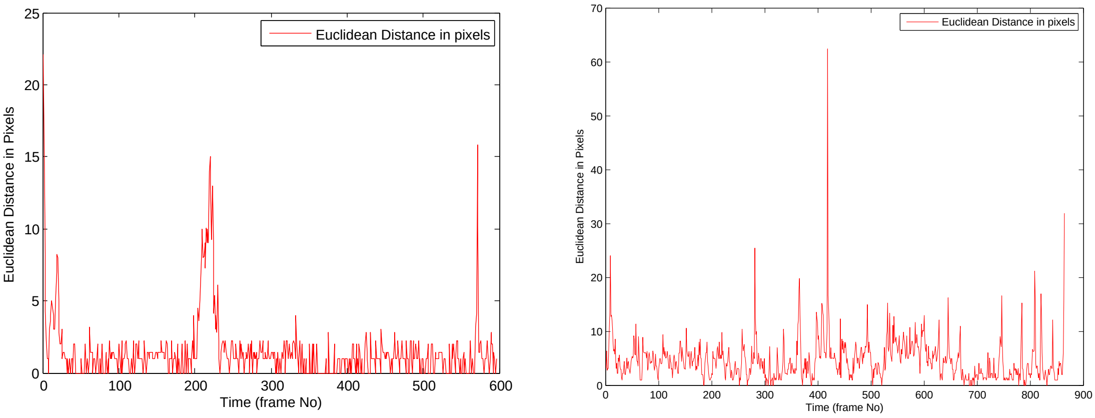
(b)

Figure 4.7: The area of each of the tracking bounding boxes computed by the two competing methods is measured and illustrated in this figure. The area of each bounding box is normalized with respect to the total image area. The green line corresponds to the true bounding box area, whereas the blue line indicates the results computed by the proposed method. Images (a) and (b) correspond to the Hand-1 and Han-2 test image sequences utilized in the quantitative assessment.



(a) (b)

Figure 4.8: The ratio of the overlapping area between the bounding boxes computed by the two competing methods is provided. The ratio of the true bounding box area to the bounding box area computed by the proposed method, indicates a measure of accuracy for the tracking performance of the proposed method. Images (a) and (b) illustrate the results for the two test image sequences utilized in the quantitative assessment, showing the high tracking performance of the proposed method (overlapping area ratio approximates the unity).



(a) (b)

Figure 4.9: The Euclidean distance in pixels between the centers of the two bounding boxes is calculated throughout each frame of each of the test image sequences. The results are illustrated in (a) and (b). The resulting distance of the bounding box centers indicates the high tracking performance of the proposed method.

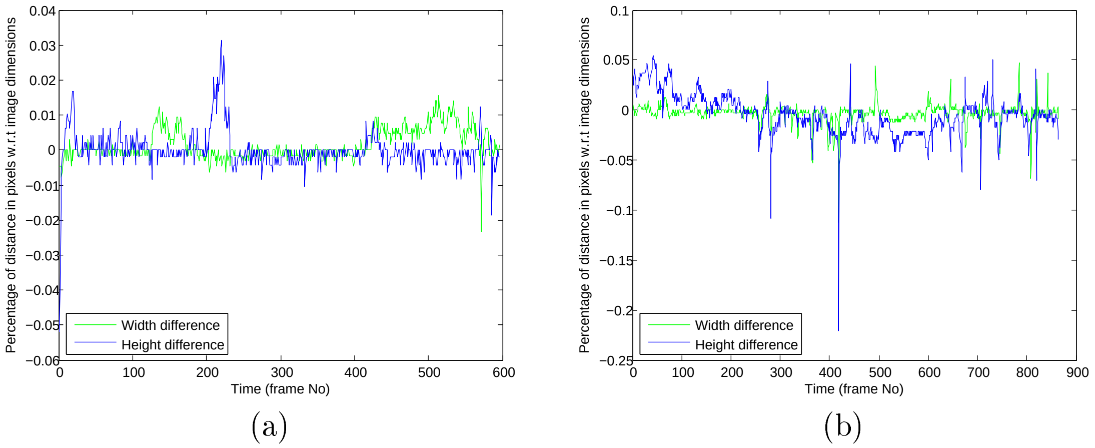


Figure 4.10: The algebraic difference in pixels per dimension of the resulting bounding boxes is calculated throughout each of the test image sequence and illustrated in (a) and (b), respectively. Red line corresponds to normalized algebraic difference of the width dimension of the true bounding box to the one computed by the proposed method. The blue line corresponds to the algebraic difference of the height dimension between the two bounding boxes.

of the Random Walker formulation for the image segmentation problem, presented in Section 3.3.5. The three formulations regard the use of seeds (Eq. (2.22)), priors (Eq. (2.24)) or a combination of them (Eq. (2.25)) to form the system of linear equations that is to be solved in order to obtain a real-valued solution for each label of the K-way segmentation. We remind that in case where prior information is incorporated to the Random Walker formulation, the γ parameter is introduced, controlling the authority of the prior information (probability) as opposed to the label information (probability) provided by random walks carried out through the biases on the graph toward the potential labels. These biases are in turn controlled by parameter β.

Finally, the third set of options, that control the segmentation performance of any selected Random Walker-based formulation, regards the appearance models utilized to model the region-specific color information after the segmentation of each frame. Two color histograms are utilized for this purpose, serving the computation of the color prior information, which is probabilistically fused with the spatial image cue to further be utilized are seeds and/or priors for the segmentation of the current frame. The selected number of bins in both color histograms is 16 per dimension.

The segmentation performance is assessed for the two selected test image sequences, based on the ground-truth data. To this end, the statistic metrics of Recall, Precision and F-measure, from the field of Information Retrieval, are utilized to evaluate the quality of the produced fine segmented mask per frame by the proposed method.

A brief description of the utilized statistic metrics is provided, blending their meaning in the context of the information retrieval and the image segmentation problem. Consider the ground truth binary image provided by the skin-color detection and tracking method for the foreground object class per frame, as the list of pixels that are known to belong to (are relevant) to that class. A set of retrieved pixels per frame refers to the pixels that are annotated to the foreground object class by the testing segmentation method in that frame. Based on these descriptions, the Precision metric is the fraction of the retrieved pixels by the testing method that are relevant to the foreground object class for a single processed frame. The Recall metric is defined by the fraction of pixels that are relevant to the foreground object class that are successfully retrieved. It is possible to interpret precision and recall as probabilities. Precision is the probability that a (randomly selected) retrieved pixel is relevant to the foreground object class. Recall is the probability that a (randomly selected) relevant pixel is retrieved through the segmentation procedure, thus correctly annotated to the foreground object class. Finally, the F-measure (or balanced F-score) combines precision and recall, yielding an harmonic mean of precision and recall defined by:

$$F_{measure} = 2\frac{Precision \cdot Recall}{Precision + Recall} \tag{4.1}$$

To start with the experimental evaluation, the influence of the parameters β and γ in the segmentation performance of the proposed method is explored, using each of the three variants of the Random Walker formulation.

We initially assess the performance of the basic Random Walker formulation for image segmentation, utilizing only seeds and solving the linear system of Eq. (2.22). The segmentation performance is tested for a set of β values, that is $[1, 20, 100, 200]$. The scores in Table 4.1 indicate the overall high performance of the proposed framework for each tested value of the β that exceed 90%. The highest scores are noticed for the medium values of $\beta = 20$ for Hand-1 and $\beta = 35$ for Hand-2 image sequence.

In the following, keeping the β value for which the highest score is exceeded using only seeds, the incorporation of prior information to the system is assessed by setting the γ parameter to values of various scale. The formulation of linear system of equations provided by Eq. (2.25) is utilized. We remind that the value of the γ parameter expresses the "degree of authority" of the provided prior information to the system as opposed to the label-wise soft assignments, which are computed by the random walks on the graph. Table 4.3 provides the resulting scores for medium values of the β, which are equal to

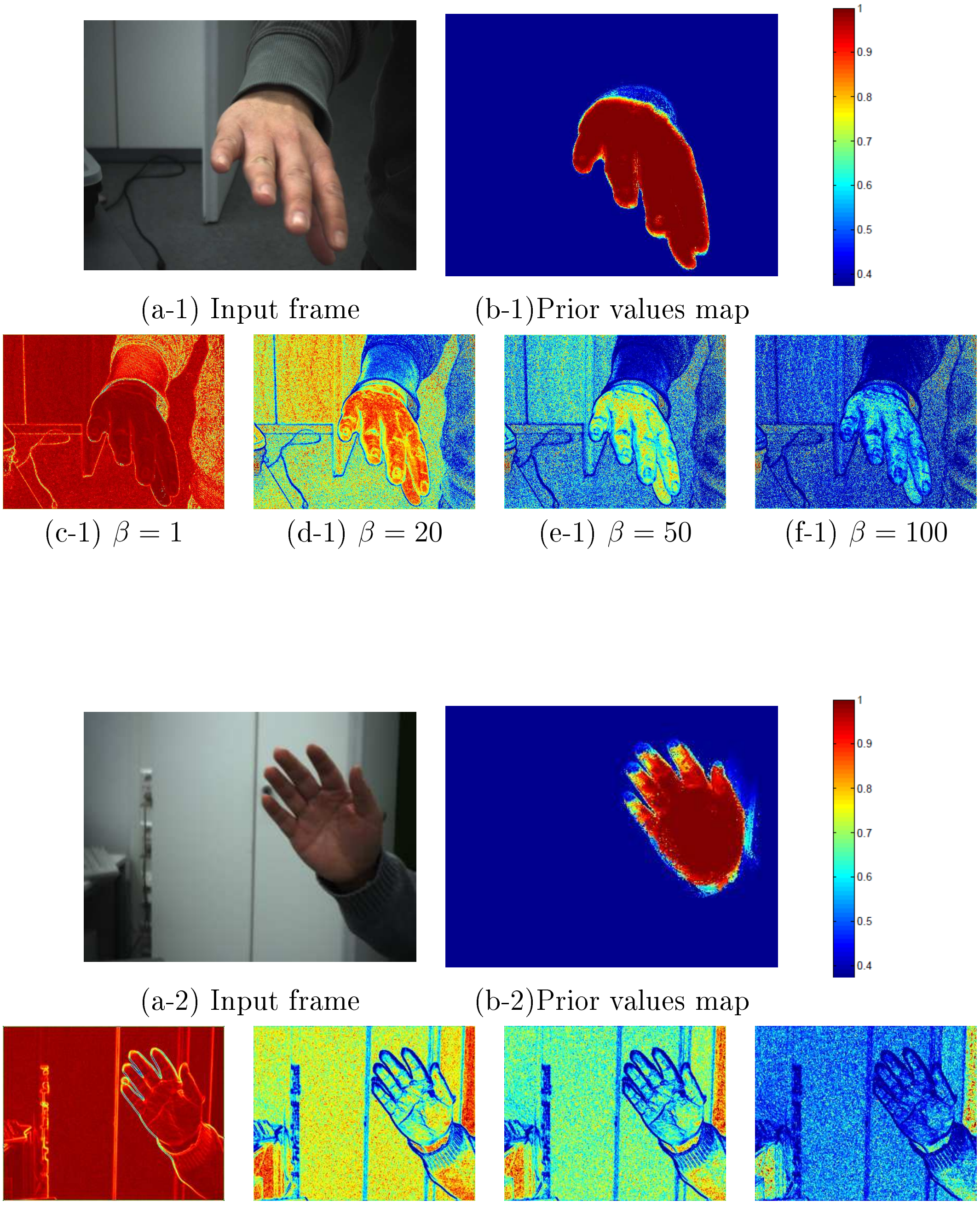
20 and 35 for Hand-1 and Hand-2 test sequences, respectively, while the γ ranges within $[0 - 0.5]$. For the Hand-1 test sequence, the higher performance of 98.8% is achieved for $\gamma = 0.05$. The highest score for the Hand-2 sequence is (94%), that is achieved for $\gamma = 0.005$. The configuration of $\gamma = 0$ indicates the usage of the Random Walker formulation, where only seeds are utilized. In that case, the linear system of equations is constructed based on Eq. (2.22).

In order to overlook the random walker biases and explore the influence of the prior information to the system, the β is set equal to 1 and the γ parameter ranges within $[0-0.5]$. The segmentation performance degrades in overall based on the resulting scores in Table 4.2. Especially, for the Hand-1 sequence the attenuation of the previously highest value is around 8%. There are slightly lower resulting scores for the Hand-2 sequence for all the tested values of γ.

The second set of experiments assess the influence of β to the segmentation performance for a fixed value of γ. Table 4.4 provides the obtained statistic scores for $\gamma = 0.05$ for both test image sequences. β ranges within $[1-200]$. The highest score for the Hand-1 sequence is validated for the configuration with $\beta = 20$ and $\gamma = 0.05$, whereas the scores for the rest of the configurations are notably lower but over 90%.

Finally, the variations on the segmentation performance using each of the three variants of the Random Walker formulation is assessed. For each test image sequence, we adopt the parameter configurations per test sequence, that resulted the highest scores throughout the already provided experimental results. Thus, β is set to 20 and γ to 0.05 for the Hand-1 sequence, whereas $\beta = 35$ and $\gamma = 0.05$ for the Hand-2 sequence.

Table 4.5 summarizes the average Precision, Recall and F-measure performance of the proposed algorithm compared to the ground truth data throughout each of the test image sequences. Although all three options perform satisfactorily, the usage of both seeds and priors together improves the segmentation performance in both test image sequences.



(a-1) Input frame (b-1)Prior values map

(c-1) $\beta = 1$ (d-1) $\beta = 20$ (e-1) $\beta = 50$ (f-1) $\beta = 100$

(a-2) Input frame (b-2)Prior values map

(c-2) $\beta = 1$ (d-2) $\beta = 35$ (e-2) $\beta = 50$ (f-2) $\beta = 100$

Figure 4.11: Inputs toward the quantitative evaluation of the two test image sequence. (a) Input frame (b) Prior pixel-wise likelihood values for the input frame, computed by the probabilistic fusion of color and spatial image cues. (c-f) The visualization of the Laplacian matrix, representing the graph weights or else the Random Walker biases for various values of β, in Eq. (3.8).

Hand-1	**Precision**	**Recall**	**F-measure**
$\beta = 1$	91.2%	91.0%	91.0%
$\beta = 20$	93.2%	92.8%	93.0%
$\beta = 50$	91.4%	90.6%	91.0%
$\beta = 100$	91.3%	90.5%	90.9%
$\beta = 200$	91.2%	90.7%	90.9%
Hand-2	**Precision**	**Recall**	**F-measure**
$\beta = 1$	92.4%	94.5%	93.3%
$\beta = 35$	94.0%	94.0%	94.0%
$\beta = 50$	94.1%	94.0%	94.0%
$\beta = 100$	94.0%	94.2%	94.0%
$\beta = 200$	93.5%	94.4%	94.0%

Table 4.1: Quantitative assessment of Random Walker based segmentation performance using only seeds on the two hand image sequences. The segmentation performance is assessed for various values of β within $[1 - 200]$.

Hand-1 $\beta = 1$	**Precision**	**Recall**	**F-measure**
$\gamma = 0.5$	91.3%	90.9%	91.7%
$\gamma = 0.25$	91.1%	91.0%	91.0%
$\gamma = 0.05$	91.3%	90.9%	91.7%
$\gamma = 0.025$	91.2%	91.0%	91.0%
$\gamma = 0.005$	91.3%	90.9%	91.7%
$\gamma = 0$ (seeds only)	91.2%	91.0%	91.0%
Hand-2 $\beta = 1$	**Precision**	**Recall**	**F-measure**
$\gamma = 0.5$	96.7%	97.8%	97.2%
$\gamma = 0.25$	96.7%	97.8%	97.2%
$\gamma = 0.05$	96.4%	98.2%	97.3%
$\gamma = 0.025$	96.4%	98.2%	97.3%
$\gamma = 0.005$	96.2%	98.2%	97.2%
$\gamma = 0$ (seeds only)	92.4%	94.5%	93.3%

Table 4.2: Quantitative assessment of Random Walker based segmentation performance using seeds and priors on the two hand image sequences. β is set to 1, whereas γ ranges within $[0 - 0.5]$.

Hand-1 $\beta = 20$	**Precision**	**Recall**	**F-measure**
$\gamma = 0.5$	90.9%	90.6%	90.7%
$\gamma = 0.25$	90.9%	90.3%	90.6%
$\gamma = 0.05$	99.2%	98.4%	98.8%
$\gamma = 0.025$	91.4%	90.7%	91.0%
$\gamma = 0.005$	91.4%	90.7%	91.0%
$\gamma = 0.0005$	94.2%	93.8%	94.2%
$\gamma = 0$ (seeds only)	93.2%	92.8%	92.9%
Hand-2 $\beta = 35$	**Precision**	**Recall**	**F-measure**
$\gamma = 0.5$	97.0%	96.5%	96.7%
$\gamma = 0.25$	97.1%	97.3%	97.2%
$\gamma = 0.05$	97.6%	97.4%	97.5%
$\gamma = 0.025$	97.8%	97.4%	97.6%
$\gamma = 0.005$	98.0%	98.0%	98.0%
$\gamma = 0.0005$	98.1%	98.0%	98.0%
$\gamma = 0$ (seeds only)	94.0%	94.0%	94.0%

Table 4.3: Quantitative assessment of Random Walker based segmentation performance using seeds and priors on the two test image sequences Hand-1 and Hand-2, where β is set to 20 and 35, respectively. γ values range within $[0 - 0.5]$ for different numerical scales.

Hand-1 $\gamma = 0.05$	**Precision**	**Recall**	**F-measure**
$\beta = 1$	91.2%	91.0%	91.0%
$\beta = 5$	91.3%	90.9%	91.0%
$\beta = 20$	99.2%	98.4%	98.8%
$\beta = 50$	90.9%	90.5%	90.7%
$\beta = 100$	90.9%	90.8%	90.8%
$\beta = 200$	90.6%	91.0%	90.8%
Hand-2 $\gamma = 0.05$	**Precision**	**Recall**	**F-measure**
$\beta = 1$	96.4%	98.2%	97.3%
$\beta = 10$	93.8%	94.1%	94.0%
$\beta = 35$	97.6%	97.4%	97.5%
$\beta = 50$	97.0%	97.3%	97.1%
$\beta = 100$	96.9%	95.6%	96.2%
$\beta = 200$	93.2%	91.7%	92.5%

Table 4.4: Quantitative assessment of Random Walker based segmentation performance using seeds and priors on the two test image sequences. γ parameter is predefined equal to 0.05 and the segmentation performance is tested for various values of β parameter.

Segmentation option Hand-1	**Precision**	**Recall**	**F-measure**
Priors ($\gamma = 0.05$)	91.3%	90.9%	91.7%
Seeds ($\beta = 20$)	93.2%	92.8%	93.0%
Seeds & Priors ($\beta = 20$,$\gamma = 0.05$)	99.2%	98.4%	98.8%
Segmentation option Hand-2	**Precision**	**Recall**	**F-measure**
Priors ($\gamma = 0.05$)	96.4%	98.2%	97.3%
Seeds ($\beta = 35$)	94.0%	94.0%	94.0%
Seeds & Priors ($\beta = 35$,$\gamma = 0.05$)	97.6%	97.4%	97.5%

Table 4.5: Quantitative assessment of segmentation performance for the three variant formulations of the Random Walker image segmentation method. Results for the Hand-1 and Hand-2 test image sequences are provided.

Chapter 5

Discussion

In this work, we presented a novel method for on-line, joint tracking and segmentation of a non-rigid object in a monocular video, captured by a possibly moving camera. The proposed approach aspires to relax several limiting assumptions regarding the appearance and shape of the tracking object, the motion of the camera and the lighting conditions. The key contribution of the proposed framework is the efficient combination of an appearance-based tracking algorithm with a Random Walker-based segmentation algorithm in a close-loop that jointly enables drift-free tracking and fine segmentation of the target object. A 2D affine transformation is computed to propagate the segmented object shape of the previous frame to the new frame exploiting the information provided by the ellipse region (iso-contour of a spatial Gaussian distribution) capturing the segmented object and the ellipse region predicted by the tracker in the new frame. A shape-band area is computed indicating an area of uncertainty where the true object boundaries lie in the new frame. Static image cues including pixel-wise color and spatial likelihoods are fused using Bayesian inference to guide the Random Walker-based object segmentation in conjunction with the brightness likelihoods between neighboring pixels.

The performance of the proposed method is qualitatively demonstrated, in a series of challenging videos in comparison with the results of the EM-shift tracking method presented in [77] and ground truth tracking and segmentation data. Moreover, the quantitative performance of the individual tracking and the segmentation parts of the proposed framework is assessed. The tracking performance of the proposed method is compared to both the stand-alone EM-shift color tracking method of [77] and ground truth data. The segmentation performance of the proposed method is compared to the ground-truth data. The experimental results validate the effectiveness of the the proposed framework as opposed to the stand-alone EM-shift color tracking method. Moreover, high performance is achieved regarding the individual tracking and segmentation results compared to ground-truth data.

The performance of the EM-shift color tracking method [77], relies on a set of options regarding the colorspace (or a subspace) that is utilized to represent the color information, the number of bins of the color histogram that act as an appearance model, the convergence criterion of the EM algorithm (maximum number of iterations, stopping threshold value or a combination or them) and finally the β_{track} parameter that controls the adaptation of the tracking Gaussian kernel of the method. A correct configuration for the β_{track} parameter depends on prior knowledge of the underlying distribution of the color that is to be tracked and the level of the noise that is present in the image sequence, according to [77]. The parameter β_{track} practically controls the iso-contour of the Gaussian parametric kernel that is determined to represent the foreground object and to build its appearance model.

Moreover, the convergence criterion of the EM procedure of the tracking method is a crucial option affecting its performance. A fixed number of EM iterations or a stopping threshold value can be determined to define the convergence of the EM procedure. A combination of a maximum number of EM iterations and a stopping threshold value is the best option for efficient tracking. A stopping threshold value can be defined regarding the number of new pixels added to the new estimated elliptical tracking region as compared tho the previous estimation of that region between consecutive iterations with respect to the image size. Many alternative heuristic functions can be utilized to implement a new stopping criterion integrating additional information regarding the underlying tracked distribution, the residual of the position of the Gaussian kernel between consecutive frames on the image plane etc.

Based on the description of the Random Walker-based image segmentation technique provided in Sections 3.3.5and 2.2.2, there are three main parts that are customizable and control its segmentation performance within the proposed framework.

The first of these parts regards the graph construction options, that is the graph connectivity policy and the weighting function that is utilized to weight the graph edges. The connectivity of the graph controls the sparsity of the Laplacian matrix, thus it affects the Random walker-based segmentation performance for fixed values of its parameters. The ubiquitous Gaussian function of Eq. (3.8) is utilized, which despite its simplicity serves efficiently for mapping nodal intensities between the image pixels to connecting weights of its undirected graph representation. Moreover, there is only the β parameter which is introduced, keeping the tuning of the algorithm less complex. See Fig. 3.12 and

Fig. 3.13 in Section 3.3.5 for more details regarding the key role of this parameter to the image segmentation procedure. A more elaborate function could easily be introduced to the system providing a alternative type of mapping of the pixel intensities or any other cue or combination of cues, to connecting weights of the Laplacian graph.

The second part regards the choice between the three variants of the Random Walker formulation for the image segmentation problem, as they presented in Section 3.3.5. The usage of seeds (Eq. (2.22)), priors (Eq. (2.24)) or a combination of them (Eq. (2.25)) is chosen to form the system of linear equations that is to be solved in order to obtain a real-valued solution for each label of the K-way segmentation. We remind that in case that prior information is incorporated to the Random Walker formulation, the γ parameter is introduced, controlling the authority of the prior information (probability) as opposed to the label information (probability) provided by random walks carried out on the graph toward the potential labels. These biases are in turn controlled by parameter β.

Finally, the third set of options affecting the segmentation performance of the Random Walker-based method regards the two appearance models, which maintain the color information of the resulting foreground object and background regions after the segmentation of each frame. The role of these appearance models is crucial towards the integration of the segmentation part with the tracking one in the proposed framework. Two multi-dimensional histograms are used as appearance models. Based on them, the color likelihoods per region are computed and are further utilized to compute the probabilistic fusion of the the color and the spatial image cues. Thus, the color information directly influence the resulting likelihoods of the fusion procedure, which in turn are used to guide the automatic seed selection and/or act as prior information on the potential labels of the segmentation. The number of dimensions of a histogram is controlled by the number of channels of the colorspace that is selected to encode the color information of each frame. The number of bins per dimension of a histogram is selected by the user. Moreover, alternative appearance models could be utilized to capture the region-specific color information, such as mixture of Gaussians.

To conclude the discussion, it is essential to present some limiting factors regarding the performance of the proposed joint tracking and segmentation method. There are some representative examples of image sequences shown in Fig. 5.1, where any of the following issues or a combination of them causes failure of the proposed tracking methodology.

- Object natural boundaries are of low contrast with regard to the underlying background

- There is a big overlap among the object and the background color-based appearances

- non-rigid objects of very small surface

The segmentation part of the method fails to extract an accurate object mask result. As a consequence, an invalid image partitioning of the foreground-object and background image regions results an invalid feed-back of color information to the appearance models.

5.1 Future Work

In this last section, we outline a number of algorithmic issues that could be investigated in more detail, as work of future interest. These issues could be incorporated to the proposed method in order to alleviate the aforementioned limiting factors, regarding its tracking and segmentation performance, and to extend its capabilities.

An immediate extension of the proposed work involves the incorporation of additional image cues such as texture and low-level motion information as prior information towards increased robustness of both tracking and segmentation components.

The performance of alternative weighting schemes regarding the construction of the graph Laplacian matrix needs to be explored, except for the Gaussian weighting function used in Eq. (3.8). Moreover, the idea of incorporating additional information to the construction of the Laplacian, such as texture and low-level motion cues besides the utilized image brightness information, provides an interesting field on investigation toward an enhanced Random Walker based segmentation performance.

Another part of the proposed method that is to be optimized refers to the computation of the shape-band area around the propagated prior object shape. The width of the shape-band area is uniformly determined around the propagated object shape and is currently defined equal to the Hausdorff distance between the previously segmented object contour points and the propagated object contour points. A point-wise computation of the shape-band width across the propagated object contour is an interesting modification of the proposed system that may lead to better and more robust segmentation performance.

In the following, a bootstrapping mechanism that will automatically determine an optimal configuration of the crucial parameters of the proposed method is an interesting extension, that will set the proposed method fully automatic requiring no tuning of these parameters by the user.

Last but not least, an interesting extension of the proposed method would be the ability to track multiple objects with partial of full, instant or long-term occlusions by unifying it with the efficient and elaborate method presented in [52].

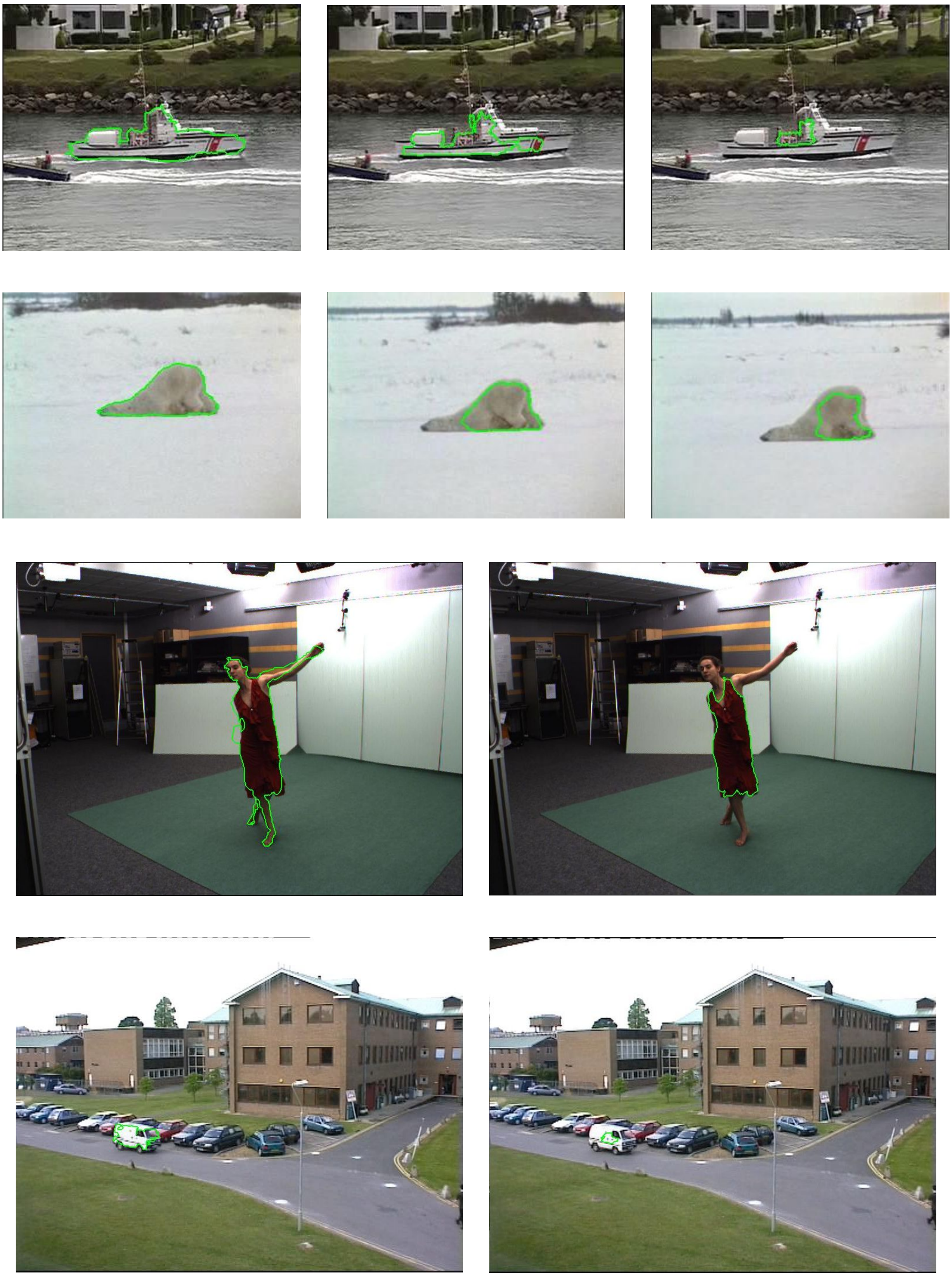


Figure 5.1: Representative example image sequence where the proposed methodology failed to perform. The segmentation failures (green contour) are illustrated in four image sequences on a variety objects that are to be tracked. From left to right in each row, the provided frames preserve temporal coherency.

Bibliography

[1] C. Aeschliman, J. Park, and A.C. Kak. A probabilistic framework for joint segmentation and tracking. *IEEE Conference on Computer Vision and Pattern Recognition 2010.*, pages 1371 –1378, 2010.

[2] A. A. Argyros and M. I. A. Lourakis. Real-time tracking of multiple skin-colored objects with a possibly moving camera. In *Computer Vision - ECCV 2004*, volume 3023 of *Lecture Notes in Computer Science*, pages 368–379. Springer Berlin / Heidelberg, 2004.

[3] S. Avidan. Support vector tracking. *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2001.*, 1:I–184 – I–191 vol.1, 2001.

[4] X. Bai, Q. Li, L.J. Latecki, W. Liu, and Z. Tu. Shape band: A deformable object detection approach. *IEEE Conference on Computer Vision and Pattern Recognition, 2009.*, pages 1335 –1342, 2009.

[5] H. Baltzakis and A. A. Argyros. Propagation of pixel hypotheses for multiple objects tracking. In *ISVC '09: Proceedings of the 5th International Symposium on Advances in Visual Computing*, pages 140–149, Berlin, Heidelberg, 2009. Springer-Verlag.

[6] Y. Bar-Shalom. *Tracking and data association.* Academic Press Professional, Inc., San Diego, CA, USA, 1987.

[7] H. Bay, T. Tuytelaars, and L. Van Gool. Surf: Speeded up robust features. In *ECCV*, 2006.

[8] M. Bertalmio, G. Sapiro, and G. Randall. Morphing active contours. *IEEE Transactions on Pattern Analysis and Machine Intelligence.*, 22(7):733 –737, 2000.

[9] C. Bibby and I. Reid. Robust real-time visual tracking using pixel-wise posteriors. In *Proceedings of the 10th European Conference on Computer Vision 2008*, pages 831–844, Berlin, Heidelberg, 2008. Springer-Verlag.

[10] M. Black and A. Jepson. Eigentracking: Robust matching and tracking of articulated objects using a view-based representation. In Bernard Buxton and Roberto Cipolla, editors, *Computer Vision ECCV '96*, volume 1064 of *Lecture Notes in Computer Science*, pages 329–342. Springer Berlin / Heidelberg, 1996.

[11] Y. Boykov and G. Funka-Lea. Graph cuts and efficient n-d image segmentation. *International Journal of Computer Vision*, 70(2):109–131, 2006.

[12] Franklin H. Branin, Jr. Computer methods of network analysis. In *DAC '67: Proceedings of the 4th Design Automation Conference*, pages 8.1–8.19, New York, NY, USA, 1967. ACM.

[13] R.T. Collins. Mean-shift blob tracking through scale space. *IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2003.*, 2:II – 234–40 vol.2, 2003.

[14] R.T. Collins, Yanxi Liu, and M. Leordeanu. Online selection of discriminative tracking features. *IEEE Transactions on Pattern Analysis and Machine Intelligence.*, 27(10):1631 –1643, 2005.

[15] D. Comaniciu and P. Meer. Mean shift analysis and applications. *The Proceedings of the Seventh IEEE International Conference on Computer Vision, 1999.*, 2:1197 –1203 vol.2, 1999.

[16] D. Comaniciu, V. Ramesh, and P. Meer. Real-time tracking of non-rigid objects using mean shift. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2000.*, 2:142 –149 vol.2, 2000.

[17] D. Comaniciu, V. Ramesh, and P. Meer. The variable bandwidth mean shift and data-driven scale selection. *Proceedings of the Eighth IEEE International Conference on Computer Vision, 2001.*, 1:438 –445 vol.1, 2001.

[18] D. Comaniciu, V. Ramesh, and P. Meer. Kernel-based object tracking. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 25(5):564 – 577, 2003.

[19] T.F. Cootes, G.J. Edwards, and C.J. Taylor. Active appearance models. *IEEE Transactions on Pattern Analysis and Machine Intelligence.*, 23(6):681 –685, 2001.

[20] I. J. Cox. A review of statistical data association for motion correspondence. *International Journal of Computer Vision*, 10(1):53–66, 1993.

[21] A. P. Dempster, N. M. Laird, and D. B. Rubin. Maximum likelihood from incomplete data via the em algorithm. *Journal of The Royal Statistical Society, Series B*, 39(1):1–38, 1977.

[22] J. Dodziuk. Difference Equations, Isoperimetric Inequality and Transience of Certain Random Walks. *Transactions of the American Mathematical Society*, 284(2):787–794, 1984.

[23] W. E. Donath and A. J. Hoffman. Lower bounds for the partitioning of graphs. *IBM Journal of Research and Development*, 17(5):420 –425, 1973.

[24] P. G. Doyle and J. L. Snell. *Random Walks and Electric Networks*. The Mathematical Association of America, Washington D.C, 1984.

[25] A. Elgammal, R. Duraiswami, D. Harwood, and L.S. Davis. Background and foreground modeling using nonparametric kernel density estimation for visual surveillance. *Proceedings of the IEEE*, 90(7):1151 – 1163, 2002.

[26] P. Fieguth and D. Terzopoulos. Color-based tracking of heads and other mobile objects at video frame rates. *IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 1997.*, pages 21 –27, 1997.

[27] D. Freedman and M.W. Turek. Illumination-invariant tracking via graph cuts. *IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2005.*, 2:10 – 17 vol. 2, 2005.

[28] G. H. Golub and C. F. Van Loan. *Matrix computations (3rd ed.)*. Johns Hopkins University Press, Baltimore, MD, USA, 1996.

[29] L. Grady. Multilabel random walker image segmentation using prior models. *IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2005.*, 1:763 – 770 vol. 1, 2005.

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

[31] L. Grady and Gareth FL. Multi-label image segmentation for medical applications based on graph-theoretic electrical potentials. In *ECCV 2004 Workshops CVAMIA and MMBIA*, Lecture Notes in Computer Science, pages 230–245. Springer, 2004.

[32] W. Hackbusch. *Iterative solution of large sparse systems of equations.* Applied mathematical sciences. Springer-Verlag, 1994.

[33] C. Harris and M. Stephens. A combined corner and edge detection. In *Proceedings of The Fourth Alvey Vision Conference*, pages 147–151, 1988.

[34] PW Holland and RE Welsch. Robust regression using iteratively reweighted least-squares. *Communications in Statistics-Theory*, 1977.

[35] D.P. Huttenlocher, J.J. Noh, and W.J. Rucklidge. Tracking non-rigid objects in complex scenes. *Proceedings of the Fourth International Conference on Computer Vision, 1993.*, pages 93 –101, 1993.

[36] M. Isard and A. Blake. Condensation - conditional density propagation for visual tracking. *International Journal of Computer Vision*, 29:5–28, 1998.

[37] M. Isard and J. MacCormick. Bramble: a bayesian multiple-blob tracker. *Eighth IEEE International Conference on Computer Vision, 2001.*, 2:34 –41 vol.2, 2001.

[38] A.D. Jepson, D.J. Fleet, and T.R. El-Maraghi. Robust online appearance models for visual tracking. *Proceedings of the 2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2001.*, 1:I–415 – I–422 vol.1, 2001.

[39] S Kakutani. Markov processes and the Dirichlet problem. *Proceedings of the Japan Academy*, 1945.

[40] S. Khan and M. Shah. Object based segmentation of video using color, motion and spatial information. *Proceedings of the 2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2001.*, 2:II–746 – II–751 vol.2, 2001.

[41] H. W. Kuhn. The hungarian method for the assignment problem. In *50 Years of Integer Programming 1958-2008*, pages 29–47. Springer Berlin Heidelberg, 2010.

[42] T. Lindeberg. Feature detection with automatic scale selection. *International Journal of Computer Vision*, 30:79–116, 1998.

[43] Laszlo Lovasz. Random walks on graphs: A survey. *Combinatorics, Paul Erdos is Eighty*, 2:1 –46, 1993.

[44] D.G. Lowe. Object recognition from local scale-invariant features. *The Proceedings of the Seventh IEEE International Conference on Computer Vision, 1999.*, 2:1150–1157 vol.2, 1999.

[45] Bruce D. Lucas and Takeo Kanade. An iterative image registration technique with an application to stereo vision. In *IJCAI'81: Proceedings of the 7th international joint conference on Artificial intelligence*, pages 674–679, San Francisco, CA, USA, 1981. Morgan Kaufmann Publishers Inc.

[46] Jacob Lurie. Review of spectral graph theory. *SIGACT News*, 30(2):14–16, 1999.

[47] RA Maronna. Robust M-estimators of multivariate location and scatter. *The annals of statistics*, 1976.

[48] K. Mikolajczyk and C. Schmid. Scale & affine invariant interest point detectors. *International Journal of Computer Vision*, 60(1):63–86, 2004.

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

[50] J. Munkres. *Topology (2nd Edition).* Prentice Hall, 2 edition, 2000.

[51] H.T. Nguyen, Qiang Ji, and A.W.M. Smeulders. Spatio-temporal context for robust multitarget tracking. *IEEE Transactions on Pattern Analysis and Machine Intelligence.*, 29(1):52 –64, 2007.

[52] V. Papadourakis and A. A. Argyros. Multiple objects tracking in the presence of long-term occlusions. *Comput. Vis. Image Underst.*, 114(7):835–846, 2010.

[53] C.P. Papageorgiou, M. Oren, and T. Poggio. A general framework for object detection. *Sixth International Conference on Computer Vision, 1998.*, pages 555 –562, 1998.

[54] K. Papoutsakis and A.A Argyros. Object tracking and segmentation in a closed loop. In *ISVC '10: Proceedings of the 6th International Symposium on Advances in Visual Computing*, Berlin, Heidelberg, 2010. Springer-Verlag.

[55] N. Paragios and R. Deriche. Geodesic active contours and level sets for the detection and tracking of moving objects. *IEEE Transactions on Pattern Analysis and Machine Intelligence.*, 22(3):266 –280, 2000.

[56] N. Paragios and R. Deriche. Geodesic active regions and level set methods for supervised texture segmentation. *International Journal of Computer Vision*, 46(3):223–247, 2002.

[57] Sangho Park. A hierarchical bayesian network for event recognition of human actions and interactions. In *Association For Computing Machinery Multimedia Systems Journal*, pages 164–179, 2004.

[58] L. R. Rabiner. A tutorial on hidden markov models and selected applications in speech recognition. *Readings in speech recognition*, pages 267–296, 1990.

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

[60] X. Ren and J. Malik. Tracking as repeated figure/ground segmentation. *IEEE Conference on Computer Vision and Pattern Recognition, 2007.*, pages 1 –8, 2007.

[61] Remi Ronfard. Region-based strategies for active contour models. *International Journal of Computer Vision*, 13:229–251, 1994.

[62] D. A. Ross, J. Lim, RS Lin, and MH Yang. Incremental learning for robust visual tracking. *International Journal of Computer Vision*, 77(1-3):125–141, 2008.

[63] H.A. Rowley, S. Baluja, and T. Kanade. Neural network-based face detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence.*, 20(1):23 –38, 1998.

[64] W. Rudin. *Real and Complex Analysis*. McGraw-Hill Education (India) Pvt Ltd, 2006.

[65] Jianbo S. and J. Malik. Normalized cuts and image segmentation. *IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 1997.*, pages 731 –737, 1997.

[66] Koichi Sato and J. K. Aggarwal. Temporal spatio-velocity transform and its application to tracking and interaction. *Journal of Computer Vision Image Understanding*, 96(2):100–128, 2004.

[67] C. Stauffer and W.E.L. Grimson. Adaptive background mixture models for real-time tracking. *IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 1999.*, 2:252 Vol. 2, 1999.

[68] Hai Tao, H. S. Sawhney, and R. Kumar. Object tracking with bayesian estimation of dynamic layer representations. *IEEE Transactions on Pattern Analysis and Machine Intelligence.*, 24(1):75–89, 2002.

[69] D. Terzopoulos and R. Szeliski. *Tracking with Kalman snakes.* MIT Press, Cambridge, MA, USA, 1993.

[70] P. Viola and M. Jones. Rapid object detection using a boosted cascade of simple features. *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2001.*, 1:I–511 – I–518 vol.1, 2001.

[71] A. Yilmaz. Object tracking by asymmetric kernel mean shift with automatic scale and orientation selection. *IEEE Conference on Computer Vision and Pattern Recognition, 2007.*, pages 1 –6, 2007.

[72] A. Yilmaz, O. Javed, and M. Shah. Object tracking: A survey. *ACM Comput. Surv.*, 38(4):13, 2006.

[73] A. Yilmaz, Xin Li, and M. Shah. Contour-based object tracking with occlusion handling in video acquired using mobile cameras. *IEEE Transactions on Pattern Analysis and Machine Intelligence.*, 26(11):1531 –1536, 2004.

[74] T. Yu, C. Zhang, M. Cohen, Y. Rui, and Y. Wu. Monocular video foreground/background segmentation by tracking spatial-color gaussian mixture models. *IEEE Workshop on Motion and Video Computing, 2007.*, pages 5 –5, 2007.

[75] Y. Zhaozheng and R.T. Collins. Shape constrained figure-ground segmentation and tracking. *IEEE Conference on Computer Vision and Pattern Recognition, 2009.*, pages 731 –738, 2009.

[76] S.C. Zhu, T.S. Lee, and A.L. Yuille. Region competition: unifying snakes, region growing, energy/bayes/mdl for multi-band image segmentation. *Fifth International Conference on Computer Vision, 1995.*, pages 416 –423, 1995.

[77] Z. Zivkovic and B. Krose. An em-like algorithm for color-histogram-based object tracking. *Proceedings of the 2004 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2004.*, 1:I–798 – I–803 Vol.1, 2004.