Stochastic geometry for automatic multiple object detection and tracking in remotely sensed high resolution image sequences

Paula Crăciun

UNIVERSITY OF NICE - SOPHIA ANTIPOLIS

DOCTORAL SCHOOL STIC

SCIENCES ET TECHNOLOGIES DE L'INFORMATION
ET DE LA COMMUNICATION

P H D T H E S I S

to obtain the title of

PhD of Science

of the University of Nice - Sophia Antipolis

Specialty : Signal and Image Processing

Defended by

Paula CRĂCIUN

Stochastic geometry for automatic object detection and tracking in remotely sensed image sequences

Thesis Advisor: Josiane ZERUBIA

prepared at INRIA Sophia Antipolis, AYIN Team

defended on November 25, 2015

Jury :

President:	Prof. Daniela ZAHARIE	-	West University of Timişoara (Romania)
Reviewers :	Prof. Alfred BRUCKSTEIN	-	Technion (Israel)
	Prof. Ba-Ngu VO	-	Curtin University (Australia)
Examinators :	Dr. Jacques BLANC-TALON	-	DGA (France)
	Dr. Xavier DESCOMBES	-	INRIA (France)
	Dr. Mathias ORTNER	-	Airbus Defence and Space (France)
	Prof. Michel SCHMITT	-	Institut Mines-Telecom (France)
Advisor :	Prof. Josiane ZERUBIA	-	INRIA (France)

Acknowledgments

First and foremost, I would like to express my special appreciation and thanks to my thesis adviser, prof. Josiane Zerubia, you have been a great mentor for me. I would like to thank you for encouraging my research, for lending support when I needed it and for allowing me to grow as a young scientist.
I would like to thank dr. Mathias Ortner from Airbus Defense and Space, France, your effective feedback and insightful ideas have helped me drive my research forward. Your active interest in my work and your enthusiasm have been very valuable to me.
I would like to thank the president of my oral defense committee and former college teacher, prof. Daniela Zaharie, for enabling my first contact with INRIA and the Ayin team. I am forever grateful. Thank you!
I would like to thank my reading committee members, prof. Alfred Bruckstein and prof. Ba-Ngu Vo for their time, interest and helpful comments. I would also like to thank the other three members of my oral defense committee, dr. Jacques Blanc-Talon, prof. Xavier Descombes and prof. Michel Schmitt for their time and interesting questions. I would like to thank my entire oral defense committee for making my defense an enjoyable moment.
I gratefully acknowledge the funding source that made my Ph.D. work possible. I was funded by Airbus Defense and Space, France, for the 3 years of my research.
Last but not least, I would like to thank my family for all their love and encouragements. A special thank you goes to my parents who raised me with a desire for scientific knowledge and supported me in all my pursuits regardless of how unconventional these might have been. Through your eyes I have seen myself as a capable, intelligent, young woman who could do anything once I have made up my mind. There are no words to describe my gratitude and appreciation for all you have done for me.

Contents

List of Tables

List of Figures

Chapter 1

Introduction

Contents

1.1 Motivation

The last decade has been a showcase for asymmetric warfare with demonstrations of highly organized attacks and advanced communication and planning methods [Kydd and Walter, 2006, Poland, 2010]. As a consequence, the civil, military and economic security is threatened by well-documented assassination attempts, hostage-taking and terrorist attacks. Popular methods such as social media analysis [Fuchs, 2009] or electronic eavesdropping [Landau, 2011] are not sufficient to prevent or detect ongoing attacks. But when these methods fail, mobile airborne and spaceborne surveillance can help to detect criminal activities before or during their execution.
Humans however are not the only ones who can benefit from overhead surveillance. In the last decade, hundreds of animal species have become endangered by excessive hunting or global warming. For example, this latter phenomenon led to an accelerated loss of Arctic sea ice during recent years [Comiso et al., 2008, Stroeve et al., 2012] which has a high impact on Arctic wildlife. Overhead surveillance can aid in assessing the trends in abundance of the Arctic species [Fretwell et al., 2012, Platonov et al., 2013] without concerns about intrusive study methods or human safety.
Satellites can be equipped with a variety of sensors such as acoustic, radar, ultrasonic or imaging sensors. Each sensor has advantages and drawbacks and the choice of which type of sensor to attach to a satellite depends on the specific applications for which that satellite is built. Satellite data has already been successfully used for search and rescue operations [Lukowski and Charbonneau, 2000, Jing and Danping, 2011], natural disaster relief [Voigt et al., 2007, Dell'Acqua et al., 2011, Bhangale

and Durbha, 2014] or environmental monitoring [Caccetta et al., 2011, Singh and Talwar, 2013]. Technological advances in the satellite industry are now enabling far more accurate and reliable imagery which has the potential to revolutionize the 21-st century surveillance and monitoring practices.
Regardless of the type of sensors mounted on the satellite, analyzing the acquired data is a difficult and tedious job for human operators characterized by high levels of fatigue and boredom [Garcia, 2007] due to the large amount of information available in the data. Appropriate algorithms for automatic and semi-automatic data processing and information retrieval can assist the human operator. Nevertheless, for most applications it is challenging to guarantee a low error rate and high confidence of an algorithm while also providing near real-time performance.
This thesis focuses on the analysis of static and dynamic images coming primarily from on-board very high resolution optical sensors. In particular, we are addressing two main problems: object detection in static images, respectively detection and tracking of moving objects in videos. Object detection and respectively tracking are two intermediary steps for automatic scene understanding. These complicated patterns of high-level information can then be used to model and detect suspicious behaviors and outliers which can be indicative of criminal behaviors. Image and video based methods provide valuable information as many properties of the detected objects such as position, size, appearance or shape can be directly derived from the data.
It is important to note that we use the term 'video' for a sequence of images taken at relatively short time intervals. The temporal frequency of the videos can range from approximately $1-2$Hz to 30Hz or more, depending on the type of sensor used. For example, a single low-orbit satellite (such as SkySat-1) can produce sub-meter resolution imagery and high-definition video capturing up to 90-second video clips at 30 frames per second. However, a large number of low-orbit satellites are needed to cover a given area for a long period of time. If this is the intent, then a geostationary satellite could provide an alternative to a constellation of low-orbit satellites. Geostationary satellites are Earth-synchronous, meaning that they rotate with the Earth (as opposed to low-orbit satellites which are Sun-synchronous) and can cover a larger area on ground. Consequently, such a satellite would produce very large images at a lower temporal frequency. This is why we consider a wide range of temporal frequencies for the videos considered in this thesis.
Satellite image analysis is not the only domain where human operators are overloaded by the amount of data available. Microscope image and video quality has experienced a large increase in recent years and the analysis of microscope data is now commonplace in fields such as medicine or biological research. The high acquisition speed of the cameras allows the real time observation of dynamic cellular and sub-cellular processes. As such, this approach can generate vast amounts of data that have to be stored, processed and analyzed. Automatic and semi-automatic methods for information retrieval are of great importance to the human operator in this field. Consequently, we also test the models developed in this thesis on synthetic and real biological image sequences. We are particularly interested in the detection

and tracking of sub-cellular structures called vesicles, which are responsible for the proper functioning of a living cell. An in-depth understanding of the dynamic and geometrical patterns of the vesicles and the cell in general, can lead to a deeper biological understanding and more adequate medical treatments.
Complicated geometrical patterns often require statistical analysis. Appropriate statistical tools and suitable mathematical models are necessary to analyze this data. Stochastic geometry is one of the areas of mathematical research which seeks to provide such methods and models. The modern theory of stochastic geometry pioneered by D.G. Kendall, K. Krickerberg and R.E. Miles examines random geometric patterns of complicated distributions [Stoyan et al., 1987, Stoyan and Stoyan, 1994, van Lieshout and Baddeley, 2002]. Spatial point processes and in particular marked point processes have already been successfully applied to object detection problems in very high resolution satellite imagery [Ortner, 2004, Perrin et al., 2005, Lacoste et al., 2005, Descamps et al., 2008, Ortner et al., 2008, Descombes et al., 2011].
This thesis focuses on the use of spatial and spatio-temporal marked point process models for object detection and tracking in various types of optical imagery. The aim is to examine if marked point processes can produce competitive results for these tasks. The underlying theoretical foundation of this thesis is based on the existing literature, but several novel ideas and approaches are introduced in order to adapt and improve existing methods with respect to object detection accuracy and to propose a new multiple object tracking approach based on marked point processes with a good tradeoff between tracking performance and run-time.

1.2 Challenges

The use of remotely sensed satellite video to detect and track objects is a challenging task. These challenges are encountered throughout the processing chain, from image acquisition to analysis. According to the occurrence time and place, they can be categorized as follows:

1. **Image / video acquisition**

    - **Sensor noise** represents a deviation from the optimal radiometric values of the image pixels. Depending on the type of sensor used, the noise can be modeled as either additive, multiplicative (speckle) or impulsive (salt-and-pepper) [Gonzalez and Woods, 2008];
    - **Artifacts** are artificial structures that are contained within the image and represent a perturbation of the signal. Examples of artifacts in this step include sensor saturation or A/D (analog/digital) conversion problems [Gonzalez, 2013].
    - **Blurred images** are a result of fast sensor / object motion and is prevalent in weak illumination settings which lead to long exposure times of the camera;

- **Weak contrast** appears due to environmental conditions. Specific weather conditions such as fog or clouds are common situations which lead to weak contrast in optical cameras;

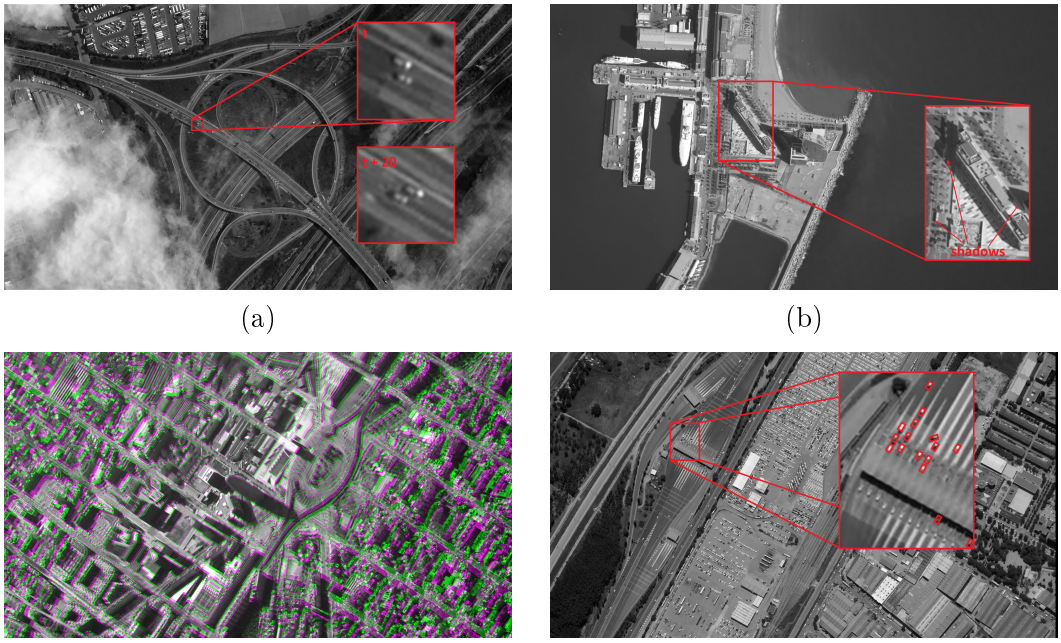
2. **Image / video transfer**

   - **Artifacts** during this step may be caused by a disturbed connection which can lead to strong artifacts or even missing images [Gonzalez, 2013];
   - **Compression-decompression artifacts** are block-like structures can significantly degrade the quality of the data [Netravali and Haskell, 1995, Antonini, 2003];

3. **Image / video analysis**

   - **Small object size** are a consequence of the ground sampling resolution of the sensors considered in this thesis. Objects are typically a few pixels in size (ranging between 10 - 100 pixels) which makes detection and recognition particularly difficult since the amount of information available on object appearance and shape is very limited. For example, urban scenes are characterized by a large number of targets located close to each other which generally results in two small, neighboring objects being detect as one large target. Vehicle traffic on a busy main road or boats in harbors are examples of such cases;
   - **Shadows** can be cast either by the targets themselves or by the surrounding objects. In an urban environment, shadows can be cast by large vehicles such as trucks and buses or by tall buildings. As a result, some targets could be missed or shadows can be identified as targets;
   - **Independent sensor and object motion** adds additional challenges to the detection of objects. Image registration and warping [Zitovà and Flusser, 2003] are used to compensate for the camera motion. Then, moving objects can be detected by their relative motion with respect to the stationary background;
   - **Time requirements** pose a great challenge to object tracking algorithms. In most applications, a trade-off between the quality of the results and the processing time has to be made. This problem becomes even more apparent in complicated scenes with large number of objects.

The challenges described above give a flavor of why the task of object detection in static images, respectively the detection and tracking of moving objects in videos is very difficult. This thesis is not intended to address all the challenges mentioned above. Recent developments in image denoising [Shao et al., 2014], image restoration [Portilla et al., 2003], image deblurring [Zhang et al., 2013] or temporal filtering [Müller and Müller, 2010] provide state-of-the-art methods to deal with most of the challenges during acquisition, transfer and initial processing of the satellite data. This thesis however, addresses nearly all the challenges of image / video analysis



(a) (b)

(c) (d)

Figure 1.1: One example for each indicated challenge of moving object detection and tracking with a moving satellite sensor (©Airbus D&S). (a) Small object size leads to limited appearance modeling. Individual object recognition is hard. Edges between the two objects are blurred. (b) Shadows cast by tall buildings and larger objects such as buses. Shadows owed to tall buildings increase the difficulty of object detection due to the lower contrast between the object and the background. Shadows owed to buses and trucks move along with the object and can be mistakenly identified as targets. (c) Camera motion accounts for large distortions between consecutive frames. (d)

presented above. The poor quality of the data is handled implicitly through the noise resistance incorporated into the algorithm. A visual representation of the challenges which arise during image / video analysis is presented in Figure B.1. Figure B.1 (a) shows two neighboring cars on the road. Modeling the appearance of the vehicles is difficult as there exists a limited amount of information. Furthermore, during the overtaking the cars are very close to each other which makes individual detection problematic as the boundaries between the vehicles become blurred.
Figure B.1 (b) illustrates the added difficulty caused by shadows. The shadow cast by a tall building could totally occlude an object passing through that area. Shadows can also be cast by large targets such as trucks. This can become a problem especially when multiple large targets are driving together in a group. The detection and tracking algorithm can misleadingly interpret the entire group as a single moving object.
Figure B.1 (c) shows the difficulty of detecting moving objects in the presence of

camera motion. The colors in the image show the camera displacement between two consecutive frames. Any motion detector that does not take the camera motion into account would produce a significant number of false alarms since static objects also appear to move due to the camera displacement.
Figure B.1 (d) shows an example of a dense urban area. The scene contains close to 100 vehicles. Each vehicle in the small window has been manually labeled with a red bounding box. The manual labeling is used as ground truth information to assess the performance of the proposed automatic detection and tracking algorithm. A time-efficient algorithm should significantly decrease the processing time required compared to a human operator, while maintaining a high accuracy level.
Several approaches have been developed in the past to meet the challenges presented above and are presented in Section 2. Nevertheless, there is a large potential to improve over the existing methods in terms of robustness and processing time.

1.3 Approach and proposed methods

The objective if this thesis is to study the use of geometric constraints in object detection and tracking methods through marked point processes. This approach enables the inclusion of two types of geometric constraints: the shape of the target can be enforced (as we are dealing mostly with man-made objects, their shape can usually be efficiently approximated using a simple parametric shape) and the geometric dispersion of the targets can be constrained.
Point processes are a class within the general domain of stochastic geometry and have already been successfully applied to object detection problems in satellite imagery. In the following, a short description of why these mathematical class is of interest in image / video processing is presented.

1.3.1 Probabilities in image / video analysis

In image analysis, marked point processes are a natural evolution of Markov fields as a result of the significant increase in the spatial resolution of the images. The reader interested in a detailed analysis of this evolution can refer to the HdR theses of Pérez [2003] and Descombes [2004].

1.3.1.1 The stochastic model of an image

An image is a collection of pixels. An image, denoted I, is a function which associates a gray value to each pixel u, $u \in K \subseteq \mathbb{Z}^2$:

$$\begin{aligned} I: \quad & K \to \mathbb{R} \\ & u \to I(u). \end{aligned}$$

This definition can easily be extended to a video by considering a 3-dimensional space such that for each pixel u, $u \in K \times T \subseteq \mathbb{Z}^3$, where K represents the support of an image and T is the time axis. For simplicity, we will only present the rationale

in the case of a 2-D image.
Let $(\Omega, \mathcal{A}, \mathbf{P})$ be a probability space, then a stochastic model of an image consists in considering the gray values of each pixel to be a realization of a stochastic version of the function I. I is now a random variable:

$$I: \quad \Omega \to \mathbb{R}^K. \tag{1.1}$$

A probability law $\mathbf{P}_I$ is associated to the random variable I such that $\mathbf{P}_I(A) = \mathbf{P}(I^{-1}(A))$. The simplest image model consists of all pixels being independent: $\mathbf{P}(I \in A) = \sum_{u \in K} \mathbf{P}(I(u) \in A_u)$.
From a physical point of view, an image can be considered as a noisy observation of an underlying phenomenon. An efficient way to capture this physical relation is to use a Bayesian framework to model the image.
A Bayesian model of an image can be written as:

$$\mathbf{P}(I = y \in A) = \sum_x \mathbf{P}(I = y \in A | X = x)\mathbf{P}(X = x) \tag{1.2}$$

where $\mathbf{P}(I = y \in A)$ is the marginal law of the observations, $\mathbf{P}(I = y \in A|X = x)$ is the law of the observation and $\mathbf{P}(X = x)$ is the apriori law which incorporates prior knowledge on the underlying phenomenon X. Using Bayes' theorem we can analyze the phenomenon X starting from observations $y_1, \cdots, y_n$ as follows:

$$\mathbf{P}(X = x|I = y) \propto \mathbf{P}(I = y|X = x)\mathbf{P}(X = x). \tag{1.3}$$

The Bayes model is interesting from a statistical point of view as it provides a class of natural estimators: the Bayes estimators [Marin, 2007, Gelman et al., 2014]. One such estimator is the Maximum A Posteriori (MAP) criterion which can be written as:

$$\hat{X}_{MAP} = \arg\max_x \prod_{i=1}^{n} \mathbf{P}(X = x|I_i = y_i) \tag{1.4}$$

when all observations $y_1, \cdots, y_n$ are independent.

1.3.1.2 Marked point processes for image analysis

A detailed description of point processes is given in Chapter 3. At this moment it is sufficient to note that a marked point process can be used to model random configurations of geometric shapes.
Let $X: \quad \mathbf{x} = \{x_1, \cdots, x_{n(\mathbf{x})}\}$, where $x_i, \quad i = \overline{1, n(\mathbf{x})}$ is a geometric object randomly placed in an image and $n(\mathbf{x})$ is the number of objects. Then, in a Bayesian framework, an observation model is given for example by:

$$\mathbf{P}(I(u) = y_u|X = \mathbf{x}) = \mathbb{1}(u \in \mathbf{x})\mathbf{P}(I(u) = y_u|u \in \mathbf{x}) + \mathbb{1}(u \notin \mathbf{x})\mathbf{P}(I(u) = y_u|u \notin \mathbf{x}). \tag{1.5}$$

In this examples, all pixels u belonging to any object of the configuration form the silhouette of the configuration. Perrin et al. [2005] proposed to use the radiometric

values of the pixels to determine whether they belong to the configuration or to the background. As such, the pixels belonging to the configuration were modeled as a Gaussian distribution with a high mean value, while the pixels belonging to the background were modeled as a Gaussian distribution with a lower mean value.
This Bayesian model encounters two difficulties in real images [Ben Hadj et al., 2010b]:

1. The background class is in general not homogeneous and to assume an Gaussian distribution for the entire background is not realistic for our problem. For example, consider an urban scene. If we want to detect cars on the road, it might be reasonable to assume that the area surrounding a single car (the road) follows a Gaussian distribution, while the entire urban area does not (as it contains shadows, buildings, trees, etc.);

2. The likelihood of the pixels which belong to the configuration does not take the morphological properties of the objects into account. For example, if we consider an image containing a bright, large rectangle and we are interested in detecting bright, small ellipses, the Bayesian model will fit as many ellipses as possible within the rectangle, which is not the desired result.

Stepping outside the Bayesian framework, these limitation have been surpassed by a detector approach in which the observation model is composed of local likelihoods for each object that is part of the configuration. This approach has enabled the use of marked point processes for various object detection tasks. Descamps et al. [2008] used a marked point process of ellipses to count the number of flamingos in large colonies containing thousands of such specimens. Perrin et al. [2005] developed a marked point process of circles for tree-crown extraction. His models could also identify the shadow cast by the tree and as such determine the position of the Sun when the image was taken. Ortner [2004] proposed a marked point process of rectangles for building detection. Alignment interactions between the rectangles were defined to reproduce the layout of the buildings. More recently, Ben Hadj et al. [2010a] introduced a marked point process of ellipses to count boats in harbors. Consequently, point process models have shown their potential in object detection.

1.3.2 Application to object detection and tracking in videos

The aim of this thesis is to study the possibility to use marked point process models for the detection and tracking of moving objects in video data. The detection and tracking of moving objects plays an important role in video analysis where it serves as a necessary step in performing high-level analysis, for instance terrestrial and maritime traffic monitoring. The problem of object detection in static images is also discussed in this thesis, with a strong emphasis on the particularly difficult problem of boat extraction in harbors.
The main data used in this these are very high resolution satellite images and videos which have been supplied by Airbus Defense and Space, France. The data sets vary

both in size and in the temporal frequency and range from short, low temporal frequency data sets (14 frames at $1-2$ Hz) to long, high temporal frequency videos (over 3000 frames at 30 Hz). The ground truth information for these data sets has been constructed by hand and validated afterward by an expert.
Secondary sets of data coming from the biological domain have also been analyzed in this thesis. These data sets have been provided by Pasteur Institute in Paris, France and Curie Institute in Paris, France. Furthermore, a free software named Icy developed by the Quantitative Analysis Unit from Pasteur Institute in Paris has been used to automatically create synthetic biological images and videos with the associate ground truth to assess the quality of the models proposed in this thesis.

1.4 Thesis organization and contributions

The thesis is organized as follows:

- **Chapter** 1 provides an overview of the thesis, describes the data used, outlines the motivation of this work and summarizes the contributions;
- **Chapter** 2 reviews state of the art approaches to the problems of object detection and object tracking.
- **Chapter** 3 presents point processes as a suitable framework for object detection, counting as well as tracking. State of the art parameter estimation and optimization techniques specifically adapted to this framework are described.
- **Chapter** 4 introduces the first important contributions of this thesis. It provides the model developed for boat detection and counting in harbor areas. The contributions of the chapter can be resumed as follows:
  - Section 4.1. presents an improved marked point process model developed for boat detection and compares it to the model previously introduced by Ben Hadj et al. [2010a];
  - Section 4.2. describes the parameter estimation techniques used to set the parameters of the model;
  - Section 4.3. proposes a new parallel implementation of the widely known RJMCMC optimization scheme.
- **Chapter** 5 introduces the main contribution of this thesis. In this chapter we describe a marked point process model specifically developed for multiple object tracking. The contributions of the chapter can be resumed as follows:
  - Section 5.1. introduces the new model and presents an in-depth description of the constraints and restrictions of this model;
  - Section 5.2. develops a novel linear programming approach to estimate the parameters of the model in the point process framework;

  - Section 5.3. proposes new perturbation kernels specifically adjusted for the problem of multiple object tracking to be used in the RJMCMC optimization scheme.

- **Chapter** 6 concludes the thesis and outlines the perspectives of this work.

1.5 Publications

The following papers have been published or submitted based on works contained in this manuscript:

- **Journal papers**:
  - P. Crăciun, M. Ortner, and J. Zerubia. A spatio-temporal marked point process model for joint detection and tracking of moving objects. Submitted to *IEEE Transactions on Image Processing*, 2015;
  - P. Crăciun and J. Zerubia. Unsupervised marked point process model for boat extraction and counting in harbors from high resolution optical remotely sensed images. In *Revue Francaise de Photogrammétrie et Télédétection*, vol. 207, pp. 33-44, 2014;
- **National and International conferences**:
  - P. Crăciun, M. Ortner, and J. Zerubia. Submitted to IEEE Computer Vision and Pattern Recognition, USA, 2016;
  - P. Crăciun and J. Zerubia. Submitted to IEEE International Symposium on Biomedical Imaging, Czech Republic, 2016;
  - P. Crăciun, M. Ortner, and J. Zerubia. Joint detection and tracking of moving objects using spatio-temporal marked point processes. In *IEEE Winter Conference on Applications of Computer Vision*, USA, 2015;
  - P. Crăciun and J. Zerubia. Towards efficient simulation of marked point process models for boat extraction from high resolution optical remotely sensed images. In *International Geoscience and Remote Sensing Symposium*, Canada, 2014;
  - P. Crăciun and J. Zerubia. Unsupervised marked point process model for boat extraction in harbors from high resolution optical remotely sensed image. In *International Conference in Image Processing*, Australia, 2013;
  - P. Crăciun, M. Ortner, and J. Zerubia. Integrating RJMCMC and Kalman Filters for multiple object tracking. In *GRETSI - Traitement du Signal et des Images*, France, 2015;
  - P. Crăciun and J. Zerubia. Boat extraction in harbors from high resolution satellite images using mathematical morphology and marked point processes. In *GRETSI - Traitement du Signal et des Images*, France, 2013.

Chapter 2

State of the art

Contents

This chapter covers related work on object detection and tracking in high resolution remotely sensed images applied to similar data sets and facing similar challenges as in this thesis. The focus of the literature review will be on satellite and aerial imagery and the considered tasks will include both detection in static images as well as detection and tracking in videos.
The image and video data considered in the literature under review are coming both from low-orbit satellites (e.g. Pleiades, GeoEye, SkySat) but also from UAVs or airplanes flying at different altitudes. The camera angle also varies between top-view [Lavigne et al., 2010, Xiao et al., 2010, Luo et al., 2012] and oblique view [Cao et al., 2011, Cheraghi and Sheikh, 2012, Siam and ElHelw, 2012] for detection and wide area surveillance [Perera et al., 2006, Reily et al., 2010, Saleemi and Shah, 2013]. Research work on satellite videos is only in its infancy since the technological advances to capture video data (whether low or high temporal frequency data) have been very recent. However, top view high altitude aerial data looks very similar to satellite data, which is why we include it in our literature review.
We start this chapter by reviewing object detection methods in static high resolution satellite images and then turn our attention to object detection methods for video data before switching to object tracking.

2.1 Object detection

In object detection, the image is usually scanned for objects of a certain class (such as boats) and all positions in the image where matches are found are marked by bounding boxes [Szelisky, 2011]. In this thesis, we address both the problem of object detection in static images, as well as in dynamic image sequences. In dynamic

sequences, the problem of object detection refers to objects that are generally moving, which enables the use of specific techniques for identifying regions of interest that are not available in the static case. Thus, we divide the object detection algorithms accordingly in two parts. The first part of this section discusses object detection methods for static images, whereas the second part presents object detection methods for dynamic sequences.

2.1.1 Object detection in static images

To identify objects in remotely sensed imagery one can usually rely on a mix of spectral and geometrical information. Spectral information includes color cues or intensity edges, while geometrical information refers to the shape of the object. Detections are obtained using color based segmentation algorithms and clustering of local features such as corners and edges. It is important to note that object detection in high resolution remote sensing imagery has different characteristics from that in ground imagery: objects are generally small relative to the ground sampling distance; the background is cluttered; rotation invariance is required.
A common method for object detection is the sliding window approach [Papageorgiou and Poggio, 2000, Wei and Tao, 2010] which has been successfully applied to human detection [Dalal and Triggs, 2005] and face detection [Viola and Jones, 2004]. A sliding window is shifted across the entire image. At each position, a series of features are computed and a classifier is used to return a decision value which represents the degree of certainty that the image area inside the window contains an object. In order to detect objects of different scales either the size of the window is varied, or the image is rescaled between the minimum and the maximum expected size of the object. The naive approach is to vary the size of the window with one separately trained classifier model for each size and no image rescaling. This approach is very time consuming and requires many training samples. The alternative approach is to rescale the image at n different scales and use a fixed window size with a single classifier model [Dalal and Triggs, 2005, Dollar et al., 2009]. After obtaining all decision values, objects are detected by applying Non-Maximum Suppression (NMS) to these values and using a minimum classifier uncertainty threshold. By modeling the object in its entirety, a holistic object representation is used in the sliding window approach. A holistic representation is usually desired when small objects have to be detected.
The sliding window approach has been extensively used in UAV aerial images for vehicle detection. Nguyen et al. [2007] use sliding windows with Haar features and orientation histograms combined with local binary patterns as vehicle descriptors and AdaBoost [Freund and Schapire, 1997] for classification. Gaszczak et al. [2011] only combine sliding windows with Haar features and cascaded AdaBoost for vehicle detection, which is very similar to the Viola and Jones [2001] approach. As vehicles could have different orientations, four discrete orientations are specified with one classifier trained for each orientation. Turmer et al. [2013] apply a sliding window with Histogram of Oriented Gradient (HoG) features and Support Vector Machines

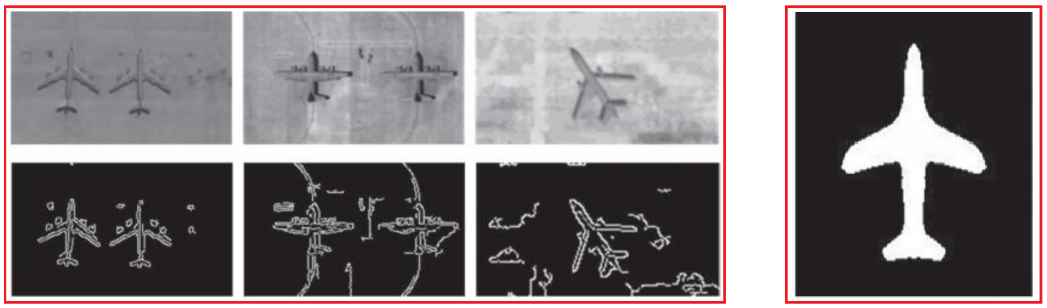


Figure 2.1: (a) Sample images of airplanes in aerial and satellite images and the edge detection results obtained using a Canny-Deriche edge detector. Image by Praveena et al. [2015]; (b) Holistic template of an airplane. A sliding window approach can be used to detect airplanes in aerial images.

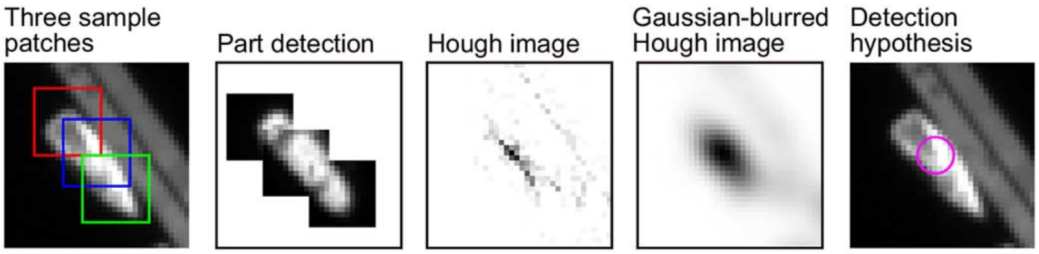


Figure 2.2: Part-based object detection procedure. First, a sliding window search is used to extract patch images. Next, parts are detected via a sparse representation of the patches. Hough voting is performed using offsets of detected parts of the target object and finally, the maxima are found in the Hough image. These maxima represent the possible targets. Image by Yokoya and Iwasaki [2015]

(SVM) to find vehicles. Finally, Proia [2010] uses a sliding window approach to detect boats in the open sea in satellite imagery. Figure 2.1 (a) shows an example feature (e.g. intensity edges) that can be used to detect airplanes. Figure 2.1 (b) shows a holistic airplane template which has been constructed from the geometric features of the airplane.

When objects are sufficiently large, part-based models can also be employed. The idea behind part-based models is that each part provides only local visual properties of the object while the geometrical arrangement of these parts is characterized by certain connections between pairs of parts. Psychological and physiological evidence for part-based representations in the brain [Logothetis and Sheinberg, 1996] as well as certain computational theories of object detection [Biederman, 1987, Ullman, 1996] support this representation. An important advantage of part-based models is their robustness to partial occlusions.

Agarwal and Roth [2002] proposed an approach for learning a sparse, part-based

representation of the object and showed its robustness to partial occlusions and background variation. Leibe et al. [2008] proposed to learn a class-specific implicit shape model (ISM) which detects the local appearances of class objects in accordance to a dictionary and then uses Hough transform to localize the objects by considering their spatial consistency. Zhang et al. [2014] propose a rotation invariant feature by extending HoG to encode the features of rotated parts and then cluster the parts based on their distance from each other. Yokoya and Iwasaki [2015] detect object parts by means of a class-specific sparse image representation of patches using learned object and background dictionaries and Hough voting to spatially integrate the co-occurrence of the parts, which enables object detection. Figure 2.2 shows the detection work-flow used by Yokoya and Iwasaki [2015].
Marked point process models have received an increased attention in the last decade for their ability to cope with a large number of objects in large data sets. Thus, such models have been applied to a wide variety of object detection problems. Perrin et al. [2005] model tree-crowns as a realization of a marked point process of ellipses, where the point process represents the locations of the trees and the mark their geometric features and determines the number of stems, their position and their size. Lacoste et al. [2005] tackle the problem of unsupervised line network extraction such as roads or rivers in satellite images. They design a marked point process of line segments and introduce a prior model that exploits the topological properties of the networks considered. Ortner [2004] detects buildings in dense urban areas using a marked point process of rectangles. Descamps et al. [2008] use a marked point process model of ellipses to detect flamingos in high resolution satellite images. The number of birds in the population is counted. Color information is used for bird detection. Finally, a first model for boat extraction has been proposed by Ben Hadj et al. [2010a]. They specifically address the problem of boat detection in harbors, which is very difficult due to the distribution of the objects. These models are of particular interest throughout this thesis and the underlying theoretical framework will be discussed in the next chapter.
A reliable appearance-based detection of objects remains nevertheless a challenging task in aerial and satellite images [Prokaj and Medioni, 2014], especially when the objects are very small. As opposed to static imagery, dynamic images (or videos) offer additional temporal information that can be exploited for object detection tasks in the case of moving objects.

2.1.2 Object detection in dynamic images

Although stationary objects still have to be detected based on their appearance (this case was described in the previous section), moving objects can be detected based on their motion. However, before moving objects can be detected and tracked, the camera motion has to be compensated for, since not only the objects, but the entire scene is moving in aerial videos. Nevertheless, camera motion compensation exceeds the scope of this thesis and thus, will not be discussed. The interested reader can refer to [Teutsch, 2015] for a global overview of camera motion compensation

methods.
Once the camera motion has been compensated for, independent motion can be detected (that is, motion that is not dependent on the camera motion). Image differencing, background learning or clustering of moving local features are examples of independent motion detection methods.
Difference images are the most common approach [Xiao et al., 2010, Cao et al., 2011, Cheraghi and Sheikh, 2012, Saleemi and Shah, 2013] and consists in computing the intensity value difference D at a pixel (x, y) in the overlapping area A_o of two registered images I_1 and I_2:

$$D(x,y) = \begin{cases} |I_1(x,y) - I_2(x,y)|, & \text{if } (x,y) \in A_o \\ 0 \quad \text{otherwise} \end{cases} \tag{2.1}$$

Strong local appearance changes caused either by moving objects or imprecise image registration are characterized by high difference values. The number of frames needed to ensure a reliable result depends on the velocity of the moving object, as well as on the sensor frame rate. For example, two consecutive images are enough for low frame-rate sensors (e.g. $1-2$Hz), as the moving object produces prominent motion blobs in the difference image [Saleemi and Shah, 2013]. In high frame-rate sensors (e.g. ≥ 25Hz), prominent blobs can be obtained by considering only every n-th image for computing the difference image [Shastry and Schozengerdt, 2005]. However, this approach can lead to an effect known in literature as *ghosting*. Ghosting means that each moving object produces two motion blobs in the difference image, one corresponding to its position in frame I_1 and a distinct one corresponding to its position in frame I_2. This effect can be handled by computing the difference image between three consecutive frames [Xiao et al., 2010, Keck et al., 2013]:

$$D(x,y) = \begin{cases} min(|I_1(x,y) - I_2(x,y)|, |I_2(x,y) - I_3(x,y)|), & \text{if } (x,y) \in A_o \\ 0, \text{ otherwise} \end{cases} \tag{2.2}$$

Background learning and foreground segmentation has been originally designed for static cameras [Piccardi, 2004, Bouwmans, 2011]. Background learning and subtraction is effective when a high amount of images with a large overlapping area and small number of moving objects are available. This is generally not the case in aerial images, since scenes may contain hundreds of moving objects and the camera is also moving such that only a limited number of frames with large overlapping areas can be extracted. Hence, well-known background modeling methods such as the Stauffer-Grimson [Stauffer and Grimson, 1999] stochastic background modeling are difficult to apply. Nevertheless, Perera et al. [2006] uses 30 successive frames for Stauffer-Grimson background modeling in the Red Green Blue (RGB) color space, while Jones et al. [2005] use a similar approach in the Hue Saturation Value (HSV) color space.
Motion blobs are identified by registering the current frame I and the learned background BG and then perform a pixel-wise subtraction, as in the case of difference

images:

$$D(x,y) = \begin{cases} |I(x,y) - BG(x,y)|, & \text{if } (x,y) \in A_o \\ 0, & \text{otherwise} \end{cases} \tag{2.3}$$

As the background learning can only be performed in the overlapping area of consecutive frames, it is important to note that the more frames are used, the smaller the background model will be. Background subtraction can be used in addition to difference images to detect stopping objects [Xiao et al., 2010].
Clustering of local motion features is completely different from difference image computation or background learning. If the latter two are typical choices for a Detect-Before-Track (DBT) tracking algorithm [Liu et al., 2005, Bugeau and Perez, 2008], local features clustering is a choice for the Track-Before-Detect (TBD) tracking approach [Davey et al., 2008, Taj and Cavallaro, 2009]. TBD is the notion used for joint detection and tracking of objects and was initially used for radar data, where the presence of noise is very strong [Ristic et al., 2004]. This is also the reason why TBD is used in aerial videos where the object motion accounts only for a small part of the entire motion of the scene [Hoseinnezhad et al., 2012, Papi et al., 2015]. The idea behind this method is that a moving object is expected to produce several non-sparse moving local features and the motion areas can be extracted by clustering these features based on additional spatial and motion constraints [Luo et al., 2012, Siam and ElHelw, 2012]. This approach has some advantages over difference images as it avoids ghosting effects but comes at a higher computational cost.
Identifying independent motion areas does not necessarily imply the detection of single moving objects. Ideally, one motion area corresponds to a single moving object. Nevertheless, a single motion area can contain either just a part of the object, multiple objects or even no object at all (e.g. noise). Independent motion detection can be regarded as a reduction of the search space, since only a few small regions need to be further processed. Reducing the search space is very important as it reduces both the computational time needed to process the video and the number of false alarms.
Once the moving blobs have been obtained, additional information such as size, shape and eccentricity can be used to classify the blob as an object. According to Teutsch [2015], most commonly appearance features such as color, shape and texture are learned and stored in a model which is then applied by classifiers such as SVM [Vapnik, 1998], Random Forests [Breiman, 2001] or boosting [Freund and Schapire, 1997]. Model knowledge such as blob motion [Cheraghi and Sheikh, 2012], blob shape and size [Zheng et al., 2013] or shape template matching [Tanaka and Saji, 2007] is used to determine the detections. Finally, sliding window approaches or part-based models can be used for detecting moving objects starting from motion blobs.

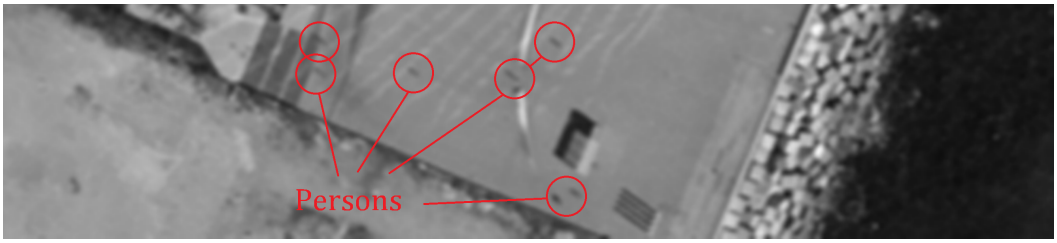


Figure 2.3: Appearance of people in top view aerial and satellite images.

2.1.3 A note on person detection

Only a few authors specifically target people detection in top-view high altitude aerial imagery. The main reason is that with a top view camera (which is the main case considered in this thesis) people can hardly be detected and recognized, as they not only appear very small (around $5 - 10$ pixels) but can easily be confused with shadows. Figure 2.3 shows a visualization of multiple persons in aerial and satellite images.
Iwashita et al. [2010] propose an approach to person detection in aerial images based on their shadows. However, shadow based approaches are useful only in the context of good lightning and weather conditions. A shadow silhouette normalization is performed using metadata such as time, camera position and sun position. Shadow features are computed and analyzed to identify the presence of a person. Reily et al. [2010] produce initial detections based on gradients, geometric constrains and object-shadow relationships and then computes Haar features for SVM classification with a Radial Basis Function kernel.
It is important to note that people detection in the context of aerial imagery is completely different from people detection in ground imagery and most of the extensive research in the latter case is not applicable for aerial imagery.

2.2 Object tracking

An object tracker is a system that generates trajectories of the objects over time, by determining their positions in every frame of the video. Ideally, an object tracker has the following characteristics:

- It should detect all objects that enter or move in the scene;
- It should differentiate between different objects that are present in the scene;
- It should assign an unique label to each object and maintain it throughout the sequence;
- It should handle partial or even total occlusion without changing the label of the object;
- It should cope with scenarios where the motion of the object changes.

A good survey on object tracking methods is provided by Yilmaz et al. [2006], while Smeulders et al. [2013] gives a more recent survey including a comprehensive experimental study.
The complexity of the object tracking methods depends on the complexity of the data. Simple tracking methods can be used in cases where only $10 - 20$ objects are present in the scene and a small number of split and merge situations occur [Teutsch, 2015]. Otherwise, one must employ more complicated tracking methods that handle split and merge situations, object occlusion and various motion models. If point tracking is desired, extended targets can be converted to point representations [Jones et al., 2005]. A combination of the position of the center of the object and its velocity is commonly used for point tracking [Perera et al., 2006, Saleemi and Shah, 2013]. Extended targets are usually described by bounding boxes [Li et al., 2009, Wu et al., 2010], ellipses [Wu et al., 2009] or blobs [Ibrahim et al., 2010]. Contour tracking is a less popular representation for object tracking in aerial images, as the targets are man-made objects with a rigid shape that can be usually represented by rectangles (for vehicles) or ellipses (for boats).
An important challenge in tracking multiple extended objects is the association of the detections to existing trajectories [Challa et al., 2011]. The detections can be split or merged which yields a necessity for increased performance of the tracking method. An object is split if two or more detections correspond to the same object, while reversely, objects are merged if a single detection corresponds to more than a single object.

2.2.1 Data association based approaches

Different data association methods can be employed depending on the complexity of the scene. The data association problem for point targets can be solved using the nearest neighbor [Perera et al., 2006], Multiple Hypothesis Tracking [Saleemi and Shah, 2013], Joint Probabilistic Data Association Filter [Kang et al., 2005, Wu et al., 2009] or data-driven MCMC-based data association Yu and Medioni [2008].
The key principle of Multiple Hypothesis Tracking (MHT) is that difficult data association decisions are deferred until more data is received. Hypotheses are propagated into the future relying on the assumption that subsequent data will resolve the uncertainty.
The Joint Probabilistic Data Association Filter (JPDAF) associates all measurements at a time t with all tracks. The measurement-to-target association probabilities are computed jointly over all targets. Based on the Markov property, the JPDAF computes these association probabilities only for the latest set of measurements. Finally, the state estimation is done either separately for each target, or in a coupled manner using a stacked state vector.
The data-driven MCMC-based data association (DD-MCMCDA) is a combinatorial optimization approach in which the enumeration of tracks is replaced by MCMC sampling methods, such as Metropolis-Hastings or Gibbs sampling. The DD-MCMCDA bases its decision to form a track on current and passed observations.

Oh et al. [2004] propose a real-time algorithm and show that this approach exhibits an increased performance compared to MHT and JPDAF under extreme conditions such as dense environments with large number of targets, high false alarm rates and low detection probabilities. This work has been improved upon by Yu et al. [2007] and Yu and Medioni [2008]. Yu et al. [2007] explicitly use the spatio-temporal smoothness in motion and appearance for their DD-MCMCDA implementation, while Yu and Medioni [2008] provide an integrated detection and tracking approach for multiple moving objects.
For extended objects, data association can be solved by computing the amount of overlap between bounding boxes in consecutive frames [Siam and ElHelw, 2012]. Other approaches for solving data association can be performed based on object re-identification using similarity measures or graph matching. Ibrahim et al. [2010] use blob centroid, area, eccentricity and color information to represent objects and then computer similarities measure for these blob features to solve the measurement-to-track association problem. Yao et al. [2008] evaluate spatial and color similarity. Xiao et al. [2010] use a graph framework to model color, shape, appearance and spatial features and associate detections with existing tracks through graph matching. Prokaj and Medioni [2014] use the position of the detection and its appearance to generate tracklets which they then combine to object tracks.
Once the data association problem has been resolved, optimal filter methods (such as the Bayes filters) are usually applied to update an existing track with an associated detection. The single target Bayes filter consists of two steps, usually called prediction and update. Suppose $Z_{1:t-1} = \{z_1, \cdots, z_{t-1}\}$ are the measurements at time $t-1$ and the posterior distribution is $p_{t-1|t-1}(x|Z_{1:t-1})$, then a prediction can be made for time t as follows:

$$p_{t|t-1}(x|Z_{1:t-1}) = \int f_{t|t-1}(x_t|x_{t-1}) p_{t-1|t-1}(x_{t-1}|Z_{1:t-1}) dx_{t-1}, \tag{2.4}$$

where $f_{t|t-1}(x_t|x_{t-1})$ is the Markov state transition model of the target state. Once a measurement z_t is obtained at time t, the posterior distribution can be updated as follows:

$$p_{t|t}(x|Z_{1:t}) = \frac{g_t(z_t|x_t) p_{t|t-1}(x|Z_{1:t-1})}{\int g_t(z_t|x_t) p_{t|t-1}(x|Z_{1:t-1}) dx}, \tag{2.5}$$

where $g_t(z_t|x)$ is the likelihood function at time t. Several implementations of the Bayes filter have been devised. The most popular are Kalman filtering Kalman [1960] and particle filtering Kitagawa [1987]. Perera et al. [2006] and Siam and ElHelw [2012] use Kalman filters to update the tracks for point targets, while for extended targets Li et al. [2009] and Shen et al. [2013] use mean shift for object representations that include motion, appearance and shape.

2.2.2 Finite Set Statistics based approaches

A different framework for multiple object tracking is represented by Random Finite Sets (RFS). A RFS can be described as a finite, set-valued random variable, in which

both the elements and the cardinality of the set are random. Thus, a realization of a RFS is a finite, unordered set of elements distributed according to a common distribution. This approach does not require any data association computations, since it directly propagates the posterior intensity of the RFS of objects in time. A detailed description of RFS and Finite Set Statistics (FISST) can be found in [Goutsias et al., 2012]. The multiple object tracking problem can be posed as a Bayesian filtering problem if the observations and objects are modeled as a single meta-observation and meta-state.
The optimal multiple object Bayes filter can be again divided into prediction and update step. Suppose that the multiple target posterior density at time $t-1$ is $p_{t-1|t-1}(X|Z_{1:t-1})$ and the cumulative measurements $Z_{1:t-1} = \{Z_1, \cdots, Z_{t-1}\}$ are known, then a prediction of the multiple target posterior density $p_{t|t-1}(X|Z_{1:t-1})$ can be made as follows:

$$p_{t|t-1}(X|Z_{1:t-1}) = \int f_{t|t-1}(X|X_{t-1}) p_{t-1|t-1}(X_{t-1|Z_{1:t-1}}) \mu_s(dX_{t-1}), \tag{2.6}$$

where $f_{t|t-1}(X|X_{t-1})$ is the Markov state transition function and $\mu_s(\cdot)$ is the dominating measure of the state space. Once the measurements Z_t at time t are obtained, the posterior can be updated as follows:

$$p_{t|t}(X|Z_{1:t}) = \frac{g_t(Z_t|X) p_{t|t-1}(X|Z_{1:t-1})}{\int g_t(Z_t|X) p_{t|t-1}(X|Z_{1:t-1}) \mu_s(dX_t)}, \tag{2.7}$$

where $g_t(Z_t|X)$ is the multiple object likelihood function.
It can be observed that the only difference to the single-target Bayes filter lies in the fact that X_t and Z_t are RFS with variable dimension that depends on t.
The complexity of the multiple object Bayes filter grows exponentially with the number of objects. Thus, there is a need for approximate solutions.
The Probability Hypothesis Density (PHD) filter is the cheapest tractable approximation of the optimal multiple-object Bayes filter proposed by Mahler [2003] that is based on the RFS framework. The PHD filter is based on the following assumptions:

- Targets evolve and generate measurements independently from each other;
- The birth RFS is independent from the current or surviving targets and is considered to be Poisson;
- Noise is a Poisson RFS, independent from the measurements RFS;
- The predicted and posterior multiple object RFS are approximated as Poisson RFSs.

If we denote the intensity functions associated to the predicted and the posterior multiple object state, $\gamma_{t|t-1}(x)$ and $\gamma_{t|t}(x)$ respectively, we can write the predicted state as follows:

$$\gamma_{t|t-1}(x) = \int \left[p_{s,t}(u) f_{t|t-1}(x|u) + b_{t|t-1}(x|u) \right] \gamma_{t-1|t-1}(u) du + \mu_t(x), \tag{2.8}$$

where:

- $p_{s,t}(u)$ is the survival probability of an object with state u;
- $f_{t|t-1}(x|u)$ is the evolution density of a single object at time t;
- $b_{t|t-1}(x|u)$ is the intensity of the point process corresponding to the new objects generated by the object in the state u at time t;
- $\mu_t(x)$ is the birth intensity function of new objects at time t.

The posterior state can be written as:

$$\gamma_{t|t}(x) = (1 - p_{d,t}(x))\gamma_{t|t-1}(x) + \sum_{z \in Z_t} \frac{p_{d,t}(x) g_t(z|x) \gamma_{t|t-1}(x)}{h_t(z) + \int p_{d,t}(u) g_t(z|u) \gamma_{t|t-1}(u) du}, \quad (2.9)$$

where $p_{d,t}(x)$ is the detection probability, $g_t(z|x)$ is the likelihood function of a single object and h_t is the noise intensity at time t.
It can be observed that the posterior involves multiple integrals that in general do not have a closed form solution. Several implementation solutions have been developed for the PHD filter. The most popular use either sequential Monte Carlo methods, as proposed by Zajic and Mahler [2003], Vo et al. [2003] and Vo et al. [2005], or a Gaussian mixture devised by Vo and Ma [2006] and Vo et al. [2006]. Under simplifying assumptions, the latter implementation greatly improves the computational efficiency of the PHD filter. For a detailed description of PHD filters, the interested reader can refer to the work of Pace [2011].
The PHD filter propagates the first-order multiple target moment. In their paper, Erdinc et al. [2006] argue for the need for a PHD-type filter which should be of first-order in the states of individual targets but of a higher order in the number of states. This led to the development of the so-called cardinalized PHD (C-PHD) filter by Mahler [2007b] which propagates not only the first-order multiple target moment, but also the entire probability distribution on the number of targets. In addition to the PHD and the C-PHD filters, a Multi-Target Multi-Bernoulli (MeMBer) recursion was also proposed by Mahler [2007a, 2014]. Under low clutter densities assumptions, this recursion is a tractable approximation to the Bayes multi-target recursion which propagates the multi-target posterior density, as compared to the PHD and C-PHD filters which only propagate moments and cardinality distributions. Vo et al. [2009] show that the MeMBer has a significant bias in the number of targets. To overcome this bias, Vo et al. [2009] propose a Cardinality-Balanced Multi-Target Multi-Bernoulli (CBMeMBer) filter which derives the cardinality bias produced in the data update step of MeMBer and uses it to produce an unbiased update.
All the above-mentioned filters assume statistical independence between objects. Nevertheless, multiple object tracking approaches such as the MHT or the JPDAF are able to model the dependence between objects. This led to recent attempts to derive a RFS-based approximation able to capture the statistical dependence between object. Indeed, Papi et al. [2015] propose a Generalized Labeled Multi-Bernoulli approximation of multi-object densities that can capture the statistical

dependence of the objects.

To conclude, in this chapter, we have discussed the most common approaches to solve the object detection and tracking problem in aerial and satellite data. Research in this domain has rapidly evolved in the last decade and the domain itself is relatively new, as the technological advances that enabled the acquisition of such data are recent.
A major contribution of this thesis is a novel detection and tracking algorithm based on marked point processes, which will be described in Chapter 5. In the following chapter, we delve into the details of the marked point process framework and show how the multiple object detection and tracking problem can be posed within this framework.

Chapter 3

Using Point processes for object detection

Contents

In the previous chapter, a broad overview of current techniques to detect objects in static and dynamic image sequences and to track them in time has been presented. Nevertheless, in the last decade, marked point processes have received great attention for object detection in large data sets, with emphasis in biological imagery and remotely sensed satellite data.
This chapter provides a short description of the theoretical framework of point processes in general and the necessary ingredients to construct a fully automated object detection model are discussed. The chapter is divided into three parts: the point process fundamentals are presented in the beginning of the chapter. As such models generally depend on a set of hyper-parameters which are difficult to be set manually, parameters estimation techniques are presented next. Finally, traditional optimization procedures and recent developments and improvements thereof, are discussed at the end of this chapter.

3.1 Fundamentals

If we consider the image support to be K, with $K \subset \mathbb{R}^d$, then a configuration of points $\mathbf{x}$ is a finite unordered set of points in K. The configuration space, Ω, can then be written as:

$$\Omega = \bigcup_{n \in \mathbb{N}} \Omega_n, \tag{3.1}$$

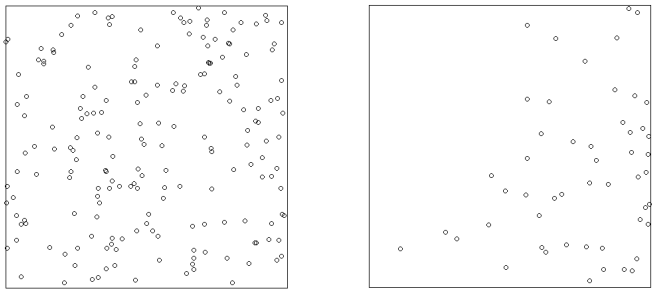


Figure 3.1: Simulation of a Poisson point process. (a) Homogeneous Poisson point process; (b) Inhomogeneous Poisson point process.

where $\Omega_0 = \emptyset$ and $\Omega_n = \{\{x_1, \cdots x_n\}, x_i \in K, \forall i\}$ are the set of configurations with no points and n unordered points, respectively.
The Lebesgue measure on K, denoted $\Lambda(K)$, is used to measure the configuration space as follows:

$$\nu(\Omega) = \sum_{n=0}^{\infty} \nu(\Omega_n) = \sum_{n=0}^{\infty} \frac{\Lambda(K)^n}{n!} = \exp^{\Lambda(K)}. \tag{3.2}$$

A configuration is called locally finite if it places at most a finite number of points in any bounded Borel set $A \subseteq K$. The family of all locally finite configurations is denoted by $N^{lf} = N_K^{lf}$.
A point process can be defined as:

Definition 3.1.1 (Point process) *A point process on K is a mapping X from the probability space $(\Omega, \mathcal{A}, P)$ into N^{lf} such that for all bounded Borel sets $A \subseteq K$, the number $N(A) = N_X(A)$ of points falling in A is a (finite) random variable.*

In other words, a point process refers to any random variable whose realizations are random configurations of points. The most well-known point process is the Poisson point process.

Definition 3.1.2 (Poisson point process) *A point process X defined on K, with intensity measure μ and intensity function ν, is a Poisson point process if:*

- *$N(A)$ is Poisson distributed with mean $\mu(A)$;*
- *conditional on $N(A)$, the points in X_A are i.i.d. with the density proportional to $\nu(u)$, $u \in A$.*

The Poisson process is called *homogeneous* if $\nu(u)$ is constant for all $u \in K$. The Poisson process is a model of complete spatial randomness or no interaction, since

X_A and X_B are independent whenever $A, B \in K$ are disjoint. Figure 3.1 shows the realization of a homogeneous (a) and a non-homogeneous (b) Poisson point process. For every Borel set A, the probability measure $\pi_\nu(A)$ associated with a Poisson process is given by:

$$\pi_\nu(A) = \exp^{-\nu(K)} \left(\mathbb{1}(\emptyset \in A) + \sum_{n=1}^{\infty} \frac{\pi_{\nu_n}(A)}{n!} \right) \tag{3.3}$$

with

$$\pi_{\nu_n}(A) = \int_K \cdots \int_K \mathbb{1}(\{x_1, \cdots, x_n\} \in A_n) \nu(dx_1) \cdots \nu(dx_n) \tag{3.4}$$

where A_n is the subset of configurations in A that contains exactly n points and $\mathbb{1}_{[\cdot]}$ is the indicator function ($\mathbb{1}(true) = 1$; $\mathbb{1}(false) = 0$).

The intensity function $\nu(\cdot)$ of the Poisson process can be used to control the point density of the configurations. Nevertheless, no correlation between points, nor constraints on their relative positions, can be enforced using this simplistic model. To model such constraints, a point process can be defined by a density function w.r.t. the Poisson measure.

Definition 3.1.3 (Density of a point process) *The probability density, f, of a point process w.r.t. the $\pi_\nu(\cdot)$ law of a Poisson process is a mapping from the configuration space Ω into $[0, \infty[$ such that:*

$$f : \Omega \to [0, \infty[, \int_\Omega f(\mathbf{x}) d\pi_\nu(\mathbf{x}) = 1. \tag{3.5}$$

For every Borel set A in Ω, the probability measure on Ω that defines a point process is defined by $P(A) = \int_A f(\mathbf{x}) d\pi_\nu(\mathbf{x})$.
An equivalence can be drawn between the probability density, f, of a point process and the belief density p_Ξ of a random finite set, Ξ, by interpreting the latter as the Radon-Nikodyn derivative of the corresponding distribution P_Ξ w.r.t. the dominating measure π_ν.
By introducing the density function w.r.t. the measure of a Poisson process, interactions between points can now be efficiently modeled. However, long-range correlations are heavy to implement. The Markov property is very useful in handling this implementation detail as it restricts the long-range correlations to a local neighborhood. In order to define the Markov property for point processes, the notion of neighborhood is needed.

Definition 3.1.4 (Neighborhood) *The neighborhood $\partial(A)$ of a set $A \subseteq K$ is defined as*

$$\partial(A) = \{x \in K : x \sim a \text{ for some } a \in A\} \tag{3.6}$$

where $\sim$ is a reflexive and symmetric relation on K.

Based on the neighborhood relation, the Markov process can be defined as follows:

Definition 3.1.5 (Markov process) *Let X be a point process with density f. X is a Markov process under the symmetric and reflexive relation $\sim$ if and only if, for every configuration $\mathbf{x} \in \Omega$ such that $f(\mathbf{x}) > 0$, X satisfies*

- *$f(\mathbf{y}) > 0$ for every $\mathbf{y} \subset \mathbf{x}$;*
- *for every point u from K, $f(\mathbf{x} \cup u)/f(\mathbf{x})$ only depends on u and its neighborhood $\partial(\{u\}) \cap \mathbf{x} = \{x \in \mathbf{x} : u \sim x\}$.*

The Hammersley-Clifford theorem allows the density of the Markov process to be decomposed as the product of local functions defined on cliques:

Theorem 3.1.1 (Hammersley-Clifford) *A point process density $f : \Omega \to [0, \infty[$ is Markov w.r.t. the neighborhood relation $\sim$ if and only if there is a measurable function $\phi : \Omega \to [0, \infty[$ such that*

$$f(\mathbf{x}) = \prod_{\mathbf{y} \subseteq \mathbf{x}, \mathbf{y} \in \mathcal{C}_{\mathbf{x}}} \phi(\mathbf{y}), \tag{3.7}$$

where the set of cliques is given by $\mathcal{C}_{\mathbf{x}} = \{\mathbf{y} \subseteq \mathbf{x} : \forall \{u, v\} \subseteq \mathbf{y}, u \sim v\}$.

In general, the normalized density function f is unreachable since the normalizing constant cannot be computed neither analytically nor numerically. Therefore, point processes defined by an unnormalized density $h(\cdot)$ are considered. The normalized density function of a point process can be written as:

$$f(X) = \frac{h(X)}{c} \tag{3.8}$$

with $c = \int_{\Omega} h(\mathbf{x}) d\mathbf{x}$. Markov models originated in statistical physics where c is called *partition function* or *normalizing constant*.
Since the density of a Markov point process is generally intractable, an approximation called the Papangelou conditional intensity is used to simulate such models Papangelou [1974], Kallenberg [1983].

Definition 3.1.6 (Papangelou conditional intensity) *The Papangelou conditional intensity for a point process X with density f is defined by*

$$\lambda^*(x, u) = \frac{f(x \cup \{u\})}{f(x)}, \quad x \in N^{lf}, u \in K \setminus x,$$

where we consider $a/0 = 0$, for $a \geq 0$.

Indeed, the Papangelou conditional intensity does not depend on the normalizing constant which cancels out and thus, $\lambda^*(x, u)$ does not depend on it. For a Poisson process, the Papangelou conditional intensity does not depend on x either ($\lambda^*(x, u) = f(u)$, since the points are spatially independent). The conditional intensity can be used to distinguish between different types of point processes. Hence,

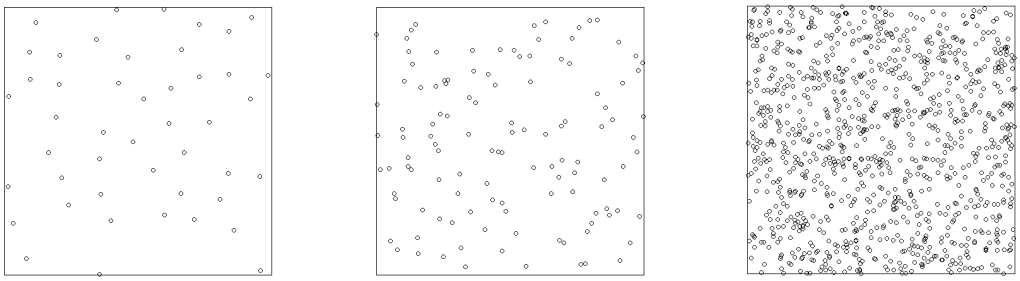


Figure 3.2: Simulation of a Strauss interaction process in a square window ($l = 10$) with fixed $\beta = 10$ and radius $r = 1$. (a) $\gamma = 0.0$; (b) $\gamma = 0.5$; (c) $\gamma = 1.0$.

attractive point processes can be used to model clustering phenomena, such as the natural clustering of plants in an area. In contrast, repulsive point processes can be used to model for example the repulsive behavior of flying airplanes. For safety reasons, airplanes are required to fly at a certain distance from one another, exhibiting thus a repulsive behavior. Attractive or repulsive point processes can be described based on the conditional intensity as follows:

Definition 3.1.7 (Attractive vs Repulsive point process) *A point process X is called attractive if*

$$\lambda^*(x,u) \leq \lambda^*(y,u), \quad \textit{whenever} \quad x \subset y$$

or repulsive if

$$\lambda^*(x,u) \geq \lambda^*(y,u), \quad \textit{whenever} \quad x \subset y.$$

A typical example of Markov point processes is the pairwise interaction process. In this case, the unnormalized density function is written as:

$$h(\mathbf{x}) = \prod_{i=1}^{n(\mathbf{x})} b(x_i) \prod_{1 \leq i < j \leq n(\mathbf{x}): x_i \sim x_j} g(x_i, x_j), \tag{3.9}$$

with $n(\mathbf{x})$ being the number of points in the configuration $\mathbf{x}$, $b(x_i)$ is a unary function of the point x_i and $g(x_i, x_j)$ is an interaction function, e.g. a non-negative function for which the right hand side of eq. 3.9 is integrable with respect to the unit rate Poisson process.

Example 3.1.1 (The Strauss process) *A simple pairwise interaction process is the Strauss process given by the density:*

$$f(\mathbf{x}) = \beta^{n(\mathbf{x})} \gamma^{s(\mathbf{x})}, \tag{3.10}$$

where $\beta > 0$ and $s(\mathbf{x})$ represents the number of pairs of points in $\mathbf{x}$ that are at a distance of r or less apart.

Based on the parameter γ, four cases can be distinguished:

- $\gamma = 0$: points are prohibited to be closer than the distance r. This type of process is called a *hard core process*;
- $\gamma < 1$: the density decreases with the number of cliques giving a repulsive effect;
- $\gamma = 1$: the points are independent. This is the Poisson process;
- $\gamma > 1$: the density is not integrable and hence, the process does not exist.

Three out of four cases are graphically displayed in Figure 3.2.
Finally, we can consider an extension of the processes defined so far. To each point x, we attach a random mark $m_x \in M$. Thus, we can now consider a marked point process X to be composed of objects, where each object i is defined by its position x_i and its mark m_i. Finally, we can introduce the Markov marked point process as follows:

Definition 3.1.8 (Markov marked point process) *X is a Markov marked point process on $W = K \times M$ w.r.t. the symmetric, reflexive relation $\sim$ on W if for all $\mathbf{x}$ such that $f(\mathbf{x}) > 0$:*

- *$f(\mathbf{y}) > 0$ for all $\mathbf{y} \subseteq \mathbf{x}$;*
- *for all $(u, l) \in W$, $f(\mathbf{x} \cup \{(u, l)\})/f(\mathbf{x})$ depends only on (u, l) and $\partial(\{(u, l)\}) \cap \mathbf{x} = \{(v, k) \in \mathbf{x} : (u, l) \sim (v, k)\}$.*

For image analysis purposes, the mark is used to define the geometry of the object, while the points in the configuration refer to the center of the objects Descombes et al. [2011]. In terms of time sequence analysis, the marks can include not only geometric features, but also labeling information.
The Strauss process example shows an important property of point processes. Based on the values of the parameters of the process, it can produce significantly different outputs. Hence, a suitable parameter setting is required for a given problem. Although these parameters could be set manually, automatic parameter estimation techniques are more desirable as they can be applied to a large variety of data sets containing huge amounts of data without the need of human intervention. Hence, such techniques will be discussed in the following section.

3.2 Parameter estimation

A generic model can be designed for solving a generic problem. In order to adapt the generic model to a specific problem, certain parameters of the model have to be adapted. These parameters can be as straight forward as setting the maximum radius of a circle that has to be detected, or as complicated as determining the weights of the different terms within the density function. Two main difficulties arise when dealing with parametric inference:

1. The configuration of objects, $\mathbf{x}$, is unknown;

2. The density function $f(\cdot)$ is known only up to a normalizing constant, c. Furthermore, the computation of this constant is analytically (and usually also numerically) intractable.

The density of a marked point process is characterized by three types of parameters according to Descombes et al. [2011]:

1. Parameters relative to the probability density of the object mark, or how the marks are distributed in the mark space;

2. Parameters relative to the interaction fields of the objects, or when do objects interact with each other;

3. Parameters relative to the interactions between the objects, or how do objects interact within a configuration.

Example 3.2.1 (Marked pairwise interaction process) *Let K be a compact set in $\mathbb{R}^2$ such that $0 < \nu(K) < \infty$, where $\nu(K)$ is the Lebesque measure on K, and $M = \{1, \cdots, I\}$, with $I \in \mathbb{N}$, be the space of marks having the uniform probability measure ν_M. A finite configuration of objects $\mathbf{x} = \{x_i, i = 1, \cdots, n\}$ that exists in $K \times M$ is formed from a collection of circles with different colors, $x_i = (k_i, m_i)$, with $k_i \in K$ and $m_i \in M$. The interaction process is defined by:*

$$f(\mathbf{x}) = \alpha \prod_{(k,m)\in\mathbf{x}} \beta(k,m) \prod_{(u,i)\sim(v,j)\in\mathbf{x}} \gamma_{ij}(||u-v||). \tag{3.11}$$

The homogeneous Poisson point process with unit intensity is the reference measurement for the point positions, to which the marks distributed according to ν_M are independently associated. The normalizing constant is denoted here with α and

$$(u,i) \sim (v,j) \leftrightarrow ||u-v|| \leq r, u \neq v. \tag{3.12}$$

The model is well defined if $\beta : K \times M \to \mathbb{R}^+$ and $\gamma_{ij} : (0,r] \to [0,1]$ for all $i, j \in M$ and $r > 0$.
The three sets of parameters for this model are as follows:

1. *Parameters relative to the probability density of the mark are given by M and ν_M. These parameters are used to determine the distribution of the mark, as well as the mark itself;*

2. *Parameters relative to the interaction fields of the objects are given by the radius r which determines the maximum distance between objects at which they interact with each other;*

3. *Parameters relative to the interactions between the objects are given by the activity parameter β and the symmetric interaction function $\gamma_{ij} = \gamma_{ji}$, $\forall i, j \in M$.*

In order to implement a fully automatic system, all of these parameters should ideally be estimated automatically. This section introduces some of the most common approaches to parameter estimation in the context of marked point processes. A more detailed overview on parameter estimation in the context of stochastic processes is given by Feldman and Ciriaco [2009].
Since the object configuration $\mathbf{x}$ is unknown and the image likelihood $f_\theta(\mathbf{y})$, with θ being the parameter vector, has no explicit expression, the extended likelihood, $f_\theta(\mathbf{x}, \mathbf{y})$, i.e. the joint likelihood of the observed data, $\mathbf{y}$, and the hidden data or the configuration of objects that we want to retrieve, $\mathbf{x}$, can be used.

3.2.1 Maximum likelihood estimation using MCMC techniques

In this estimation method (short MLMCMC [Pelillo and Hancock, 1997]), instead of considering directly the log-likelihood, the ratio between the likelihood for the current parameter, θ, and the likelihood for a fixed parameter, $\psi \in \Theta$ is considered. For the general case of a non-normalized density function $h_\theta(\mathbf{z})$, this ratio can be expressed in terms of the expectation of the non-normalized densities:

$$\frac{c(\theta)}{c(\psi)} = \frac{1}{c(\psi)} \int h_\theta(\mathbf{z})\mu(d\mathbf{z}) = \frac{1}{c(\psi)} \int \frac{h_\theta(\mathbf{z})}{h_\psi(\mathbf{z})} h_\psi(\mathbf{z})\mu(d\mathbf{z}) \tag{3.13}$$

$$= \int \frac{h_\theta(\mathbf{z})}{h_\psi(\mathbf{z})} f_\psi(\mathbf{z})\mu(d\mathbf{z}) = \mathbb{E}_\psi \left[\frac{h_\theta(Z)}{h_\psi(Z)} \right]. \tag{3.14}$$

The normalizing constant for the observation likelihood $f_\theta(\mathbf{y})$ can be written as:

$$c(\theta) = \int \int h_\theta(\mathbf{x}, \mathbf{y})\mu(d\mathbf{x})\mu(d\mathbf{y}). \tag{3.15}$$

Similarly, the normalizing constant associated to the conditional density $f_\theta(\mathbf{x}|\mathbf{y})$ of the hidden data $\mathbf{x}$, given the data $\mathbf{y}$, can be written as:

$$c(\theta|\mathbf{y}) = \int h_\theta(\mathbf{x}, \mathbf{y})\mu(d\mathbf{x}). \tag{3.16}$$

Thus, the observation likelihood can be written as:

$$f_\theta(\mathbf{y}) = \frac{1}{c(\theta)} \int h_\theta(\mathbf{x}, \mathbf{y})\mu(d\mathbf{x}) = \frac{c(\theta|\mathbf{y})}{c(\theta)} \tag{3.17}$$

and the ratio of the log-likelihoods can be expressed as:

$$l(\theta) = \log \frac{c(\theta|\mathbf{y})}{c(\theta)} - \log \frac{c(\theta)}{c(\psi)} \tag{3.18}$$

Using eq. 3.14, we obtain the following expression for the ratio of the log-likelihoods:

$$l(\theta) = \log \mathbb{E}_\psi \left[\frac{h_\theta(X, Y)}{h_\psi(X, Y)} | Y = \mathbf{y} \right] - \log \mathbb{E}_\psi \left[\frac{h_\theta(X, Y)}{h_\psi(X, Y)} \right]. \tag{3.19}$$

Importance sampling Doucet et al. [2001] has to be used to approximate the expression in eq. 3.19. This approximation consists of two steps:

1. simulate n realizations of the process with parameters ψ w.r.t. which the expectation is computed;

2. approximate the expectation by an empirical mean obtained from the n realizations.

In eq. 3.19, the expectation on the left can be approximated using MCMC techniques [van Lieshout, 2000], while the expectation of the right can be obtained by simulating both the observations $\mathbf{y}$ and the hidden data $\mathbf{x}$.
Although this approach has been successfully used for interpolating and extrapolating point processes by Geyer [1999] and van Lieshout and Baddeley [2002], it has obvious limitations in the case of object detection. The main limitation is that the observed data in the case of object detection is not a configuration of the point process within a restricted area of the image, but rather the radiometric values of the image pixels. Thus, in order to approximate the expectation on the right in eq. 3.19, one must simulate both a configuration $\mathbf{x}$ and the associated radiometry $\mathbf{y}$. This proves to be really difficult for real images. Thus, these approaches cannot be used in the case of object detection, nor tracking, and other methods have to be considered.

Algorithm 1 Expectation Maximization algorithm proposed by Dempster et al. [1977].

Inputs
- initial value θ^0, image $\mathbf{y}$
$i = 0$
Repeat:

1. (E)-step: compute

$$Q(\theta, \theta^i; \mathbf{y}) = \mathbb{E}_{\theta^i}[\log f_\theta(X, Y)|Y = \mathbf{y}], \tag{3.20}$$

   where θ^i is the value at iteration i of the parameter vector θ that needs to be estimated;

2. (M)-step: $\theta^{i+1} = \arg\max_\theta Q(\theta, \theta^i; \mathbf{y})$

$i = i + 1$
until $|\theta^{(i+1)} - \theta^{(i)}| \leq \varepsilon$

3.2.2 Expectation Maximization-like algorithms

Expectation Maximization (EM) algorithms are simple to use and efficient to determine the estimators of the maximum likelihood. The EM was introduced by Dempster et al. [1977] and is an iterative method, where each iteration consists of

two steps, as shown in algorithm 1. It is proven that at each iteration, the likelihood is increased until it converges to a local maximum.
Nevertheless, the EM algorithm has two main drawbacks:

1. the algorithm converges only to a local maximum, and thus, the initialization θ^0 has a crucial impact on the results;

2. the convergence speed can be very low.

Nevertheless, a stochastic version of the EM algorithm can be used. However, before going into the stochastic versions of the EM algorithm, a brief discussion on the approximation of the likelihood with a pseudo-likelihood, which makes computations easier by removing the normalizing constant from the computations is necessary.

3.2.2.1 Approximation of the likelihood with a pseudo-likelihood

Different solutions can be provided in cases when the likelihood is intractable, impossible to be formulated explicitly or expensive to evaluate. One such solution is to propose a likelihood-free inference, initially proposed in statistical literature by Diggle and Gratton [1984] and also referred to as Approximate Bayesian Computation (ABC), a term coined by Beaumont et al. [2002]. ABC based methods approximate the likelihood function by simulated data and compares the output with the observed data [Beaumont, 2014, Bertorelle et al., 2010, Csilléry et al., 2010, DelMoral et al., 2011]. Another solution is to substitute the likelihood with a pseudo-likelihood, first proposed for point processes by Mateu and Montes [2001].
Two reasons are at the heart of approximating the likelihood with a pseudo-likelihood:

1. given a configuration $\mathbf{x}$, the pseudo-likelihood is a function very close to the likelihood function (and in some cases the two are the same, such as in the trivial case of a Poisson point process);

2. the estimators for the maximum likelihood and the maximum pseudo-likelihood have similar values for the point processes of interest.

The pseudo-likelihood of a point process $\mathbf{x}$, given the observation $\mathbf{y}$ can be defined as [Mateu and Montes, 2001]:

$$PL(\theta; \mathbf{x}, \mathbf{y}) = \left[\prod_{x_i \in \mathbf{x}} \lambda_\theta(x_i; \mathbf{x}, \mathbf{y}) \right] \exp\left(- \int \lambda_\theta(u; \mathbf{x}, \mathbf{y}) \Lambda(du) \right), \tag{3.21}$$

where $\Lambda(\cdot)$ is the intensity measure of the reference Poisson point process and $\lambda_\theta(\cdot)$ is the Papangelou conditional intensity function defined in eq. 3.1.6 for the parameter vector θ. The pseudo-likelihood can be used to approximate the likelihood of a configuration by considering the objects within that configuration to be independent from each other. The pseudo-likelihood can be inserted in the expectation step of the EM algorithm to facilitate computations. In the following, stochastic versions of the EM are discussed.

Algorithm 2 Stochastic Expectation Maximization algorithm proposed by Celeux and Diebolt [1985].

Inputs
- initial value θ^0, image $\mathbf{y}$
$i = 0$
Repeat:

1. (S)-step: Simulate $\mathbf{x}^i \sim f_{\theta^i}(\mathbf{x}, \mathbf{y})$;
2. (E)-step: Compute $Q(\theta, \theta^i; \mathbf{y}) = \log PL(\theta; \mathbf{x}^i, \mathbf{y})$;
3. (M)-step: $\theta^{i+1} = \arg\max_\theta Q(\theta, \theta^i; \mathbf{y})$

$i = i + 1$
until $|\theta^{(i+1)} - \theta^{(i)}| \leq \varepsilon$

3.2.2.2 Stochastic EM

The most natural approach for a stochastic version of the EM algorithm is to use Monte Carlo methods to approximate the likelihood in eq. 3.20. This idea was proposed by Wei and Tanner [1990] and is known in literature under the name *Monte-Carlo Expectation Maximization* (MCEM). The expectation step is thus computed as follows:

1. simulate n configurations $\mathbf{x}^{(1)}, \cdots, \mathbf{x}^{(n)}$, conditioned by the observation $\mathbf{y}$ under the parameter vector θ^i and
2. compute for each configuration $\mathbf{x}^{(j)}$ the logarithm of the joint likelihood $f_{\theta^i}(\mathbf{x}^{(j)}, \mathbf{y})$ and approximate the likelihood with the empirical mean of the n configurations.

The resemblance to the MLMCMC algorithm can be observed. Nevertheless, MCEM has a significant advantage in that there is no need to simulate the observations $\mathbf{y}$. On the other hand, this method has a significant drawback consisting in the computation cost needed to simulate n configurations and approximating the likelihood for each one of them.
Hence, another stochastic version of the EM algorithm was proposed by Celeux and Diebolt [1985]. Their idea was to replace the computation of the likelihood in eq. 3.20 with a much easier computation of the log-likelihood $Q(\theta, \theta^i; \mathbf{y}) = \log f_{\theta^i}(\mathbf{x}^i, \mathbf{y})$, where the configuration $\mathbf{x}^i$ is simulated conditioned on the observations $\mathbf{y}$ under the parameter vector θ^i. This approach is known in literature under the name Stochastic EM (SEM) and is a particular case of the MCEM algorithm, where the number of simulated configurations is fixed to $n = 1$.
Besides the decreased computational cost, the SEM presents a significant theoretical advantage compared to the EM. Given its stochastic nature, SEM is less likely to converge to a local maximum, instead of a global one. The SEM is presented in

Algorithm 3 Stochastic Approximation Expectation Maximization algorithm proposed by Deylon et al. [1999].

Inputs

- initial value θ^0, image $\mathbf{y}$

$i = 0$

Repeat:

1. (S)-step: Simulate $\mathbf{x}^i \sim f_{\theta^i}(\mathbf{x}, \mathbf{y})$;
2. (E)-step: Compute $Q(\theta, \theta^i; \mathbf{y}) = (1 + \tau_i)Q(\theta, \theta^{(i-1)}; \mathbf{y}) + \frac{\tau_i}{n}\sum_{j=1}^{n} \log PL(\theta; \mathbf{x}_i^j, \mathbf{y})$;
3. (M)-step: $\theta^{i+1} = \arg\max_\theta Q(\theta, \theta^i; \mathbf{y})$

$i = i + 1$

until $|\theta^{(i+1)} - \theta^{(i)}| \leq \varepsilon$

Algorithm 2, where the likelihood in eq. 3.20 is replaced by the pseudo-likelihood described in eq. 3.21.

Yet another variant is the Stochastic Approximation EM (SAEM) algorithm proposed by Deylon et al. [1999] as an alternative to the stochastic versions previously mentioned. The main difference in the SAEM is the way in which the expectation is computed, which is based on the stochastic approximation method proposed by Robbins and Monro [1951]. The expectation thus corresponds to:

$$Q(\theta, \theta^i; \mathbf{y}) = (1 + \tau_i)Q(\theta, \theta^{(i-1)}; \mathbf{y}) + \frac{\tau_i}{n}\sum_{j=1}^{n} \log PL(\theta; \mathbf{x}_i^j, \mathbf{y}), \tag{3.22}$$

where τ_i is the forgetting factor and n is the number of configurations simulated. This approach keeps a memory of past simulations. The forgetting factor has a crucial influence on the convergence properties of the SAEM. A good value for τ_i is $\tau_i = j^{-\alpha}$, with $\alpha \in (0.5, 1)$ ([Jank, 2006]). The SAEM is presented in algorithm 3, with $n = 1$.

The SAEM has been recently proposed by Boisbunon and Zerubia [2014] for parameter estimation for the specific problem of boat extraction in harbors. Once the parameters of the point process model have been estimated/learned, optimization methods can be applied to find the best configuration of objects for a given image. In the following section, appropriate optimization methods for the point process framework are discussed.

3.3 Optimization

This section tackles the problem of marked point process simulation. As opposed to a point process, the intensity measure of a marked point process is written as

a product of two measures, one measuring Borel sets on the image support K and one measuring Borel sets on the mark space M. Hence, the intensity measure of a marked Poisson point process can be written as:

$$\nu(\cdot) = (\Lambda_K \times \mathbb{P}_M)(\cdot) \tag{3.23}$$

where Λ_K is the Lebesgue measure on K and $\mathbb{P}_M$ is some probability measure on M.
The interest lies in finding the most probable configuration of objects, $\mathbf{x}$, that fits the given data $\mathbf{y}$. Therefore, an optimization problem has to be solved. Furthermore, point process models represent a particular type of optimization problems in which the number of variables to be estimated is itself a random variable. Thus, dimensional jumps are necessary to reach an optimal solution, meaning that the optimization procedure must be able to add or remove objects from a configuration. The classical Markov chain Monte Carlo samplers ([Robert and Casella, 2005]) developed to tackle trans-dimensional simulation problems, as well as recent developments in this area and alternative algorithms are introduced in this section.
Markov chain Monte Carlo (MCMC) sampling was invented in the 50s shortly after the ordinary Monte Carlo algorithm in Los Alamos, one of the few places where computers existed at the time. In their attempt to simulate fluid dynamics, Metropolis et al. [1953], realized that they do not need to simulate the exact dynamics, since it sufficed to simulate a Markov chain with the same equilibrium distribution. The algorithm became known as the *Metropolis algorithm* and gained popularity as computers became widely available. Later, Hastings [1970] generalized the algorithm and transformed it into what became known as the *Metropolis-Hastings algorithm*. Finally, Green [1995] generalized further the Metropolis-Hastings algorithm by introducing dimensional jumps into the algorithm which became known as the *Metropolis-Hastings-Green algorithm* or alternatively *Reversible Jump MCMC algorithm* (RJMCMC).
The main idea of this algorithm is to simulate a Markov chain $(X_i)_{i\in\mathbb{N}}$ over the configuration space Ω which converges to a target distribution π. Nevertheless, the Markov chain must possess certain properties in order to guarantee its convergence to the desired target distribution π.

3.3.1 The reversible jump MCMC sampler

Before describing the sampler generalized by Green [1995], the necessary theoretical foundation is presented.

3.3.1.1 Markov chains and convergence properties

Definition 3.3.1 (Markov chain) *Let (X_n) be a sequence of random variables, with values taken from a space Ω associated with its sigma-algebra $\mathcal{B}$. (X_n) is said to be a Markov chain if:*

$$p(X_{i+1} \in A | X_i = \mathbf{x}_i, \cdots, X_0 = \mathbf{x}_0) = p(X_{i+1} \in A | X_i = \mathbf{x}_i),\ \forall A \in \mathcal{B}. \tag{3.24}$$

According to the definition, the evolution of a Markov chain depends only on the current state.

Definition 3.3.2 (Homogeneous chain) *If the evolution of a Markov chain does not depend on the time parameter i, then the chain is said to be homogeneous.*

Only homogeneous Markov chains will be considered from now on.
In order for the Markov chain to evolve, a perturbation (or transition) kernel has to be associated to it.

Definition 3.3.3 (Perturbation kernel) *A perturbation kernel is a function P defined on $\Omega \times \mathcal{B}$ such that:*

- $\forall x \in \Omega$, $P(x, \cdot)$ *is a probability measure;*
- $\forall A \in \mathcal{B}$, $P(\cdot, A)$ *is measurable.*

A transition kernel P associated to a Markov chain $(X_i)_{i\in\mathbb{N}}$ is given by:

$$P(\mathbf{x}, A) = p(X_{i+1} \in A | X_i = \mathbf{x}). \tag{3.25}$$

Apart from the homogeneity constraint, in order to successfully simulate a Markov chain using an MCMC sampler, the chain must be:

- **Stationary**. A measure π is stationary for the Markov chain of perturbation kernel P if:
$$\pi(A) = \int P(\mathbf{x}, A) d\pi(\mathbf{x}),\ \forall A \in \mathcal{B}. \tag{3.26}$$
Indeed, a stationary measure π is needed for the chain to converge.

- **Reversible**. A Markov chain is reversible if its perturbation kernel P satisfies the equality:
$$\int_B P(\mathbf{x}, A) d\pi(\mathbf{x}) = \int_A P(\mathbf{x}', B) d\pi(\mathbf{x}'),\ \forall A, B \in \mathcal{B}. \tag{3.27}$$
Simply put, the probability of going from B to A under π is the same as going from A to B.

- **Irreducible**. A Markov chain is irreducible if for all pairs of states $(\mathbf{x}, \mathbf{x}')$, there is a positive probability of reaching $\mathbf{x}$ from $\mathbf{x}'$ in a finite number of transitions:
$$\forall \mathbf{x}, \mathbf{x}' \in \Omega, \exists m < \infty : p(X_{i+m} = \mathbf{x} | X_i = \mathbf{x}') > 0. \tag{3.28}$$
Irreducibility means that the chain has a strictly positive probability of reaching any π-probable set in a finite time, independent of the initial conditions.

- **Aperiodic**. The period d of a Markov chain is defined as:

$$d = \gcd\{k : p(X_k = \mathbf{x}|X_0 = \mathbf{x}) > 0\}. \tag{3.29}$$

  where $\gcd\{\cdot\}$ is the greatest common divisor. A Markov chain is aperiodic if $d = 1$. A Markov chain needs to be aperiodic in order to converge.

- **Harris recurrent**. A Markov chain is Harris recurrent if

$$p(\exists t : X_t \in A|X_0 = \mathbf{x}) = 1 \; \forall \mathbf{x} \in \Omega, \; \forall A \in \mathcal{B} : \pi(A) > 0. \tag{3.30}$$

  The Harris recurrence property is needed to ensure that the Markov chain converges, regardless of the initial conditions.

- **Ergodic**. A Markov chain with stationary measure π converges ergodic towards π if it is aperiodic and Harris recurrent.

The Markov chain needs to be designed so as to be ergodic, in order to converge to the desired distribution π. In his work, Green [1995] proposes the use of a mixture of kernels to simulate a Markov chain.

3.3.1.2 State dependent mixing

Consider a finite set of perturbation kernels $Q_m(\mathbf{x}, A)$, $m \in M$, such that:

$$Q(\mathbf{x}, A) = \sum_{m \in M} Q_m(\mathbf{x}, A). \tag{3.31}$$

The perturbation kernels must satisfy the following constraints:

- $Q_m(\mathbf{x}, \Omega)$ is known, $\forall m \in M$;
- $\sum_{m \in M} Q_m(\mathbf{x}, \Omega) \leq 1$, $\forall \mathbf{x} \in \Omega$;
- there exists a symmetric measure $\phi_m(d\mathbf{x}, d\mathbf{x}')$, $\forall m \in M$, $\phi_m(\cdot, \cdot) : \Omega \times \Omega$ such that $\pi(d\mathbf{x})Q_m(\mathbf{x}, d\mathbf{x}')$ is absolutely continuous w.r.t. $\phi_m(d\mathbf{x}, d\mathbf{x}')$;
- $\forall m \in M$, $\forall \mathbf{x} \in \Omega$, realizations can be simulated starting from the normalized perturbation distribution:

$$P_m(\mathbf{x}, \cdot) = \frac{Q_m(\mathbf{x}, \cdot)}{Q_m(\mathbf{x}, \Omega)}. \tag{3.32}$$

- The associated Radon-Nikodym derivative is denoted by $f_m(\cdot, \cdot)$ and is written as:

$$f_m(\mathbf{x}, \mathbf{x}') = \frac{\pi(d\mathbf{x})Q_m(\mathbf{x}, d\mathbf{x}')}{\phi_m(d\mathbf{x}, d\mathbf{x}')}. \tag{3.33}$$

For a given state $X_i = \mathbf{x}$, we can write the updating scheme as follows:

1. With probability $Q_m(\mathbf{x}, \Omega)$ choose a kernel Q_m or with probability $1 - \sum_m Q_m(\mathbf{x}, \Omega)$ let the state unchanged $X_{i+1} = \mathbf{x}$;
2. Simulate $\mathbf{x}'$ according to the normalized kernel chosen:

$$\mathbf{x}' \sim \frac{Q_m(\mathbf{x}, \cdot)}{Q_m(\mathbf{x}, \Omega)} \tag{3.34}$$

3. Compute the Green ratio:

$$R_m(\mathbf{x}, \mathbf{x}') = \frac{f_m(\mathbf{x}', \mathbf{x})}{f_m(\mathbf{x}, \mathbf{x}')} \tag{3.35}$$

4. Accept the perturbation with probability $\alpha_m(\mathbf{x}, \mathbf{x}') = \min(1, R_m(\mathbf{x}, \mathbf{x}'))$ or reject otherwise.

3.3.1.3 Detailed balance

For the chain to be reversible we require that the probability of the chain in a general set A and moving to a general set B to be the same with A and B reversed (see eq. 3.27). This is also known as the *detailed balance condition* (DB). In order to prove the DB condition for a kernel $P(\cdot, \cdot)$, it suffices to show that each sub-kernel $P_m(\cdot, \cdot)$ maintains the DB condition [Green, 1995, Ortner, 2004], where:

$$P(\mathbf{x}, A) = \int_A Q_m(\mathbf{x}, d\mathbf{x}')\alpha_m(\mathbf{x}, \mathbf{x}'). \tag{3.36}$$

Using eq. 3.33, we can write:

$$\int_A \int_B \pi(d\mathbf{x})Q_m(\mathbf{x}, d\mathbf{x}')\alpha_m(\mathbf{x}, \mathbf{x}') = \int_A \int_B f_m(\mathbf{x}, \mathbf{x}')\alpha_m(\mathbf{x}, \mathbf{x}')\phi_m(d\mathbf{x}, d\mathbf{x}'). \tag{3.37}$$

By definition of $\alpha_m(\mathbf{x}, \mathbf{x}')$ we have that:

$$f_m(\mathbf{x}, \mathbf{x}')\alpha_m(\mathbf{x}, \mathbf{x}') = f_m(\mathbf{x}', \mathbf{x})\alpha_m(\mathbf{x}', \mathbf{x}). \tag{3.38}$$

The symmetry of $\phi_m(\cdot, \cdot)$ together with this property give the DB condition in eq. 4.6.

3.3.1.4 Standard perturbation kernels

The choice of perturbation kernels is an important point when using MCMC samplers. In this section we present the most common perturbation kernels used in literature [Green, 1995], [Ortner, 2004], Descombes et al. [2011] and show that the detailed balance (DB) condition is maintained. The DB for these kernels has already been proven by Ortner [2004]. We reiterate the proof for the birth and death kernel and only state the results obtained by Ortner [2004] for the other kernels.
Standard perturbation kernels for simulating marked point process models can be categorized as follows:

- **Birth and Death**: The birth and death kernel is used to add or remove objects from a configuration and hence, modify the dimension of the configuration space. This kernel is in itself a mixture of two sub-kernels: a birth kernel (used to add a new object to a configuration) and the death kernel (used to remove an object from a configuration). If we suppose that the birth kernel generates a new object in W according to the probability law $\frac{\nu(\cdot)}{\nu(W)}$, where $\nu(W)$ is a measure on W, and the death kernel uniformly chooses one object to be deleted from the current configuration, then we can write the kernel as follows:

$$Q(\mathbf{x}, \cdot) = p_b(\mathbf{x})Q_b(\mathbf{x}, \cdot) + p_d(\mathbf{x})Q_d(\mathbf{x}, \cdot), \tag{3.39}$$

with the birth kernel defined by:

$$Q_b(\mathbf{x}, A) = \int_{u \in W} \mathbb{1}_A(\mathbf{x} \cup u) \frac{\nu(du)}{\nu(W)} \qquad A \in N^{lf} \tag{3.40}$$

and

$$Q_d(\mathbf{x}, A) = \sum_{u \in \mathbf{x}} \mathbb{1}_A(\mathbf{x} \setminus u) \frac{1}{n(\mathbf{x})}, \tag{3.41}$$

except if $n(\mathbf{x}) = 0$, when $Q_d(\mathbf{x}, \cdot) = I(\mathbf{x}, \cdot)$.

Proof of DB. Let us consider the following measure ϕ, where A and B are two measurable subsets of $\mathcal{C}$:

$$\phi(A \times B) = \int_{\mathcal{C}} \int_{u \in W} \mathbb{1}_A(\mathbf{x}) \mathbb{1}_B(\mathbf{x} \cup u) \nu(du) \mu(d\mathbf{x}) + \int_{\mathcal{C}} \mathbb{1}_A(\mathbf{x}) \sum_{u \in \mathbf{x}} \mathbb{1}_B(\mathbf{x} \setminus u) \mu(d\mathbf{x}). \tag{3.42}$$

The symmetry of this measure comes from the fact that $\nu(\cdot)$ is the intensity measure of the underlying Poisson process governed by the law $\mu(\cdot)$. Let $A_n = A \cap N_n^{lf}$ where N_n^{lf} is the subset of N_W^{lf} corresponding to configurations containing exactly n objects. Then, by definition of the Poisson point process we obtain:

$$\begin{aligned}
\phi(A_n \times B_{n-1}) &= \frac{e^{-\nu(W)}}{n!} \int_{W^n} \sum_{u \in \mathbf{x}} \mathbb{1}_{A_n}(\mathbf{x}) \mathbb{1}_{B_{n-1}}(\mathbf{x} \setminus u) \nu^n(d\mathbf{x}) \\
&= \frac{e^{-\nu(W)}}{n!} \int_{W^n} n \mathbb{1}_{A_n}(\{x_1, \cdots, x_n\}) \mathbb{1}_{B_{n-1}}(\{x_1, \cdots, x_{n-1}\}) d\nu^n(\mathbf{x}) \\
&= \frac{e^{-\nu(W)}}{(n-1)!} \int_{W^{n-1}} \int_W \mathbb{1}_{B_{n-1}}(\mathbf{x}') \mathbb{1}_{A_n}(\mathbf{x}' \cup u) \nu^{n-1}(d\mathbf{x}') \nu(du) \\
&= \phi(B_{n-1} \times A_n).
\end{aligned}$$

The symmetry of ϕ is obtained by writing $\phi(A, B)$ as an infinite sum $\phi(A, B) = \sum (\phi(A_n, B_{n-1}) + \phi(A_n, B_{n+1}))$. Note that $A \times B$ has a strictly positive measure $\pi(\cdot)Q(\cdot, \cdot)$ and that its measure ϕ is also strictly positive. We now have

to show that $\phi(d\mathbf{x}, d\mathbf{x}')$ dominates $\pi(d\mathbf{x})Q(\mathbf{x}, d\mathbf{x}')$ and compute the Radon-Nikodym derivative. There are two cases to consider:
Birth: If $\mathbf{x}' = \mathbf{x} \cup u$ we obtain the following expressions:

$$\pi(d\mathbf{x})Q(\mathbf{x}, d\mathbf{x}') = h(\mathbf{x})\mu(d\mathbf{x})p_b(\mathbf{x})\frac{\nu(du)}{\nu(W)} \quad \phi(d\mathbf{x}, d\mathbf{x}') = \mu(d\mathbf{x})\nu(du) \tag{3.43}$$

where $h(\mathbf{x})$ is the non-normalized density function of $\mathbf{x}$ and from where we can deduce the domination of $\phi(d\mathbf{x}, d\mathbf{x}')$ over $\pi(d\mathbf{x})Q(\mathbf{x}, d\mathbf{x}')$. The Radon-Nikodym derivative yields:

$$f(\mathbf{x}, \mathbf{x}') = p_b(\mathbf{x})\frac{h(\mathbf{x})}{\nu(W)}. \tag{3.44}$$

Death: If $\mathbf{x}' = \mathbf{x} \setminus u$ we obtain the following expressions:

$$\pi(d\mathbf{x})Q(\mathbf{x}, d\mathbf{x}') = h(\mathbf{x})\mu(d\mathbf{x})p_d(\mathbf{x})\frac{1}{n(\mathbf{x})} \quad \phi(d\mathbf{x}, d\mathbf{x}') = \mu(d\mathbf{x}) \tag{3.45}$$

where $h(\mathbf{x})$ is the non-normalized density function of $\mathbf{x}$ and from where we can again deduce the domination of $\phi(d\mathbf{x}, d\mathbf{x}')$. The Radon-Nikodym derivative in this case if given by:

$$f(\mathbf{x}, \mathbf{x}') = p_d(\mathbf{x})\frac{h(\mathbf{x})}{n(\mathbf{x})} \tag{3.46}$$

This ends the proof.

Green ratio. The Green ratio has two different expressions, depending on whether an object is added or removed from the configuration.

- In case of a **Birth** ($\mathbf{x}' = \mathbf{x} \cup u$), the Green ratio is given by:

$$R(\mathbf{x}, \mathbf{x}') = \frac{f(\mathbf{x}', \mathbf{x})}{f(\mathbf{x}, \mathbf{x}')} = \frac{p_d(\mathbf{x}')}{p_b(\mathbf{x})}\frac{h(\mathbf{x}')}{h(\mathbf{x})}\frac{\nu(W)}{n(\mathbf{x}')}. \tag{3.47}$$

- In case of a **Death** ($\mathbf{x}' = \mathbf{x} \setminus u$), the Green ratio is given by:

$$R(\mathbf{x}, \mathbf{x}') = \frac{f(\mathbf{x}', \mathbf{x})}{f(\mathbf{x}, \mathbf{x}')} = \frac{p_b(\mathbf{x}')}{p_d(\mathbf{x})}\frac{h(\mathbf{x}')}{h(\mathbf{x})}\frac{n(\mathbf{x})}{\nu(W)}. \tag{3.48}$$

A variant of this perturbation kernel is the **Birth and Death in a Neighborhood**, which allows the creation or removal of an object in the predefined neighborhood of an existing object. For the type of applications considered, we can assume that if an object exists at a given location, it is more likely that additional objects exist in that area. The Birth and Death in a Neighborhood kernel incorporates this assumption by proposing the creation of objects in areas where other objects already exist and as such, increases the convergence of the Markov chain.
Let $\sim$ be a symmetric relation on W such that $\forall u, v \in W\ u \neq v$:

$$u \sim v \leftrightarrow d_K(u, v) \leq d_{max}, \tag{3.49}$$

where $d_K(\cdot,\cdot)$ is the Euclidean distance on K. Let $\mathcal{N}(\mathbf{x})$ be the set of pairs of related objects:

$$\mathcal{N}(\mathbf{x}) = \{\{u, v\},\ u, v \in \mathbf{x} \text{ s.t. } u \sim v\}. \tag{3.50}$$

The neighborhood of an object u is defined as:

$$\partial(u) = \{v \in W \text{ s.t. } v \sim u\},\ \partial(u) \subseteq W \tag{3.51}$$

and the neighborhood of a configuration is defined as:

$$\partial(\mathbf{x}) = \{u \in W : u \sim v \text{ for some } v \in \mathbf{x}\}. \tag{3.52}$$

The birth and death kernel in a neighborhood can be written as:

$$Q(\mathbf{x}, \cdot) = p_b(\mathbf{x})Q_b(\mathbf{x}, \cdot) + p_d(\mathbf{x})Q_d(\mathbf{x}, \cdot) \tag{3.53}$$

The birth in a neighborhood kernel is written as:

$$Q_b(\mathbf{x}, A) = \sum_{u \in \mathbf{x}} \eta_b^{\mathbf{x}}(u) Q_b^u(\mathbf{x}, A), \tag{3.54}$$

where $\eta_b^{\mathbf{x}}(u)$ is the discrete distribution according to which a new object is proposed and $Q_b^u(\mathbf{x}, \cdot)$ adds objects to the neighborhood of u.
The death in a neighborhood kernel is written as:

$$Q_d(\mathbf{x}, A) = \sum_{u \in \mathbf{x}} \eta_d^{\mathbf{x}}(u) \mathbb{1}_A(\mathbf{x} \setminus u), \tag{3.55}$$

where $\eta_d^{\mathbf{x}}(u)$ is null for any object u not belonging to any pair in $\partial(\mathbf{x})$.
As the birth and death in a neighborhood is a special case of the previously introduced birth and death kernel, we can keep the same symmetric measure ϕ described in eq. 3.42.

Birth: Before we state the Radon-Nikodym derivative, we briefly present how to generate a neighbor of an already chosen object u. First, a vector z is generated according to the law of the random variable Z on the space Σ. Second, an injection $\xi_u(\cdot)$ is applied on z with

$$\xi_u : \quad \Sigma \to W \tag{3.56}$$

$$z \to v. \tag{3.57}$$

The pair (z, ξ_u) must give an object v that is a neighbor of u and thus $\xi_u(\Sigma) = \partial(u)$. If we suppose that Z follows a law P_Z on Σ, we can detail Q_b^u:

$$\begin{aligned} Q_b^u(\mathbf{x}, A) &= P_Z(\mathbf{x} \cup \xi_u(Z) \in A) \\ &= P_Z(\xi_u(Z) \in A_{\mathbf{x}}) \end{aligned}$$

where $A_{\mathbf{x}} \subseteq W$ corresponds to the set $A_{\mathbf{x}} = \{v \in W \text{ s.t. } \mathbf{x} \cup v \in A\}$ which allows us to write:

$$Q_b^u(\mathbf{x}, A) = \int_\Sigma \mathbb{1}_{A_{\mathbf{x}}}(\xi_u(z)) dP_Z(z). \tag{3.58}$$

In his calculation Ortner [2004] uses two hypotheses:

- $\xi(\cdot)$ is a diffeomorphism, which implies that Σ has the same dimension as $\partial(u)$, and Σ and W have also the same dimension;
- the law of Z is dominated by the Lebesgue measure and denotes $f_Z(\cdot)$ its Radon-Nikodym derivative, for which he takes the uniform law as an example: $f_Z(\cdot) = \frac{1}{\lambda_Z(\Sigma)}$.

Ortner [2004] shows that under these hypotheses, the Radon-Nikodym derivative of the birth kernel can be written as:

$$f(\mathbf{x}, \mathbf{x} \cup v) = \frac{p_b}{f_\nu(v)} h(\mathbf{x}) \left(\sum_{u \in \mathbf{x}} \eta_b^{\mathbf{x}}(u) f_Z(\xi_u^{-1}(v)) \Lambda_u(v) \right), \tag{3.59}$$

with

$$\Lambda_u(v) = \begin{cases} 0 & \text{if } v \notin \partial(u) = \xi_u(\Sigma) \\ |J_{\xi_u^{-1}}(v)| & \text{otherwise} \end{cases} \tag{3.60}$$

where $J_{\xi_u^{-1}}(\cdot)$ is the Jacobian of $\xi_u^{-1}(\cdot)$.

Death: The expression of the Radon-Nikodym derivative obtained by Ortner [2004] is:

$$f(\mathbf{x}, \mathbf{x} \setminus u) = h(\mathbf{x}) p_d \eta_d^{\mathbf{x}}(u). \tag{3.61}$$

Green ratio. As in the case of the uniform birth and death kernel, the Green ratio has two different expressions, depending on whether an object is added or removed from the configuration.

- In case of a **birth** ($\mathbf{x}' = \mathbf{x} \cup v$), the Green ratio has the form:

$$R(\mathbf{x}, \mathbf{x} \cup v) = \frac{h(\mathbf{x} \cup v)}{h(\mathbf{x})} \frac{p_d(\mathbf{x} \cup v)}{p_b(\mathbf{x})} \frac{\eta_d^{\mathbf{x} \cup v}(v) f_\nu(v)}{\sum_{u \in \mathbf{x}} \eta_b^{\mathbf{x}}(u) f_Z(\xi_u^{-1}(v)) \Lambda_u(v)} \tag{3.62}$$

- In case of a **death** ($\mathbf{x}' = \mathbf{x} \setminus v$), the Green ratio has the form:

$$R(\mathbf{x}, \mathbf{x} \setminus v) = \frac{h(\mathbf{x} \setminus v)}{h(\mathbf{x})} \frac{p_b(\mathbf{x} \setminus v)}{p_d(\mathbf{x})} \frac{\sum_{u \in \mathbf{x} \setminus v} \eta_b^{\mathbf{x} \setminus v}(u) f_Z(\xi_u^{-1}(v)) \Lambda_u(v)}{\eta_d^{\mathbf{x}}(v) f_\nu(v)}. \tag{3.63}$$

- **Local perturbations**: These perturbations do not change the dimension of the configuration space, but rather modify the parameters of an object in the current state. Typical examples include: rotation, translation and scaling. Figure 3.3 depicts these type of perturbations. Ortner [2004] discusses these kernels in detail. We will briefly mention the main results obtained by Ortner [2004].
  Let Σ be a countable subset of $\mathbb{R}^d$ and $s(\cdot)$ the associated Lebesgue measure. Given a configuration of objects $\mathbf{x}$ and an object u, let $Z_{(\mathbf{x},u)}$ be a random

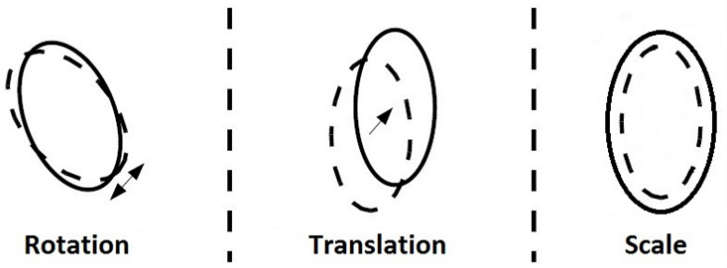


Figure 3.3: Example of a modification of the paramters of an ellipse with local perturbation kernels; (left) Rotation; (middle) Translation; (right) Scale.

variable living on a set $\Sigma(\mathbf{x}, u) \subset \Sigma$. Denote $\mathbb{P}_Z^{\mathbf{x},u}(\cdot)$ the distribution of $Z_{(\mathbf{x},u)}$ on Σ and $f_Z^{(\mathbf{x},u)}(\cdot)$ the associated density function. Finally, consider an injection

$$\zeta_{\mathbf{x}} : \quad W \times \Sigma \to W \tag{3.64}$$

$$(u, z) \to v. \tag{3.65}$$

Then, for a current configuration $\mathbf{x} = \{u_i\},\ i = \overline{1, n(\mathbf{x})}$, the local perturbation kernel can be described as follows:

1. Choose an object $u \in \mathbf{x}$ according to a discrete probability law $\eta^{\mathbf{x}}(u_i)$;
2. Generate z using the distribution of $Z_{(\mathbf{x},u)}$;
3. Compute $v = \zeta_{\mathbf{x}}(u, Z)$;
4. Propose $\mathbf{x}' = \mathbf{x} \setminus u \cup v$.

The perturbation kernel can be written as:

$$Q(\mathbf{x}, A) = \sum_{u \in \mathbf{x}} \eta^{\mathbf{x}}(\mathbf{x}, u) \mathbb{P}_Z^{(\mathbf{x},u)}(\mathbb{1}_A(\mathbf{x} \setminus u \cup \zeta_{\mathbf{x}}(u, z))). \tag{3.66}$$

We can consider the following measure:

$$\phi(A \times B) = \int_A \sum_{u \in \mathbf{x}} \int_{\Sigma(\mathbf{x},u)} \mathbb{1}_B(\mathbf{x} \setminus u \cup \zeta_{\mathbf{x}}(u, z)) \nu(dz) \mu(d\mathbf{x}). \tag{3.67}$$

To obtain the symmetry of this measure, we suppose the symmetry of the transformation:

$$v = \zeta_{\mathbf{x}}(u, z) \Leftrightarrow \exists \tilde{z} \in \Sigma(\mathbf{x}', v) \text{ s.t. } \mathbf{x}' = \mathbf{x} \setminus u \cup v,\ \ u = \zeta_{\mathbf{x}'}(v, \tilde{z}). \tag{3.68}$$

The uniqueness of $\tilde{z}$ comes from the injection $\zeta_{\mathbf{x}}(u, \cdot)$.

Green ratio. The following Green ratio is obtained:

$$R(\mathbf{x}, \mathbf{x}') = \frac{f(\mathbf{x}', \mathbf{x})}{f(\mathbf{x}, \mathbf{x}')} = \frac{h(\mathbf{x}')}{h(\mathbf{x})} \frac{\eta^{\mathbf{x}'}(v)}{\eta^{\mathbf{x}}(u)} \frac{f_Z^{(\mathbf{x}',v)}(\tilde{z})}{f_Z^{(\mathbf{x},u)}(z)}. \tag{3.69}$$

3.3.1.5 Reversible jump MCMC

The reversible jump MCMC sampler introduced by Green [1995], describes an algorithm that allows the simulation of a distribution in which the number of variables is itself random by producing an ergodic Markov chain. The RJMCMC sampler consists of two steps:

- **1:** A perturbation kernel is chosen from a mixture of perturbation kernels and the current state of the Markov chain is perturbed using this kernel;
- **2:** An acceptance ratio is computed and a decision is made to either keep the Markov chain in the current state or evolve it to a new state after the perturbation.

The RJMCMC sampler is presented in Algorithm 4. This sampler can be used to effectively simulate marked point process models. Nevertheless, this effectiveness comes at a price. The sampler requires a high number of iterations until convergence and usually results in very high computation times. On one hand, this computa-

Algorithm 4 Reversible jump MCMC sampler proposed by Green [1995].

1. Initialize $X_0 = \mathbf{x_0}$ and $i = 0$;
2. At iteration i with $X_i = \mathbf{x}$:
   - Choose a kernel type $m \in M$ according to the probability $p_i(\mathbf{x})$
     - Perturb $\mathbf{x}$ to a configuration $\mathbf{x}'$ according to $P_m(\mathbf{x} \to \cdot)$;
     - Compute the Green ratio:

$$R = \frac{\mathcal{Q}_m(\mathbf{x}' \to \mathbf{x})}{\mathcal{Q}_m(\mathbf{x} \to \mathbf{x}')} \frac{\pi(d\mathbf{x}')}{\pi(d\mathbf{x})}$$

     - Choose $X_{i+1} = \mathbf{x}'$ with probability $\min(1, R)$ and $X_{i+1} = \mathbf{x}$ otherwise;

tional burden has motivated many researchers to investigate other algorithms to effectively and efficiently simulate marked point processes. Such algorithms will be presented later in this section. On the other hand, the rapid development of technology led to new software implementation paradigms. Nowadays, the traditional single-processor sequential computing is replaced by fast paced multiple core parallel computing paradigms. This technological evolution inspired Verdié and Lafarge [2012] to propose a parallel implementation of the RJMCMC. Verdié and Lafarge [2012] showed that this implementation leads to a significant reduction in computation time in the case of image analysis.

3.3.1.6 Parallel implementation of RJMCMC

The RJMCMC performs successive perturbations to the current configuration. It can be easily observed that such a procedure can take a large amount of time to complete, especially in the case of large scenes with a significant number of objects. Verdié and Lafarge [2012] proposed to perform multiple perturbations in parallel by exploiting the conditional independence of objects outside the local neighborhood. Thus, they partitioned the image space K into a regular mosaic of independent cells using a data-driven partitioning scheme. Verdie [2013] demonstrates that the transition probability of two successive perturbations that occur in independent cells equals the product of the transition probabilities of the individual perturbations. Two cells, c_s and $c_{s'}$ are said to be independent if and only if:

$$\min_{p \in c_s, p' \in c_{s'}} ||p - p'|| \geq \varepsilon + 2\delta_{max}, \tag{3.70}$$

where p and p' are pixels within the cells c_s and $c_{s'}$ respectively, ε is the width of the neighboring relationship between cells and δ_{max} is the largest perturbation move allowed. In other words, two cells are independent if the transition probability of a random perturbation performed in one cell does not depend on the objects contained in the second cell. This property is visually depicted in Figure 3.4.
The cells are then regrouped into $2^{\dim K}$ sets such that no two neighboring cells belong to the same set. This regrouping scheme is used to ensure the mutual independence between cells from the same set, generically called a *mic-set* (e.g. Mutually Independent Cells set). Figure 3.5 shows a regular partition scheme on K, with $\dim K = 2$ (left) and $\dim K = 3$ (right) respectively. By using a data-driven partitioning, Verdié and Lafarge [2012] succeeded in developing a natural and efficient scheme for simulating non-uniform point distributions. Figure 3.6 shows the difference between a regular partition scheme on a given image (with $\dim K = 2$) (c) and a data-driven partitioning scheme for the same image (d).
Finally, they formulated the general transition kernel Q as a mixture of uniform sub-kernels $Q_{c,m}$, each kernel being defined on a cell c of $\mathcal{K}$, the kernel type $m \in M$ such that:

$$\forall \mathbf{x} \in \Omega, Q(\mathbf{x} \to \cdot) = \sum_{c \in \mathbf{K}} \sum_{m \in M} q_{c,m} Q_{c,m}(\mathbf{x} \to \cdot), \tag{3.71}$$

where $q_{c,m}$ is the probability of choosing kernel $Q_m(\mathbf{x} \to \cdot)$ in the cell c, given by:

$$q_{c,m} = \frac{P(m)}{\#\ cells\ in\ \mathcal{K}}. \tag{3.72}$$

The kernel defined in eq. 3.71 is embedded into the MCMC dynamics. This sampler enables the execution of parallel perturbations to intrinsically non-uniform point distribution based on the data-driven partitioning scheme. The sampling procedure is presented in Algorithm 5.

Algorithm 5 Parallel sampler proposed by Verdié and Lafarge [2012], based on the data-parallel distributed processing model.

1. Initialize $X_0 = \mathbf{x_0}$ and $i = 0$;
2. Compute the data-driven space partitioning tree $\mathcal{K}$;
3. At iteration i with $X_i = \mathbf{x}$:
   - Choose a mic-set $S_{mic} \in \mathcal{K}$ and a kernel type $m \in M$ according to the probability $\sum_{c \in S_{mic}} q_{c,m}$
   - For each cell c in S_{mic}:
     - Perturb $\mathbf{x}$ in cell c to a configuration $\mathbf{x}'$ according to $\mathcal{Q}_{c,m}(\mathbf{x} \to \cdot)$;
     - Compute the Green ratio:

$$R = \frac{\mathcal{Q}_{c,m}(\mathbf{x}' \to \mathbf{x})}{\mathcal{Q}_{c,m}(\mathbf{x} \to \mathbf{x}')} \frac{\pi(d\mathbf{x}')}{\pi(d\mathbf{x})}$$

     - Choose $X_{i+1} = \mathbf{x}'$ with probability $\min(1, R)$ and $X_{i+1} = \mathbf{x}$ otherwise;

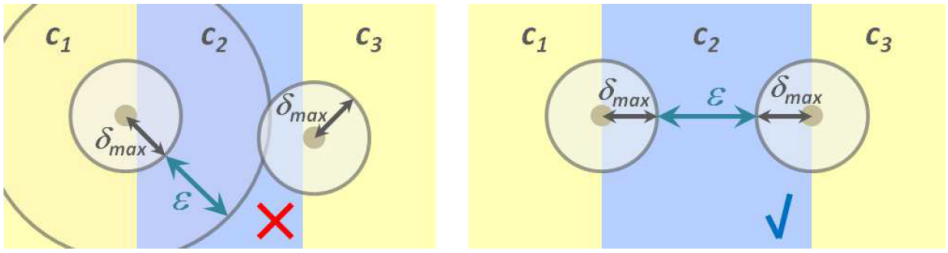


Figure 3.4: Cell independence. Left: Cell independence is not ensured. The width of the cell is not large enough and hence, a perturbation of the object in cell c_1 depends on the object in cell c_3; Right: Cell independence is ensured. The width of the cell is large enough so that any perturbation of an object in cell c_1 would not depend on an object in cell c_3. Image courtesy of Verdie [2013].

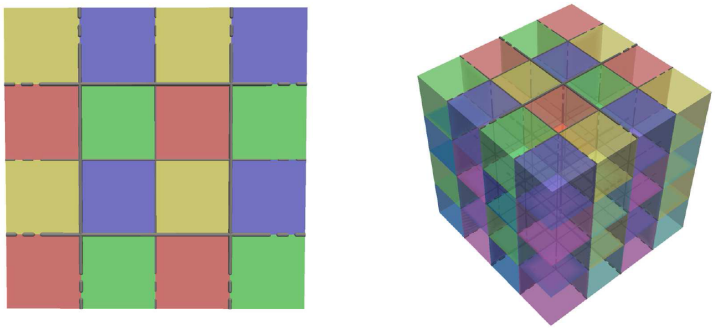


Figure 3.5: Regular partition scheme on K, with $\dim K = 2$ (left) and $\dim K = 3$ (right). Image courtesy of Verdie [2013].

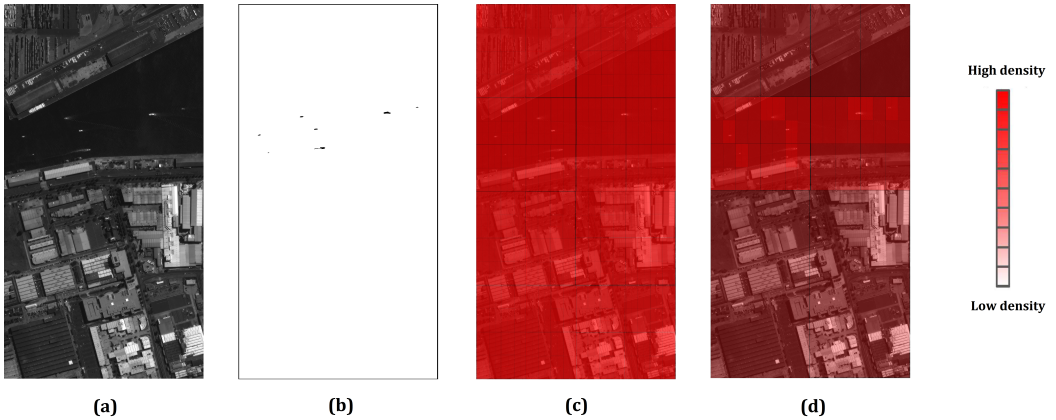


Figure 3.6: (a) An image of boats; (b) The class of interest is estimated from the input image (for example, a birth map can be used to obtain this initial estimation); (c) The corresponding probabilities $q_{c,i}$, when a regular partitioning scheme is applied; (d) The corresponding probabilities $q_{c,i}$ when a data-driven partitioning scheme is applied.

3.3.2 Multiple birth and death algorithm

A different attempt to simulate point processes is the multiple birth and death (MBD) algorithm proposed by Descombes et al. [2009]. The idea is that at each iteration i, a new random configuration $\mathbf{x}'$ (multiple objects) is added and then non-fitting objects are removed from $\mathbf{x} = \mathbf{x}_i \cup \mathbf{x}'$. The MBD algorithm has been designed as a continuous-time reversible process, which is then discretized using a Markov chain. The authors show that this approximation guarantees weak convergence to the desired distribution π.
The algorithm is presented in Algorithm 6, where as before Λ is the intensity of the underlying Poisson process and T is the temperature parameter used for the simulated annealing scheme, while α_Λ and α_T are parameters used to decrease the intensity of the Poisson process and the temperature respectively.

Algorithm 6 Multiple birth and death sampler proposed by Descombes et al. [2009].

1. Initialize $X_0 = \mathbf{x_0}$ and $i = 0$;
2. $\Lambda = \Lambda_0$ and $T = T_0$
3. Repeat:
   - Birth-step: generate $\mathbf{x}'$, a realization of a Poisson process of intensity Λ
   - $\mathbf{x} = \mathbf{x}_i \cup \mathbf{x}'$
   - Death-step: For each $x_i \in \mathbf{x}$, compute $a_T(x_i) = \exp^{\frac{U(\mathbf{x} \setminus x_i) - U(\mathbf{x})}{T}}$, and draw p from a uniform distribution
   - If $p < \frac{\Lambda a_T(x_i)}{1 + \Lambda a_T(x_i)}$ then remove x; $\mathbf{x} \leftarrow \mathbf{x} \setminus x$
   - Set $\mathbf{x}_{i+1} = \mathbf{x}$, $\Lambda_{i+1} = \Lambda_i a_\Lambda$, $T_{i+1} = T_i a_T$, $i = i + 1$ and go to Birth-step.

A modified version of the MBD is the multiple birth and cut (MBC) algorithm developed by Gamal-Eldin et al. [2010]. MBC proposes to use the graph-cut algorithm for energy minimization. Each object in the configuration represents a node in the graph, while the edge weights are assigned based on the energy of each object. Then, the min-cut max-flow algorithm is used to find the minimum cut that maximizes the flow within the graph and thus, minimizes the energy function. More details can be found in [Gamal-Eldin et al., 2010, Gamal Eldin et al., 2011].

3.3.3 Jump-diffusion processes

Starting with the seminal paper of Merton [1976] jump-diffusion processes have been intensively studied in the academic finance community. In computer vision, these processes became known through the work of Grenander and Miller [1994] on building algorithms for automatic hypothesis formation and was successfully applied afterward in applications such as target tracking [Srivastava et al., 1995, 2002], image

segmentation [Han et al., 2004] or more recently in texture analysis [Lafarge et al., 2010a] and 3D stereo reconstruction [Lafarge et al., 2010b].
The jump-diffusion sampler combines the MCMC [Hastings, 1970, Green, 1995] and the Langevin equations [Geman and Huang, 1986]. Each dynamic plays a different role within the sampler: the jump dynamics are used to perform reversible dimensional jumps between the subspaces of Ω, with $\Omega = \bigcup_{n\in\mathbb{N}} \Omega_n$, whereas the diffusion dynamics are used to conduct a stochastic diffusion within each continuous subspace Ω_n driven by Brownian motions.

3.3.3.1 Jump process

The jump process is a MCMC algorithm, generally a RJMCMC algorithm [Descombes et al., 2011]. Reversible jumps are performed between the different subspaces of Ω according to the birth and death kernels discussed in Section 3.3.1.4. Local perturbations are no longer necessary as they are substituted by the diffusion process that allows for an efficient exploration of the subspace.

3.3.3.2 Diffusion process

The diffusion process is used to explore the subspaces continuously. In contrast to the Metropolis-Hastings-Green algorithm, these algorithms are characterized by the absence of rejection. The dynamics through which the current configuration evolves at every iteration can be interpreted as a stochastic gradient descent. The diffusion process is based on the Langevin equations which we will further detail in this section.
Let X be a point process on W and suppose that X follows as Gibbs distribution such that $f(X) = \frac{1}{c}\exp^{-U(X)}$. An alternative to the Metropolis-Hastings-Green algorithm is to define a diffusion equation which converges toward the target distribution $\pi(dX)$. The following stochastic differential equation gives such dynamics:

$$dX_t = -\nabla U(X_t)dt + W_t, \tag{3.73}$$

where X_t, $t \in \mathbb{R}^+$ and W_t is a Brownian motion.
The diffusion equation has to be discretized in order to be simulated. These dynamics are known as the Langevin dynamics. Let $\tau(\delta) = \{\tau_i, i = \overline{0,t}\}$ be a discretization of the interval $[0,t]$ by the time steps $\delta_i = \tau_{i+1} - \tau_i$, then the discrete approximation of the process X can be written $\forall u \in W$ as:

$$\begin{cases} z_u(0) & = x_u(0) \\ z_u(i+1) & = z_u(i) + a_u(Z(i))\delta_i + W_u(\tau_{i+1}) - W_u(\tau_i), \end{cases} \tag{3.74}$$

where $a_u(Z(i)) = -\nabla_u U(Z(i))$. A centered Gaussian distribution with variance 1 can be used to simulate the discrete version of the Brownian motion $W_u(\tau_{i+1}) - W_u(\tau_i)$.
According to Descombes et al. [2011], the jump-diffusion algorithm is usually more effective than the MCMC sampler. Nevertheless, the Gibbs energy associated with

the density function must satisfy the Lipschitz-continuity condition [Geman and Huang, 1986].

3.3.3.3 Jump and diffusion coordination

The global process is controlled by a relaxation parameter called temperature. The continuous diffusion is interrupted by jumps at times t_i which are distributed according to a Poisson distribution. As the diffusion is discretized using a time step δ_i, the discrete waiting times, w, can be computed as:

$$w = \frac{t_{i+1} - t_i}{\delta_i} \sim p(w) = \frac{\tau^w}{w!} \exp^{-\tau}, \tag{3.75}$$

where $\tau = \mathbb{E}(w)$ represents the expected waiting time which controls the frequency of the jumps [Descombes et al., 2011].

3.3.4 Simulated annealing

The convergence towards an optimum for all the samplers presented above is guaranteed by embedding the sampler in a simulated annealing scheme. This is true when using the Maximum Likelihood estimator, although it is not mandatory if other estimators are used. Simulated annealing was developed by Metropolis et al. [1953] and proposes to reach the global maximum of a density $h(\cdot)$ by constructing a sequence of densities of the form:

$$h_i(\mathbf{x}) \propto h^{\frac{1}{T_i}}(\mathbf{x}), \tag{3.76}$$

where T_i is a sequence of temperatures that tends towards 0 as i goes to infinity. Simulated annealing has been proven to converge for reversible jump MCMC algorithm by Stoica et al. [2005] and for birth and death processes by van Lieshout [1994]. However, the choice of the temperature decrease function has a significant impact on the quality of the simulated annealing. This temperature parameter is controlled using a cooling schedule. The cooling schedule is composed of four stages:

1. **The temperature is increased**. As a result, the density becomes close to a uniform distribution. This step ensures that the initial temperature is adequate for the given density function. If the starting temperature is too low, the process will most likely end up in a local minimum. If the starting temperature is too high, the convergence is not affected, however, the computation cost increases;

2. **The temperature decreases slowly**. During this step the process explores the different modes of the density and becomes more restrictive with the number of iterations. If the temperature decrease is slow enough, the process will identify one of the global maxima of the density function;

3. **The temperature is near zero**. During this step small adjustments are made to the configuration of objects. The density finally reaches one of the global optima;

4. **Convergence is reached**. The stopping condition has been attained. In general, the cooling schedule stops when the energy of the process has not evolved for a certain number of iterations or when a certain temperature is reached.

It can easily be observed that the choice of the temperature decrease function is crucial for reaching the global maximum. The most popular decrease functions are discussed below.

3.3.4.1 Logarithmic decrease

The logarithmic decrease given by

$$T_i = \frac{C}{log(i+1)} \tag{3.77}$$

has been proven in theory to ensure the convergence towards the global optimum when the constant C is larger than the deepest local maximum. However, in practice it is impossible to foresee the deepest local maximum and the decrease of the temperature is very slow which renders this schedule impractical for implementation purposes. Instead, a geometric decrease function is usually preferred in practice. Note that in this case however, the convergence to the global optimum is no longer guaranteed.

3.3.4.2 Geometric decrease

The geometric decrease is given by

$$T_{i+1} = \begin{cases} T_i & \text{if } i \mod N \neq 0 \\ \alpha T_i & \text{otherwise,} \end{cases} \tag{3.78}$$

where $0 < \alpha < 1$ is a coefficient that is used to drive the speed of the temperature decrease. This scheme proposes the use of a plateau of length N. This plateau is particularly useful to test the convergence of the process. Finally, the temperature decrease function does not have to be fixed, it can be adaptive as proposed by Ortner et al. [2007].

3.3.4.3 Adaptive decrease

The adaptive decrease function allows the temperature to accustom itself based on the evolution of the energy of the system. Several such schemes have been proposed in literature. The interested reader can refer to [Ortner et al., 2007] for more details.

3.4 Conclusions

The framework of point processes and in particular spatial and spatio-temporal marked point processes is a suitable tool for extracting and tracking objects in high resolution satellite images. Geometric properties can be easily represented into the framework by marks and thus, the image or video analysis can be performed at object level which is more robust to noise as opposed to pixel-based approaches.
Mathematical tools for optimizing marked point processes models have been successfully developed over the past decade which allow an efficient simulation of such models. As a result, a data-parallel implementation of the RJMCMC sampler embedded in a simulated annealing scheme has been developed which not only has good convergence properties but does this in a relatively short amount of time.
The increased flexibility in model design, the good convergence properties of the RJMCMC sampler and the relatively short computation time required for the optimization process to reach a strong local maximum make this framework particularly attractive for object detection and tracking in remotely sensed satellite data.

Chapter 4

Spatial marked point process model for boat extraction in harbors

Contents

As presented in chapter 1, boat extraction in harbors is a difficult problem due to the particular distribution of the boats. A first model to detect boats in harbors in optical satellite images was developed by Ben Hadj et al. [2010a]. The authors placed three strong constraints on the position and orientation of boats, as follows:

1. all objects must have the same orientation;
2. objects should be tangent to each other and their centers should be aligned;
3. objects must not overlap more than a given extent.

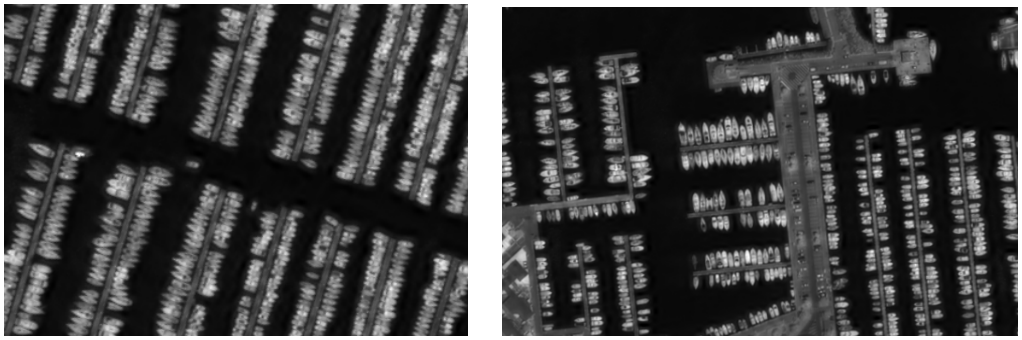


Figure 4.1: Left: Particular case of harbor where all boats have the same orientation. Right: General case of harbors where boats have different orientations, based on their position in the harbor.

Though, this model might be adequate for particular cases of harbors, the first two constraints are too restrictive for the general case. First, the orientation of boats in a harbor varies, as shown in Figure 4.1. Second, the size of neighboring objects can differ and thus, their centers will be shifted. In this chapter, a new model is developed that handles general cases such as the ones previously presented. The assumptions made on the objects are the following:

1. the orientation of each boat is determined locally;
2. neighboring boats should have similar orientation;
3. boats should not overlap more than a given extent.

The first assumption implies that the orientation of boats is determined locally by using cues available in the image. The second assumption restricts neighboring boats from having completely different orientations which is generally true in harbors. Finally, as in the model of Ben Hadj et al. [2010a], the amount of overlap between objects should be kept relatively small.

4.1 Model

Consider a marked point process of ellipses. The object space, $\mathcal{W}$, is a bounded set in $\mathbb{R}^5$ defined as:

$$\mathcal{W} = \mathcal{K} \times \mathcal{M} = [0, IW_M] \times [0, IH_M] \times [a_m, a_M] \times [b_m, b_M] \times [0, \pi]. \tag{4.1}$$

Here, IW_M and IH_M represent the width and height of the image $\mathbf{y}$, respectively, $[a_m, a_M]$ is the range for the length of the semi-major axis, $[b_m, b_M]$ is the range for the length of the semi-minor axis and $\omega \in [0, \pi]$ is the orientation of the ellipse. The image observation $\mathbf{y} = [0, IW_M] \times [0, IH_M]$ is a 2-dimensional array of $IW_M \times IH_M$ pixels. As we consider only gray-scale images, the value of a pixel belonging to $\mathbf{y}$ is given by a real number. This model is a particular case of the observation model

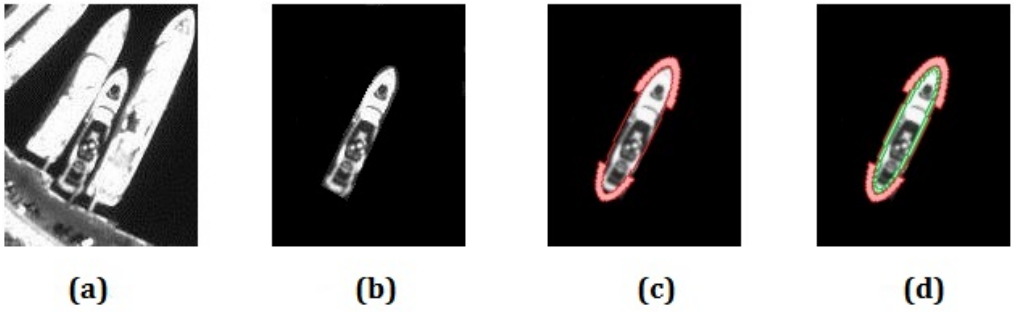


Figure 4.2: (a) Close-up on an area with neighboring boats within the harbor; (b) A typical example of a boat where the pilothouse and superstructures appear as dark spots within the boat; (c) In red: the exterior border $\mathcal{F}^{\rho}(u)$ used to compute the contrast distance to the interior of the ellipse; (d) In red: the exterior border $\mathcal{F}^{\rho}(u)$ and in green: the interior border $\mathcal{I}^{1}(u)$.

used by Vo et al. [2010] to jointly detect and estimate multiple objects from image observations.
A Gibbs process is used to model the density function f which is defined as follows:

$$f_{\theta}(X = \mathbf{x}|\mathbf{y}) = \frac{1}{c(\theta|\mathbf{y})} \exp^{-U_{\theta}(\mathbf{x},\mathbf{y})} \tag{4.2}$$

with

$$c(\theta|\mathbf{y}) = \int_{\Omega} \exp^{-U_{\theta}(\mathbf{x},\mathbf{y})} \mu(d\mathbf{x}) \tag{4.3}$$

being the normalizing constant.
The model is described in terms of the energy function $U_{\theta}(\mathbf{x}, \mathbf{y})$, which is composed of two terms: an external energy term which reflects how good the model fits the actual data, denoted $U^{ext}_{\theta_{ext}}(\mathbf{x}, \mathbf{y})$, and an internal energy term including constraints imposed on the configuration, denoted $U^{int}_{\theta_{int}}(\mathbf{x})$. The vector θ_{ext} contains the parameters of the external energy term that have to be estimated, while θ_{int} contains the parameters of the internal energy term. Hence, the parameter vector for the entire model can be written as $\theta = \{\theta_{ext}, \theta_{int}\}$.

4.1.1 External energy term

In this approach, the computation of the external energy term takes place locally, for each ellipse. The external energy of the entire configuration is computed as the sum of the individual external energies:

$$U^{ext}_{\theta_{ext}}(\mathbf{x}, \mathbf{y}) = \gamma_{cnt} \sum_{u \in \mathbf{x}} U^{ext}(u, \mathbf{y}), \tag{4.4}$$

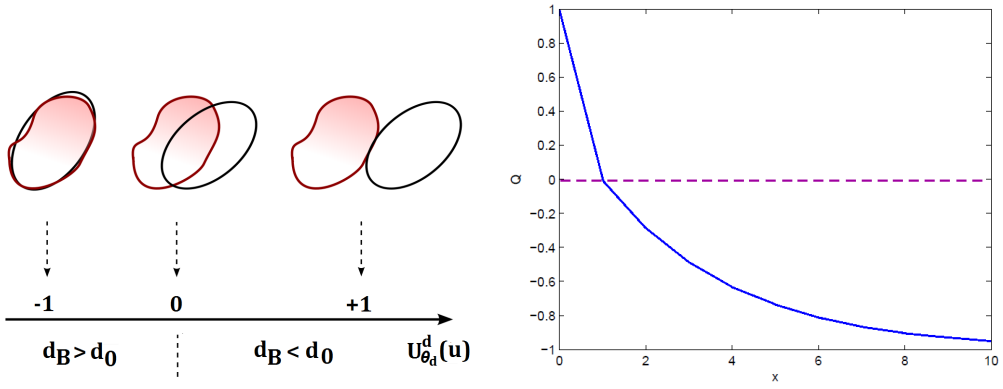


Figure 4.3: Left:Visual representation of the quality function used to construct the external term in equation 4.5. The more an ellipse from a configuration $\mathbf{x}$ overlaps an object in the image, the lower the value of the quality function and thus, the lower the external energy term. If an ellipse does not overlap with an object in the image, the quality function returns a high value which in turn leads to a higher value for the external term, making the ellipse less probable to be part of the true configuration. Right: Visual representation of the quality function w.r.t. its argument.

where γ_{cnt} represents the external energy weight and has to be estimated.
The computation of the local external energy term, $U^{ext}(u, \mathbf{y})$, relies on the use of a contrast distance measure $d_B(\cdot, \cdot)$ between the interior of the ellipse, denoted by u, and two predefined borders, as shown in Figure 4.2. The external border, $\mathcal{F}^\rho(u)$, is used to detect the object with respect to the background. Yet, when seen from above, most boats have a dark spot where the pilothouse or superstructures are located. This leads to a lower contrast distance and thus a lower detection probability. Therefore, an interior border, $\mathcal{I}^1(u)$, is also considered. The effects of computing the distance between the interior and external borders are twofold: first, the overall detection probability increases when this distance is high; and second, the false alarm rate decreases. The external term becomes therefore:

$$U^{ext}(u, \mathbf{y}) = \mathcal{Q}\left(\frac{d_B(u, \mathcal{F}^\rho(u))}{d_0(\mathbf{y})}\right) + \gamma_c \mathcal{Q}\left(\frac{d_B(\mathcal{I}^1(u), \mathcal{F}^\rho(u))}{d_0(\mathbf{y})}\right), \tag{4.5}$$

where γ_c is the weight of the contrast measure computed using the interior border. The contrast distance measure, $d_B(\cdot, \cdot)$, is similar to the Bhattacharyya distance (see Goudail et al. [2004]):

$$d_B(p, q) = \left[\frac{(\mu_p - \mu_q)^2}{4\sqrt{\sigma_p^2 + \sigma_q^2}} - \frac{1}{2}\log\left(\frac{2\sqrt{\sigma_p^2 \sigma_q^2}}{\sigma_p^2 + \sigma_q^2}\right)\right] \tag{4.6}$$

where $(\mu_\cdot, \sigma_\cdot^2)$ represent empirical means and variances.
The threshold $d_0(\mathbf{y})$ for the contrast is determined based on the image, $\mathbf{y}$ and will

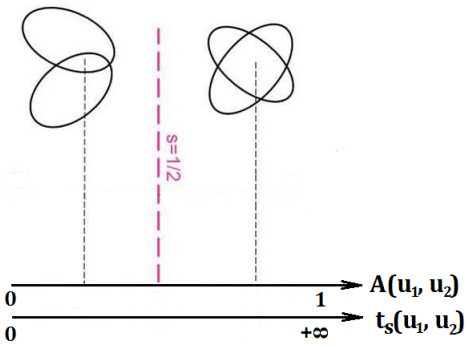


Figure 4.4: Visual representation of overlapping ellipses w.r.t. a fixed overlapping ratio $s = 0.5$. The energy of the configuration increases as the amount of overlapping between objects increases.

be discussed in section 4.2.
Finally,

$$\mathcal{Q}(x) = \begin{cases} 1 - x^{1/3} & \text{if } x < 1 \\ exp(-\frac{x-1}{3}) - 1 & \text{if } x \geq 1 \end{cases} \tag{4.7}$$

is a quality function, depicted in Figure 4.3, that is defined on $\mathcal{Q} : \mathbb{R}^+ \rightarrowtail [-1, 1]$. The lower the energy of one configuration, the better the configuration fits the image. Thus, the quality function attributes a negative value to well placed objects (e.g. those objects u for which $d_B(\cdot, \cdot)$ is higher than the threshold $d_0(\mathbf{y})$) and a positive value to misplaced objects. The use of the cubic root allows for a moderate penalization when the output of the distance measure is near the threshold.

4.1.2 Internal energy term

The internal energy term is decomposed into two parts that incorporate the three assumptions made in the beginning of this chapter. The first part takes care of overlapping, meaning that objects are not allowed to overlap more than a given extent. The second part handles the constraints related to the orientation and alignment of the ellipses. The local orientation of boats is determined based on image cues and ellipses which have orientations close to the expected one are favored. Moreover, configurations where neighboring objects have similar orientations are also favored.

4.1.2.1 Non-overlap constraint

The first part of the internal energy term corresponds to a penalization of overlapping ellipses, avoiding the detection of the same object several times. The proposed model uses a *hard core* process to handle object overlapping, meaning that all configurations containing ellipses that overlap more than a given extent will be disregarded

(i.e. the energy value assigned to such configurations will be infinitely high). Hence, denoting by $A(u_i, u_j) = \frac{Area(u_i \cap u_j)}{\min(Area(u_i), Area(u_j))}$ the area of intersection between the objects u_i and u_j, the non-overlapping energy term can be defined as:

$$U_o^{int}(\mathbf{x}) = \sum_{1 \leq i \neq j \leq n(\mathbf{x})} t_s(u_i, u_j) \tag{4.8}$$

with:

$$t_s(u_i, u_j) = \begin{cases} 0 & \text{if } A(u_i, u_j) < s \\ +\infty & \text{otherwise} \end{cases} \tag{4.9}$$

where $s \in [0, 1]$ corresponds to the amount of overlapping allowed by the model and $n(\mathbf{x})$ is the number of objects in the configuration $\mathbf{x}$. A visual representation of the non-overlapping constraint is presented in Figure 4.4. Accordingly, all configurations containing at least two objects that overlap to a higher ratio than specified by s are prohibited.

4.1.2.2 Determining the local orientation of the objects

A method proposed by Li and Briggs [2009] for road extraction in high resolution satellite images of urban areas, is used in order to locally determine the orientation of the docks using the water. The idea is to apply an edge detector and then identify long, straight lines using the Hough transform based on the edge detector response. The orientation of the lines can then be easily determined.
The Canny-Deriche edge detector which was initially proposed by Canny [1986] and then extended by Deriche [1987] is used to separate the water area from the other structures in the image. Since all pixels in the water area have similar values in the absence of high waves, most edges are due to the high contrast between the water area and the land structures or boats. Any edge detector can be applied. The Canny-Deriche edge detector was selected for its moderate edge output. Both high oversegmentation or undersegmentation leads to lower accuracy in the following steps.
Li and Briggs [2009] described two key concepts: *reference circles* and *central pixels*. For each pixel p, its reference circle $C(p)$ is the circle with the largest radius, centered at p, that does not contain edge points. In other words, the radius of the reference circle is the maximum distance from the pixel to the closest edge point. A pixel is considered a central pixel, if it has the largest reference circle among the neighboring pixels. These concepts are illustrated in Figure 4.5. The idea behind these concepts is that all the central pixels form the center line of the water area. The incomplete or inaccurate edges will slightly affect the location of the central pixels, but the method has high tolerance over such error. The central pixels are found using the distance transform based on the result of the edge detection. The central pixels are plotted onto a binary image, where a pixel value of 1 signals the presence of a central pixel at that location and a pixel value of 0 signals the absence of a central pixel. After extracting the central pixels, a Hough transform is applied to detect the lines in the resulted binary image and compute their orientation. A visual description of the

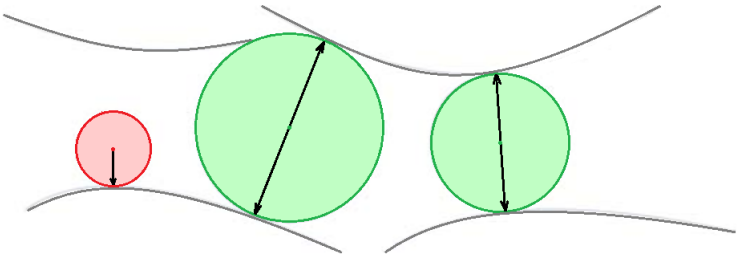


Figure 4.5: Visual representation of the concepts of *reference circles* and *center pixels* used to determine the local orientations of objects in the scene.

method is presented in Figure 4.6. Thus, the orientation of the water area has been established locally and a set of relevant pixels in the image has been obtained. Each such pixel contains information about the orientation of the water in that area.

4.1.2.3 Alignment and orientation constraints

As stated in the beginning of this chapter, neighboring boats should have similar orientations. Thus, an alignment interaction between two neighboring ellipses u_1 and u_2 is defined in the following way:

$$u_1 \sim_{al} u_2 \Leftrightarrow \begin{cases} d_\omega(u_1, u_2) \leq d_{\omega_{max}} \\ d_C(u_1, u_2) \leq d_{C_{max}} \end{cases} \tag{4.10}$$

where $d_\omega(u_1, u_2) = |\omega_1 - \omega_2|$ is the difference of the orientations of the two ellipses u_1 and u_2 and $d_{\omega_{max}}$ is the maximum angle allowed between two neighboring objects, while $d_C(u_1, u_2) = |d(c_1, c_2) - (b_1 + b_2)|$, where $d(c_1, c_2)$ stands for the Euclidean distance between the centers of the ellipses and $d_{C_{max}}$ is the maximum distance allowed.
Then, an internal energy term that promotes this alignment is designed in the following way:

$$U_{al}(u_1, u_2) = \begin{cases} \delta\varpi(d_\omega(u_1, u_2), d_{\omega_{max}}) \\ \qquad \text{if } u_1 \sim_{al} u_2 \\ 0 \qquad \text{otherwise} \end{cases} \tag{4.11}$$

where $\varpi(x, x_{max})$ is a reward function introduced by Ortner et al. [2008], that favors alignment and is defined as:

$$\varpi(x, x_{max}) = -\frac{1}{x_{max}^2}\left[\frac{1 + x_{max}^2}{1 + x^2} - 1\right], \; for \; x \leq x_{max}. \tag{4.12}$$

In the previous subsection, the local orientation of objects in a scene has been determined based on the method proposed by Li and Briggs [2009]. This predetermined

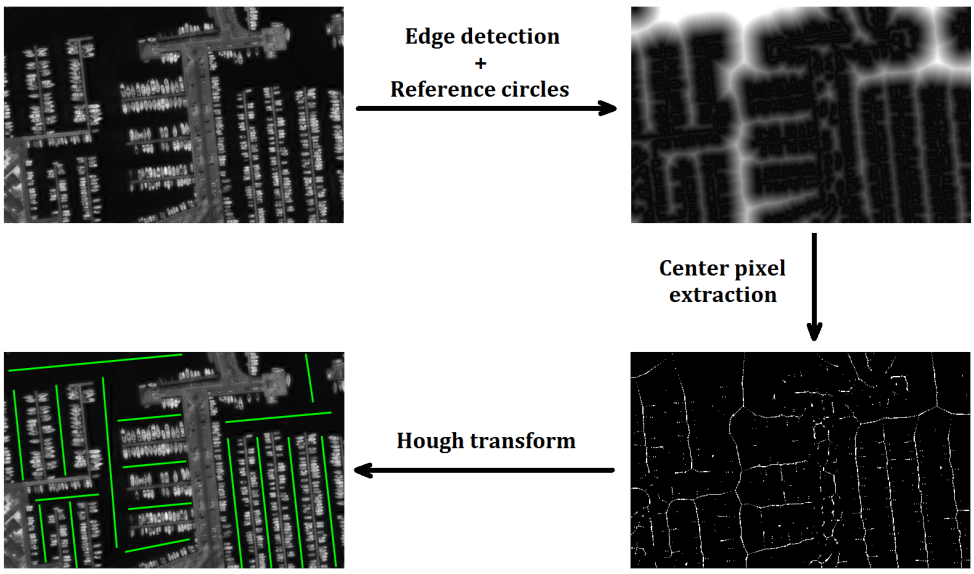


Figure 4.6: Work-flow of the local orientation detection method. Top-left: Initial image. Top-right: Visualization of the reference circles. The brighter the pixel values, the larger the radius of the reference circle. Bottom-right: Center pixel extraction based on the radius of the reference circles. Bottom-left: Lines obtained using the Hough transform. The orientation of these lines is used to determine the local orientation of the objects.

orientation can be used to favor ellipses which tend to align with it. Given an ellipse u, the nearest relevant pixel in its spatial neighborhood is identified. If no relevant pixel is found within a certain distance, ellipses that have similar orientations to other ellipses in their neighborhood are favored. If a relevant pixel is found, its position w.r.t. the semi-axes of the ellipse is determined and ellipses perpendicular to the orientation indicated by the relevant pixel are favored. Thus, the final internal energy term that refers to both orientation and alignment becomes:

$$U^{int}_{al_{\omega_l}}(\mathbf{x}) = \sum_{u \in \mathbf{x}} U^{int}_{al_{\omega_l}}(u) \tag{4.13}$$

where:

$$U^{int}_{al_{\omega_l}}(u) = \begin{cases} 0 & \text{if a relevant pixel is found and} \\ & ||\omega_u - \omega_l|| > d_{\omega_{max}} \\ \gamma_{al} \sum_{v \in \mathbf{x}} U_{al}(u,v) & \text{otherwise} \end{cases} \tag{4.14}$$

where ω_u is the orientation of object u and ω_l is the orientation perpendicular to the one retained in the relevant pixel.

4.1.2.4 Total internal term

The total internal term is the sum of its two parts and is therefore written as:

$$U^{int}_{\theta_{int}}(\mathbf{x}) = U^{int}_{o}(\mathbf{x}) + U^{int}_{al_{\omega_l}}(\mathbf{x}) \tag{4.15}$$

with $\theta_{int} = \{\gamma_{al}\}$, the weight of the alignment interaction, present in equation 4.14.

4.1.3 Total energy term

The total energy term is written as the sum of the external energy and internal energy terms, as follows:

$$\begin{aligned} U_{\theta}(\mathbf{x},\mathbf{y}) &= U^{ext}_{\theta_{ext}}(\mathbf{x},\mathbf{y}) + U^{int}_{\theta_{int}}(\mathbf{x}) \\ &= \gamma_{cnt} \textstyle\sum_{u\in\mathbf{x}} U^{ext}(u,\mathbf{y}) + U^{int}_{o}(\mathbf{x}) + U^{int}_{al_{\omega_l}}(\mathbf{x}). \end{aligned} \tag{4.16}$$

The parameters of the model are described by the parameter vector $\theta = [\theta_{ext}, \theta int] = [\gamma_{cnt}, \gamma_c, d_0, \rho, s, \gamma_{al}]$. The next section presents a detailed description of how each of these parameters are determined.

4.2 Parameter estimation

Modeling an image using a marked point process usually results in a heavy model with a large number of parameters. In this section, the most important parameters of the previously described model are presented in more detail. As a reminder, the parameter vector is $\theta = [\gamma_{cnt}, \gamma_c, d_0, \rho, s, \gamma_{al}]$, where

- γ_{cnt} is the weight of the external energy term w.r.t. the internal energy term;
- γ_c is the weight of the contrast distance between the interior border and the exterior border w.r.t. the contrast distance between the interior of the ellipse and the exterior border;
- d_0 is the threshold used in the quality function $\mathcal{Q}(\cdot)$ to assess the contrast distance;
- ρ is the width of the exterior border of an ellipse;
- $s \in [0, 1]$ is the allowed overlapping ratio;
- γ_{al} is the weight of the alignment term with respect to the overlapping term within the internal energy.

Estimating automatically the entire parameter vector θ is a difficult task. Therefore, the parameters that have to be estimated need to be narrowed down. Indeed, the parameters ρ and s can be easily set empirically as they have a straight forward physical interpretation. As such, ρ is set to $\rho = 2$. This value can be easily motivated: if $\rho = 1$, the number of pixels that make up the border is small, resulting

in poor estimates of the mean and the variance of the border, when computing the external energy term; if $\rho \geq 3$, the border will contain pixels from the neighboring objects, which will lead to a higher mean value of the border and thus, a lower contrast distance. A visual representation of the border width ρ is shown in Figure 4.7. The overlapping ratio s is set to $s = 0.1$, meaning that only configurations with objects overlapping less than 10% are allowed. Allowing a small amount of overlapping is crucial for detecting tightly clustered objects, as can be seen in Figure 4.7.

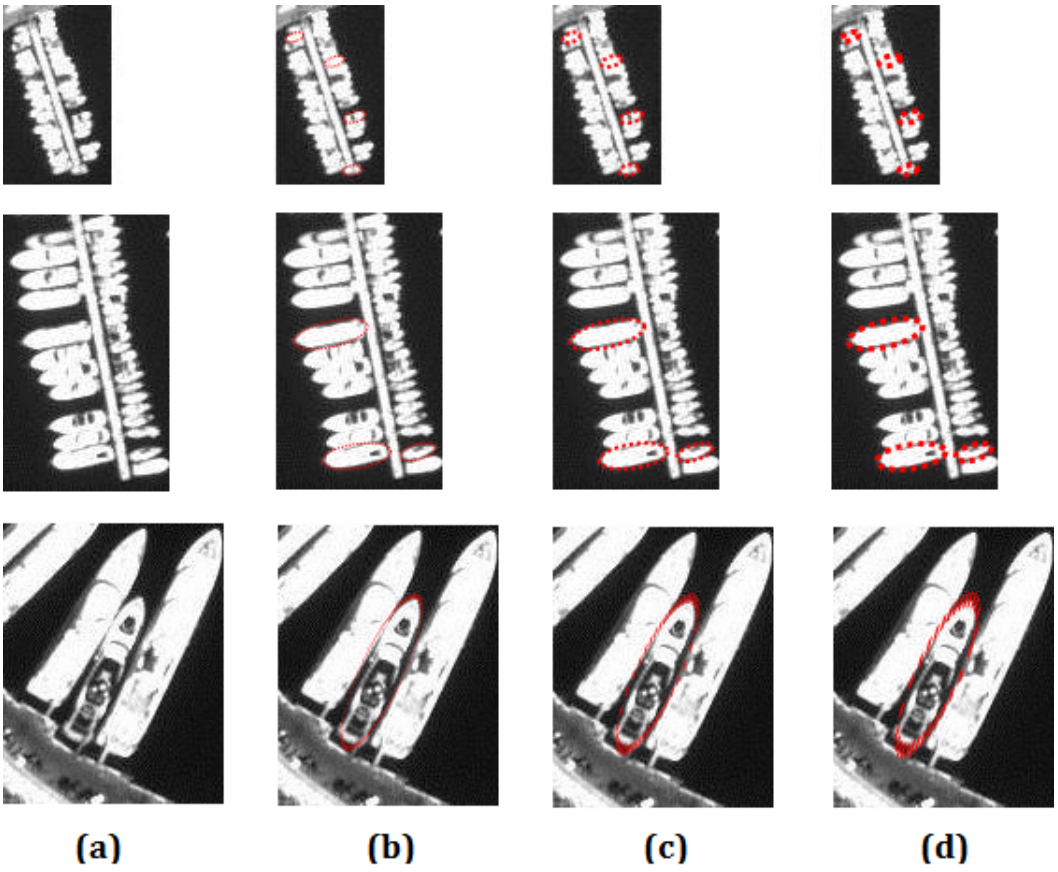


Figure 4.7: A visual representation of the border width ρ for a few selected boats of different sizes. (a) The input image. (b) $\rho = 1$. This is a good value for very small boats, however, it does not provide a sufficient number of pixels to compute the relevant statistics. (c) $\rho = 2$. This is the best value that fits both small and large boats. Mean and variance of the border pixels can be computed for small boats as well. (d) $\rho = 3$. In this case, the border significantly overlaps with neighboring objects (both boats and docks) and the contrast value drops accordingly. For large objects, the additional pixel in the border width does not lead to significant changes in the border statistics.

The other parameters however are more difficult to set by hand. Nevertheless, trying to estimate all remaining parameters with EM-like methods results in a high computational complexity and low convergence properties. Therefore, the weights γ_{cnt} and γ_{al} are estimated using the SEM algorithm presented in chapter 3, while heuristics are devised for the parameters d_0 and γ_c.

4.2.1 Determining the weights γ_{cnt} and γ_{al}

The main difficulty in estimating these parameters draws from the fact that both the configuration of objects $\mathbf{x}$ and the marginal density of the observations $f_\theta(\mathbf{y})$ are unknown. In such situations, EM-like algorithms are known to be efficient. This type of iterative algorithms provide an estimate of θ based on the search for a local maximum of the image likelihood $f_\theta(\mathbf{y})$. Nevertheless, the EM algorithm is intractable for the proposed model. Thus, a stochastic variant of the EM, called SEM [Celeux and Diebolt, 1985] (see Chapter 3), is an appropriate solution in this case. Nevertheless, it is not possible to obtain a tractable expression of the expectation since the normalizing constant in eq. 4.3 is itself intractable. Thus, the likelihood is approximated using the pseudo-likelihood defined in eq. 3.21, where the extended Papangelou intensity $\lambda_\theta(u;\mathbf{x},\mathbf{y})$ associated with object u is written as:

$$\lambda_\theta(u;\mathbf{x},\mathbf{y}) = \beta \exp\left(-\gamma_{cnt} U^{ext}(u,\mathbf{y}) - \sum_{v\in\mathbf{x}, v\neq u} t_s(u,v) - \gamma_{al} \sum_{v\in\mathbf{x}, v\neq u} U_{al}(u,v)\right). \tag{4.17}$$

The initialization step is very important for the convergence of the SEM to a good solution [Robert and Casella, 2005]. For the proposed model, γ_{cnt} and γ_{al} are initialized such that the objects in the first configuration are independent. More precisely, γ^0_{cnt} and γ^0_{al} are unique non-zero roots of $\int_{\mathcal{W}} \lambda_\theta(u;\emptyset,\mathbf{y})\Lambda(du) - \beta$, where $\lambda_\theta(u;\emptyset,\mathbf{y})$ is the Papangelou intensity associated with the empty configuration $\mathbf{x} = \emptyset$. Empirical tests have shown that γ^0_{al} should have a lower value than γ^0_{cnt}.

4.2.2 Determining the weight γ_c

The weight γ_c determines the influence of the contrast distance between the interior border and the exterior border within the total data energy term. As shown in Figure 4.2, a dark spot usually appears in the middle or lower part of the object, where the pilothouse or superstructures are. This dark spot decreases the contrast distance between the exterior border and the interior of the ellipse. Therefore, an interior border, $\mathcal{I}^1(\cdot)$ of fixed width equal to 1 is also considered. Given the small width of $\mathcal{I}^1(\cdot)$, generally only a few pixels make up the interior border. Thus, the main role of this term is to shift the balance in cases where, given an object u and its border $\mathcal{F}^\rho(u)$, the contrast distance $d_B(u,\mathcal{F}^\rho(u))$ is close to the threshold d_0. Trial and error experiments have shown that for $\gamma_c \leq 0.6$, $\mathcal{Q}\left(\frac{d_B(\mathcal{I}^1(u),\mathcal{F}^\rho(u))}{d_0(\mathbf{y})}\right)$ does not have a significant effect on the overall external energy term. On the other hand,

if $\gamma_c \geq 1.5$, the overall external energy term for true positives increases. Thus, good values for γ_c are $\gamma_c \in [0.7, 1.4]$.

4.2.3 Determining the threshold d_0

The threshold d_0 for the contrast distance measure is determined using a heuristic approach. Since the images considered contain mostly water and the objects of interest (i.e. boats), the pixels are roughly divided into two classes in the following way:

1. Choose an initial threshold, $D_{i=0} = D_0$. Usually, D_0 is taken to be the middle of the grey-level range;
2. Segment the image into two classes, objects and background, as follows:
   - object class: $C_o = \{I(m,n) : I(m,n) > D\}$, where $I(m,n)$ represents the pixel grey-level at location (m,n) in the image I.
   - background class: $C_b = \{I(m,n) : I(m,n) \leq D\}$.
3. Compute the average values of the two classes such that $m_1 = avg(C_1)$ and $m_2 = avg(C_2)$;
4. The threshold at iteration i is computed as the arithmetic mean of the two averages: $D_i = (m_1 + m_2)/2$;
5. Repeat steps 2 - 4 using the newly computed threshold until it converges ($D_{i+1} = D_i$).

The final threshold is obtained using the formula: $d_0 = (m_1/m_2) * (g_{max}/(g_{max} - D_i))$, where g_{max} is the maximum grey-level value.

4.3 Optimization

A marked point process is fully defined by its unnormalized density $f(\mathbf{x})$ w.r.t. to a reference measure, which in this model is the homogeneous Poisson measure [Stoyan and Stoyan, 1994]. Simulating point processes is usually done using the reversible jump Markov chain Monte Carlo (RJMCMC) sampler [Green, 1995], presented in chapter 3. RJMCMC simulates a discrete Markov chain $(X_i)_{i \in N}$ on the configuration space Ω that converges towards an invariant measure specified by the energy U of the model. At each step, the current configuration $\mathbf{x}$ of the chain is perturbed to a new configuration $\mathbf{x}'$ according to a transition kernel $\{Q_m(\mathbf{x}, \cdot)\}_{m \in M}$. The new configuration $\mathbf{x}'$ is accepted with a certain probability, called the Green ratio. For better results, the RJMCMC is usually embedded into a simulated annealing scheme [Descombes et al., 2011] and thus, the Green ratio is written as:

$$R = \frac{Q_m(\mathbf{x} \to \mathbf{x}')}{Q_m(\mathbf{x}' \to \mathbf{x})} \exp\left(\frac{U(\mathbf{x}) - U(\mathbf{x}')}{T_i}\right), \tag{4.18}$$

where T_i is a relaxation parameter, also called temperature. The standard transition kernels employed to simulate the proposed model are: object creation/removal (generally known as the Birth and Death kernel) and local perturbation kernels (e.g. rotation, translation and scale).
The efficiency of the optimization procedure depends on several parameters such as the size of the search space $\mathcal{K}$, the size of the object space $\mathcal{W}$, the distribution and number of objects in the scene, the sampler used, etc. The size of the object space $\mathcal{W}$ is predetermined by the model and the distribution and number of objects is considered to be unknown [Ortner, 2004]. However, the size of the search space in the particular case of boat detection can be reduced by using the simple heuristic that boats are expected to appear only in water areas. Furthermore, a parallel implementation of the RJMCMC sampler has already been introduced in chapter 3.
In this section, we describe the perturbation kernels used to simulate the marked point process model presented in Section 4.1. Then, the drawbacks of using the parallel implementation devised by Verdié and Lafarge [2012] are illustrated and an alternative multiple-core implementation that addresses the identified drawbacks is described. Finally, a water / land discrimination algorithm to reduce the search space $\mathcal{K}$ is presented.

4.3.1 Perturbation kernels used for object detection

Consider the marked point process model presented in this chapter. The object space is $W = K \times M \subset \mathbb{R}^2 \times \mathbb{R}^3$ and an object u is given by $u = (x_u, y_u, a_u, b_u, \omega_u)$, where (x_u, y_u) is the position of the object and (a_u, b_u, ω_u) are the semi-major axis, semi-minor axis and orientation of the ellipse. To simulate the model presented in Section 4.1, we have used the following perturbation kernels:

- **Uniform Birth and Death**. This kernel is used to create or delete objects from the configuration and thus, changes the dimension of the configuration space. This standard kernel is a mixture of two sub-kernels, one which creates objects and another which deletes objects from the configuration. This kernel assumes a uniform generation of objects in W, according to the probability law $\frac{\nu(\cdot)}{\nu(W)}$ where $\nu(W)$ is a measure on W, and a uniform removal of objects from the current configuration. The kernel can be written as:

$$Q_{BD}(\mathbf{x}, \cdot) = p_b(\mathbf{x}) Q_B(\mathbf{x}, \cdot) + p_d(\mathbf{x}) Q_D(\mathbf{x}, \cdot), \tag{4.19}$$

  where we take $p_b = p_d = 0.5$ in our experiments, $Q_B(\cdot, \cdot)$ is the birth sub-kernel and $Q_D(\cdot, \cdot)$ is the death sub-kernel. The Green ratios for these sub-kernels can be written as follows:
  Birth: The kernel proposes a new configuration $\mathbf{x} \cup u$ and the associated Green ratio is given by:

$$R_B(\mathbf{x}, \mathbf{x} \cup u) = \frac{h(\mathbf{x} \cup u)}{h(\mathbf{x})} \frac{p_d}{p_b} \frac{\nu(W)}{n(\mathbf{x})}. \tag{4.20}$$

As a reminder, $n(\mathbf{x})$ represents the number of objects in the configuration $\mathbf{x}$.
Death: The kernel proposes a new configuration $\mathbf{x} \setminus u$ and the associated Green ratio is given by:

$$R_D(\mathbf{x}, \mathbf{x} \setminus u) = \frac{h(\mathbf{x} \setminus u)}{h(\mathbf{x})} \frac{p_b}{p_d} \frac{n(\mathbf{x})}{\nu(W)}. \tag{4.21}$$

- **Local perturbations**: It is more efficient to perform small perturbations on an existing object rather than propose a death, immediately followed by a birth. Local perturbations are specifically designed to achieve this efficiency. We use symmetric perturbations in order to guarantee the detailed balance condition. Let $\mathcal{T} = \{T_a : a \in A\}$ be a family of symmetric perturbations parametrized by a. The perturbation kernel associated to this family consists in choosing an object u uniformly and at random from the configuration $\mathbf{x}$ and proposing a perturbation by applying T_a to u such that $v = T_a(u)$. Given that both the perturbation T_a and the object u are chosen randomly, we can write the Green ratio for this kernel as:

$$R_L(\mathbf{x}, (\mathbf{x} \setminus u) \cup v) = \frac{h((\mathbf{x} \setminus u) \cup v)}{h(\mathbf{x})}. \tag{4.22}$$

We have used three types of symmetric perturbations in our experiments: translation, rotation and scale.
The family of **translations** is defined in $[-\delta_x, \delta_x] \times [-\delta_y, \delta_y]$ and a transition $T_{[d_x,d_y}$ translates the center (x_u, y_u) of the ellipse u to a new position, provided that this position is still in $K = [0, I_{h_{max}}] \times [0, I_{w_{max}}]$:

$$T_{[d_x,d_y]} \begin{pmatrix} x_u \\ y_u \\ a_u \\ b_u \\ \omega_u \end{pmatrix} = \begin{bmatrix} (x_u + d_x)[I_{h_{max}}] \\ (y_u + d_y)[I_{w_{max}}] \\ a_u \\ b_u \\ \omega_u \end{bmatrix}. \tag{4.23}$$

A **rotation** is parametrized by a vector $[d_\omega]$, with $d_\omega \in [-\delta_\omega, \delta_\omega]$. This perturbation changes the orientation of an ellipse u:

$$T_{[d_\omega]} \begin{pmatrix} x_u \\ y_u \\ a_u \\ b_u \\ \omega_u \end{pmatrix} = \begin{bmatrix} x_u \\ y_u \\ a_u \\ b_u \\ (\omega_u + d_\omega)[\pi] \end{bmatrix}. \tag{4.24}$$

Finally, a **scale** is parametrized by a vector $[d_a, d_b]$ with $d_a \in [-\delta_a, \delta_a]$ and $d_b \in [-\delta_b, \delta_b]$. A scale perturbation modifies the length of the semi-major and

semi-minor axis of an object u as follows:

$$T_{[d_a,d_b]}\begin{pmatrix} x_u \\ y_u \\ a_u \\ b_u \\ \omega_u \end{pmatrix} = \begin{bmatrix} x_u \\ y_u \\ (a_{min} + (a_u - a_{min} + d_a))[a_{max} - a_{min}] \\ (b_{min} + (b_u - b_{min} + d_b))[b_{max} - b_{min}] \\ \omega_u \end{bmatrix}. \quad (4.25)$$

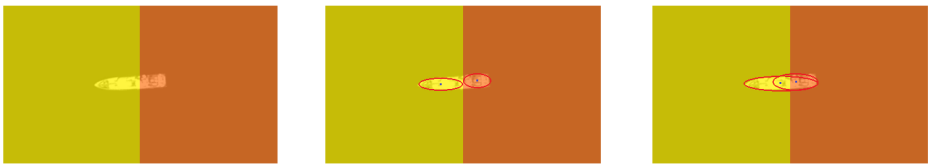


Figure 4.8: Left: A large boat split in two parts due to the space partitioning; Middle: The object is split at cell boundary leading to the detection of two smaller boats; Right: The boat is detected twice if ellipses are allowed to cross the cell boundary.

4.3.2 Efficient implementation of RJMCMC - multiple cores

The Graphical Processing Unit (GPU) parallel implementation of the RJMCMC sampler proposed by Verdié and Lafarge [2012] is based on the underlying assumption that objects do not cross the boundary of a cell more than a certain small value ε. This assumption enables the use of a data-parallel processing model [Mattson et al., 2005]. In other words, there is no need for individual processors to communicate and share data between each other, nor to access a shared memory. Their assumption is generally true for small objects. However, if the objects are large, this assumption is no longer valid. A large object can be contained in two neighboring cells, as shown in Figure 4.8 (left). Imposing a data-parallel processing in each cell independent from the configurations in the neighboring cells results in two possible types of detection errors:

1. The object is split at cell boundary and thus, two smaller objects are detected, as shown in Figure 4.8 (middle);

2. Ellipses are allowed to cross the cell boundary which would lead to several detections of the same object, as shown in Figure 4.8 (right).

4.3.2.1 Challenges in exploiting parallelism

Several challenges have to be considered which arise with the increase in processing elements and the need to coordinate access to shared resources. Parallel programming (i.e. providing a correct and efficient program that takes full advantage of the

underlying parallel architecture) is considerably more challenging than sequential programming due to the added complexity of coordination between the processing elements [Foster, 1995, Trinder et al., 2013].
The following aspects are key issues that have to be considered when developing a parallel implementation [Belikov et al., 2013]:

- **Coordination**. Coordination tackles the access to shared resources, parallelism and inter-process communication. Either the computation or the data (or both) have to be partitioned to facilitate parallel execution;
- **Performance portability**. The performance portability refers to the ability of software to maintain high performance across different parallel architectures;
- **Productivity**. High-level programming models generally offer high productivity and flexibility;
- **Scalability**. Scalability is the ability to increasingly use available resources such as memory size or the number of processing elements to solve larger problems.

When devising a parallel implementation, the programmer must take all these aspects into account. High-level parallel programming models offer an abstraction of the computer system architecture [Mattson et al., 2005]. The most common high-level parallel programming models are:

- **Shared memory model** is a multi-threaded model whose practical implementation is OpenMP. OpenMP [Chapman et al., 2007, OpenMP, 2013] enables a highly structured use of threads. More precisely, the switch between sequential and parallel regions is performed through a fork/join model [Andrews, 1999]. The control thread uses a fork command to split into several independent threads. After the threads finish their execution, they join to resume the sequential execution. The parallel region allows the replication of a single task across a set of threads;
- **Message passing model** is specifically designed to enable communication between processes by interchanging messages. This is a common alternative for distributed systems, where shared variables cannot be used for communication. The Message Passing Interface (MPI) is the most widely used specification for message passing operations [Pacheco, 1996, Gropp et al., 1999]. Send/Receive operations are encoded to facilitate inter-process communication;
- **Heterogeneous models** have been recently developed due to the appearance of heterogeneous systems (i.e. systems that have one or more host CPUs and one or more GPUs). CUDA is such a parallel programming model developed by NVIDIA [NVidia, 2013] and designed to develop applications that scale with the increasing number of cores provided by GPUs. The parallel system consists of a host (i.e. a CPU) and a computation resource (i.e. a GPU) and

the tasks are performed on the GPU through a number of threads. The main advantage of using the GPU is its massively parallel structure. As CUDA is specifically developed for NVIDIA graphical cards, the community worked on producing a royalty-free standard, known as OpenCL [Khronos, 2013], which is designed for general purpose parallel computing across CPUs, GPUs and other processors. OpenCL distinguishes between devices (mainly CPUs or GPUs) and the host (CPU) and the key idea behind this model is to write kernels (e.g. functions that run on OpenCL devices) and application programming interfaces (APIs) for creating and managing these kernels which are then compiled for the targeted device.

Verdié and Lafarge [2012] use CUDA for their GPU implementation. They dedicate a thread to each simultaneous perturbation and show that the more cells in the partition tree, the more efficient the sampler. They also optimize the implementation through memory coalescing (i.e. combining multiple memory accesses into a single transaction) and object indexing.
In terms of computational efficiency, the communication costs of loading the data into memory accounts for a large amount of time. When handling very large data sets, the memory transfer time between the CPU and the GPU can be very long, especially when repeated transfers have to be made. A multi-core implementation does not encounter this lag. Furthermore, portability is still a heavily debated issue for GPU implementations. Hence, we opted for a multi-core implementation of the parallel sampler.

4.3.2.2 Parallel implementation of RJMCMC using multiple cores with shared memory

A solution to handle split objects at cell boundary is not obvious if only the configuration within a cell is considered. However, if the configurations of objects within the neighboring cells are also considered such errors would be easily solvable. Indeed, a shared memory model can be efficiently used to tackle this problem. At each iteration, for each cell in the chosen mic-set S_{mic}, the configurations from the neighboring cells are taken into account when performing perturbations, as can be observed in eq. 4.26. Note that this does not raise synchronization issues between computing units when accessing the shared memory, since the computing units only read the information in the neighboring cells and do not modify it. The sampler we proposed is described in Algorithm 7.
Two additional steps are added to the sampler to make it specific to the problem of boat detection:

1. A pre-processing step is included into the sampler that computes the water mask. We will discuss this step in detail in Section 4.3.3;

2. The space partitioning tree is pruned based on the water mask (step 3).

A generic parallel RJMCMC sampler based on the shared memory distributed processing model can be easily obtained by disregarding the two additional steps mentioned above.

Algorithm 7 Parallel sampler based on the distributed shared memory processing model.

1. Compute the water mask;
2. Initialize $X_0 = \mathbf{x_0}$ and $i = 0$;
3. Compute the data-driven space partitioning tree $\mathcal{K}$ and truncate it based on the water mask;
4. At iteration i with $X_i = \mathbf{x}$:
   - Choose a mic-set $S_{mic} \in \mathcal{K}$ and a kernel type $m \in M$ according to the probability $\sum_{c \in S_{mic}} q_{c,m}$
   - For each cell c in S_{mic}:
     - Perturb $\mathbf{x}$ in cell c to a configuration $\mathbf{x}'$ according to $\mathcal{Q}_{c,m}(\mathbf{x} \to \cdot)$;
     - Retrieve the configuration $\mathbf{z}$ from the neighboring cells;
     - Compute the Green ratio:

$$R = \frac{\mathcal{Q}_{c,\gamma}(\mathbf{x}' \cup \mathbf{z} \to \mathbf{x} \cup \mathbf{z})}{\mathcal{Q}_{c,\gamma}(\mathbf{x} \cup \mathbf{z} \to \mathbf{x}' \cup \mathbf{z})} \exp \frac{U(\mathbf{x} \cup \mathbf{z}) - U(\mathbf{x}' \cup \mathbf{z})}{T_t} \tag{4.26}$$

     - Choose $X_{i+1} = \mathbf{x}'$ with probability $\min(1, R)$, else $X_{i+1} = \mathbf{x}$;
   - Update $T_{i+1} = \alpha T_i$ (in our tests $\alpha = 0.95$).

4.3.3 Water / Land discrimination

Boats are a particular type of object that only appear in the presence of water (with the exception of dry docs, which are not of interest here). Thus, when presented with a large satellite image, the first step in searching for boats is to restrict the search to water areas. The water area can be generally identified as a large area of low radiometric values. However, a simple threshold is not sufficient. Tall buildings usually cast shadows onto the ground, which appear as dark areas as shown in Figure 4.9 (top). Figure 4.9 (middle) shows the result when applying a threshold, based on the Otsu method ([Otsu, 1979]). The false alarm rate caused by shadows is very high. Choosing size as a critical feature to distinguish between water and shadows is not sufficient, since rivers and lakes can be easily missed. Indeed, the interest lies in extracting the water areas under the condition that the number of missed detections should be zero and the number of false alarms should be minimal. The difficulty of this problem lies in finding appropriate features to do the separation.

Algorithm 8 Algorithm for creating a water mask

Input: Input image;
Output: Binary mask displaying the water area in the input image.

1. Threshold the input image using the threshold obtained by bimodal histogram splitting;
2. Identify the connected components in the threshold image;
3. For each connected component, compute its size and the intra-component variance of the radiometric values;
4. Classify the connected components into water and shadows based on the size and variance features.

4.3.3.1 Water mask computation

Dare [2005] puts forward an analysis of the shadows that appear in high resolution optical images of urban areas. He identifies one main feature that can be used to distinguish between shadows and water: intra-area variance. As Dare [2005] points out, the variance of the radiometric values within the shadow areas tends to be larger when compared with the variance within water areas, due to the underlying structures on which the shadow is cast. This feature is used in conjunction with the size feature to construct an algorithm (see Algorithm 8) to identify the water area. The algorithm is straight forward: after applying a threshold to the initial image, the

Component type	Size	Mean	Variance
Shadow 1	8641	12.8962	1.03389
Shadow 2	8211	13.0898	1.22846
Shadow 3	9986	13.0068	1.00696
Water	969675	13.2784	0.377285

Table 4.1: Comparison of characteristics of water and shadow components

connected components are identified. For each connected component, its size and the variance of the radiometric values inside the component are computed. Finally, only those components that exhibit a large size and small variance are selected as water areas. Table 4.1 shows the statistics for three shadow areas compared to water area. Although the mean radiometric value is similar for both class types, a clear difference can be observed in terms of size and radiometric variance. Finally, morphological dilation is used to extend the water area to include boats that are anchored near the land. The resulting water mask Ξ is depicted in Figure 4.9 (bottom).

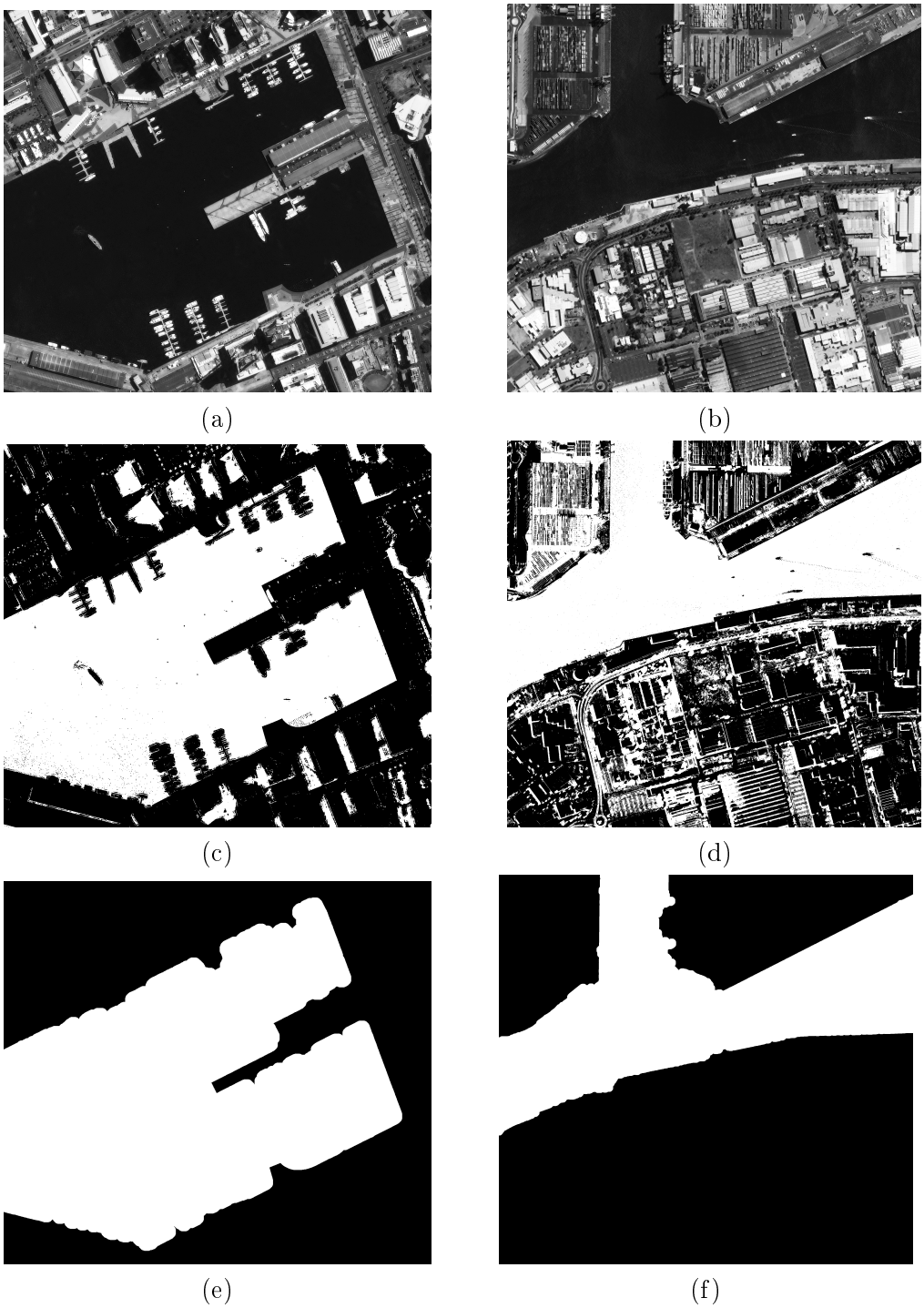


Figure 4.9: (a), (b) Image of boats inside and outside a harbor ©Airbus D&S; (c),(d) Water/Land discrimination results after applying a threshold to the input image. High false alarm rate due to shadows; (e),(f) Water/Land discrimination results using the algorithm described in Algorithm 8.

4.3.3.2 Water mask integration into the RJMCMC sampler

Once the water mask has been created, we need to integrate it into the RJMCMC sampler. The water mask Ξ represents the area where objects are expected to appear. We assume that no objects exist outside this area. We integrate the water mask by modifying the underlying Poisson intensity $\nu(\cdot)$ such that for an object u:

$$\nu(u) = \begin{cases} \varepsilon & \text{if } u \in \Xi, \\ 0 & \text{otherwise} \end{cases} \tag{4.27}$$

Hence, objects will be created only in the pre-determined water area.

4.4 Results

In the following, a detailed analysis is presented of the results obtained on two different types of satellite images:

- Spot 5 images with 2.5m ground sampling resolution provided by CNES;
- Pleiades images with a ground sampling resolution of 0.5m provided by Airbus D&S.

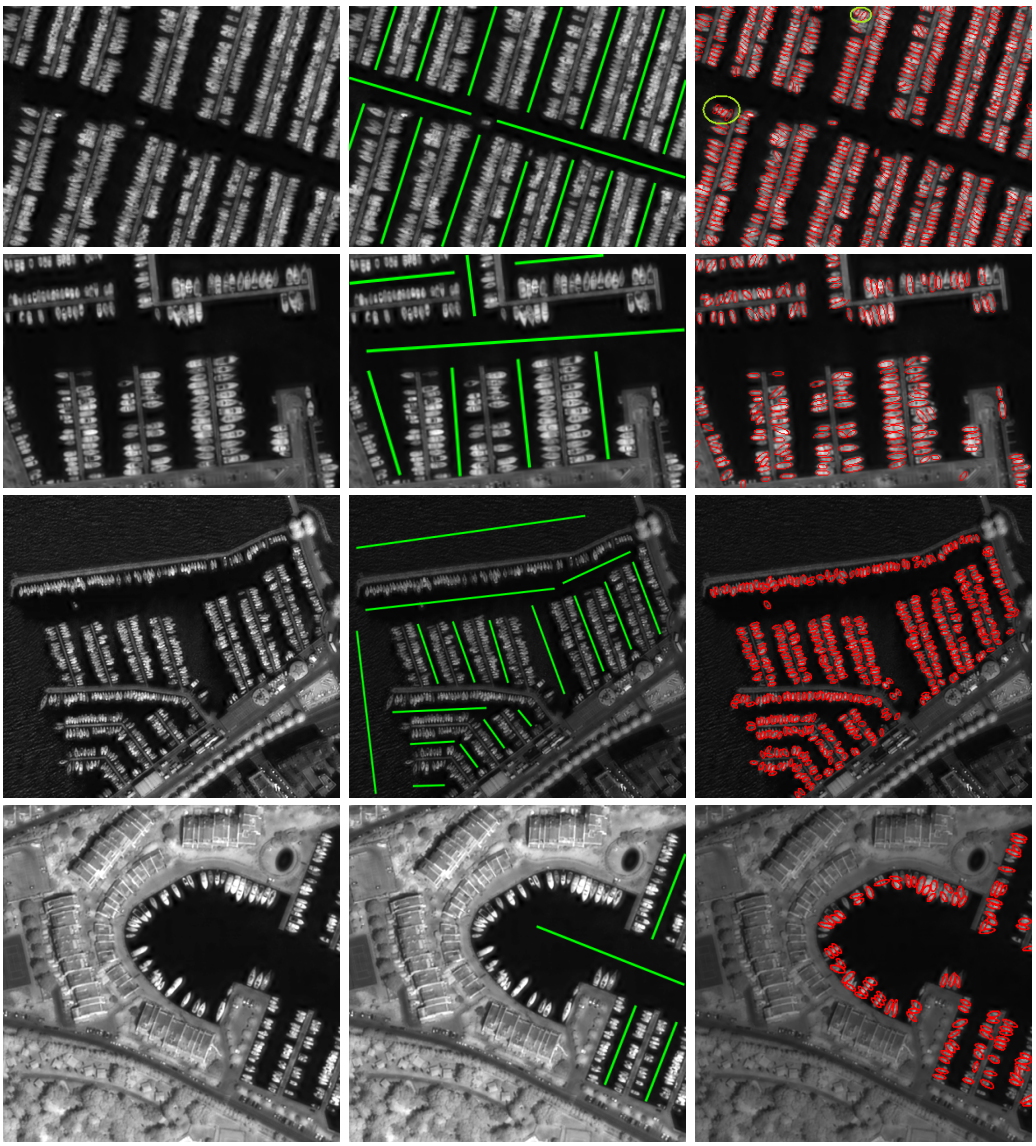
The computational efficiency of the proposed method is also discussed below.

4.4.1 Detection accuracy of the proposed model

In the Figure 4.10 the results obtained on Spot 5 images over the French coast are depicted. The first image is 385×275 pixels in size and all the boats in the scene have the same orientation.

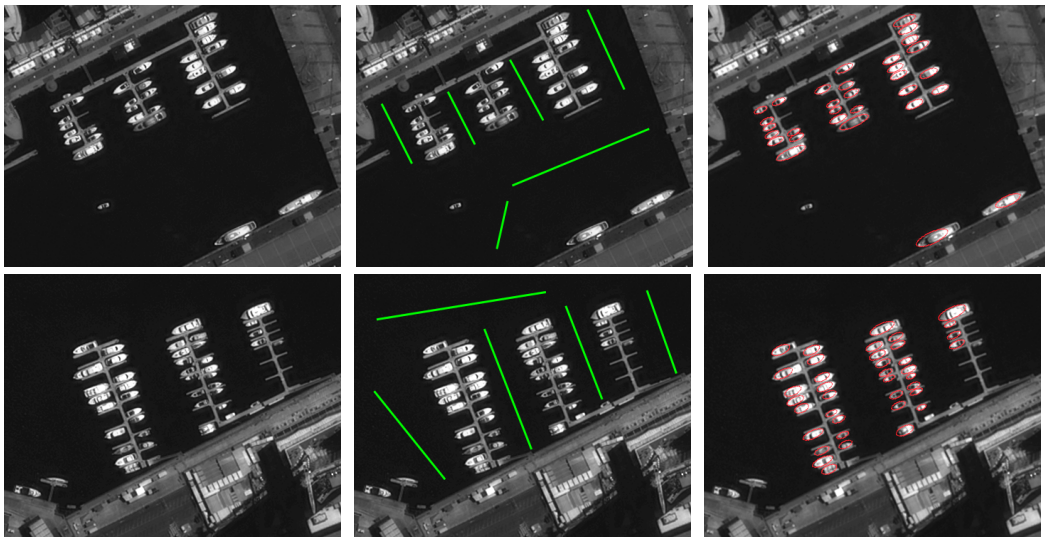
	Fig.4.10 (top)	Fig.4.10 (second row)	Fig.4.10 (third row)	Fig.4.10 (bottom)
Number of boats (GT)	518	178	572	131
Number of detected boats	501	169	427	96
Precision	97.6%	94.6%	97.6%	87.5%
Recall	88.9%	87.9%	84.2%	80.0%
Accuracy	85.5%	81.7%	79.4%	65.4%

Table 4.2: Quantitative analysis of the proposed boat extraction algorithm for Spot 5 images. The detection accuracy decreases as the information on the orientation of the boats becomes less reliable. This is particularly true in Figure 4.10 (bottom) where the orientation of the boats in the curved area is practically non-existent.



(a) (b) (c)

Figure 4.10: (a) Spot 5 images of boats in harbor on the French coast ©CNES; (b) Hough lines obtained for the extraction of local orientation information for the boats; (c) Boat extraction results. A total of 501 (top), 169 (second row), 427 (third row) and 96 (bottom) boats are detected respectively.



(a) (b) (c)

Figure 4.11: (a) Pleiades image of boats in harbor in Melbourne, Australia ©Airbus D&S; (b) Hough lines obtained for the extraction of local orientation information for the boats; (c) Boat extraction results. A total of 31 (top) and 39 (bottom) boats are detected respectively.

	Fig.4.11 (top)	Fig.4.11 (bottom)
Number of boats (GT)	32	42
Number of detected boats	31	39
Precision	100%	100%
Recall	96.8%	92.8%
Accuracy	99.7%	99.2%

Table 4.4: Quantitative analysis of the proposed boat extraction algorithm for Pleiades images. Due to the very high resolution of these images, the objects no longer appear to be tangent. Thus, the overall accuracy is very good.

The second image is 326×226 pixels in size and contains boats that have two different orientations. The third and forth images are 501×451, respectively 410×410 pixels in size and the orientation of the boats has a large variance within the scenes. The quantitative results are summarized in Table 4.2. The ground truth information, the number of boats detected and the detection errors for each image are shown. The detection results are very good and the algorithm correctly detects boats of different sizes. This is mainly due to the orientation constraint within the internal term. This constraint favors boats that are oriented perpendicular to the Hough lines.
The detection results on two close-ups of a Pleiades image of a harbor area in Melbourne, Australia are illustrated in Figure 4.11. Since the boats are well separated from each other, the detection accuracy is very good. Table 4.4 shows the quantitative results for these two close-ups. The image on which these close-ups were taken is 1620×1450 pixels in size. The close-ups are used for a better visual assessment.

# CPU's	Computation times for Figure 4.10 (top)		Computation times for Figure 4.10 (second row)	
	No water mask	Water mask	No water mask	Water mask
1	2h 05min	2h 05min	1h 40min	1h 40min
2	1h 17min	1h 17min	58h 12sec	58h 12sec
8	25min 25sec	25min 25sec	14min 46sec	14min 46sec
24	18min 58sec	18min 18sec	10min 04sec	10min 04sec
# CPU's	Computation times for Figure 4.10 (third row)		Computation times for Figure 4.10 (bottom)	
	No water mask	Water mask	No water mask	Water mask
1	2h 51min	2h 12min	1h 51min	1h 27min
2	1h 31min	1h 17min	1h 02min	49min 39sec
8	29min 51sec	27min 09sec	10min 56sec	6min 34sec
24	16min 11sec	17min 34sec	7min 47sec	5min 13sec

Table 4.3: Quantitative analysis of the computational efficiency for boat extraction on Spot 5 images.

4.4.2 Computational efficiency

In section 4.3, two ways have been discussed which increase the computational efficiency of the proposed algorithm: computing a water mask to reduce the search space and using a parallel implementation of the RJMCMC sampler to perform multiple independent perturbations at the same time. Here, the computational

efficiency of the model presented in this chapter is analyzed. Tables 4.3 and 4.5 summarize the results w.r.t. the computational efficiency.
The results show a significant decrease in computation times for images containing vast land areas when the water mask is used. The reasons behind rely on the fact that the usage of a water mask reduces the search space, in some images up to more than 50%. This effect is not visible in Figure 4.10 (top) and (second row) because the land area is practically nonexistent. The sustained decrease in computational times with the number of processors shows that the proposed optimization scheme is adapted to parallel implementation. Note however, that the speed-up is less significant in images with reduced size (i.e. the images in Figure 4.10). The underlying idea behind the parallel implementation is space-partitioning. The smaller the images, the less cells are in the partitioning tree and hence, a smaller number of cores can actually perform parallel computations while maintaining the cell-independence conditions outlined in section 4.3.2.

# CPU's	Computation times for Figure 4.9 (a)	
	No water mask	Water mask
1	2h 4min 5sec	1h 38min 57sec
2	1h 45min 41sec	1h 20min 23sec
8	49min 23sec	44min 7sec
24	12min 54sec	9min 41sec

Table 4.5: Quantitative analysis of the computational efficiency for boat extraction on Pleiades images.

4.5 Conclusions

In this chapter, a new marked point process model has been presented that draws on stochastic geometry to extract boats in harbors. This problem is hard due to the particular distribution of the boats. Most of the boats are very close to each other or even tangent, which makes individual detection difficult. Furthermore, boats can have different sizes or orientations, as well as dark spots which makes template matching techniques inefficient.
The model proposed in this chapter has significant advantages:

- It can handle dark spots inside the objects through the use of an internal border to compute the contrast distance w.r.t. the external border. Thus, the dark spots within the boats have less influence on the overall contrast distance used in the data likelihood;
- It can handle objects of varying sizes through the use of a large interval for

the semi-major and semi-minor axis of the ellipses used to extract the boats;

- It can handle objects with different orientations through the orientation constraint used in the internal energy term. The general heuristic w.r.t. the orientation of the boats in a harbor is that they tend to be anchored perpendicular to the keys. The orientation constraint present in the model favors ellipses that are oriented perpendicular to the water areas that are formed between the keys of the harbor.

The model has been used to detect boats in two types of images: Spot 5 images, taken over the French coast and provided by CNES; and Pleiades images, taken over Melbourne, Australia and provided by Airbus D&S. The detection accuracy for the Spot 5 images is good. Missed detections appear mainly due to the proximity of the boats, which decreases the contrast distance in the external energy term and thus, makes objects less likely to be detected, but also due to the low radiometric value of the boat itself. False alarms appear mainly on the keys that have a high radiometric value.
The computational efficiency of the proposed model increases as the number of processors used for the optimization increases. The complexity of the model leads to the necessity of higher computational efforts w.r.t. other object extraction methods, but the extraction accuracy compensates for the computational burden. This model might not be desirable in the case when only a rough estimate of the locations of the objects of interest is required. Nevertheless, the model proves to be superior when an exact assessment of the scene is necessary.

Chapter 5

From object detection to object tracking using a spatio-temporal marked point process model

Contents

In the previous chapter the use of marked point processes for object detection has been investigated. A point process of ellipses was used to detect boats in harbors which is a challenging case due to the particular distribution of the objects.

Nevertheless, the detection results were very promising with high accuracy levels and both good precision and recall rates.
This chapter revolves around the following question: How can marked point process models be successfully applied to the joint detection and tracking of multiple objects? The word 'successfully' should be interpreted not only as merely solving the tracking problem using such processes, but doing so with high accuracy and efficiency rates which are competitive with state of the art methods in the computer vision field.
Supplementary to the object detection problem, object tracking involves a new challenge in the form of associating individual detections with one object. The independent detections of a single object throughout the image sequence have to be grouped together into meaningful tracks in order to infer the trajectories and motion characteristics of that object. This additional task can become particularly difficult in dense areas with multiple objects crossing their paths or being very close to each other.
This chapter gives the answer to the question raised above and sustains it with detailed analysis and convincing examples. The chapter is organized as follows: first, a marked point process model proposed for the joint detection and tracking is presented. The application-dependent parameters of the model are identified and a procedure for learning these parameters is discussed. Next, optimization methods are investigated and a new optimization procedure specifically adapted for tracking is proposed. Finally, results are shown on a large variety of image sequences and the performance of the novel method is discussed in detail. Conclusions are drawn at the end of this chapter w.r.t. the suitability of marked point process models to the joint detection and tracking of multiple objects, their advantages and their drawbacks.

5.1 Model

Compared to a static image, an image sequence has an additional dimension: the dimension of time. In the context of marked point processes, time can be modeled in at least two conceptually different ways:

- **Time as a mark**. The point process can be viewed as a counting process defined on the 2D space. The time mark could represent the number of appearances of an event in a given location. In the case of satellite images, this approach can be efficiently used for applications such as traffic density estimation.

- **Time as a dimension**. In this case, the point process exists in the 3D space (two spatial dimensions and a temporal one). Spatial as well as temporal correlations can be explicitly described in this approach, which makes it suitable for tracking purposes. Hence, time is modeled as an additional dimension in this work.

The 3D image cube is modeled as a bounded set $K = [0, I_{h_{max}}] \times [0, I_{w_{max}}] \times \{1, \cdots, T\}$. A marked point process of ellipses is considered on K, with the mark space M defined as $M = [a_m, a_M] \times [b_m, b_M] \times (-\frac{\pi}{2}, \frac{\pi}{2}] \times [0, L]$, where a_m, a_M and b_m, b_M are the minimum and maximum length of the semi-major and semi-minor axis respectively, $\omega \in]-\frac{\pi}{2}, \frac{\pi}{2}]$ is the orientation of the ellipse and $l \in [0, L]$ is its label. Thus, an ellipse u can be defined as $u = (c_h, c_w, t, a, b, \omega, l)$ and a marked point process of ellipses X is a point process on $W = K \times M$. While the semi-axes a and b and the orientation ω describe the physical properties of an ellipse, the label l is used as an identifier. Objects with the same label across the image sequence form a track. Individual trajectories are extracted by grouping objects according to their label. In the framework of stochastic geometry, the idea of using labels to distinguish between different trajectories has already been proposed and implemented by Vo and Vo [2013], Vu et al. [2014] and Vo et al. [2014].
A point process distribution can be defined by its probability density function where the Poisson point process plays the analogue role of the Lebesgue measure on $\mathbb{R}^d$, with d being the dimension of the object space. The Gibbs family of processes is used to define the probability density as follows:

$$f_\theta(X = \mathbf{X}|\mathbf{Y}) = \frac{1}{c(\theta|\mathbf{Y})} \exp^{-U_\theta(\mathbf{X},\mathbf{Y})} \tag{5.1}$$

where:

- $\mathbf{X} = \{\mathbf{x_1} \cup \mathbf{x_2} \cup \cdots \cup \mathbf{x_t} \cup \cdots \cup \mathbf{x_T}\}$ is the configuration of ellipses, with $\mathbf{x_t}$ being the configuration of ellipses at time t;
- $\mathbf{Y}$ represents the 3D image cube;
- θ is the parameter vector;
- $c(\theta|\mathbf{Y}) = \int_\Omega \exp^{-U_\theta(\mathbf{X},\mathbf{Y})} \mu(d\mathbf{X})$ is the normalizing constant, with Ω being the configuration space and $\mu(\cdot)$ being the intensity measure of the reference Poisson process;
- $U_\theta(\mathbf{X}, \mathbf{Y})$ is the energy term.

The solution being sought for is the configuration which maximizes the density $f_\theta(\mathbf{X})$, e.g. the most likely configuration of objects corresponds to the global minimum of the energy:

$$X \in arg \max_{\mathbf{X}\in\Omega} f_\theta(X = \mathbf{X}|\mathbf{Y}) = arg \min_{\mathbf{X}\in\Omega} [U_\theta(\mathbf{X}, \mathbf{Y})]. \tag{5.2}$$

This configuration is obtained using the framework of simulated annealing which results in iteratively simulating from the distribution $f(\mathbf{X})^{\frac{1}{T}} d\pi(\mathbf{X})$, where the temperature parameter T is slowly decreasing to zero. Hence, the reference measure $\pi(\mathbf{X})$ will only influence the dynamics of the process, but not the solution.
The energy function is divided in two parts: an external energy term, $U^{ext}_{\theta_{ext}}(\mathbf{X}, \mathbf{Y})$

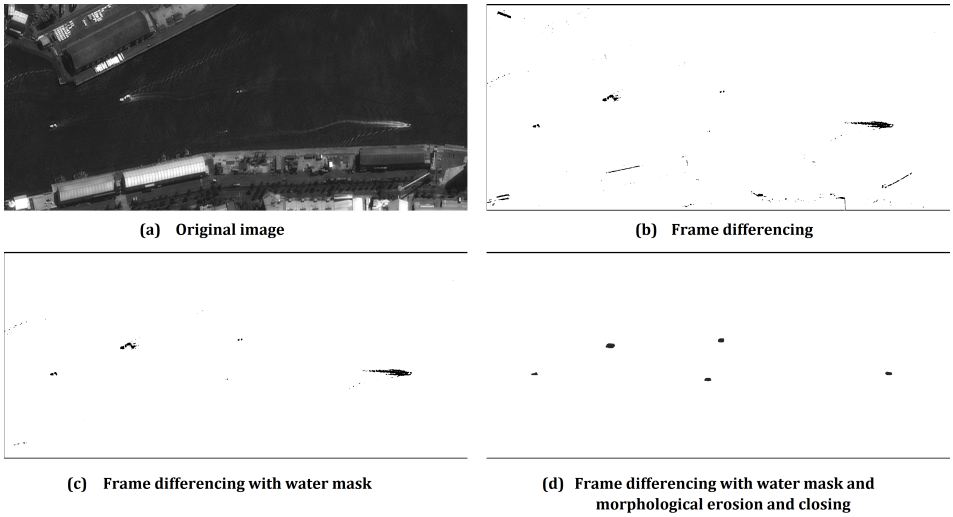


Figure 5.1: Results of frame differencing for computing the object evidence term. (a) Image of boats in Melbourne, Australia ©Airbus D&S; (b) Frame differencing of two consecutive frames; (c) Frame differencing results after applying a water mask; (d) Frame differencing results after applying a water mask and performing morphological erosion and closing operations to smooth the boundaries of the foreground regions.

which determines how good the configuration fits the input sequence, and an internal energy term, $U^{int}_{\theta_{int}}(\mathbf{X})$, which incorporates knowledge about the interaction between objects in a single frame and across the entire batch considered. The total energy can be written as the sum of these two terms:

$$U_\theta(\mathbf{X}, \mathbf{Y}) = U^{ext}_{\theta_{ext}}(\mathbf{X}, \mathbf{Y}) + U^{int}_{\theta_{int}}(\mathbf{X}). \tag{5.3}$$

The parameter vectors of the external and internal energy terms are θ_{ext} and θ_{int} respectively and $\theta = \{\theta_{ext}, \theta_{int}\}$.
The similarity to the object detection approach described in the previous chapter is evident. Note however that in this case, $\mathbf{Y}$ represents a 3D image cube containing both spatial and temporal information, as opposed to a 2D image in Chapter 4. Furthermore, $\mathbf{X}$ is composed of configurations of objects which exist in each time frame t. Finally, the internal energy $U^{int}_{\theta_{int}}(\mathbf{X})$ must model not only the interactions of objects within a single time frame, but also the interactions of objects across time.
In the following sections, the two parts of the energy are described in detail.

5.1.1 Internal energy term

The internal energy term consists of a set of constraints meant for a correct detection of objects and to facilitate tracking. These constraints target the layout of the

objects. For instance geometric and physical consistency should be maintained (e.g. interpenetration among objects should be avoided).

5.1.1.1 The dynamic model

A defining property of tracking (as opposed to individual detections per frame) is that in most cases object trajectories are smooth. This allows to favor configurations where objects exhibit a motion described by a dynamic model. This motion model, denoted by $d(\cdot,\cdot,\cdot)$, depends on the application. The dynamic model used throughout this work is a constant velocity model. Let $v \in \mathbf{x}_{t-1}$, $u \in \mathbf{x}_t$, $w \in \mathbf{x}_{t+1}$ be three objects, then the dynamic model can be written in terms of their positions $pos(v)$, $pos(u)$ and $pos(w)$ as follows:

$$d(v,u,w) := ||pos(v) - 2pos(u) + pos(w)||^2 \tag{5.4}$$

Note however that any motion model can be inserted into the model. Nevertheless, the constant velocity model is general enough to cover a large variety of tracking applications.
An energy term is designed to favor objects which follow a given motion model s.t. for an object u that exists at time t it is written as:

$$U^{int}_{dyn}(u) = \begin{cases} d(\cdot,u,\cdot) - dyn_0 & \text{if } \exists\ (v \in \mathbf{x}_{t-1} \text{ and } w \in \mathbf{x}_{t+1}) \text{ s.t. } d(v,u,w) \leq dyn_0 \\ 0 & \text{otherwise} \end{cases} \tag{5.5}$$

where dyn_0 is a threshold that describes how much objects can deviate from the motion model but still be awarded. Note however that this motion model does not restrict the velocity of the objects in terms of magnitude. For instance, this energy term will reward equally both configurations depicted in Figure 5.2. If the maximum velocity of objects is known a priori, it can be incorporated into the model by imposing a maximum distance between the objects v and u, respectively u and w.
The energy term that awards configurations which follow the dynamic model is the sum over all objects in the configuration:

$$U^{int}_{dyn}(\mathbf{X}) = \gamma_{dyn} \sum_{u \in \mathbf{X}} U^{int}_{dyn}(u). \tag{5.6}$$

This term favors the creation of objects where the data evidence is reduced but the dynamic model motivates the existence of an object.

5.1.1.2 Label persistence

In order to distinguish between distinct trajectories a label is added to the mark of each object. This label can be viewed as a trajectory identifier. Different labels mean different trajectories. Thus, the number of labels has to be kept closely related to the number of trajectories in the data set. Ideally, the large number of objects u scattered across the image sequence should be assigned to a rather small number of

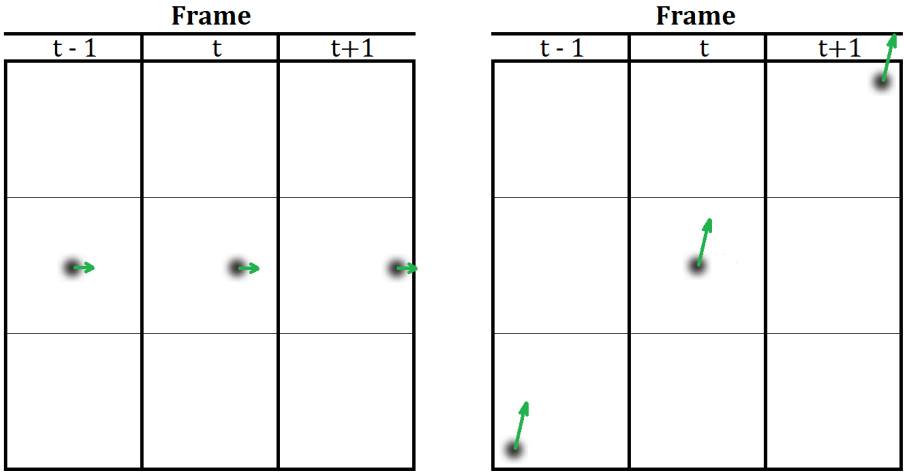


Figure 5.2: Two examples of tracks over three frames that will be identically rewarded by the dynamic energy term, as this term favors objects that follow a constant velocity model, regardless of the actual velocity of the object. The green arrow shows the direction and the magnitude of the velocity. Left: The object depicted has a slow velocity over the three frames; Right: The object has a very high velocity over the three frames.

labels. In this regard, the set of labels present in a configuration $\mathbf{X}$ is constructed by $labels(\mathbf{X}) = \bigcup_{u \in \mathbf{X}} l(u)$, where $l(u)$ is the label of object u. Configurations where the number of distinct labels is small are favored in the following way:

$$U_{label}^{int}(\mathbf{X}) = -\gamma_{label}\Big(\frac{1}{|labels(\mathbf{X})|}\Big) \tag{5.7}$$

where $|labels(\mathbf{X})|$ represents the cardinality of the set.

5.1.1.3 Mutual exclusion

Handling object collision or overlapping in a given frame is a crucial aspect when detecting and tracking objects. In this model, an infinite penalty is attributed to any configuration that contains objects that overlap more than a given extent s. Thus, the probability of selecting such a configuration is zero. The same non-overlapping constraint as in Section 4.1.2.1 is used.
As a reminder, denoting by $A(u_i, u_j) = \frac{Area(u_i \cap u_j)}{\min(Area(u_i), Area(u_j))}$ the area of intersection between the objects u_i and u_j, with $u_i, u_j \in \mathbf{x}_t$, the non-overlapping energy term can be defined as:

$$U_{overlap}^{int}(\mathbf{X}) = \sum_{t \in \{0,\cdots,T\}} \sum_{1 \leq i \neq j \leq n(\mathbf{x}_t)} t_s(u_i, u_j) \tag{5.8}$$

with:

$$t_s(u_i, u_j) = \begin{cases} 0 & \text{if } A(u_i, u_j) < s \\ +\infty & \text{otherwise} \end{cases} \tag{5.9}$$

where $s \in [0, 1]$ corresponds to the amount of overlapping allowed by the model and $n(\mathbf{x}_t)$ is the number of objects in the configuration $\mathbf{x}_t$ at time t. This energy term is used to restrict the number of objects that cover the same location in a given frame.

5.1.2 External energy term

We have developed two alternative approaches for linking the data likelihood of a configuration. The first is an extension of the external energy presented in Chapter 4, which combines the contrast distance measure presented in that chapter, with a new term, called object evidence which is computed based on the difference of consecutive frames to which a threshold is applied. We call this model **Quality model**. The second approach is based on applying statistical tests to determine the likelihood of an object being present. We call this model **Statistical model**.

5.1.2.1 Quality model

The external energy term is composed of the local external energies of each object u in the configuration. In order to enhance the image evidence, two local terms are computed for each object: an object evidence and a contrast distance measure.

5.1.2.1.1 Object evidence

Likely locations of moving objects are identified by frame differencing. At each pixel location, the mean over time of the radiometric values, denoted p_m, is computed. Next, for each frame f, for all pixels belonging to frame f, the difference between their radiometric value p_f and the mean value p_m at that location is calculated. Finally, only those pixels for which this difference is higher than a predefined threshold are retained as foreground: $\forall f \in [0, T], \forall p_f \in f : p_f \in foreground \iff |p_f - p_m| \geq threshold$. Morphological erosion and closing operations are used to enhance the filter response and smooth the boundaries of the foreground regions (Schmitt and Mattioli [2013]). A visual representation of the frame differencing results after applying morphological operations can be seen in Figure 5.1. The class of a pixel p can be defined as $\nu(p) = \{foreground, background\}$. Finally, the evidence of object u is computed in the following way:

$$\mathcal{E}(u|\mathbf{Y}) = -\frac{1}{|u|} \sum_{p \in u} \mathbb{1}\{\nu(p) = foreground|\mathbf{Y}\} \tag{5.10}$$

where $|u|$ marks the cardinality of object u (e.g. the number of pixels that belong to u) and $\mathbb{1}\{\cdot\}$ marks the indicator function ($\mathbb{1}\{true\} = 1$, $\mathbb{1}\{false\} = 0$). The object evidence $\mathcal{E}(u|\mathbf{Y})$ is used to favor the detection of smaller objects.

Figure 5.3: (a) A typical example of a boat; (b) In red: the exterior border $\mathcal{F}^{\rho}(u)$ and in green: the interior border $\mathcal{I}^{1}(u)$ used to compute the contrast distance measure for object detection in Chapter 4; (c) In red: the exterior border $\mathcal{F}^{\rho}(u)$ used to compute the contrast distance to the interior of the ellipse for tracking purposes in this chapter.

5.1.2.1.2 Contrast distance measure

The aim of this term is to further refine the detection and extract information such as the orientation and the size of the objects. The objects of interest appear as bright structures on a dark background. Hence, a contrast distance measure is computed between the interior of the ellipse and its border. The contrast distance measure has been previously defined in chapter 4. However, due to the existence of the evidence term, the impact of the contrast distance is greatly reduced, as compared to the previously described model. Thus, only the contrast distance between the interior of an object u, and its ρ-border $\mathcal{F}^{\rho}(u)$ is computed as follows:

$$U^{ext}(u, \mathbf{Y}) = \mathcal{Q}\left(\frac{d_B(u, \mathcal{F}^{\rho}(u))}{d_0(\mathbf{Y})}\right), \tag{5.11}$$

where again $d_B(\cdot,\cdot)$ is the contrast distance measure defined in eq. 4.6 and $\mathcal{Q}(\cdot)$ is the quality function defined in eq. 4.7. A reminder of the graphical representation of the border is shown in Figure 5.3.
The two terms computed in eq. 5.10 and eq. 5.11 are further combined into a local external energy for an object u:

$$U^{ext}_{local}(u|\mathbf{Y}) = \gamma_{ev}\mathcal{E}(u|\mathbf{Y}) + \gamma_{cnt}\mathcal{Q}\Big(\frac{d_B(u, \mathcal{F}^{\rho}(u))}{d_0(\mathbf{Y})}\Big). \tag{5.12}$$

Finally, the external energy term for the configuration $\mathbf{X}$ is:

$$U^{ext}_{\theta_{ext}}(\mathbf{X}, \mathbf{Y}) = \sum_{u \in \mathbf{X}} U^{ext}_{local}(u|\mathbf{Y}). \tag{5.13}$$

The parameter vector of the external term, $\theta_{ext} = \{\gamma_{ev}, \gamma_{cnt}\}$, consists of the weight γ_{ev} of the evidence term and the weight γ_{cnt} of the quality of the contrast distance.

5.1.2.2 Statistical model

This method relies on Bayesian decision theory and consists in statistically testing two hypotheses and choose the most likely one. In our setting, the aim is to determine whether an object is present or absent.
First, we compute the difference between two consecutive frames, FD_t. This will result is a gray level image in which the intensity of a pixel represents the amount of change present at that location between the frames. The larger the amount of change (resp. the higher the intensity of a pixel), the more likely it is for a moving target to be present in that location. To determine this, we use a sliding window u, which is translated over every pixel p. We have the following two hypotheses:

- $\mathbf{H_0}$: the ellipse covers only the background without any target being present and denote the area with $\mathcal{A}$;
- $\mathbf{H_1}$: the ellipse is placed in the center of a target. The area corresponding to the interior of a target is denoted $\mathcal{I}_u$, while the area corresponding to the exterior of a target is denoted $\mathcal{F}_u$.

We are interested in the following probabilities:

$$\begin{cases} P(H_0|FD_t)\text{: The probability of } H_0\text{, knowing the scene } FD_t; \\ P(H_1|FD_t)\text{: The probability of } H_1\text{, knowing the scene } FD_t. \end{cases} \tag{5.14}$$

We define the probabilities ratio in the following way:

$$R = \frac{P(H_1|FD_t)}{P(H_0|FD_t)} = \frac{P(FD_t|H_1)}{P(FD_t|H_0)} \frac{P(H_1)}{P(H_0)}, \tag{5.15}$$

where $P(H_0)$ is the probability that the window u covers only the background, independent of the observation and $P(H_1)$ is the probability of a target being located at the center of the window, independent of the observation.
We define the likelihood ratio

$$\Lambda = \frac{P(FD_t|H_1)}{P(FD_t|H_0)}. \tag{5.16}$$

The decision of whether a target is or not present at a given location is taken as follows:

$$\begin{cases} \Lambda > \frac{P(H_0)}{P(H_1)} & (\Leftrightarrow R > 1) \text{ A target is present;} \\ \Lambda < \frac{P(H_0)}{P(H_1)} & (\Leftrightarrow R < 1) \text{ A target is not present.} \end{cases} \tag{5.17}$$

The quantities $P(H_0)$ and $P(H_1)$ are hard to evaluate. Thus, we use the Neyman-Pearson decision rule ([Lehmann and Romano, 2008, Proia, 2010]) to decide upon Λ.

5.1.2.2.1 Neyman-Pearson decision rule

In order to define the Neyman-Pearson decision rule, we need to introduce two performance measures:

- **Probability of detection (PD)**: is the probability of deciding that a target is present within the window, if it is actually present;
- **Probability of false alarms (PFA)**: is the probability of deciding that a target is present within the window, when actually it is not.

The Neyman-Pearson decision rule ([Lehmann and Romano, 2008]) consists in fixing a level of acceptable risk α for the probability of false alarms (PFA) and maximizing the detection probability (PD). According to the Neyman-Pearson lemma, eq. 5.16 maximizes the PD for all chosen PFAs. If we take the logarithm of the likelihood ratio

$$\log \Lambda = \log(P(FD_t|H_1)) - \log(P(FD_t|H_0)) \tag{5.18}$$

then, according to the Neyman-Pearson decision rule, we can rewrite eq. 5.17 as follows:

$$\begin{cases} \log \Lambda > \tau \text{ A target is present;} \\ \log \Lambda < \tau \text{ A target is not present.} \end{cases} \tag{5.19}$$

5.1.2.2.2 Detection model

To compute $P(FD_t|H_0)$ and $P(FD_t|H_1)$ we will make the following simplifying assumptions:

- The intensity of each pixel in FD_t is independent and uniformly distributed;
- Each pixel within a target follows a Gaussian distribution of unknown parameters (mean and variance);
- Each pixel within the background follows a Gaussian distribution of unknown parameters (mean and variance);
- The variance within the target and the background is equal.

On the one hand, it is essential to realize that these constraints are not very restrictive when a small sliding window is used, even in the presence of inhomogeneous background. On the other hand, it is clear that this model is not well adapted when the background presents a visible texture. We will see later why the last constraint is very useful.
By using this model, we can write the probability of a pixel p from an area S as:

$$P(p) = \frac{1}{\sqrt{2\pi\sigma_S^2}} \exp - \frac{(p - m_S)^2}{2\sigma_S^2}, \tag{5.20}$$

where σ_S^2 is the variance of an area S and m_S is the expectation of the area S. As the pixels are independent, we can write the probability of the area S as:

$$P(S) = \prod_{i \in S} \frac{1}{\sqrt{2\pi\sigma_S^2}} \exp - \frac{(p_i - m_S)^2}{2\sigma_S^2} \tag{5.21}$$

and the logarithm as:

$$L(S) = -\frac{1}{2} \sum_{p_i \in S} \left[\log(2\pi\sigma_S^2) + \frac{(p_i - m_S)^2}{\sigma_S^2} \right]. \tag{5.22}$$

We can now compute $P(FD_t|H_0)$ and $P(FD_t|H_1)$ using eq. 5.21 and eq. 5.22:

$$P(FD_t|H_0) = P(\mathcal{A}) \Leftrightarrow \log(P(FD_t|H_0)) = L(\mathcal{A}) \tag{5.23}$$

$$P(FD_t|H_1) = P(\mathcal{I}_u) \times P(\mathcal{F}_u) \Leftrightarrow \log(P(FD_t|H_1)) = L(\mathcal{I}_u) + L(\mathcal{F}_u). \tag{5.24}$$

Hence, the logarithm of the likelihood ratio can be written as:

$$\log \Lambda = L(\mathcal{I}_u) + L(\mathcal{F}_u) - L(\mathcal{A}). \tag{5.25}$$

As the quantities m_S and σ_S are unknown (here S stands for the areas $\mathcal{A}$, $\mathcal{I}_u$ and $\mathcal{F}_u$), a classical solution to solve this problem is to replace them by their maximum likelihood estimates $\hat{m_S}$ and $\hat{\sigma_S}$ respectively. After a few simplifications we obtain the following expression for eq. 5.25:

$$\log \Lambda = -\frac{1}{2} \left(N_{\mathcal{I}_u} \log(\hat{\sigma}^2_{\mathcal{I}_u}) + N_{\mathcal{F}_u} \log(\hat{\sigma}^2_{\mathcal{F}_u}) - N_{\mathcal{A}} \log(\hat{\sigma}^2_{\mathcal{A}}) \right) \tag{5.26}$$

where:
- $N_{\mathcal{I}_u}$, $N_{\mathcal{F}_u}$, $N_{\mathcal{A}}$ are the number of pixels in the areas $\mathcal{I}_u$, $\mathcal{F}_u$ and $\mathcal{A}$ respectively;
- $\hat{\sigma}^2_{\mathcal{I}_u}$, $\hat{\sigma}^2_{\mathcal{F}_u}$, $\hat{\sigma}^2_{\mathcal{A}}$ are the estimated variances of the areas $\mathcal{I}_u$, $\mathcal{F}_u$ and $\mathcal{A}$ respectively.

As we have made the assumption that $\sigma^2_{\mathcal{I}_u} = \sigma^2_{\mathcal{F}_u} = \sigma^2_{\mathcal{A}} = \sigma$, we can further simplify eq. 5.26 to:

$$\log \Lambda = N_{\mathcal{I}_u} \hat{m}_{\mathcal{I}_u} + N_{\mathcal{F}_u} \hat{m}_{\mathcal{F}_u} - N_{\mathcal{A}} \hat{m}_{\mathcal{A}}, \tag{5.27}$$

where:
- $N_{\mathcal{I}_u}$, $N_{\mathcal{F}_u}$, $N_{\mathcal{A}}$ are the number of pixels in the areas $\mathcal{I}_u$, $\mathcal{F}_u$ and $\mathcal{A}$ respectively;
- $\hat{m}_{\mathcal{I}_u}$, $\hat{m}_{\mathcal{F}_u}$, $\hat{m}_{\mathcal{A}}$ are the estimated means of the areas $\mathcal{I}_u$, $\mathcal{F}_u$ and $\mathcal{A}$ respectively.

5.1.2.2.3 Choosing a threshold τ

We have obtained a simple and relatively fast pre-detection algorithm that can be described in the following way: For each time instance t, for each pixel of the difference image FD_t, compute the generalized likelihood ratio given by eq. 5.27 for each sliding ellipse u. For each time t, we obtain a 2D matrix which we denote with $\mathbf{M}^t_{GLRT}$. A threshold τ is applied to this matrix based on the desired PFA.

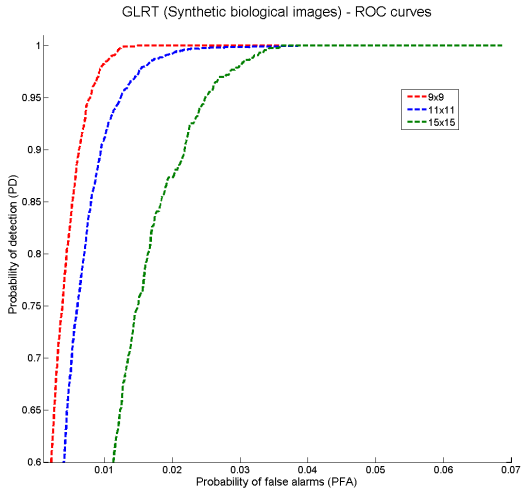


Figure 5.4: ROC curves for the synthetic biological data set with 3 different window sizes: 9×9 (red), 11×11 (blue) and 15×15 (green). The ROC curves show that a window size of 9×9 gives the best results.

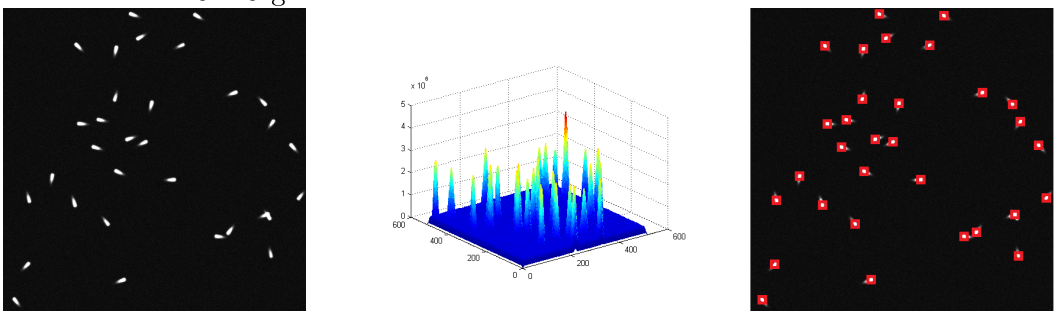


Figure 5.5: Left: Synthetic biological image containing 32 targets. Middle: The output of $\mathbf{M}^t_{GLRT}$. Right: Pre-detection results.

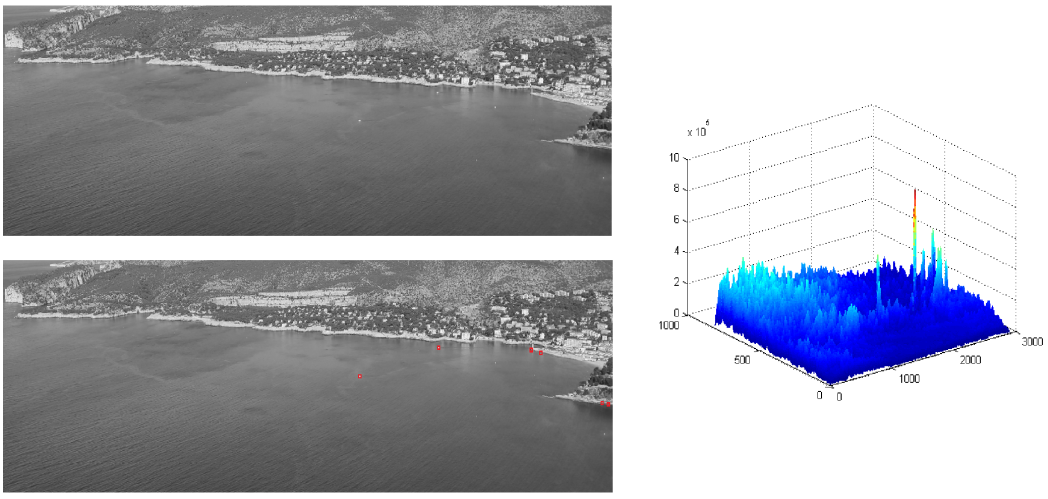


Figure 5.6: Left top: Satellite image of Toulon containing 2 targets (boats). Left bottom: Pre-detection results. The false alarms are caused by waves hitting the shore, resulting in high changes between consecutive frames. Right: The output of $\mathbf{M}^t_{GLRT}$.

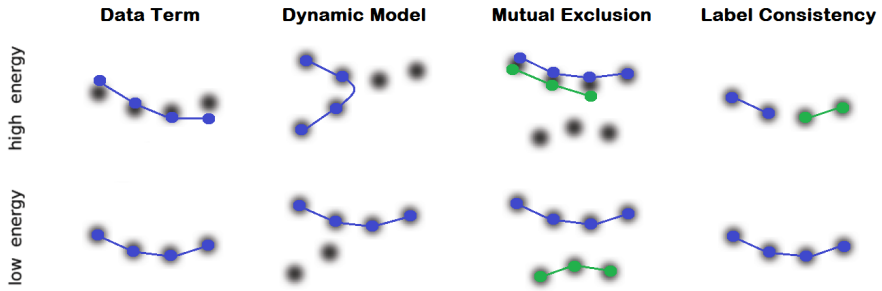


Figure 5.7: The effects of different components of the energy terms. The upper row shows a configuration with a higher energy value for each individual term. The bottom row shows a configuration with a lower energy value for each individual term. The dark spots denote target locations at different time frames. Different colors on the targets represent different labels assigned to each.

The choice of the threshold depends entirely on the application.
We have computed a threshold τ for each data set (the different data sets will be presented in detail in Section 5.4). We have computed the ROC (Receiver Operating Characteristic) curves for windows of size 9×9, 11×11 and 15×15 in order to choose the appropriate one. The ROC curves for the synthetic biological data set are displayed in Figure 5.4. We observe that the best results are obtained for a window size of 9×9 pixels (red curve). The same window size is used for satellite data. Thus, we can now set the threshold based on the desired level of false alarms. In our experiments, we have chosen τ as the threshold which maximizes the probability of detection (PD) with the smallest probability of false alarms (PFA).
In Figure 5.5 we show a detection example on a synthetic biological image. Similar results are obtained for satellite images. Here again a window size of 9×9 pixels gives the best results. We show a detection example on the Toulon data set in Figure 5.6.

5.1.2.2.4 External energy formulation

The external energy term for an object $u = (x_u, y_u, t_u, a_u, b_u, \omega_u, l_u)$ is written as follows:

$$U^{ext}_{stat}(u) = \begin{cases} -1 & \text{if } \mathbf{M}^{t_u}_{GLRT}(x_u, y_u) > \tau, \\ +1 & \text{if } \mathbf{M}^{t_u}_{GLRT}(x_u, y_u) < \tau. \end{cases} \tag{5.28}$$

The external energy term for a configuration $\mathbf{X}$ associated to the pre-detection result is:

$$U^{ext}_{stat}(\mathbf{X}) = \gamma_{stat} \sum_{u \in \mathbf{X}} U^{ext}_{stat}(u) \tag{5.29}$$

The parameter vector for the external energy for this model is $\theta_{ext} = \{\gamma_{stat}\}$.

5.1.3 Total energy term

The total energy term can be written as a sum of all the energy terms defined in section 5.1.2 and section 5.1.1:

- **Quality model**:

$$\begin{aligned} U_\theta(\mathbf{X},\mathbf{Y}) &= \gamma_{dyn}U^{int}_{dyn}(\mathbf{X}) + \gamma_{label}U^{int}_{label}(\mathbf{X}) + \gamma_o U^{int}_{overlap}(\mathbf{X}) + \\ &\quad \gamma_{ev}\mathcal{E}(u|\mathbf{Y}) + \gamma_{cnt}\textstyle\sum_{u\in\mathbf{X}}\left(\mathcal{Q}\left(\frac{d_B(u,\mathcal{F}^\rho(u))}{d_0(\mathbf{Y})}\right)\right). \end{aligned} \tag{5.30}$$

- **Statistical model**:

$$\begin{aligned} U_\theta(\mathbf{X},\mathbf{Y}) &= \gamma_{dyn}U^{int}_{dyn}(\mathbf{X}) + \gamma_{label}U^{int}_{label}(\mathbf{X}) + \gamma_o U^{int}_{overlap}(\mathbf{X}) + \\ &\quad \gamma_{stat}U^{ext}_{stat}(\mathbf{X}|\mathbf{Y}). \end{aligned} \tag{5.31}$$

An intuition of how each energy term influences the output result is presented in Figure 5.7.
A total of five (resp. four) weight parameters $\gamma_.$ are needed to balance the individual terms in the final energy of the Quality model (resp. Statistical model). In Chapter 4 it was argued that the stability of the model depends on the number of parameters that have to be estimated when EM-like algorithms are used for the estimation. The proposed energy contains a high number of parameters which cannot be set empirically. Hence, alternative parameter learning techniques are discussed in the following section in order to improve the stability of the model and reduce the computational burden.

5.2 Parameter estimation

Parameter learning using EM-like estimation algorithms tends to become unstable as the number of parameters increases. Moreover, the computational complexity of such methods increases exponentially with the number of parameters which results in very high computational loads. Thus, alternative learning techniques have to be considered.
A key property of the energy described in eq. C.4 and eq. C.5 is its linearity in the weight parameters. This property is crucial in determining a suitable parameter learning algorithm. The linearity is exploited in this section by proposing a parameter learning procedure based on linear programming.
After a brief introduction to linear programming, the estimation of the weight parameters is discussed.

5.2.1 Linear programming

Several optimization problems fall into the linear programming class, among which are minimum spanning trees or shortest paths. But probably the most widely known problem is the maximum flow / minimum cut problem in graph theory. Linear

programming was formalized and applied behind the Iron Curtain by Kantorovich to problems in economics, while in the western society, Koopmans [1951] applied it to shipping problems.
Linear programming problems consist of a linear objective function which has to be minimized (or maximized) subject to a number of constraints. The constraints are linear inequalities of the variables used in the objective function.
A linear programming problem with n variables and m constraints is in standard form when it is written as:

$$\text{Maximize: } \sum_{i=1}^{n} c_i x_i \tag{5.32}$$

$$\text{Subject to: } \sum_{i=1}^{n} a_{ij} x_i \leq b_i, \qquad j = 1, \cdots, m \tag{5.33}$$

$$x_i \geq 0, \qquad i = 1, \cdots, n \tag{5.34}$$

If the interest lies in minimizing an objective function, the problem can be rewritten in standard form by negating the coefficients c_i.
The linear program can also be written in vector form:

$$\text{Maximize: } \mathbf{a}^T \mathbf{x} \tag{5.35}$$

$$\text{Subject to: } \mathbf{A}^T \mathbf{x} \leq \mathbf{b}, \quad \mathbf{x} \geq 0. \tag{5.36}$$

Any vector $\mathbf{x}$ that satisfies the constraints of the linear programming problem is called a feasible solution. A linear programming problem can be either:

- **Infeasible**. If there is no vector $\mathbf{x}$ that satisfies the constraints of the linear programming problem, the problem is said to be infeasible;
- **Unbounded**. If the linear programming problem is not sufficiently constrained (for any given feasible solution, another feasible solution that further improves the objective function can be found), the problem is said to be unbounded;
- **Having an optimal solution**. The linear programming problem has an unique maximum (or minimum). This does not mean however, that the values of the variables that yield the optimal solution are unique.

The first algorithm to solve linear programming problems was developed by Dantzig in 1947 (Dantzig [1948, 1963]) and is known as the Simplex algorithm. The simplex method has two steps: first, a feasible solution to the linear programming problem is found. Often, a trivial solution such as $\mathbf{x} = 0$ is a feasible solution. Once a feasible solution is found, the method iteratively improves the value of the objective function. Graphically, this can be seen as moving along the edges of a feasible set from corner to corner.
Two issues have to be kept in mind when solving linear programming problems using the simplex method:

- **Initialization**. For many linear programming problems, the trivial solution $\mathbf{x} = 0$ is a feasible solution to the problem. Nevertheless, if this is not the case, a feasible solution has to be found to initialize the simplex method. An auxiliary linear programming problem can be formulated by subtracting additional variables from the constraints of the original problem and changing the objective function to incorporate these new variables. The solution obtained by solving the auxiliary problem (with all additional variables being zero) is a feasible solution for the original problem. If no solution to the auxiliary problem is found where all additional variables are zero, then the original problem is infeasible. This procedure is also known as the two-phase simplex algorithm;

- **Degeneracy**. If a corner on the edge of the feasible set is degenerate, the simplex method can get stuck is a local minima, simply because all the neighbors of the corner are identical and thus, the objective function can not be improved.

In terms of computation time, theoretical studies show that the simplex method is exponential in the number of variables. Nevertheless, such exponential examples rarely appear in practice. The general consensus is that the number of iterations of the simplex method increases linearly with the number of constraints and only logarithmically with the number of variables, which is why the simplex method is so widely used.
Recently, linear programming approaches have experienced an increase of popularity for multiple object tracking. Jiang et al. [2007] propose a linear programming relaxation scheme for multiple object tracking problems with a convex inter-object interaction metric where the intra-object state continuity can be used as a metric. Berclaz et al. [2009] propose to use linear programming to solve the data association in a tracking-by-detection framework. They suggest a greedy search approach by reformulating the data association as a constraint flow optimization problem resulting in a convex problem for which the authors use linear programming to solve it. Finally, McLaughlin et al. [2015] propose to enhance linear programming approaches with motion modeling for multiple object tracking.

5.2.2 Weight estimation of individual energy terms as a linear programming problem

Equation C.4 and eq. C.5 are linear in the weight parameters. Thus, a linear programming approach can be envisioned for learning these weights. In order to pose the problem of weight estimation as a linear programming problem, an objective function needs to be formulated and the constraints under which to optimize it must be determined.
On the one hand, the posterior density function is a linear combination of the weight parameters for any given configuration $\mathbf{X}$. On the other hand, as shown in Chapter 3, only the ratio $\pi(\mathbf{X}')/\pi(\mathbf{X})$ has to be computed in the Markov chain transition without having to know the exact value of neither $\pi(\mathbf{X}')$ nor $\pi(\mathbf{X})$. Starting from

these properties, a set of constraints $\pi(\mathbf{X}')/\pi(\mathbf{X}) \geq 1$ (or ≤ 1) can be established, if one configuration is known to be better than another. These constraints can then be transformed into linear inequalities of the weight parameters. Finally, after collecting sufficiently many linear inequalities, linear programming can be applied to find a feasible solution for the weight parameters.
One way to know whether one configuration is better than another is to start from a small sample of ground truth information. Ground truth information is assumed to be available and the ground truth configuration is denoted with $\mathbf{X}^*$. The ground truth configuration $\mathbf{X}^*$ contains tracks with correct labels and locations. Different perturbation kernels are applied to degrade the ground truth configuration to the configurations $\mathbf{X}_i$. A general overview on perturbation kernels has been given in Section 3.3 and a more detailed discussion of perturbation kernels for object tracking will be given in Section 5.3. Currently, it is important to know that the perturbations are degrading the ground truth configuration to a less optimal one. As such, for each configuration $\mathbf{X}_i$, the following constraint holds:

$$\frac{\pi(\mathbf{X}^*)}{\pi(\mathbf{X}_i)} \geq 1. \tag{5.37}$$

Given one configuration, the log function of the posterior $f(\mathbf{C}|\mathbf{X}) \overset{not.}{=} \log(h(\mathbf{X}|\mathbf{Y}))$ is a linear function in terms of the weight parameters. Equation 5.37 provides one such linear inequality, i.e. $f(\mathbf{C}|\mathbf{X}^*) - f(\mathbf{C}|\mathbf{X}_i) \geq 0$. Indeed, by collecting multiple constraints the weight estimation problem can be posed as a linear programming problem as follows:

$$\text{Maximize: } \mathbf{a}^T\mathbf{C} \tag{5.38}$$

$$\text{Subject to: } A^T\mathbf{C} \leq \mathbf{b}, \quad \mathbf{C} \geq 0, \tag{5.39}$$

where $\mathbf{C} = [\gamma_{dyn}, \gamma_{label}, \gamma_o, \gamma_{ev}, \gamma_{cnt}]$, $\mathbf{a} = [1,1,1,1,1]^T$ for the Quality model (resp. $\mathbf{C} = [\gamma_{dyn}, \gamma_{label}, \gamma_o, \gamma_{stat}]$, $\mathbf{a} = [1,1,1,1]^T$ for the Statistical model) and each row of the form $A^T\mathbf{C} \leq \mathbf{b}$ encodes one constraint from eq. 5.37.
Care must be taken when adding inequalities to the set of constraints. As the ground truth information may contain small ambiguities, a small number of conflicting constraints may exist. It is crucial to identify these conflicting constraints in order to ensure that the objective function is feasible. Thus, any constraint that is in conflict with the current set of constraints is ignored.
Once the set of constraints has been created, the simplex algorithm is used to find the best solution to the optimization problem. Note that any vector $\mathbf{a}$ containing positive numbers is a solution to the problem. Thus, it is important to determine how many inequalities are necessary to ensure an accurate estimate of the parameters. In order to determine the necessary number of constraints, we simulate a density function with 5 parameters (for the Quality model) and respectively 4 parameters (for the Statistical model). For a given number of constraints, we independently generate multiple sets of constraints and evaluate the average normalized error as proposed by Yu and Medioni [2009]. Figure 5.8 depicts the correlation between the

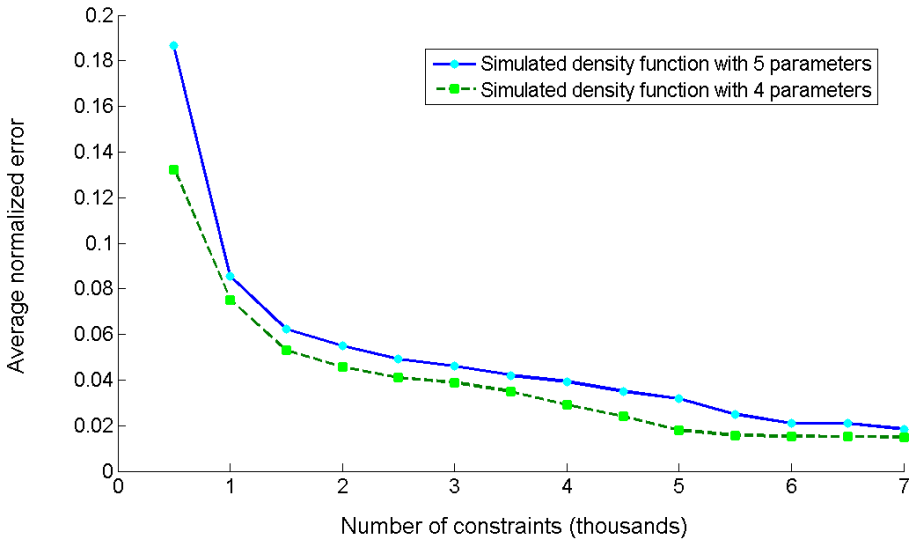


Figure 5.8: The average normalized error $||\hat{\mathbf{C}} - \mathbf{C}||/||\mathbf{C}||$ with respect to the number of constraints.

number of constraints and the average normalized error $||\hat{\mathbf{C}} - \mathbf{C}||/||\mathbf{C}||$. Based on these results, we set the number of constraints to 6.000 in our experiments for the Quality model and 5.000 constraints for the Statistical model as the average normalized error does not decrease significantly by using a higher number of constraints, as shown in Figure 5.8.
Once the parameters of the model have been trained, the density function can be optimized to determine the best configuration of objects that describe a given data set. In the following section, we will describe the optimization procedure used to simulate the spatio-temporal models presented in this chapter.

5.3 Optimization

The energies described in eq. C.4 and eq. C.5 are clearly not convex. It is easy to construct examples that have two virtually equal minima, separated by a wall of high energy values. The dependence caused by the high-order physical constraints is the main reason that drives the energy to be non-convex.
The target distribution is the posterior distribution of $\mathbf{X}$, i.e. $\pi(\mathbf{X}) = f(\mathbf{X}|\mathbf{Y})$, defined on a union of subspaces of different dimensions. The most widely known optimization method for non-convex energy functions and an unknown number of objects is the reversible jump Markov Chain Monte Carlo (RJMCMC) sampler developed by Green [1995]. RJMCMC uses a mixture of perturbation kernels $Q(\cdot,\cdot) = \sum_m p_m Q_m(\cdot,\cdot)$, $\sum_m p_m = 1$ and $\int Q_m(\mathbf{X}, \mathbf{X}')\mu(d\mathbf{X}') = 1$, to create tunnels through the walls of high energy (see Section 3.3 for further details).
Simulated annealing is used to find a minimizer of the energy function. The density

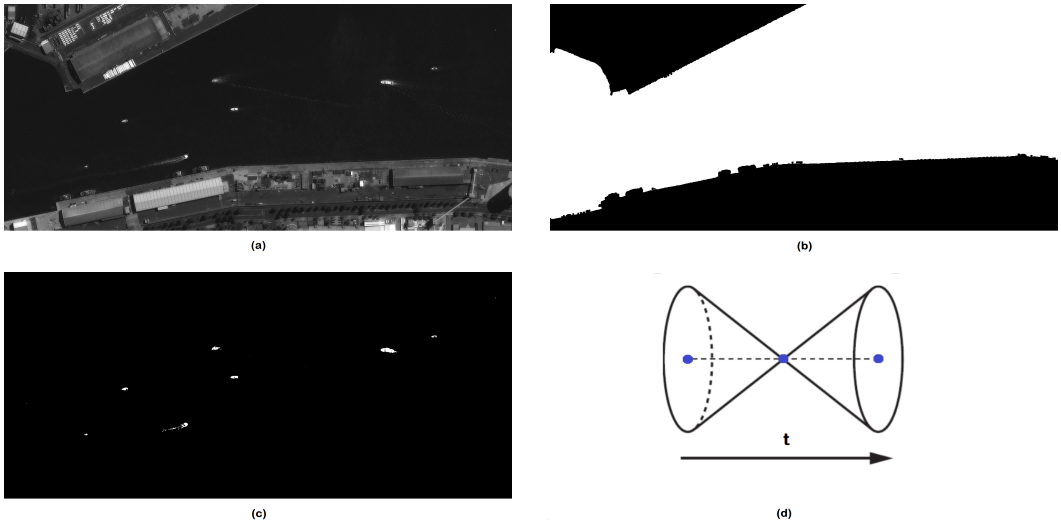


Figure 5.9: (a) Example input image from the Melbourne data set (©Airbus D&S); (b) Extracted water area using the algorithm presented in 4; (c) Initial birth map obtained using a simple threshold on the input image and restricted to the water area using the water mask; (d) Visualization of an event cone. The cone reaches both forwards and backwards in time. The event cone is used to increase the probability of a detection in the volume it influences.

function in eq. C.1 can be rewritten as:

$$f_{\theta,i}(X = \mathbf{X}|\mathbf{Y}) = \frac{1}{c_{Temp_i}(\theta|\mathbf{Y})} \exp^{-\frac{U_\theta(\mathbf{X},\mathbf{Y})}{Temp_i}} \tag{5.40}$$

where $Temp_i$ is a temperature parameter that tends to zero when i tends to infinity. If $Temp_i$ decreases in logarithmic rate, then X_i tends to a global optimizer of $f_{\theta,i}$. In practice however, a logarithmic law is not computationally feasible and hence, a geometric law is used instead to decrease the temperature. Therefore, a proper design of the perturbation kernels is needed to ensure a good exploration of the state space.

5.3.1 RJMCMC in 2D + T

The efficiency of RJMCMC depends on the variety of the perturbation kernels. The standard perturbation kernels described in Chapter 3 are sufficient to ensure the convergence of the RJMCMC sampler. Nevertheless, convergence can be very slow when using only standard kernels. Thus, the design of adapted perturbation kernels is crucial to ensure good convergence properties. The following perturbation kernels are used for joint object detection and tracking:

- *Birth and death according to a birth map*: As opposed to the classical birth and death kernel, the use of birth maps increases the efficiency of the optimization

scheme by proposing births and deaths in areas where objects are more likely to exist. Two types of maps are created in a pre-processing step:

1. *Birth maps*: since objects are supposed to have higher radiometric values than the background, a simple threshold technique is used to identify probable locations of objects in every frame and attribute higher probabilities to these locations for the birth proposition kernel. An example map for satellite images is shown in Figure 5.9 (c);
2. *Water mask*: In the case of tracking boats in satellite images the water area can be detected as shown in Figure 5.9 (b) and the search can be limited to such areas. Water/Land discrimination was discussed in Section 4.3.3. This mask is only computed for the Melbourne data set.

As a reminder, the birth and death according to a birth map kernel first chooses with probability p_b and $p_d = 1 - p_b$ whether an object u should be added to (birth) or deleted from (death) the configuration. If a birth is chosen, the kernel generates a new object u according to the birth map and proposes $\mathbf{X}' = \mathbf{X} \cup u$. If a death is chosen, the kernel selects one object u in $\mathbf{X}$ according to the birth map and proposes $\mathbf{X}' = \mathbf{X} \setminus u$;

- *Birth and death in a neighborhood*: this kernel is used to propose the addition or removal of an interacting pair of objects. To define the neighborhood of an object we introduce the notion of *event cones*. This notion was previously introduced by Leibe et al. [2008] to search for plausible trajectories in the space-time volume by linking up event cones. Following the idea of Leibe et al. [2008], the event cone of an object $u = (c_h, c_w, t, a, b, \omega, l)$ is defined as depicted in Figure 5.9 (d), to be the space-time volume it can physically influence from its current position;

- *Local transformations*: local transformations are transformations that randomly select an object u in the current configuration and then propose to replace it by a perturbed version of the object v: $\mathbf{X}' = (\mathbf{X} \setminus u) \cup v$. Translation, rotation and scale are examples of such transformations.

A mapping $R_m(\cdot, \cdot) : \mathcal{C} \times \mathcal{C} \to (0, \infty)$, called the Green ratio (or acceptance ratio), is associated to each of these perturbation kernels. At iteration i, the proposition $X_i = \mathbf{X}'$ is accepted with probability $\alpha_m = \min(1, R_m(\mathbf{X}, \mathbf{X}'))$. Otherwise $X_i = \mathbf{X}$ [Robert and Casella, 2005] (see Section 3.3 for further details).
An important aspect of multiple object tracking is not only the consecutive detection of the objects within the image sequence, but also the correct labeling of the objects such that consistent trajectories can be extracted. In our setting, the label of the object determines the trajectory to which it belongs. The perturbation kernels presented above do not act on the label of the objects to make them more consistent. We have analyzed two different approaches to object relabeling:

1. **Split/Merge perturbation moves**. A **split** move considers an existing trajectory and proposes to split it in two different trajectories at a given split

point. A **merge** move proposes to unite two distinct trajectories and relabel the objects such that they form a single, longer track. We describe the two perturbation moves, together with the associated Green ratios in Section 5.3.3.1;

2. **Deterministic labeling during the Birth move**. This labeling approach consists in assigning the label of a newly created object based on its neighborhood. When a new object u is created, a search is performed to find the nearest object v in its neighborhood and the label of u is set to the label of v. The computational speed-ups coupled with the good performance obtained motivate the use of such an approach. Further details are given in Section 5.3.3.2.

In the following section, we define the perturbation kernels used for object tracking and give their acceptance ratios in the particular case of ellipses. We then discuss the labeling approaches in Section 5.3.3.

5.3.2 Proposition kernels used for object tracking

We propose to perform the optimization of the spatio-temporal marked point process model presented in this chapter using a Metropolis-Hastings-Green (or RJMCMC) sampler whose general structure has been introduced in Chapter 3 (Section 3.3). As explained in Section 3.3.1, an interesting property of this sampler is the fact that the proposition kernel Q can be decomposed into a set of sub-kernels, each corresponding to a reversible perturbation. Although a single Birth and Death kernel is sufficient to guarantee the convergence of the sampler ([Green, 1995]), it is important to construct meaningful perturbations in order to speed up the convergence of the Markov chain. In this section, we present different sub-kernels: birth and death according to a birth map, birth and death in the neighborhood, local transformations and object relabel. The acceptance ratios for all perturbations are also described in this section. The birth and death moves change the dimension of the configuration by adding or deleting an object, while local perturbations and object relabel change the parameters of existing objects without modifying the number of objects in the configuration.
Consider the spatio-temporal marked point process models proposed in this chapter. The object space is $W = K \times M \subset \mathbb{R}^2 \times \{1, \cdots, T\} \times \mathbb{R}^4$ and an object u is given by $u = (x_u, y_u, t_u, a_u, b_u, \omega_u, l_u)$, where (x_u, y_u) is the object's location in space, t_u is the time frame in which the object exists, (a_u, b_u, ω_u) are the semi-major axis, semi-minor axis, respectively the orientation of the ellipse and l_u is the label associated to the object which is used to extract the trajectories.

- **Birth and death using a birth map**. A birth map is a mapping $b(\cdot)$ that assigns to each pixel in a frame the probability of that pixel containing the center of an ellipse. Each frame in a sequence has its associated birth map $b_t(\cdot)$. The birth maps transform the homogeneous Poisson intensity measure

$\nu(\cdot)$ into an inhomogeneous one, based on the location and time frame. Hence, objects are no longer chosen uniformly in W.
The kernel can be written as $Q_{BDM}(\mathbf{X},\cdot) = p_b(\mathbf{X})Q_{BM}(\mathbf{X},\cdot) + p_d(\mathbf{X})Q_{DM}(\mathbf{X},\cdot)$, where $p_b = p_d = 0.5$ represent the probability of a birth (resp. a death) and $Q_{BM}(\mathbf{X},\cdot)$ is the birth sub-kernel which creates new objects based on the associated birth maps and $Q_{DM}(\mathbf{X},\cdot)$ deletes an object accordingly.
The Green ratios can be written as follows:
Birth:
The kernel proposes a new configuration $\mathbf{X} \cup u$ and the Green ratio is given by:

$$R_{BM}(\mathbf{X}, \mathbf{X} \cup u) = \frac{h(\mathbf{X} \cup u)}{h(\mathbf{X})} \frac{p_d}{p_b} \frac{\nu(W) b_{t_u}(x_u, y_u)}{\sum_{v \in \mathbf{X}} b_{t_v}(x_v, y_v)}. \tag{5.41}$$

Death:
The kernel proposes a new configuration $\mathbf{X} \setminus u$ and the Green ratio is given by:

$$R_{DM}(\mathbf{X}, \mathbf{X} \setminus u) = \frac{h(\mathbf{X} \setminus u)}{h(\mathbf{X})} \frac{p_b}{p_d} \frac{\sum_{v \in \mathbf{X}} (b_{t_v}(x_v, y_v))}{\nu(W) b_{t_u}(x_u, y_u)}. \tag{5.42}$$

- **Birth and death in a neighborhood**. The birth and death kernel can be written as $Q_{BDN}(\mathbf{X},\cdot) = p_b(\mathbf{X})Q_{BN}(\mathbf{X},\cdot) + p_d(\mathbf{X})Q_{DN}(\mathbf{X},\cdot)$, where again $p_b = p_d = 0.5$ represent the probability of a birth (resp. a death) and $Q_{BN}(\mathbf{X},\cdot)$ is the birth sub-kernel which adds a new object to the configuration, while $Q_{DN}(\mathbf{X},\cdot)$ is the death kernel which removes an existing object from the configuration.
We are interested in the following neighborhood relation $\sim_t$:

$$u \sim_t v \leftrightarrow ||u - v||_2 \leq d_{max} \quad \text{with} \quad u \in \mathbf{x}_t, \quad v \in \mathbf{x}_{t\pm1} \quad ||u||_2 = \sqrt{x^2 + y^2} \tag{5.43}$$

The neighborhood relation $\sim_t$ exists between objects in adjacent frames. d_{max} describes the maximum distance at which an object v can be located in an adjacent frame of object u and still be in relation to u. This kernel can be synthesized as follows:
Birth:

1. Choose an object u among $\mathbf{X} = \{\mathbf{x}_1, \cdots, \mathbf{x}_T\}$, with $\mathbf{x}_t = \{u_1, \cdots, u_{n(\mathbf{x}_t)}\}$, $t \in \{1, \cdots, T\}$ according to the discrete law $\eta_b^{\mathbf{X}}(\cdot)$;
2. Generate v such that $v \in \partial(u)$, where $\partial(u)$ describes the neighborhood of object u according to a symmetric relation $\sim_t$ which is described in eq. 5.43;
3. Propose $\mathbf{X}' = \mathbf{X} \cup v$.

To generate a new object v in relation $\sim_t$ with u, we use a random vector z on the space Σ according to the law of a random variable Z and apply an

injection $\xi_u(\cdot)$ to the obtained result as presented in Section 3.3.1.4:

$$\Sigma \to W \tag{5.44}$$

$$\xi_u : \quad z \to v. \tag{5.45}$$

In our case, we can write:

$$\Sigma = \{(x,y) \in \mathbb{R}^2 \quad ||(x,y)||_2 \leq d_{max}\} \times \{t \pm 1\} \times M \tag{5.46}$$

and

$$z = (x,y,t,a,b,\omega,l) \quad \xi_u(z) = (x_u + x, y_u + y, t_u \pm 1, a, b, \omega, l). \tag{5.47}$$

The Jacobian in this case is 1 and if a uniform generator is used on Σ, then the Green ratio can be written as:

$$R_{BN}(\mathbf{X}, \mathbf{X} \cup v) = \frac{h(\mathbf{X} \cup v)}{h(\mathbf{X})} \frac{p_d}{p_b} \frac{1}{2\pi d_{max}^2} \frac{\eta_d^{\mathbf{X} \cup v}(v)}{\sum_{u \in \mathbf{X}} \eta_b^{\mathbf{X}}(u) \mathbb{1}(||u - v||_2 \leq d_{max})}, \tag{5.48}$$

where $\eta_b^{\mathbf{X}}(u) = \frac{1}{n(\mathbf{X})}$ and $\eta_d^{\mathbf{X}}(v) = \frac{\frac{1}{2} card\{\{u,w\} \in \mathcal{N}(\mathbf{X}), \quad v \in \{u,w\}\}}{card\{\mathcal{N}(\mathbf{X})\}}$., where $\mathcal{N}(\mathbf{X})$ represents the set of pairs $\{u, v\}$ that are in relation $\sim_t$ in $\mathbf{X}$.
The only missing ingredient is to built an uniform generator within a circle. The simplest solution is to generate objects in a square of length $2d_{max}$ until one object falls within the circle.
Death:

1. Choose an object $u \in \mathbf{X}$ such that $\partial(u) \cap (\mathbf{X} \setminus u) \neq \emptyset$ according to the law $\eta_d^{\mathbf{X}}(\cdot)$;
2. Propose $\mathbf{X}' = \mathbf{X} \setminus v$, where v is chosen uniformly from $\partial(u)$.

The Green ratio in this case can be written as:

$$R_{DN}(\mathbf{X}, \mathbf{X} \setminus v) = \frac{h(\mathbf{X} \setminus v)}{h(\mathbf{X})} \frac{p_b}{p_d} \frac{2\pi d_{max}^2}{1} \frac{\sum_{u \in \mathbf{X}} \eta_b^{\mathbf{X} \setminus v}(u) \mathbb{1}(||u - v||_2 \leq d_{max})}{\eta_d^{\mathbf{X}}(v)} \tag{5.49}$$

- **Local perturbations**. Small perturbations of an object are more efficient than a death immediately followed by a birth in terms of computation time. This type of perturbations are symmetric in order to guarantee the reversibility of the chain. Let $\mathcal{T} = \{T_a : a \in A\}$ be a family of symmetric perturbations parametrized by a. The kernel associated to this family consists in randomly choosing an object $u \in \mathbf{X}$ and proposing a perturbation by applying T_a to u. If both the object u and the transformation T_a are uniformly chosen, then the Green ratio can be written as:

$$R_L(\mathbf{X}, (\mathbf{X} \setminus u) \cup v) = \frac{h((\mathbf{X} \setminus u) \cup v)}{h(\mathbf{X})}. \tag{5.50}$$

We have used three types of local perturbations: translation, rotation and scale.

A **translation** is parametrized by a vector $[d_x, d_y]$, with $d_x \in [-\delta_x, \delta_x]$ and $d_y \in [-\delta_y, \delta_y]$. A transition $T_{[d_x,d_y]}$ corresponds to a translation of the center (x_u, y_u) of the ellipse u such that the new center is still in $K = [0, I_{h_{max}}] \times [0, I_{w_{max}}]$:

$$T_{[d_x,d_y]} \begin{pmatrix} x_u \\ y_u \\ t_u \\ a_u \\ b_u \\ \omega_u \\ l_u \end{pmatrix} = \begin{bmatrix} (x_u + d_x)[I_{h_{max}}] \\ (y_u + d_y)[I_{w_{max}}] \\ t_u \\ a_u \\ b_u \\ \omega_u \\ l_u \end{bmatrix} \tag{5.51}$$

The family of **rotations** is defined in $[-\delta_\omega, \delta_\omega]$ and a rotation T_{d_ω} modifies the orientation ω of an object u as follows:

$$T_{d_\omega} \begin{pmatrix} x_u \\ y_u \\ t_u \\ a_u \\ b_u \\ \omega_u \\ l_u \end{pmatrix} = \begin{bmatrix} x_u \\ y_u \\ t_u \\ a_u \\ b_u \\ (\omega_u + d_\omega)[\pi] \\ l_u \end{bmatrix}. \tag{5.52}$$

Finally, a **scale** is parametrized by a vector $[d_a, d_b]$ with $d_a \in [-\delta_a, \delta_a]$ and $d_b \in [-\delta_b, \delta_b]$. A scale perturbation modifies the length of the semi-major and semi-minor axis of an object u as follows:

$$T_{[d_a,d_b]} \begin{pmatrix} x_u \\ y_u \\ t_u \\ a_u \\ b_u \\ \omega_u \\ l_u \end{pmatrix} = \begin{bmatrix} x_u \\ y_u \\ t_u \\ (a_{min} + (a_u - a_{min} + d_a))[a_{max} - a_{min}] \\ (b_{min} + (b_u - b_{min} + d_b))[b_{max} - b_{min}] \\ \omega_u \\ l_u \end{bmatrix}. \tag{5.53}$$

5.3.3 Consistent labeling

In the previous section we mentioned two ways of assigning consistent labels to the objects within a configuration $\mathbf{X} = \mathbf{x}_1 \cup \cdots \cup \mathbf{x}_T$. The first option is to create two additional perturbation kernels that will either split a given trajectory in two, or merge two given trajectories in a single one. The second option is to deterministically assign labels to objects at birth. Both approaches have their advantages and drawbacks which we will discuss in the following subsections.

5.3.3.1 Split/Merge perturbations

We develop a perturbation kernel that acts only on the labels of the objects. For this, we define a relation $\sim_t$ such that:

$$\begin{aligned} u \sim_t v \Leftrightarrow & u \in \mathbf{x}_t, v \in \mathbf{x}_{t+1}, \forall t \in 1, \cdots, T-1 \text{ and} \\ & ||u,v||_2 < ||u,w||_2, \quad \forall w \in \mathbf{x}_{t+1}, ||u||_2 = \sqrt{x^2+y^2}. \end{aligned} \tag{5.54}$$

We define the neighborhood $\partial(u)$ of an object u in $\mathbf{x}_t$ as follows:

$$\partial(u) = \{v \in \mathbf{x}_{t+1} \text{ s.t. } u \sim_t v\}. \tag{5.55}$$

We define a trajectory τ, $\tau = \bigcup_{i=1}^{n(\tau)}\{u_i\}$, such that $u_i \sim_t u_{i+1}$ and $label(u_i) = label(u_{i+1}), \forall i = \overline{1, n(\tau)-1}$, where $n(\tau)$ is the length of the trajectory τ.
We define a maximal trajectory to be a trajectory such that no object with the same label is in relation $\sim_t$ with the starting object u_s at time t_s^τ or the ending object u_e at time t_e^τ of the trajectory:

$$\tau = \tau_m \Leftrightarrow \forall v \in \mathbf{x}_{t_s^\tau - 1}, \quad v \nsim_t u_s \text{ and } \forall w \in \mathbf{x}_{t_e^\tau + 1}, \quad w \nsim_t u_e. \tag{5.56}$$

We can now define a relationship $\sim$ between two maximal trajectories $\tau_{m,1}$ and $\tau_{m,2}$ if the last object $u_e^{\tau_{m,1}}$ is in relation $\sim_t$ with the first object $u_s^{\tau_{m,2}}$:

$$\tau_{m,1} \sim \tau_{m,2} \Leftrightarrow u_e^{\tau_{m,1}} \sim_t u_s^{\tau_{m,2}}. \tag{5.57}$$

Let $\Upsilon = \{\tau_{m,1}, \cdots, \tau_{m,k}\}$ be the set of all maximal trajectories that exist within a configuration $\mathbf{X}$. Furthermore, let T_{d_l} be a symmetric transformation such that $d_l \in [0, L]$ and T_{d_l} changes the label of an object u as follows:

$$T_{d_l}\begin{pmatrix} x_u \\ y_u \\ t_u \\ a_u \\ b_u \\ \omega_u \\ l_u \end{pmatrix} = \begin{bmatrix} x_u \\ y_u \\ t_u \\ a_u \\ b_u \\ \omega_u \\ d_l \end{bmatrix}. \tag{5.58}$$

We can now define the split/merge perturbation kernel $Q_{SM}(\mathbf{X}, \cdot) = p_S(\Upsilon)Q_S(\mathbf{X}, \cdot) + p_M(\Upsilon)Q_M(\mathbf{X}, \cdot)$, as follows:

- **Split** $(Q_S(\mathbf{X}, \cdot))$:
  1. Select a maximal trajectory τ_m;
  2. Select a split point u_τ according to a probability law $\eta^\tau(u_i)$;
  3. For all objects $u_i \in \tau$ that exist after the split point u_τ, apply a transformation T_{d_l};
  4. Propose $(\mathbf{X} \setminus \{u_i \in \{u_\tau, \cdots, u_e^\tau\}\}) \cup \{T_{d_l}(u_i \in \{u_\tau, \cdots, u_e^\tau\})\}$;

- **Merge** ($Q_M(\mathbf{X}, \cdot)$):
  1. Select two maximal trajectories $\tau_{m,1}$ and $\tau_{m,2}$ such that $\tau_{m,1} \sim \tau_{m,2}$;
  2. For all objects $u_i \in \tau_{m,2}$, apply a transformation $T_{d_{l_{\tau_{m,1}}}}$;
  3. Propose $(\mathbf{X} \setminus \{u \in \tau_{m,2}\}) \cup \{T_{d_{l_{\tau_{m,1}}}}(u \in \tau_{m,2})\}$

We denote by $n(\Upsilon)$, the number of trajectories in Υ, by $n_\sim(\Upsilon)$, the number of second order cliques of trajectories in Υ with respect to the $\sim$-relation and by $n(\tau)$ the length of a trajectory τ (more precisely, the number of objects $u \in \mathbf{X}$ that form the trajectory τ). The probability of proposing a split or a merge can be computed as:

$$p_S(\Upsilon) = \frac{n(\Upsilon)}{n_\sim(\Upsilon) + n(\Upsilon)} \tag{5.59}$$

and

$$p_M(\Upsilon) = \frac{n_\sim(\Upsilon)}{n_\sim(\Upsilon) + n(\Upsilon)}. \tag{5.60}$$

The associated perturbation kernels are:

- In case of a **split**:

$$Q_S(\mathbf{X}, A) = \frac{1}{n(\Upsilon)} \sum_{\tau_{m,i}} \frac{1}{n(\tau_{m,i})} \sum_{u \in \tau_{m,i}} \mathbb{1}_A((\mathbf{X} \setminus \{u_i \in \{u_\tau, \cdots, u_e^\tau\}\}) \cup \{T_{d_l}(u_i \in \{u_\tau, \cdots, u_e^\tau\})\}) \tag{5.61}$$

- In case of a **merge**:

$$Q_M(\mathbf{X}, A) = \frac{1}{n_\sim(\Upsilon)} \sum_{\tau_{m,i} \sim \tau_{m,j}} \sum_{u \in \tau_{m,2}} \mathbb{1}_A((\mathbf{X} \setminus \{u \in \tau_{m,2}\}) \cup \{T_{d_{l_{\tau_{m,1}}}}(u \in \tau_{m,2})\}). \tag{5.62}$$

The acceptance ratio for the split/merge perturbation kernel depends on whether a split or a merge perturbation is proposed:

- In case of a **split**, the Green ratio is given by:

$$R(\mathbf{X}, \mathbf{X}') = \frac{p_M(\mathbf{X}')}{p_S(\mathbf{X})} \frac{h(\mathbf{X}')}{h(\mathbf{X})} \frac{n_{\sim,\mathbf{X}}(\Upsilon)}{(n_\mathbf{X}(\Upsilon) - 1)n(\tau)} \tag{5.63}$$

  where $\mathbf{X}' = (\mathbf{X} \setminus \{u_i \in \{u_s^\tau, \cdots, u_e^\tau\}\}) \cup \{T_{d_l}(u_i \in \{u_s^\tau, \cdots, u_e^\tau\})\}$.

- In case of a **merge**, the Green ratio is given by:

$$R(\mathbf{X}, \mathbf{X}') = \frac{p_S(\mathbf{X}')}{p_M(\mathbf{X})} \frac{h(\mathbf{X}')}{h(\mathbf{X})} \frac{n_\mathbf{X}(\Upsilon)(n(\tau_{m,1}) + n(\tau_{m,2}))}{n_{\sim,\mathbf{X}}(\Upsilon) + 1} \tag{5.64}$$

  where $\mathbf{X}' = (\mathbf{X} \setminus \{u \in \tau_{m,2}\}) \cup \{T_{d_{l_{\tau_{m,1}}}}(u \in \tau_{m,2})\}$.

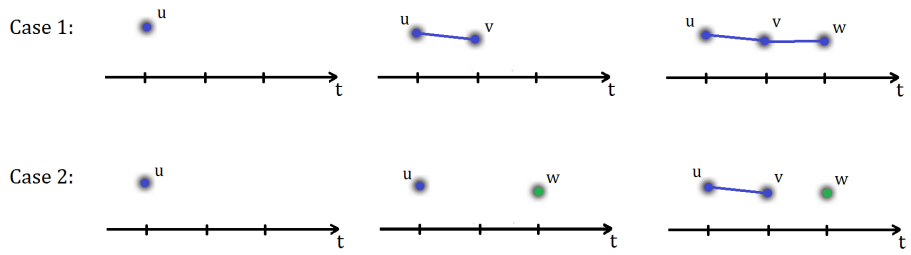


Figure 5.10: The state of the Markov chain depends on the order in which the objects are added to the configuration. Case 1: $\mathbf{X} = ((\mathbf{X} \cup u) \cup v) \cup w$ (first u, then v and then w was created). Case 2: $\mathbf{X} = ((\mathbf{X} \cup u) \cup w) \cup v$ (first u, then w and then v was created).

5.3.3.2 Deterministic label assignment

A different approach of assigning labels to the objects in order to build consistent trajectories is by setting them in a deterministic way. Hence, instead of randomly assigning a label to the object when it is created, we determine the label of the object at birth, based on its neighborhood.
The labels are assigned based on the motion model. Given object u centered at location $pos(u) = (c_h(u), c_w(u))$, the objects in the adjacent frames are identified that are likely to satisfy the motion model. The distance between u and these objects is computed and compared to a threshold. If the distance is smaller than the threshold, the label of object u is set to the label of the object in the previous frame. Otherwise, a new random label from $[0, L] \setminus labels(\mathbf{x_t})$ is assigned to u. Configurations $\mathbf{X}$ that contain two or more objects with the same label at any time instance t are not permitted, meaning that an infinite energy is assigned to such configurations.
The drawback of this approach is that the labels depend on the order of exploration. For instance, consider three neighboring objects u, v, w according to some neighborhood relation $\sim_t$, such that $u \in \mathbf{x}_t$, $v \in \mathbf{x}_{t+1}$, $w \in \mathbf{x}_{t+2}$ and $u \sim_t v$ and $v \sim_t w$. Let us analyze the labels of the objects, depending on the order in which they are created:

1. $\mathbf{X} = ((\mathbf{X} \cup u) \cup v) \cup w$ (first u, then v and then w were created): If we assume that u is not in relation to any object before the creation of v, then the label of u is set randomly, $label(u) = l$. When object v is created, it is in relation with u and thus, $label(v) = label(u) = l$. Finally, when object w is created, it is in relation to object v and hence, $label(w) = label(v) = label(u) = l$. In the end, we obtain a configuration with all three objects sharing the same label l;

2. $\mathbf{X} = ((\mathbf{X} \cup u) \cup w) \cup v$ (first u, then w and then v were created): Again, if we assume that u is not in relation to any object at its birth, then the label of u

is set randomly, $label(u) = l_1$. Now, object w is added to the configuration. Note that $w \in \mathbf{x}_{t+2}$ while $u \in \mathbf{x}_t$ and hence, u and w are not neighbors. Accordingly, the label of w is set randomly, $label(w) = l_2$. Finally, v is added and since it is in relation to u, the label of v is $label(v) = label(u) = l_1$. In this case however, at the end we obtain a configuration where the labels of the three objects are not the same anymore.

This simple example, depicted in Figure 5.10, shows that the state of the Markov chain depends on the order in which the objects are added to the configuration. Nevertheless, as we will show in Section 5.4, this approach offers a high tracking accuracy while reducing the computational burden compared to the split/merge approach.

Finally, the kernels presented in Section 5.3.2 together with the labeling approaches have been designed to improve the convergence speed of the sampler. The sampler has been embedded into a simulated annealing scheme for better performance. Nevertheless, this also led to an increased computational cost. In order to tackle this problem, we have proposed two novel approaches:

- **Integrating Kalman like moves in RJMCMC**. Sequential filters have proven to provide relatively fast and reliable tracking performances in particular for single target tracking. The properties of sequential filters can be efficiently exploited within the RJMCMC sampling scheme. The filter is used to generate more meaningful perturbation proposals which are then evaluated using an appropriate Green acceptance ratio. Better perturbation proposals increase the acceptance probability of the overall RJMCMC sampling scheme which in turn leads to a faster convergence.

- **Parallel implementation of RJMCMC**. A parallel implementation similar to the one presented in Chapter 4 is proposed. Special attention needs to be given to ensure that the parallel perturbations are independent in both the spatial dimensions as well as in time.

The two techniques are further investigated in this section. First, the integration of sequential filters-like moves within the RJMCMC sampling scheme is discussed. A dedicated Birth and Death kernel using Kalman-like moves is introduced. The Kalman filter is chosen for its computational efficiency and ease of implementation. However, particle filters can also be used instead. Next, a parallel implementation of the RJMCMC sampler is proposed. The implementation considers the perturbation kernels previously introduced in this section and does not extend to the hybrid birth and death kernel with sequential filters-like moves.

5.3.4 Integrating Kalman like moves in RJMCMC

As opposed to the classical birth and death kernel, a problem-specific birth and death kernel is proposed that uses a Kalman filter within the birth step. This

allows to create tracklets (i.e. ellipses with the same label in consecutive frames) in a single step. The Kalman filter dates back to 1960, when Kalman [1960] described a recursive solution to the discrete-data linear filtering problem. This filter became very popular and multiple variations and extensions of it have been designed to adjust the filter to diverse problems ([Bar-Shalom and Fortmann, 1988, Haug, 2012]).

5.3.4.1 Kalman filter design

The Kalman filter (KF) is applied to estimate the state of an object, where the state is assumed to be linearly Gaussian distributed in time. The continuity of the motion serves as a strong prediction criterion in object tracking. To generate new tracklets, the system is modeled as linear Gaussian, with the state parameters of the Kalman filter given by the ellipse location, its velocity, its size and its orientation. For a single object, the discrete-time dynamic equation is given by:

$$\mathbf{KX}_{t+1} = \mathbf{F} \cdot \mathbf{KX}_t \tag{5.65}$$

where the state vector is given by $\mathbf{KX} = [c_w, c_h, \dot{c_w}, \dot{c_h}, a, b, \omega]$, where c_w and c_h are the predicted coordinates of the ellipse, $\dot{c_w}$ and $\dot{c_h}$ are the velocities in the respective direction, a and b are the semi-major and semo-minor axis of the ellipse and ω is the orientation; and

$$\mathbf{F} = \begin{pmatrix} 1 & 0 & dt & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & dt & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix}, \tag{5.66}$$

is the a priori known transition matrix, where dt is set to $dt = 1$ in the experiments. The measurement vector $\mathbf{M}$ is obtained using a simple threshold applied to the frame differencing output to identify moving objects in every frame. At each time frame, the foreground blobs are identified and their center location, width, height and orientation are obtained. A measurement vector $\mathbf{M_t}$ is constructed, that can be injected into the measurement model of the KF. Accordingly:

$$\mathbf{M}_t = \mathbf{H} \cdot \mathbf{KX}_t + q_t \tag{5.67}$$

with $q_t \sim \mathcal{N}(0, \mathbf{R_t})$ being white Gaussian noise with covariance matrix $\mathbf{R}_t$ and $\mathbf{H}$ is the measurement function such that:

$$\mathbf{H} = \begin{pmatrix} 1 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix}. \tag{5.68}$$

Since a known model is assumed for the dynamics of an object, the KF can be used to predict the position of the object in the next frame. The KF state prediction $\bar{\mathbf{KX}}_{t+1}$ and the state covariance prediction $\bar{\mathbf{P}}_{t+1}$ are defined by:

$$\bar{\mathbf{KX}}_{t+1} = \mathbf{F} \cdot \hat{\mathbf{KX}}_t \tag{5.69}$$

$$\bar{\mathbf{P}}_{t+1} = \mathbf{F} \cdot \hat{\mathbf{P}}_t \cdot \mathbf{F}^T \tag{5.70}$$

where $\hat{\mathbf{KX}}_t$ and $\hat{\mathbf{P}}_t$ are respectively the estimated state vector and error covariance matrix at time t.
Then, the KF update step is as follows:

$$\mathbf{K}_{t+1} = \bar{\mathbf{P}}_{t+1} \cdot \mathbf{H}^T \left(\mathbf{H} \cdot \bar{\mathbf{P}}_{t+1} \cdot \mathbf{H}^T + \mathbf{R}_{t+1}\right)^{-1} \tag{5.71}$$

$$\hat{\mathbf{KX}}_{t+1} = \bar{\mathbf{KX}}_{t+1} + \mathbf{K}_{t+1} \left(\mathbf{M}_{t+1} - \mathbf{H}_{t+1} \cdot \bar{\mathbf{KX}}_{t+1}\right) \tag{5.72}$$

$$\hat{\mathbf{P}}_{t+1} = (\mathbf{I} - \mathbf{K}_{t+1} \cdot \mathbf{H}) \cdot \bar{\mathbf{P}}_{t+1}. \tag{5.73}$$

The KF starts with the initial conditions given by $\mathbf{K}_0$ and $\bar{\mathbf{P}}_0$. $\mathbf{K}_t$ is called Kalman gain and defines the updating weight between the new measurements and the prediction from the dynamic model.

5.3.4.2 Birth and death using the Kalman filter

The **Birth and Death using a Kalman filter** kernel first chooses with probability p_b and $p_d = 1 - p_b$ whether an object u should be added to (birth) or deleted from (death) the configuration. If a death is chosen, the kernel selects one object u in $\mathbf{X}$ and proposes $\mathbf{X}' = \mathbf{X} \setminus u$. However, if a birth is chosen, the kernel generates a new object u and proposes $\mathbf{X}' = \mathbf{X} \cup u$. If the birth is accepted, a Kalman filter is initialized to the location, size and orientation of u. The state at time t is updated using the measurements $\mathbf{M}_t$ and then a prediction step is executed. The kernel generates a new object v based on the state vector $\bar{\mathbf{KX}}_{t+1}$ of the KF and proposes $\mathbf{X}'' = \mathbf{X}' \cup v$. The state of the filter is updated using $\mathbf{M}_{t+1}$ and a new prediction is made. The process is repeated until a birth proposal is rejected. Note that the standard birth and death kernel is a particular case of the proposed perturbation kernel, when the proposal $\mathbf{X}'' = \mathbf{X}' \cup v$ is rejected.

5.3.4.3 Perturbation kernel and corresponding acceptance ratio

The Birth and death using a Kalman filter can be divided into two stages:

- **first stage**: a birth and death using a birth map perturbation is performed;
- **second stage**: depending on the outcome of the first stage, we have the following cases:
  - If a birth was performed and accepted at the first stage, then propose a new birth based on the Kalman filter output;
  - Otherwise, do nothing.

The first stage of this kernel is identical to the birth and death using a birth map perturbation which was discussed in Section 5.3.2. We can define a kernel for the second stage of the perturbation as follows:

$$Q_{KF}(\mathbf{X},\cdot) = p_b(\mathbf{X})Q_{BKF}(\mathbf{X},\cdot) + p_d(\mathbf{X})Q_{DKF}(\mathbf{X},\cdot) \tag{5.74}$$

where the birth kernel is defined using the prediction of the Kalman filter:

$$Q_{BKF}(\mathbf{X}\cup u, A) = \int_{v\in W} \mathbb{1}_A(\mathbf{X}\cup u\cup v)\frac{\mathcal{N}(v;\overline{\mathbf{KX}}_{v|u},\mathbf{P}_{v|u})\nu(dv)}{\nu(W)}, \tag{5.75}$$

where $\mathcal{N}(v;\mathbf{KX},\mathbf{P})$ is the Gaussian probability density function with mean $\mathbf{KX}$ and covariance matrix $\mathbf{P}$ evaluated at v; and the death kernel consists in choosing an object from the configuration based on the birth map and deleting it:

$$Q_{DKF}(\mathbf{X}, A) = \sum_{v\in\mathbf{X}} \mathbb{1}_A(\mathbf{X}\setminus v)\frac{b_{t_v}(x_v,y_v)}{\sum_{w\in\mathbf{X}} b_{t_w}(x_w,y_w)}. \tag{5.76}$$

We can consider the symmetric measure $\phi(A\times B) = \int_{\mathcal{C}}\int_{v\in W}\mathbb{1}_A(\mathbf{X})\mathbb{1}_B(\mathbf{X}\cup v)\nu(dv)\mu(d\mathbf{X}) + \int_{\mathcal{C}}\mathbb{1}_A(\mathbf{X})\sum_{v\in\mathbf{X}}\mathbb{1}_B(\mathbf{X}\setminus v)\mu(d\mathbf{X})$. We have shown in Chapter 3 that this measure is symmetric and dominates $\pi(d\mathbf{X})Q(\mathbf{X},d\mathbf{X}')$. Hence, we can write the acceptance ratios for the two cases as follows:
Birth:

$$R(\mathbf{X}\cup u, \mathbf{X}\cup u\cup v) = \frac{p_d}{p_b}\frac{h(\mathbf{X}\cup u\cup v)}{h(\mathbf{X}\cup u)}\frac{b_{t_v}(x_v,y_v)\mathcal{N}(v;\overline{\mathbf{KX}}_{v|u},\mathbf{P}_{v|u})}{\sum_{w\in\mathbf{X}} b_{t_w}(x_w,y_w)}; \tag{5.77}$$

Death:

$$R(\mathbf{X}\cup u, \mathbf{X}\cup u\setminus v) = \frac{p_b}{p_d}\frac{h(\mathbf{X}\cup u\setminus v)}{h(\mathbf{X}\cup u)}\frac{\sum_{w\in\mathbf{X}} b_{t_w}(x_w,y_w)}{b_{t_v}(x_v,y_v)\mathcal{N}(v;\overline{\mathbf{KX}}_{v|u},\mathbf{P}_{v|u})}. \tag{5.78}$$

Another way to reduce the computation time is to propose a parallel implementation of the RJMCMC sampler. Nowadays, conventional computers have usually more than two processing units and thus, parallel implementations are becoming ubiquitous. In chapter 4 a parallel implementation of RJMCMC for object detection in static images has been described. The parallel implementation is further extended to 2D + T dimensions for tracking purposes.

5.3.5 Efficient implementation of RJMCMC in 2D + T - multiple cores

The first major step towards a time-efficient approach to simulate MPPs came with the development of the parallel sampler devised by Verdié and Lafarge [2012]. The main idea in the $2D$ case is to divide the search space using a quadtree $\mathcal{K}$. The cells of the quadtree are divided into 4 independent sets called mic-sets and denoted

$S_{mic} = S_k$, with $k = \overline{1,4}$. At each iteration, one mic-set is selected and the optimization step is performed in parallel within all the cells, c, contained in it. The general proposition kernel $\mathcal{Q}$ is formulated as a mixture of uniform sub-kernels $\mathcal{Q}_{c,\gamma}$, where $\gamma \in \Gamma = \{\text{birth and death, translation, rotation, scale}\}$. The probability of each proposition kernel is computed as $q_{c,t} = \frac{Pr(\gamma)}{\#\text{ cells in } \mathcal{K}}$. The computational efficiency of the sampler has been proven for a large number of applications. The sampler proposed by Verdié and Lafarge [2012] makes use of GPU computing to perform the optimization.

In this work, a multiple-core version of the sampler is implemented with a shared

Algorithm 9 Parallel version of the RJMCMC sampler

1. Initialize $X_0 = \mathbf{X_0}$ and $i = 0$;
2. Compute the water mask (for the Melbourne data set only);
3. Compute the data-driven space partitioning tree $\mathcal{K}$ and truncate it based on the water mask (if it exists);
4. At iteration i with $X_i = \mathbf{X}$:
   - For all even frames:
   - Choose a mic-set $S_{mic} \in \mathcal{K}$ and a kernel type $\gamma \in \Gamma$ according to the probability $\sum_{c \in S_{mic}} q_{c,\gamma}$
   - For each cell c in S_{mic}:
     - Perturb $\mathbf{X}$ in cell c to a configuration $\mathbf{X}'$ according to $\mathcal{Q}_{c,\gamma}(\mathbf{X} \to \cdot)$;
     - Retrieve the configuration $\mathbf{Z}$ from the neighboring cells;
     - Compute the Green ratio:

$$R = \frac{\mathcal{Q}_{c,\gamma}(\mathbf{X}' \cup \mathbf{Z} \to \mathbf{X} \cup \mathbf{Z})}{\mathcal{Q}_{c,\gamma}(\mathbf{X} \cup \mathbf{Z} \to \mathbf{X}' \cup \mathbf{Z})} \exp \frac{U(\mathbf{X} \cup \mathbf{Z}) - U(\mathbf{X}' \cup \mathbf{Z})}{T_i}$$

     - Choose $X_{i+1} = \mathbf{X}'$ with probability $\min(1, R)$ and $X_{i+1} = \mathbf{X}$ otherwise;
   - Update $T_{i+1} = \alpha T_i$ (in this work $\alpha = 0.95$);
   - Repeat for uneven frames.

memory between the cores. At each iteration, for each cell in a chosen mic-set S_k, $k = \overline{1,4}$, the configurations within the neighboring cells are taken into consideration. Note that this does not lead to synchronization problems when accessing the shared memory since processors only read information in the neighboring cells and do not modify it. The reasoning behind the use of a multiple-core sampler was described in Section 4.3.2.

In this section, the parallel version of the RJMCMC sampler is extended to the

$2D + T$ framework, the parallel sampler being detailed in Algorithm 9. The idea behind this extension is based on the fact that independent perturbations in time can be performed for any frame t, if the configurations $\mathbf{x_{t-1}}$ and $\mathbf{x_{t+1}}$ are kept constant. These configurations have to remain constant because the energy term that incorporates the motion model into the energy for an object at time t is computed using three consecutive frames, as defined in eq. 5.4. Based on this observation, one iteration of the sampler can be divided in two parts as shown in Algorithm 9 (step 4): first, perturbations are performed in parallel on all even frames in the batch by keeping all uneven frames constant; second: the process is repeated for uneven frames by maintaining the even frames constant.
The two optimization approaches described in this section are fundamentally different, but both serve the same purpose: a faster optimization of the dedicated energy function previously described.
In the following section, results on various types of data sets from two different fields, remote sensing and fluorescent microscopy, are presented.

5.4 Results

The proposed models were applied to a large variety of data sets, which can be categorized as follows:

- **Synthetic benchmarks used by the biological processing community**. Several sequences were generated with various levels of noise and different number of objects. The objects can enter or exist the region of interest at any time and location. These data sets have been extensively used to analyze the performance of the method, given the existence of precise ground truth information;

- **Real biological data**. One sequence of real biological data, provided by courtesy of J. Salamero, PICT IBiSA, UMR 144 CNRS Institut Curie ([Basset et al., 2014]), is used to show the applicability of the proposed method for real applications in the biological context;

- **Simulated satellite data**. These data sets are real satellite image sequences which were altered to resemble the output of a geostationary satellite. A large variety of samples is explored ranging from static cameras to moving cameras, high temporal frequencies to low temporal frequencies, a single object type to multiple object types that have to be detected and tracked over the sequence.

The results for each category of data are illustrated using a small number of representative examples. More specifically, since the benchmark biological data can be generated with the associated ground truth, the limits of the proposed approach are tested w.r.t. dense environments in sequences with either very low temporal frequencies ($1 - 2$Hz) as well as very high temporal frequencies (up to 60Hz).

5.4.1 Tracking results on synthetic benchmarks used by the biological processing community

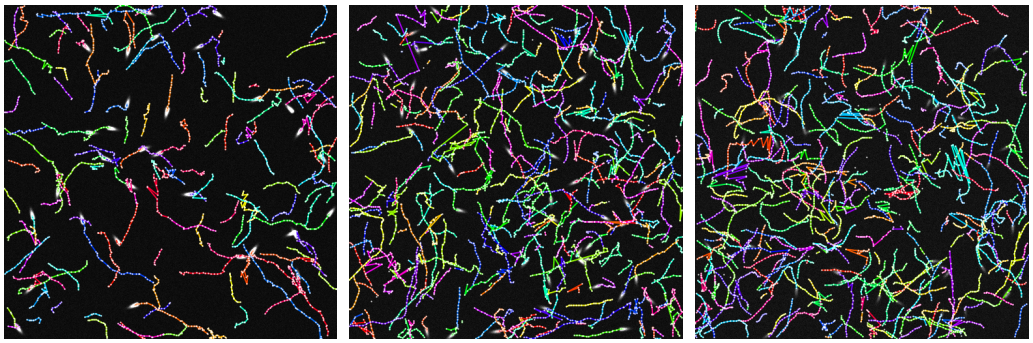


Figure 5.11: Detection and tracking results obtained on three synthetic biological image sequences of 100 frames each (©INRIA). Each color represents a different track. Objects can appear and disappear at any location and time instance. Each sequence has a different level of Gaussian noise (left: no noise; middle: $\mu = 25$, $\sigma = 2.5$; right: $\mu = 50$, $\sigma = 5.0$).

Data set	No. ground truth tracks	No. tracks			Similarity between tracks			Similarity between detections		
		MHT	A. Milan et al. [2014]	Proposed algorithm	MHT	A. Milan et al. [2014]	Proposed algorithm	MHT	A. Milan et al. [2014]	Proposed algorithm
Seq. 1	160	363	221	197	43.7%	61.7%	**70.1%**	38.2%	57.8%	**72.4%**
Seq. 2	318	715	461	395	42.9%	60.9%	**70.0%**	35.9%	55.6%	**69.7%**
Seq. 3	335	752	474	385	43.6%	61.3%	**71.9%**	37.0%	56.9%	**61.5%**

Table 5.1: Quantitative analysis of the detection and tracking results obtained using the built-in MHT tracker within Icy ([de Chaumont et al., 2012]), the continuous energy minimization algorithm developed by Milan et al. [2014] and the proposed method for the three synthetic biological image sequences. The proposed algorithm has the highest similarity scores w.r.t. the ground truth, both in terms of tracks and detections. The proposed algorithm outperforms current state of the art methods by more than 5%.

The proposed approach is first tested on three synthetic biological image sequences. The image sequences have been generated with Icy ([de Chaumont et al., 2012]), an online available toolbox for biological image sequences developed by the Quantitative Image Analysis Unit at the Pasteur Institute in Paris. The ISBI Cell Tracking Challenge in 2013, led to the development of an Icy plug-in called "ISBI Challenge" which can be used to generate synthetic sequences with the associated ground truth.
The sequences consist of 100 images, each 512×512 pixels, with approximately 35 objects per frame. The three sequences exhibit different levels of Gaussian noise. Targets can appear and disappear at any location and time instance. The extracted

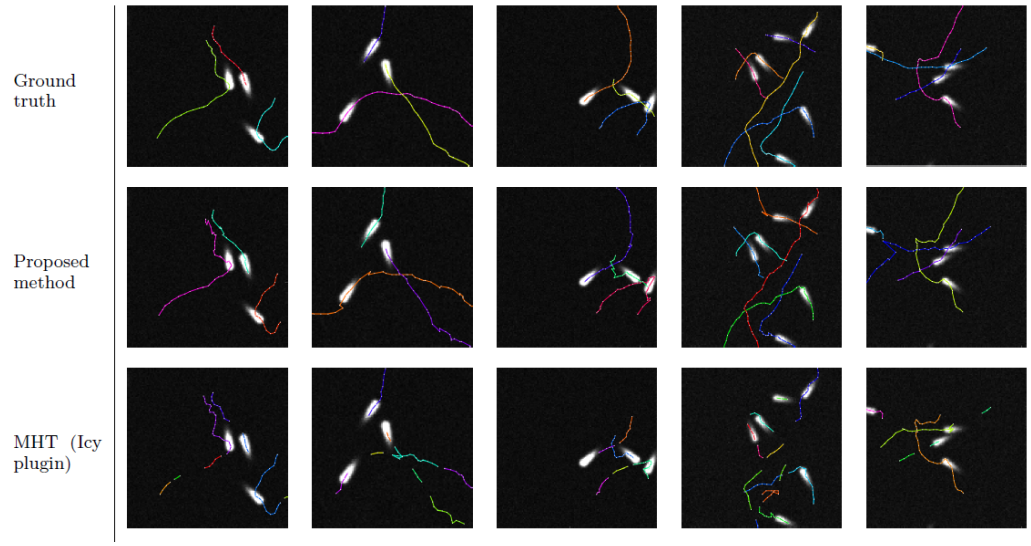


Table 5.2: Difficult scenarios with two or more crossing or by-passing trajectories in synthetic biological image sequences. First row: Ground truth trajectories. Second row: Trajectories obtained using proposed method (©INRIA). The trajectories closely resemble the ground truth trajectories. Labels are correctly preserved during the crossings or by-passing. Third row: Trajectories obtained using MHT (Icy plugin). The trajectories are highly fragmented. The labels are switched or objects are lost during crossings or by-passing.

tracks are depicted in Figure 5.11 and the quantitative tracking results are displayed in Table 5.1. The objects are considered to exhibit a directed motion.

5.4.1.1 Tracking results of the models

The proposed methods are compared with the built-in Multiple Hypothesis Testing (MHT) particle tracker that comes as a plug-in to the Icy software [De Chaumont et al., 2012]. The superiority of the proposed models can be easily observed from the detection and tracking similarity values in Table 5.1. The detection and tracking similarity w.r.t. the ground truth is almost double as compared to that of the built-in tracker. The tracking results of the MHT plug-in have also been fed to the recently developed continuous energy minimization scheme proposed by Milan et al. [2014] for multiple target tracking. This led to better detection and tracking results. However, the proposed approach outperforms this result by more than 5% in terms of detection and track similarity with the ground truth.
The joint detection and tracking method extracts long, smooth tracks. Moreover, it can correctly handle object appearances and disappearances. In Table 5.2 several zoom-ins are shown on very difficult scenarios where two or more trajectories are crossing or passing close to each other. The proposed method correctly maintains the labels and does not segment the tracks as the MHT tracker.

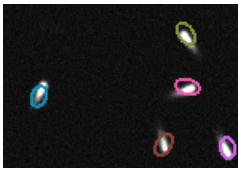
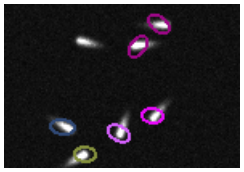
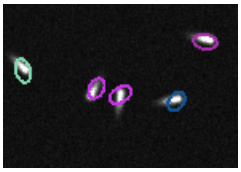
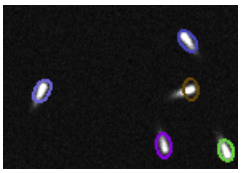
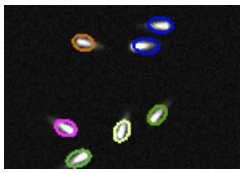
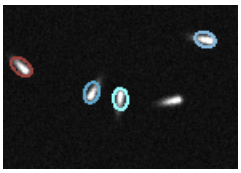
Statistical model			
Quality model			

Table 5.3: Quality of the detection using the Statistical model (top) and the Quality model (bottom).

5.4.1.2 Quality model vs. Statistical model

The main difference between the Quality model and the Statistical model introduced in Section 5.1.2 lies in the ability to correctly identify the objects and determine the parameters of the corresponding ellipses as closely as possible to the actual parameters of the targets.
On the one hand, the Quality model has two external energy terms: one favors the detection of a target (the evidence term) and the second favors a strong alignment between the ellipse and the target (the contrast distance measure). Hence, a low external energy will correspond to a configuration that has the largest overlap with the target configuration. This model is very efficient when the interest lies in both tracking the targets and segmenting them accurately. However, it is difficult to identify the proper parameters of the model when a predetermined probability of detection/false alarms is desired.
On the other hand, the Statistical model has a single external energy term that links the motion present in consecutive frames to the probability of detecting a target. The model can be easily adapted to the needs of the user in terms of detection/false alarms probability by adjusting the threshold τ used for the matrix $\mathbf{M}_{GLRT}$ (see Section 5.1.2). The drawback of this model w.r.t. the Quality model lies in the quality of the detection. Whereas this model only determines the presence/absence of a target, the Quality model produces a better segmentation of the target. Table 5.3 shows the detection results using the two models. A clear difference is observable in terms of the quality of the detection. Nevertheless, the tracking performance is not influenced, as shown in Table 5.1.

5.4.1.3 Split/Merge perturbation kernel vs. Deterministic labeling

In Section 5.3 we have developed a RJMCMC framework for simulating the proposed models. An important aspect in tracking is the ability to assign consistent labels to objects throughout the image sequence. As such, we have proposed two

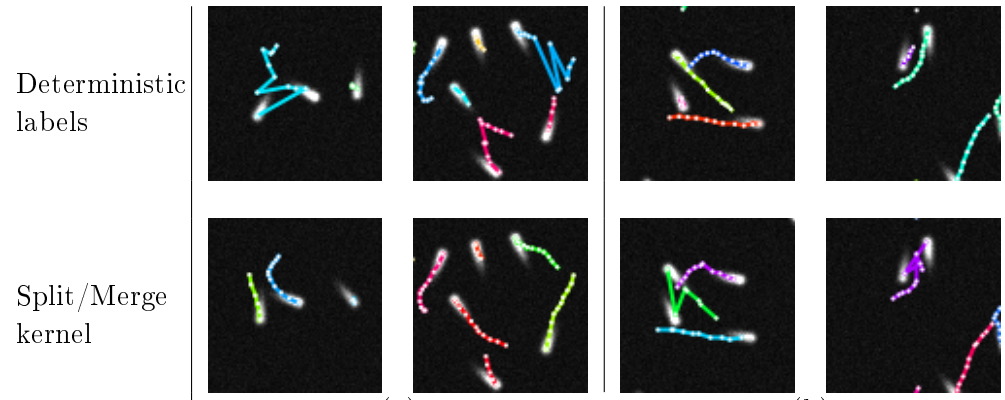


Table 5.4: (a) Examples when the deterministic labeling outperforms the split/merge approach. (b) Examples when the reverse is true.

ways of assigning the labels of the objects: a split/merge perturbation kernel which acts on the trajectories existing in a configuration and a deterministic approach of setting the labels according to a predefined rule.
The split/merge perturbation kernel is in tone with the general framework presented in this work. It provides a method to modify the labels of the objects of the simulated chain. Deterministic labeling on the other hand depends on the order of the exploration as shown in Section 5.3. Nevertheless, a deterministic labeling approach significantly reduces the computational burden that is linked to performing split/merge moves.
Experimental results have shown that the performance of the two labeling approaches is comparable in terms of quality. In Table 5.4 (a) we show examples when the deterministic labeling outperforms the split/merge approach, while in Table 5.4 (b) we show the reverse.
The subsequent results showed in this section have been obtained using the Quality model and a deterministic labeling approach, unless otherwise stated.

5.4.2 Tracking results on real biological data

The main objective however, is to apply the proposed algorithm on real data. Thus, the algorithm has been applied on a real biological sequence of 300 images of 350 by 1340 pixels, by courtesy of J. Salamero, PICT IBiSA, UMR 144 CNRS Institut Curie ([Basset et al., 2014]). The TIRF image sequence shows a cell, with the corresponding vesicles that transport substances inside it. One frame consists of two images of the cell taken at the same time instance but at different wavelengths. The vesicles can appear or disappear at any location and time instance since a frame contains only a 2D representation of the actual 3D scene. The main objective is to detect and track these vesicles in order to understand how they move within the cell. The SNR for this data set is very low. Furthermore, the intensity of the radiometric

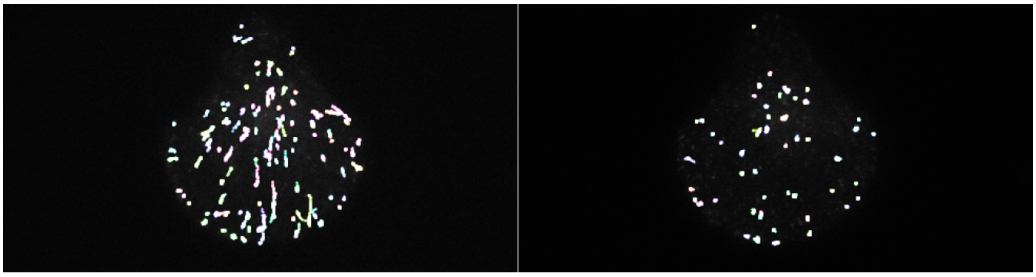


Figure 5.12: Detection and tracking results on a real biological TIRF sequence of 300 images (by courtesy of J. Salamero, PICT IBiSA, UMR 144 CNRS Institut Curie ([Basset et al., 2014])) (©INRIA). The image sequence shows a cell, with the corresponding vesicles that transport substances inside it. The goal is to detect and track these vesicles. The visual assessment of the results reveals their very good quality.

Data set	No. reference tracks (MHT)	No. candidate tracks (Proposed algorithm)	Similarity between tracks	Similarity between detections
Real biological sequence	512	346	40.4%	24.5%

Table 5.5: Comparison between the results obtained using the built-in MHT tracker within Icy ([de Chaumont et al., 2012]) and the proposed method for the real TIRF image sequence ([Basset et al., 2014]). Note however, that the output of the MHT tracker should not be taken as ground truth information. The visual assessment of the results reveals that the tracks obtained using the proposed algorithm are more consistent and less fragmented than those obtained using the MHT plug-in.

value of the vesicles fades in time due to the data acquisition process.
The detection and tracking results are depicted in Figure 5.12. The visual assessment of the results is very good. Long, smooth tracks are identified which correspond to the linear motion of the vesicles. This result is in agreement with current research in cell biology w.r.t. the motion of vesicles within a cell. The proposed approach is again compared to the built-in particle tracker of the Icy software. However, in the absence of ground truth information, only the detection and tracking similarity of this model w.r.t. the MHT tracker is shown in Table 5.5.

5.4.3 Tracking results on simulated low temporal frequency satellite data

The acquisition rate of satellite images has experienced a significant increase in the last years. Therefore, object tracking using high resolution satellite images can be regarded as a new application in remote sensing, complementary to object

detection and land-cover classification. Hence, the proposed approach is tested on challenging real high resolution optical satellite image sequences. The frame rate of these sequences is around $1 - 5$Hz.

Metrics. Conducting an objective comparison between different tracking algorithms is a challenging task for various reasons. First, the importance of individual tracking failures is application dependent. Second, classifying tracker outputs as correct or incorrect may as well be very ambiguous and usually requires additional parameters (e.g. thresholds) to assess the correctness and precision of the trackers.

To evaluate the multi-object tracking accuracy, we compute three types of errors: false positives (FP), false negatives (FN) and identity switches (ID). The three types of errors are weighted equally. The number of true positives (TP) is also stated and the total number of moving objects (TO) is provided. The total number of moving objects (TO) is the sum over all frames of the objects that change their position in two consecutive frames. Additionally, mostly tracked (MT) and mostly lost (ML) scores are computed on the entire number of distinct trajectories (TT) to measure how many ground truth trajectories are tracked successfully (tracked for at least 80 percent) or lost (tracked for less than 20 percent). Finally, the precision (TP / (TP + FP)) and recall (TP / (TP + FN)) of each algorithm is stated.

The results are compared to two classical trackers: Kalman filter and smoother ([Welch and Bishop, 2001]) and Histogram-Based Tracker ([Dalal and Triggs, 2005]).

5.4.3.1 Melbourne sequences

Two high resolution optical satellite image sequences are considered for evaluation. Each sequence consists of 14 frames taken at a low temporal frequency. The sensor is moving at a relatively slow speed. The acquisition angle of two consecutive images changes significantly. Figure 5.14 shows the displacement between two frames.

Object appearance. Objects exhibit strong variations in appearance due to the changing angle at which the images were taken. Table 5.6 (top) depicts the appearance variation of one boat over the data set. However, the proposed method correctly identifies the target even under these conditions, as shown in Table 5.6 (middle). This is mainly due to the use of a contrast distance measure, without taking into account additional information within the ellipse. This is the main reason why the histogram-based tracker fails to identify the boat as the same object across time, as shown in Table 5.6 (bottom).

Quantitative assessment. Figure 5.13 shows the results of our method on the two image sequences considered, while the quantitative results for each image sequence are presented in Table 5.7. The results of three trackers are shown: the full model including dynamic birth maps and the water mask used for optimization presented in this chapter, denoted (ST-MPP + BM), the Kalman filter and smoother (KFS) and the histogram-based tracker (HBT). For comparison, the detection results of the spatial marked point processes model developed in chapter 4 and denoted (MPP) which was applied independently on each frame to extract boats is also listed in Table 5.7.

The first image sequence has 14 frames of 1840 × 820 pixels each and contains a

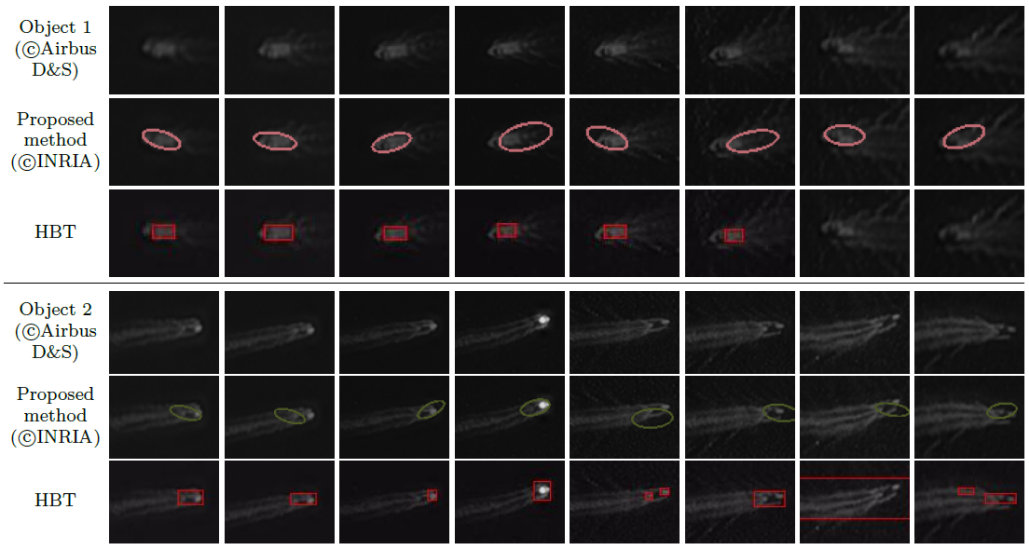


Table 5.6: Two hard boat detection and tracking cases. Object 1 almost blends with the background by the end of the sequence and is hard to distinguish from noise (waves). The proposed method successfully detects the object, due to the joint optimization over detections and tracks, while all other trackers fail to distinguish the object from the background noise. Object 2 exhibits strong appearance changes due to its increase in speed. The tail of this object is mistaken for a new object by all trackers except the proposed method.

total number of 8 moving objects throughout the entire sequence. The use of the object evidence term leads to better estimates of the locations of the objects, while the contrast distance measure is used to obtain accurate values for the size and orientation of the objects. Moreover, the labels are preserved throughout the image sequence.
The second image sequence has 14 frames of 830 × 730 pixels each. The sequence contains a total number of 4 moving objects as well as a large number of static objects of the same type. The evidence term plays a decisive role in distinguishing dynamic objects from static ones. The proposed model yields a higher tracking performance compared to the classical trackers, due to the better detection results. The labels of the objects are generally preserved throughout the sequence.
The proposed method (ST-MPP + BM) outperforms both the Kalman filter and smoother (KFS) and the histogram-based tracker (HBT). The lower performance of the Kalman filter is described by the lower performance of the detector used which produces a high number of false alarms due to background noise (waves). The performance of the histogram-based tracker is highly influenced by the change in illumination due to the different angles of acquisition of the satellite images. The appearance of the objects changes throughout the sequence and thus, the precision

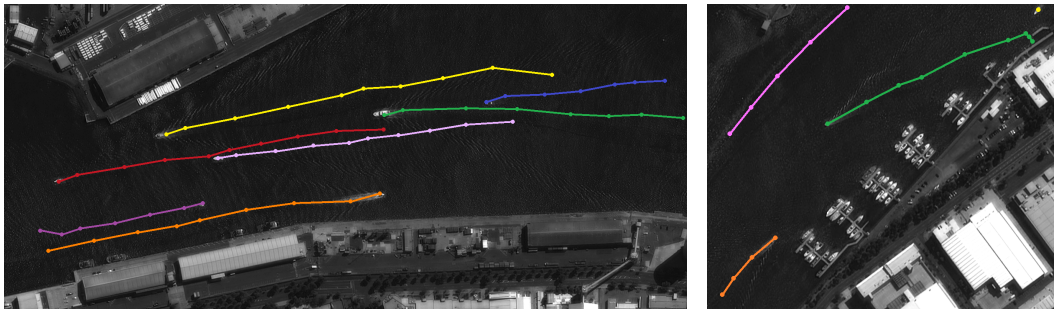


Figure 5.13: Detection and tracking results on two sequences of real satellite images taken at different angles (©INRIA). Each color represents a different track. Left: Tracking results on the first image sequence up to frame 10. Right: Tracking results up to frame 13 of the second image sequence.

Data set	Method	FP	FN	TP	TO	ID	MT	ML	TT	Precision	Recall
Seq. 1	ST-MPP + BM	**1**	**6**	**85**	91	0	7	1	8	**98.8%**	93.4%
	KFS	3	34	57		0	4	2		95.0%	62.6%
	HBT	5	14	77		3	6	2		93.9%	84.6%
	MPP	7	5	84		–	–	–		92.3%	**94.4%**
Seq. 2	ST-MPP + BM	**1**	**1**	**24**	25	0	4	0	4	**96.2%**	**96.2%**
	KFS	2	4	21		0	2	1		91.3%	84.0%
	HBT	2	3	22		1	2	0		91.6%	88.0%
	MPP	3	1	**24**		–	–	–		88.9%	96.1%

Table 5.7: Quantitative results for the two sequences of real satellite images. The proposed method has the highest precision and the second highest recall scores for the first sequence, as well as the highest precision and recall scores for the second sequence. The proposed methods succeeds in tracking more targets in each sequence than the other methods (MT) and also has the best detection rates for both sequences (TP).

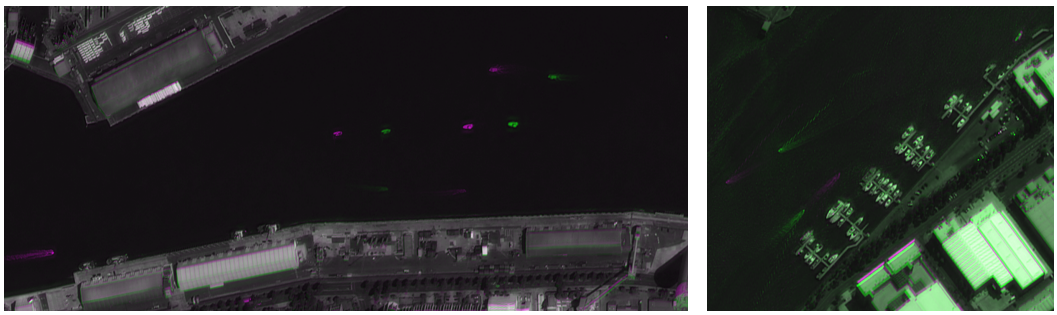


Figure 5.14: Displacement between two consecutive frames. The objects exhibit large jumps from one frame to the next. Distortions are also visible on the land area.

of the tracker is affected. In terms of appearance, the proposed tracker however only relies on the contrast between the objects and their border. Therefore, its performance is not affected by appearance changes. The spatio-temporal model is more accurate than a per-frame detection using a spatial model (MPP), since the temporal model explicitly uses temporal information to increase detection results. The increased quality of detections leads in turn to better-formed trajectories.

5.4.3.2 Barcelona sequence - low temporal frequency

The results on one high resolution sequence taken over the port area of Barcelona is analyzed. The sequence contains 420 frames, each 1600×900 pixels in size.
Large geometric distortions appear due to the movement of the imaging sensor. The object evidence presented in section 5.1.2 is based on frame differencing. Such a technique can be efficiently used when the camera is static or near-static. The performance of background modeling approaches drops with the increase in camera motion. Hence, such an approach is not suited in this case. The frame differencing approach described in 5.1.2 is replaced by a more sophisticated motion detection algorithm developed at Airbus D&S which takes the geometric distortions into account. The output of this algorithm is fed also to the Kalman filter and the Histogram based tracker used to evaluate the results. Apart from the geometric distortions which are depicted in Figure 5.15 (a), the sequence also exhibits a high level of noise. The noise is mainly due to the water waves which lead to a very high false alarm rate of the motion detection algorithm.
Object appearance. In this sequence, the interest lies in detecting and tracking all objects that exhibit a consistent motion pattern throughout the sequence. Objects are characterized by a significant contrast with the background. Moving objects include cars, buses, boats and pedestrians. However, no distinction is made between the different object types. A post-processing step can be used to distinguish between the different object classes based on their size and velocity.
Qualitative evaluation. The motion detection algorithm used in the evidence term of the model has a high false alarm rate. This can be seen in Figure 5.15 (b). Each circle is a detection and the color of the circle represents the probability of the detection being a true target. A yellow circle shows a small probability, whereas a red circle shows a high probability of being a target. The size of the circle relates to the size of the object detected. The majority of the false alarms are produced by waves in the sea. This is a typical problem when dealing with a turbulent sea within a sequence of images. Figure 5.15 (c) shows the flow of the objects during the entire sequence. This Figure is particularly important, since it visually demonstrates that the performance of the proposed model is not affected by the high number of false alarms obtained while pre-detecting the targets. Figure 5.15 (d), shows the tracks obtained in the current frame using the proposed method. Both KFS and HBT are strongly affected by the detection results, yielding a large number of spurious tracks on water, as can be seen in Figure 5.15 (e) and Figure 5.15 (f).
Although the lack of ground truth data does not allow an in-depth, quantitative

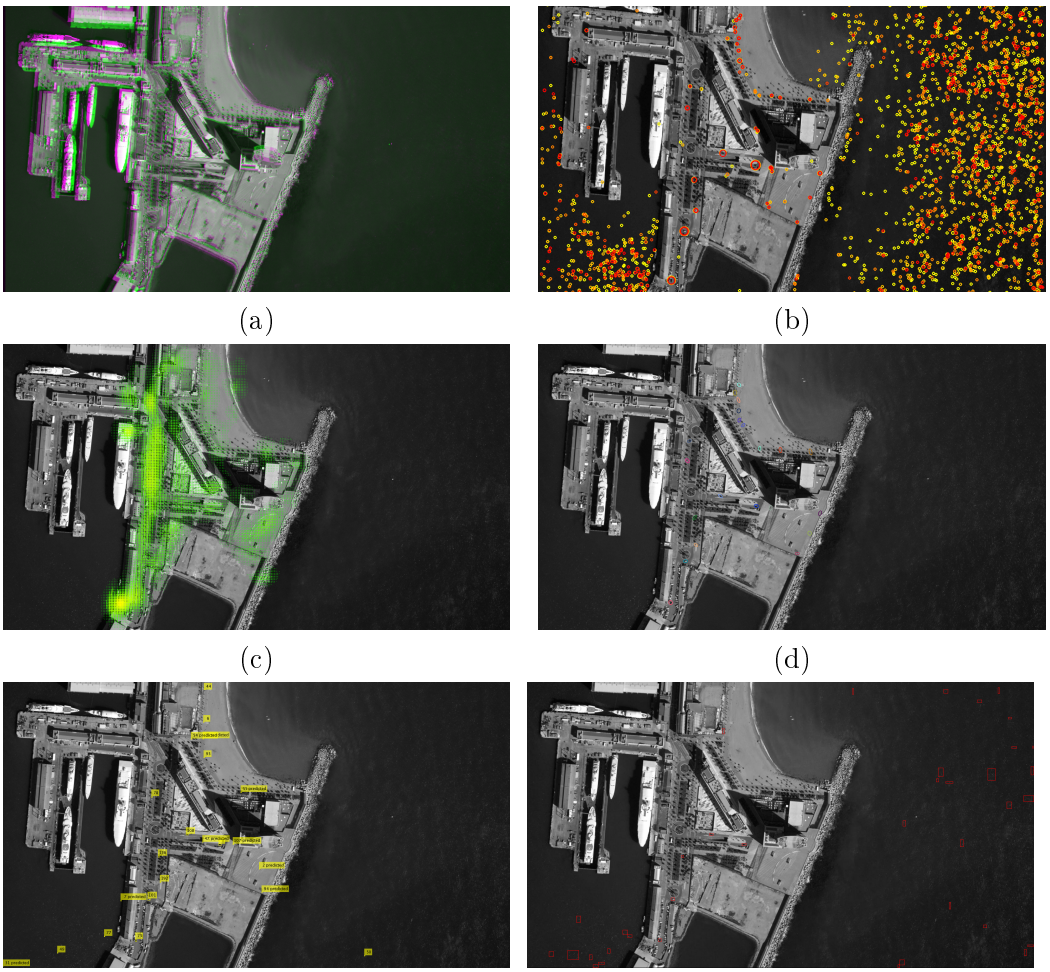


Figure 5.15: (a) Image displacement between two images 20 frames apart in the Barcelona sequence at a 3Hz temporal resolution; (b) Detection results of possible moving targets obtained using the motion detector provided by Airbus D&S. The detector can handle geometric distortions and camera movements. The circles represent possible detections with various probabilities (yellow = low, red = high); (c) The motion flow in the entire sequence. Lighter areas highlight locations with intense motion throughout the sequence. The proposed model mainly identifies the objects which are on land with a few false alarms around the coast. However, by taking into account the temporal information, the high number of false alarms generated by the detection pre-processing step is considerably reduced; (d) Targets detected at frame 170 of the sequence using the proposed method. The distinct colors of the objects represent different tracks; (e) Targets detected at frame 170 using the extended Kalman filter and smoother. Each object is assigned a number to represent different tracks; (f) Targets detected at frame 170 using the histogram-based tracker. All objects have the same color. Labeling is not shown on this image.

analysis, the proposed method outperforms visually the KFS and HBT. The proposed algorithm tracks both cars and pedestrians. As stated in the beginning of this section, no distinction is made between different classes of objects is implicitly made within the model. Nevertheless, a post-processing step based on the size and speed of the objects can be used to distinguish between object classes.

5.4.4 Tracking results on simulated high temporal frequency satellite data

Video quality can also be obtained by high resolution optical satellite sensors for a short period of time. In this section, several videos are analyzed when the imaging sensor is both static and moving. The frame rate is around 30Hz for each sequence.

5.4.4.1 Toulon sequences

Two sequences are acquired by a static camera over the Toulon harbor area. Each sequence contains 150 frames. Given the fact that the camera is static, the object evidence is determined based on frame differencing, as shown in section 5.1.2.
The objects of interest are moving boats. The boats have different colors, shapes and can be partially occluded or blended with the background.
Qualitative and quantitative evaluation. The detection and tracking results are shown in Table 5.8. The proposed algorithm consistently tracks the moving boats over the entire sequence considered. A single object is missed by the algorithm in the second sequence. This object moves very slowly and thus, the frame differencing approach does not identify any movement where that boat is located. This can be also observed in the displacement image shown in Figure 5.17. Hence, the evidence term has a very low value for this particular object. As it is never detected as a possible candidate object, the motion model does not help in increasing the detection probability. Therefore, this object is consistently missed.
Both the KFS and HBT also miss this object in sequence 2. Furthermore, these trackers have difficulties identifying the moving object in sequence 1 as a consistent target, as it is constantly partially occluded by the surrounding objects. The similarity of the moving target with its surroundings leads to an unreliable localization of the target by the HBT. This similarity also affects the proposed method which is why the resulted trajectory is rather unsteady. Nevertheless, the proposed algorithm is less sensitive to it than the HBT.

5.4.4.2 Barcelona sequences - high temporal frequency

Three videos captured by a moving camera over the Barcelona area are analyzed. These sequences also suffer from large geometric distortions. However, given the high temporal frequency of the videos, the geometric distortions are relatively small within a narrow batch of frames. The displacement between two images 20 frames apart in a Formula 1 sequence is depicted in Figure 5.16 (a). The displacement of the background areas is almost non-existent. Thus, the evidence term still relies on

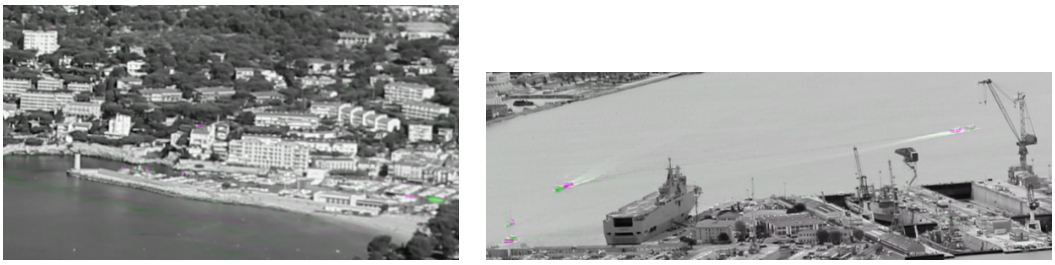


Figure 5.17: Image displacement between two images 20 frames apart in the Toulon sequence at a 30Hz temporal resolution. The only visible displacement is in the motion of the boats. Note however, that the displacement of the undetected boat is not visible even after 20 frames.

Data set	Method	FP	FN	TP	TO	ID	MT	ML	TT	Precision	Recall
Seq. 1	ST-MPP + BM	**0**	**0**	**150**		0	1	0		**100.0**%	**100.0**%
	KFS	5	6	109	150	4	1	0	1	95.6%	94.8%
	HBT	12	25	110		11	1	0		90.2%	81.4%
Seq. 2	ST-MPP + BM	**0**	**150**	**450**		0	3	1		**100.0**%	**75.0**%
	KFS	12	180	408	600	6	3	1	4	97.1%	69.3%
	HBT	23	201	376		16	3	1		94.2%	65.2%

Table 5.8: Quantitative results for the two sequences of satellite sequences of high temporal frequency (30Hz). The proposed method has excellent precision and better recall than the Kalman filter and the Histogram based tracker. Note that all trackers miss one boat (image on the right). Since the velocity of this boat is very small, frame differencing fails to identify the object as foreground. Thus, the object is never detected throughout the sequence.

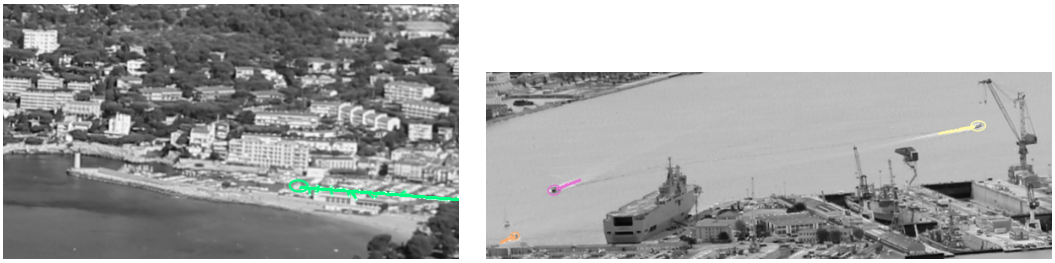


Figure 5.18: Detection and tracking results on two sequences of simulated satellite images of Toulon. The proposed method consistently misses one boat (image on the right) because of its very small velocity. The image sequences are by courtesy of Airbus Defense & Space, France.

frame differencing to detect possible objects.

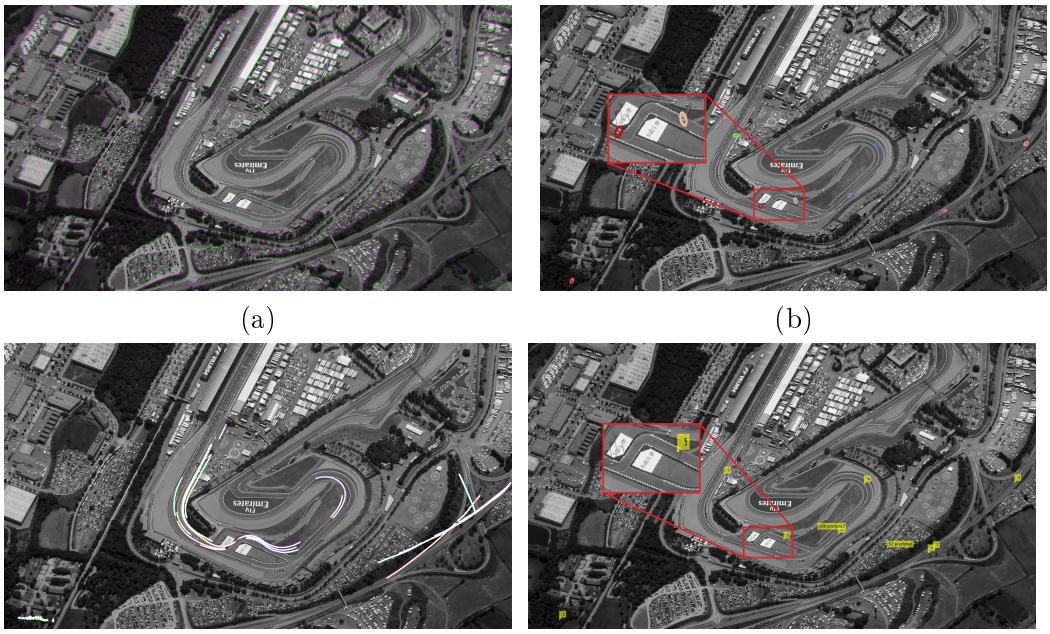


Figure 5.16: (a) Displacement between two images 20 frames apart in a sequence with high temporal frequency. The image sequence is by courtesy of Airbus Defense & Space, France. (b) Detection and tracking results for the frame 94. (c) Traffic density after 500 frames. (d) Tracking results obtained by KFS in frame 94.

Qualitative evaluation. The detection and tracking results are very good. On the one hand, the proposed method is very robust to noise and outperforms the KFS and HBT in this regard. Figure 5.16 (b) shows the tracking results using the proposed method, while Figure 5.16 (d) shows the tracking results using KFS. A large number of spurious tracks are wrongly detected by the KFS. On the other hand, a general tendency of the proposed model to miss objects that do no exhibit a consistent pattern is observed. The automatic parameter learning technique proposed in this chapter is based on available ground truth information. As the parameters are learned, they are optimized to fit an almost zero false alarm scenario. Thus, the model is more likely to dismiss objects that are not good fits, leading to a better precision but at the expense of recall rates. An alternative solution to increase the detection probability is the use of the Statistical model. This model can be easily modified to provide the desired detection/false alarms rates.
Qualitative evaluation using the Statistical model vs. the Quality model. The main advantage of the Statistical model over the Quality model is the ability to set a desired probability of detection/false alarms. The parameter learning technique remains the same and thus, the parameters obtained fit an almost zero false alarms

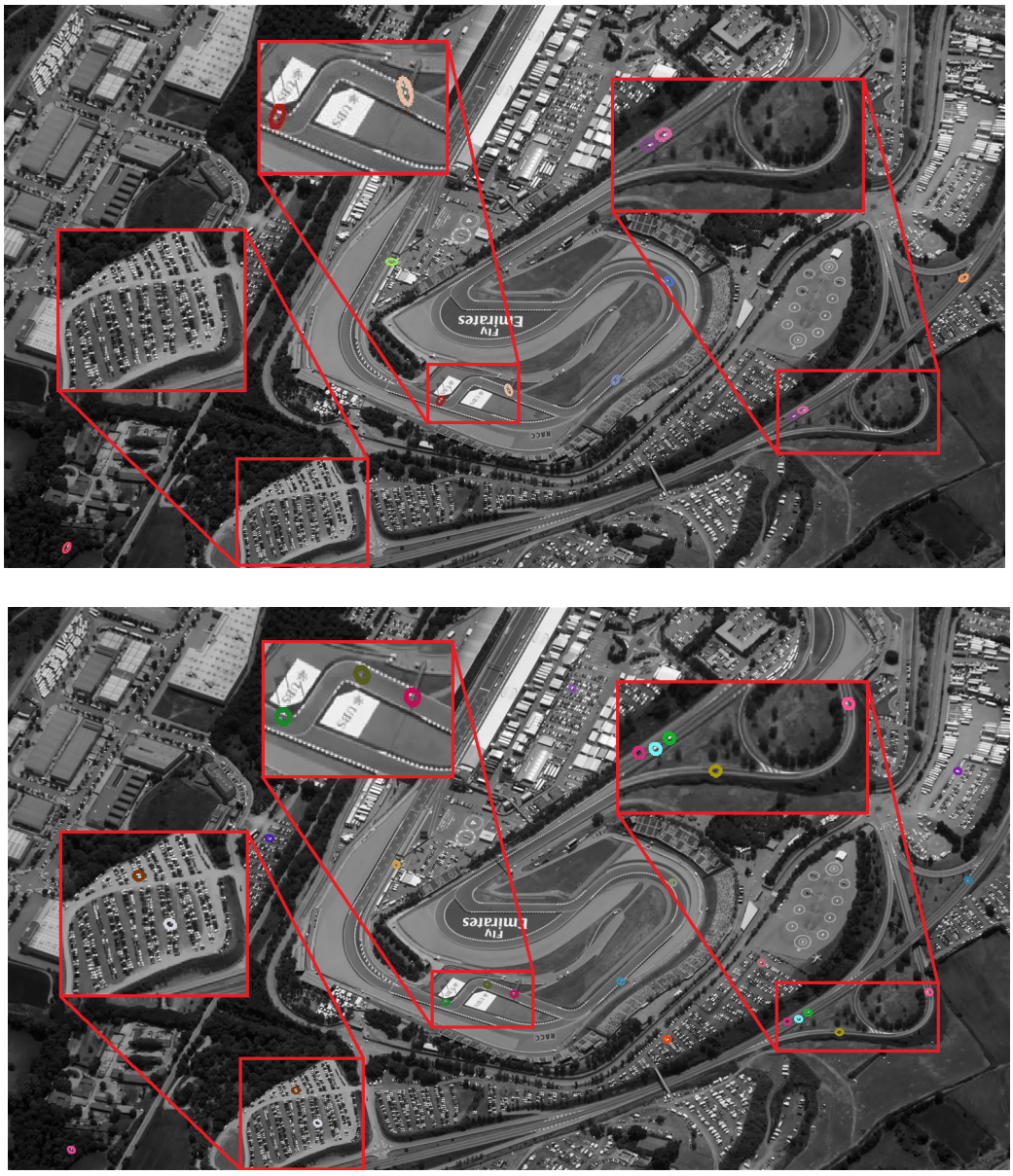


Figure 5.19: Detection and tracking results for the frame 94 of the Formula 1 sequence. Top: Results obtained using the Quality model. The model offers a good detection rate with zero false alarms. Bottom: Results obtained using the Statistical model. The model offers a better detection rate at the cost of an increased false alarms rate. The image sequences are by courtesy of Airbus Defense & Space, France.

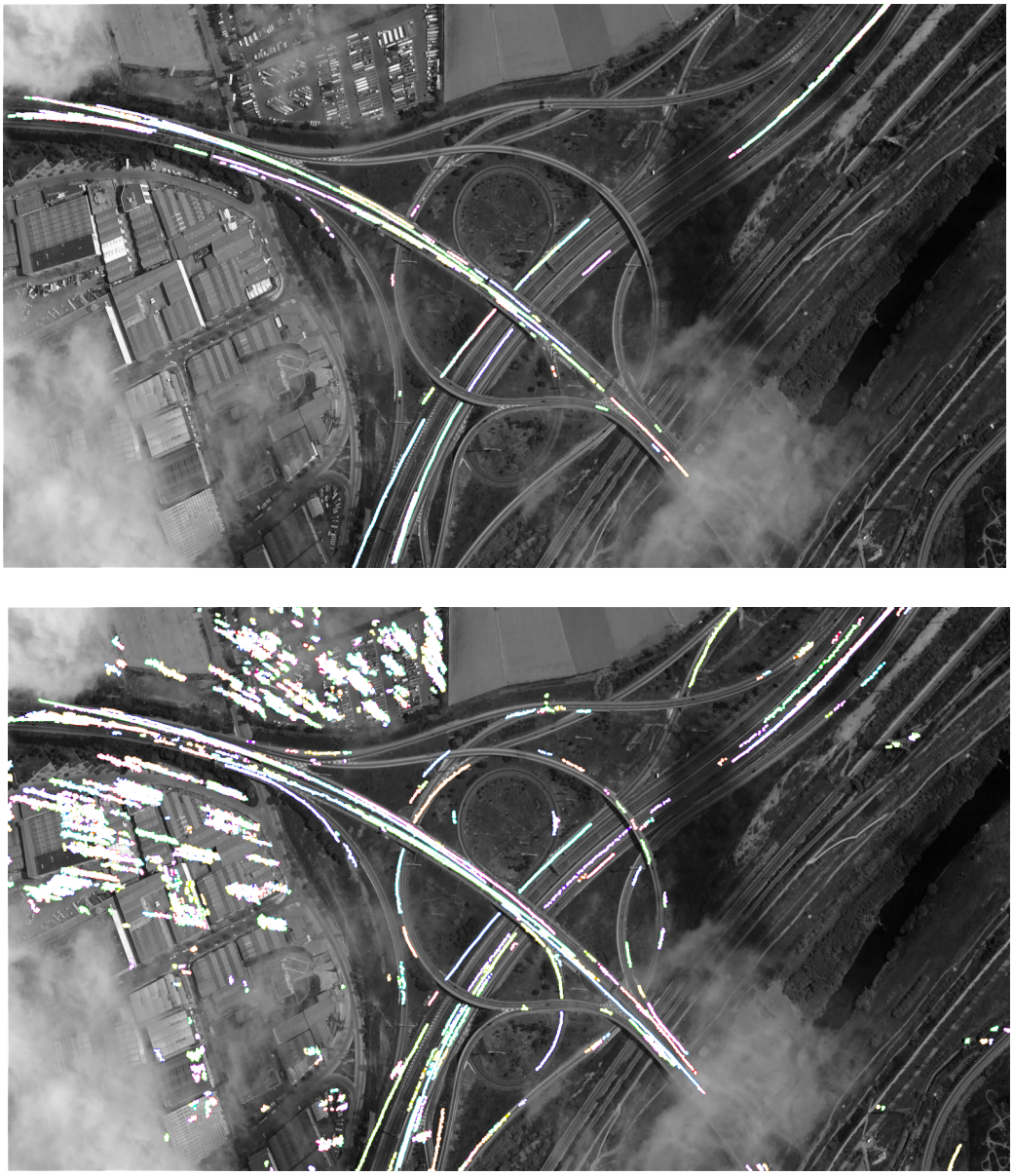


Figure 5.20: Detection and tracking results on a sequence of simulated satellite images of Barcelona. The traffic density after 500 can be observed. Top: The traffic density obtained using the Quality model. Bottom: The traffic density obtained using the Statistical model with a larger number of detected targets at the cost of a high false alarm rate. The image sequence is by courtesy of Airbus Defense & Space, France.

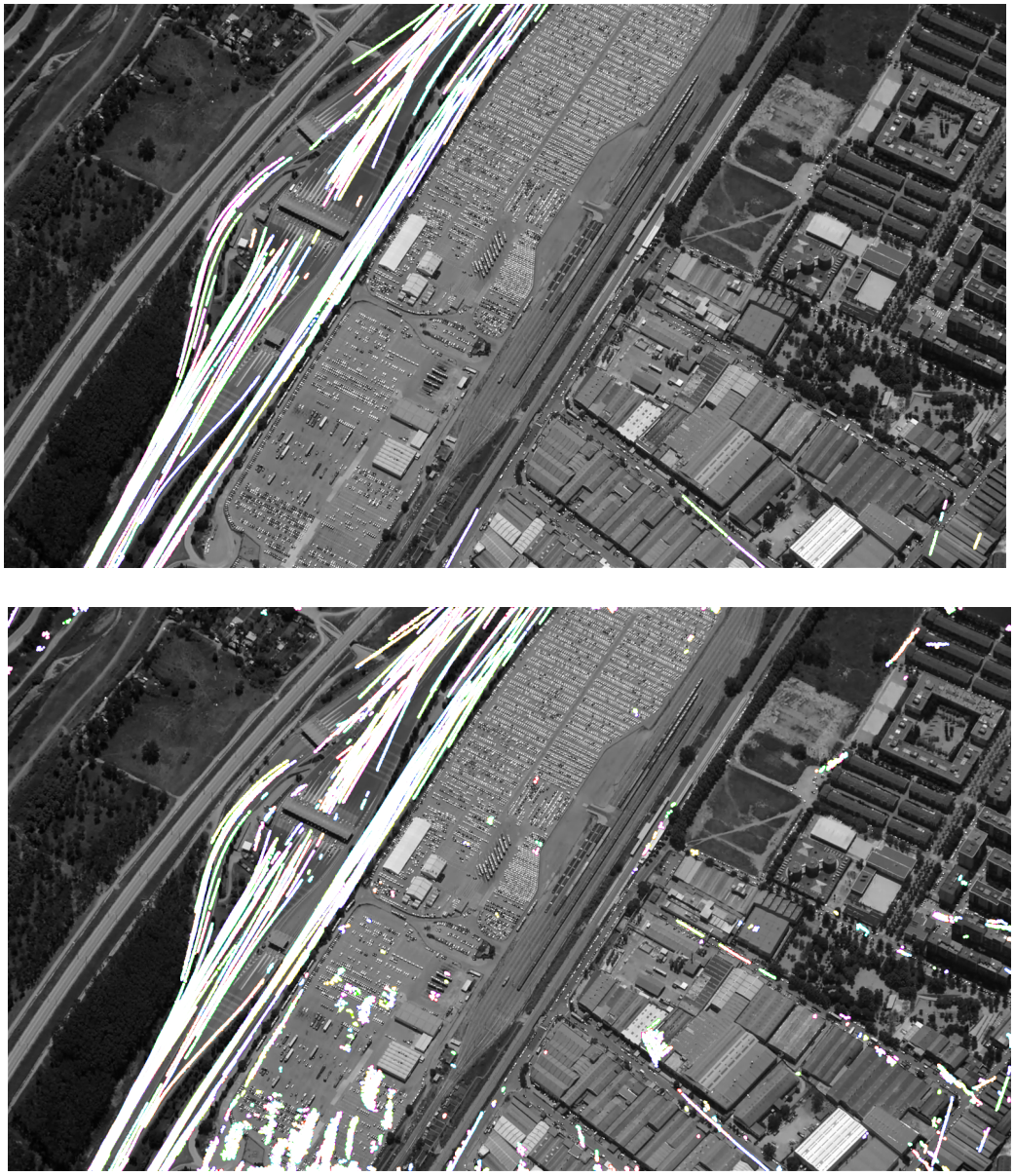


Figure 5.21: Detection and tracking results on a sequence of simulated satellite images of Barcelona. The traffic density after 500 can be observed. Top: The traffic density obtained using the Quality model. Bottom: The traffic density obtained using the Statistical model with a larger number of detected targets at the cost of a high false alarm rate. The image sequence is by courtesy of Airbus Defense & Space, France.

scenario. Nevertheless, in this case, the choice of the threshold τ of the matrix $\mathbf{M}_{GLRT}$ can be modified after parameter learning to ensure a better detection probability. Figure 5.19 shows the results obtained on the same Formula 1 sequence, with a higher probability of false alarms. Objects previously undetected using the Quality model are successfully detected using the Statistical model. However, the increased detection probability comes at the cost of higher false alarms rate. Most of these false alarms appear in the parking lot, due to the reflection of the sunlight which produces a considerable amount of motion (difference) between consecutive frames.
Figure 5.20 and Figure 5.21 show the traffic density after 500 frames on two image sequences. The tracker assigns consistent labels throughout the sequence. ID changes are more frequent in dense environments where objects tend to be very close to each other. If we use the Statistical model we can set a higher false alarm rate to detect more targets as shown in the bottom images of Figure 5.20 and Figure 5.21. A significant amount of false alarms appear in the parking lots and at the edges of the buildings, as these areas are responsible for a large intensity changes within the pixels in consecutive frames.

5.4.5 Computational efficiency

The computational efficiency of the proposed framework for multiple object detection and tracking depends on three factors:

- **The number of objects**. The proposed model depends on the number of objects in the scene, or more precisely on their density in a given area. The denser the objects are in an area, the more interactions between the objects. Thus, whenever an ellipse is created or locally perturbed, the interactions with a large number of neighbors have to be computed. This in turn increases the computational effort during a single iteration;

- **The size of a frame**. The larger the size of the frame, the bigger the sampling space and thus, the more iterations are needed;

- **The length of the sequence**. As in the previous case, the longer the sequences, the more frames have to be processed and thus, the longer it takes until the entire sequence is analyzed.

First, the computational efficiency of the hybrid implementation of the RJMCMC sampler with Kalman-like moves is discussed and compared to the RJMCMC sampler in 2D + T as presented in section 5.3.1. Then, the performance of the multiple cores implementation of RJMCMC is presented and the speed-up is analyzed.

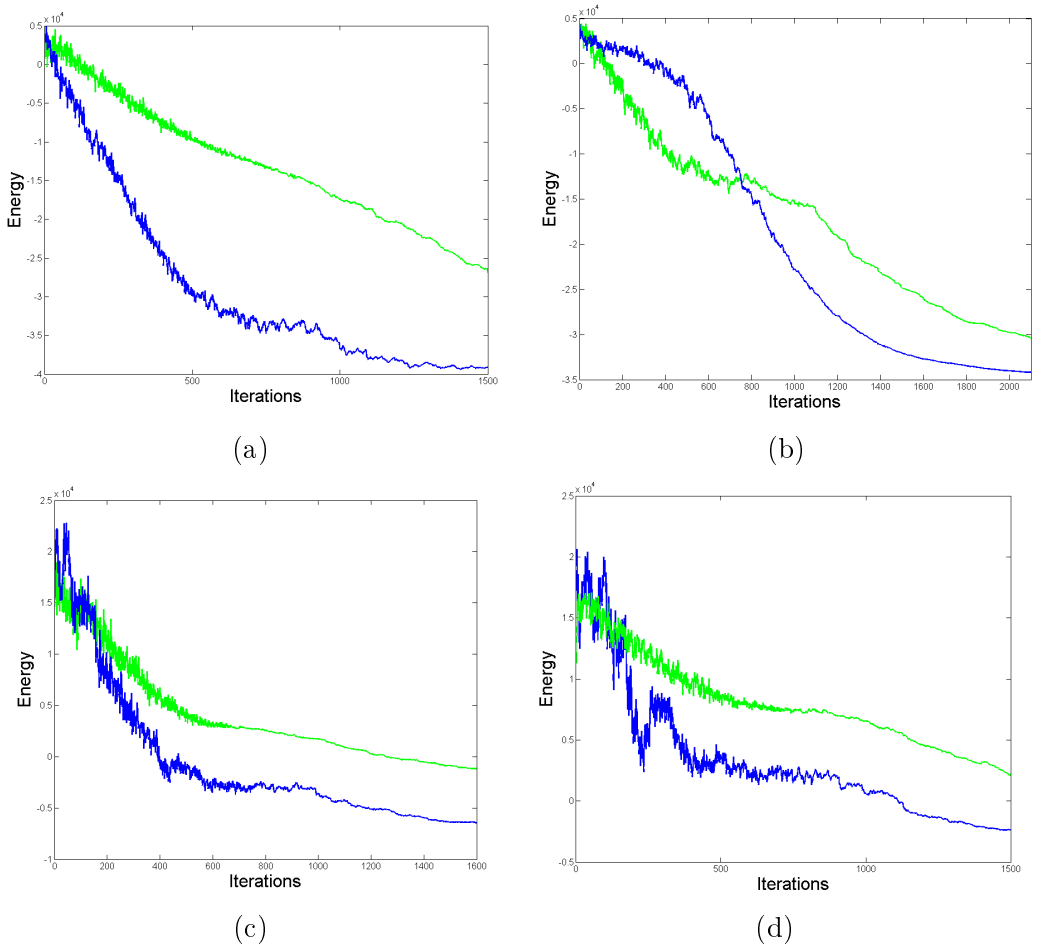


Figure 5.22: Energy evolution with the number of iterations for the standard RJMCMC sampler (green) and the RJMCMC with Kalman-like moves (blue). The efficiency of the standard RJMCMC sampler is significantly increased by incorporating the properties of sequential filters into the birth and death perturbation kernel. (a)-(b) Results on two benchmark biological sequences; (c)-(d) Results on the two high resolution satellite image sequences with high temporal frequency from the Toulon data set.

5.4.5.1 RJMCMC using Kalman like moves

The birth and death kernel using Kalman-like moves efficiently generates several birth proposals in a single step. This leads to a high birth rate especially at the beginning of the simulated annealing procedure. The high temperature at the beginning of the annealing process increases the acceptance probability of otherwise unlikely perturbations. Figure 5.22 depicts the energy evolution during the optimization process when using the Kalman-like moves as opposed to the standard birth and death kernels. Figure 5.22 (a) and (b) show the energy variation for two benchmark biological sequences, while Figure 5.22 (c) and (d) show the energy vari-

ation for the two sequences in the Toulon data set.
The energy values at the beginning of the optimization are slightly higher than compared to the standard RJMCMC. The reason for this initial increase lies in the high number of births that are initially accepted. The Kalman filter is used to increase the overall acceptance ratio of the sampler by proposing more meaningful perturbations. However, as the temperature in the beginning is very high and the distribution function is similar to a uniform distribution, a large number of perturbations is accepted. As the temperature decreases, the sampler quickly removes badly fitted objects which is experienced as a rapid drop in the energy variation. Finally, towards the end of the optimization process, mostly non-jumping transformations are performed.
Intuitively, the birth and death kernel using Kalman-like moves offers a better and faster global exploration of the object space at the beginning of the optimization, compared to standard RJMCMC. As the temperature decreases, the sampler settles in a strong local minimum. Towards the end of the optimization, the non-jumping transformations can be seen as local adjustments.
The state vector of the Kalman filter contains not only the location and velocity of the objects, but also information about the size and orientation of the ellipse. Consequently, the Kalman filter generates proposals of ellipses that are similar in size and orientation and hence, a smaller number of non-jumping transformations are needed to fit the data. By combining the ability to generate multiple birth proposals in a single iteration and more similar ellipses within a single trajectory, the hybrid implementation of RJMCMC generally results in a smaller number of iterations needed until convergence and thus, an increased computational efficiency. It is important to notice that the term "iteration" does not directly relate to the computation time. Consider for instance the birth of an object in the first time frame u_{t_1}. In the standard RJMCMC setting, a birth iteration represents the time needed to propose a new configuration $\mathbf{X} \cup u_{t_1}$ and compute the acceptance ratio of this perturbation. In the hybrid sampler setting, the amount of time needed to perform a birth iteration using Kalman-like moves can take any value from the time needed to propose and compute the acceptance ratio of $\mathbf{X} \cup u_{t_1}$ to the time needed to propose and compute the acceptance ratios of $\mathbf{X} \cup u_{t_1} \cup u_{t_2} \cdots \cup u_{t_T}$. The latter case is reached when the initial proposal $\mathbf{X} \cup u_{t_1}$ is accepted and then all subsequent perturbations proposed using the Kalman filter are accepted, except for u_{t_T} for which the acceptance is not mandatory. Hence, it is important to analyze both the number of iterations needed until convergence (Figure 5.22), but also the computation time of the two samplers (Table 5.9).
A different approach to increasing the efficiency of the sampler is a parallel implementation of the sampling procedure.

5.4.5.2 RJMCMC 2D + T sampler - multiple cores

Embedding the RJMCMC sampler into a simulated annealing framework ensures an optimal accuracy and at the same time, a high computational time cost even if

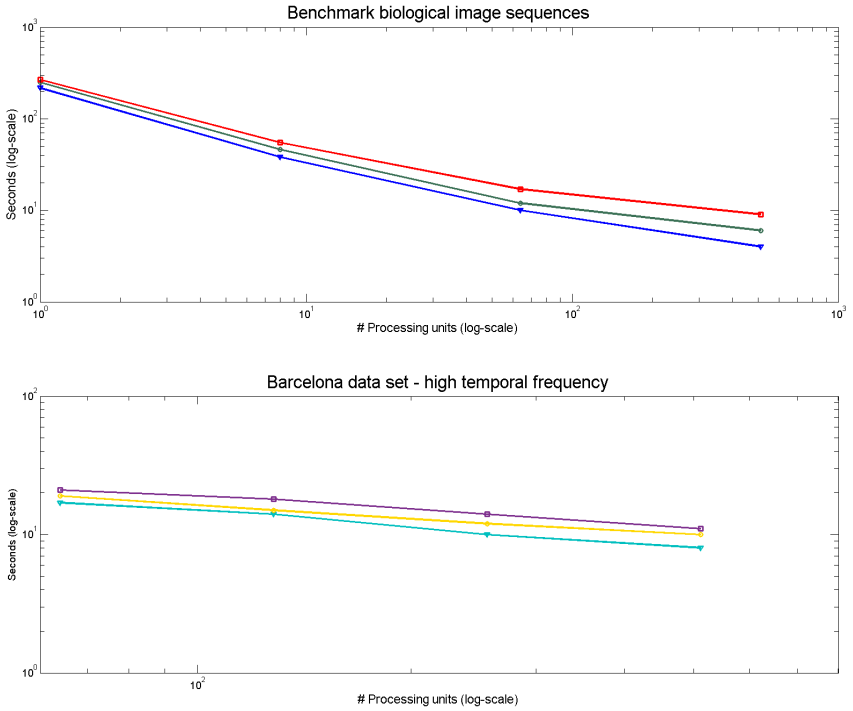


Figure 5.23: Increase in computational efficiency of the proposed multiple object tracker w.r.t. the number of cores (CPU's): (left) for benchmark biological image sequences; (right) for the Barcelona data set with high temporal frequency.

# CPU's	Synthetic biological data		
	Seq. 1	Seq. 2	Seq. 3
1 (std)	3.62min / frame	4.18min / frame	4.47min / frame
1 (hybrid)	3.11min / frame	3.42min / frame	3.61min / frame
8	38sec / frame	46sec / frame	55sec / frame
64	10sec / frame	12sec / frame	17sec / frame
512	4sec / frame	6sec / frame	9sec / frame
# CPU's	Barcelona data set - high temporal frequency		
	Formula 1	Road Junction	Toll Station
64	17sec / frame	21sec / frame	19sec / frame
128	14sec / frame	18sec / frame	15sec / frame
256	10sec / frame	14sec / frame	12sec / frame
512	8sec / frame	11sec / frame	10sec / frame

Table 5.9: Computation times of the proposed multiple target tracker w.r.t. the number of cores (CPU's). The obtained results reveal that the proposed method scales very well with a large number of cores.

the properties of sequential filters are incorporated into the sampler. This can be observed in Table 5.9 in the case of a single processor. However, the readily available computation power (either in the form of computer clusters or clouds) and a good distributed design can significantly decrease the computational burden.
The marked point process models described in this chapter feature an energy term that is computed over the entire sequence, namely the label persistence described in eq. 5.7. Thus, in a distributed setting, special care needs to be taken to ensure that new labels are unique throughout the sequence. To do this, the frame number and the identity of the computing unit is used as a prefix for a newly assigned label. This ensures that new labels are unique and do not appear twice in the same frame which would violate the mutual exclusion constraint in Section 5.1.1. In the case of deterministic labeling, the last iteration is performed in a sequential way to ensure a consistent labeling throughout the sequence.
The parallel implementation of the sampler proposed in this chapter has been tested on up to 512 cores on a computer cluster. The computation times shown in Table 5.9 and depicted in logarithmic scale in Figure 5.23 reveal a drastic increase in the computational efficiency w.r.t. the number of cores used. These results confirm that the proposed implementation scales very well with the number of cores. The scalability of the proposed method depends on the size of the image sequence, on the size of the frame within the sequence and the number of objects present on the scene. Thus, long sequences with large frames and a small number of objects are the best candidates for the parallel sampling procedure and exhibit the highest speed-ups w.r.t. the number of cores.

5.4.5.3 Influence of the perturbation kernels on the convergence speed

The computation times shown in Figure 5.23 are obtained using a combination of specifically designed perturbation kernels to increase the efficiency of the sampling. It is interesting to evaluate the influence of the proposed perturbation kernels on the convergence speed. In this regard, we consider a total of 50 frames from a synthetic biological sequence characterized by a large number of targets and a high temporal frequency (Seq.1). We use the quality of the reached energy to evaluate and compare the performance of RJMCMC samplers with different perturbation kernels. The combinations of perturbation kernels are as follows:

- **BDM: Birth and death according to a birth map**. This kernel relies on the pre-computation of birth maps for each frame of a sequence. The birth maps can be seen as a tool that changes the underlying homogenous Poisson measure of the point process into an inhomogeneous Poisson measure based on some desired properties. For instance, we have computed the birth map for frame t as a linear combination of the absolute difference between the frame $t-1$ and t and the radiometric values of the pixels in frame t. The frame differencing is used to identify locations with large changes from one frame to the next which is used as an indicator for the presence of a moving object. The radiometric values within the frame are used to identify possible locations of

objects that do not exhibit a clear motion between frames. Hence, the kernel favors interesting locations for the creating/deleting objects;

- **BDN: Birth and death in a neighborhood coupled with BDM**. This kernel proposes to create/remove objects from the current configuration in the neighborhood of an existing object. This approach is motivated by the assumption that objects tend to cluster together and hence, it is more likely for objects to appear close to other objects than to appear totally isolated;
- **BDML: BDM coupled with local perturbations**. As a reminder, examples of local perturbations are rotation, translation and scale;
- **BDNL: BDN coupled with local perturbations**. This sampler has been used in Section 5.4.5.1;
- **BDKF: Birth and death using Kalman-like moves**. This hybrid kernel relies on a Kalman filter for the proposal of new objects. This approach is motivated by the proven effectiveness and efficiency of the Kalman filter for object tracking.
- **BDKFL: BDKF coupled with local perturbations**. This sampler is used in our experiments in Section 5.4.5.1.

An indicator of the mixing properties of the sampler is to monitor the acceptance rate, i.e. the fraction of proposed perturbations that are accepted (Note: To not be confused with acceptance ratio. Although the acceptance rate is linked to the acceptance ratio, the acceptance ratio refers to the probability of accepting a proposed perturbation). If the acceptance rate is close to 1, then virtually all proposed moves will be accepted at the cost of moving very slowly in the state space. If the acceptance rate is close to 0 the chain will virtually not move at all. The choice of the proposal distribution should aim to balance the acceptance rate with the convergence of the chain. Although some theoretical results about the optimal acceptance rate exist in literature for simplified setups ([Gelman et al., 1997], [Roberts and Rosenthal, 2001]), a good rule of thumb is to have an acceptance rate far from 0 and far from 1.
Figure 5.25 shows the acceptance rate w.r.t. to the temperature parameter $T \in [0, 100]$, for all the moves for the samplers presented above. A temperature parameter $T \in [0, 15]$ offers good proposal distributions to sample from. Thus, in our experiments we set the initial temperature to $T = 15$ to obtain a balance between the acceptance rate and convergence speed.
A way to assess the performance of the samplers is to monitor the evolution of the energy of the model. Figure 5.24 shows the energy levels reached w.r.t. time. Since the number of iterations can be misleading in terms of computation efficiency (and we explained why in Section 5.4.5.1), we have used the computation time as a measure to evaluate the performance of the samplers.
The results in Figure 5.24 reveal two important aspects:

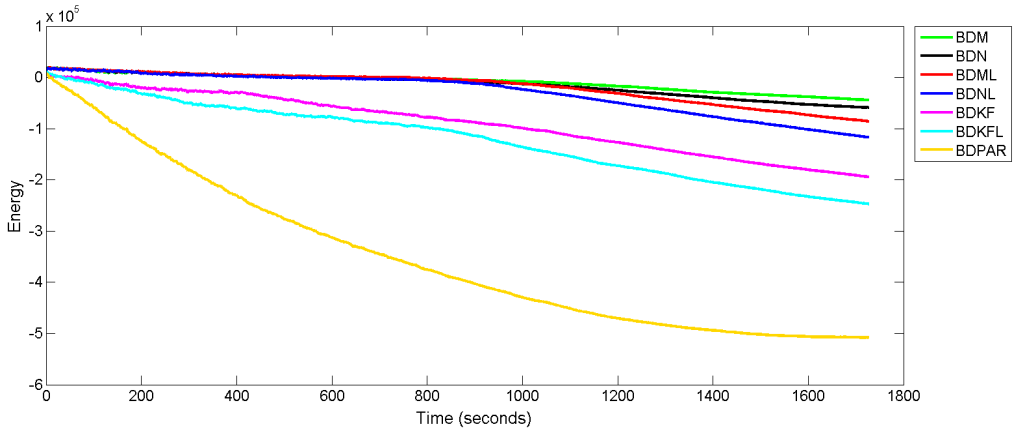


Figure 5.24: Energy levels reached w.r.t. time for different samplers.

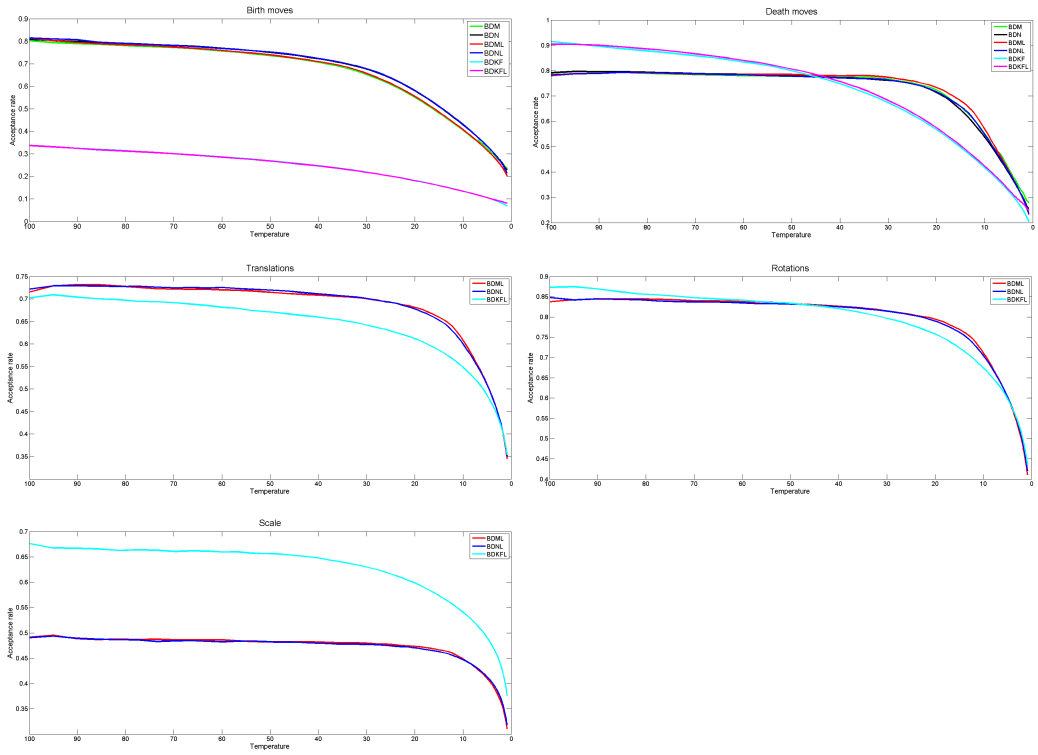


Figure 5.25: The acceptance rate w.r.t. to the temperature parameter $T \in [0, 100]$, for all the moves for the samplers presented. A temperature parameter $T \in [0, 15]$ offer good proposal distributions to sample from. Consequently, in our experiments we set the initial temperature to $T = 15$ to obtain a balance between the acceptance rate and convergence speed.

1. Local perturbations are more efficient than a death immediately followed by a birth. This can be observed both in the BDNL and the BDKFL sampler. Even though the computational gain from a single move is insignificant, it becomes apparent over a large number of iterations;

2. The Kalman-filter provides not only a significant gain in terms of iterations (as shown in Section 5.4.5.1), but also a gain in terms of computation times.

We have also displayed the results for the parallel implementation of BDNL proposed in Section 5.3.5 (BDPAR). Although a comparison between BDPAR and the other samplers would not be fair, we only state the result to give the reader a feeling about the actual gain that can be obtained using parallel implementations.

5.5 Conclusions

Two novel marked point process models for the joint detection and tracking of moving objects have been described in this chapter: Quality model and Statistical model. The models rely on three major building blocks:

- A dedicated energy function has been defined to jointly detect and track objects in image sequences. The proposed models are robust to appearance changes, illumination changes, high displacements between consecutive frames and partial occlusions of objects. The method has a high tolerance to noise;

- A powerful procedure for learning parameters from the data sets has been proposed to tackle the large number of parameters of the highly non-convex energy function. The parameter learning is posed as a constrained optimization problem and linear programming is used to search for a feasible solution;

- An enhanced RJMCMC sampler. Two approaches to improve the performance of the sampler have been investigated. First, the RJMCMC sampler was blended with the Kalman filter into a hybrid sampler that benefits from the advantages of both. The good convergence properties of the RJMCMC sampler are combined with the efficiency of the Kalman filter to produce an efficient sampler specifically adapted to multiple object tracking. Second, the parallel implementation of the RJMCMC sampler has been extended to 2D + T dimensions and the good scaling properties of this sampler w.r.t. the number of processing units has been shown.

The Statistical model is based on statistical hypotheses testing to determine whether an object is present at a given location. Through this approach we can configure the parameters of the model to reach an application-dependent level of false alarms. Nevertheless, this model does not control how an ellipse fits the target object which leads to poorer segmentation results.
The Quality model uses an evidence term to determine the presence of a target coupled with a contrast distance measure which aims to correctly fit the ellipse over

the target. The link between the external energy term and the level of false alarms is not obvious. Thus, it is more difficult to configure the parameters of the model in order to reach a desired level of detections/false alarms. Nevertheless, the importance of the contrast distance measure becomes apparent when the segmentation results are considered. This model offers a high control power over the fitness of a configuration w.r.t. the target configuration.

The proposed method has been applied to a large number and various types of data sets within the scientific fields of remote sensing and fluorescent microscopy: benchmark synthetic biological data sets have been extensively used to test, validate and compare the models. Real biological sequences have been used to show that the models are suited for real applications. Finally, results on high resolution satellite image sequences with high and low temporal resolution have shown the advantages of the proposed method.

The high detection and tracking rates show that spatio-temporal marked point process models are well suited for real-life applications and outperform current state of the art methods in these two domains.

Chapter 6

Conclusions and perspectives

Contents

6.1 Thesis contributions

During this Ph.D. thesis we studied how stochastic geometry can be efficiently employed for the analysis of high resolution optical images and videos taken by a remote satellite sensor. The aim of this work was to investigate the possibility of using marked point processes to detect and track multiple moving objects, where the targets appear small in the image with a usual size of 10 - 100 pixels per object. The main contributions of this thesis are:

1. **A novel spatio-temporal marked point process model for the detection and tracking of moving objects**. We have developed an intuitive model based on a solid theoretical background which incorporates constraints on the spatial and temporal distribution of the objects. The temporal information has allowed us to restrict the search for moving objects to areas that exhibit significant changes in consecutive frames. We have applied the model to detect and track various types of objects including cars, boats and pedestrians in satellite videos as well as vesicles in microscopy images. These applications have enabled the use of a simple yet sufficiently general constant velocity motion model. We have used the obtained trajectories to infer higher level information such as the traffic density estimation (in urban areas);

2. **Improved spatial marked point process model for boat extraction in harbors**. Boat counting in harbors is a very difficult task due to the particular distribution of the objects. In harbors boats are anchored very close to each other. Consequently, two or more neighboring boats are usually detected as one large object. Ben Hadj et al. [2010a] proposed a model for this extraction problem when all objects have the same orientation. We build on

the model presented by Ben Hadj et al. [2010a] and develop a more general and efficient model for counting boats in harbors regardless of their orientation. In particular, we discard the constraints which make the model of Ben Hadj et al. [2010a] too restrictive and introduce simple heuristics to improve the quality of the result;

3. **Automatic parameter estimation**. The performance of marked point process models for a specific application depends on a well suited and efficient choice of parameters. The models presented in this work are a linear combination of several weighted energy terms. These weights are generally hard to set manually as they lack a direct physical interpretation. We have developed an automatic parameter estimation technique based on linear programming that can be efficiently used to set the parameters of such models;

4. **Integrated RJMCMC sampler with Kalman-like moves**. RJMCMC dynamics are used to simulate the models presented in this thesis. RJMCMC is an iterative sampler that uses a set of perturbation kernels to simulate a Markov chain that converges to the desired distribution. This sampler has very good convergence properties which is why it is of great interest to the scientific community. Nevertheless, these properties come at a high computational cost. We have proposed a novel hybrid sampler by integrating the RJMCMC sampler with Kalman-like moves. The Kalman-like moves are used to direct the path taken by the RJMCMC sampler to converge more rapidly towards the desired distribution;

5. **Efficient parallel implementation of the RJMCMC sampler**. A successful attempt to increase the computational efficiency of the sampler was made by Verdié and Lafarge [2012]. Verdié and Lafarge [2012] propose a parallel implementation of the sampler on GPU and obtain great improvements in terms of computation time over the sequential version. They use a hierarchical partitioning scheme to divide the image space into independent cells and perform independent perturbations in these cells. Consequently, difficulties arise in correctly extracting objects at the cell boundary. We have proposed a new multiple core implementation of the RJMCMC that keeps a shared memory between processors so that the perturbations in a cell are performed by taking into account its neighborhood.

This research work shows that marked point process models can be used for tracking purposes. A qualitative analysis of the results reveals high detection and tracking performances for all the videos presented in this thesis. The advantages of this approach are discussed in the following section.

6.2 Advantages of the proposed methods

The proposed models presented in Chapter 4 and Chapter 5 enable the detection and tracking of moving objects and provide high level information about the objects

such as their trajectory, their size, shape and orientation. The integrated energy term coupled with the batch optimization process enables the detection of weakly contrasted objects or temporarily occluded objects while maintaining consistent trajectories.
This approach is automatic and permits to structure the extraction of trajectories based on the interactions between the objects. Complex object interactions can be efficiently incorporated into the models.
An important advantage of this method is its robustness to the type and quality of the data used. In particular, parameter training is performed only once for each data set. The model can then be successfully applied to each sequence in a data set. We also experiment with using the same parameters across multiple similar data sets and show that the quality of the result is not significantly affected.
Although the advantages of such a method are solid, several drawbacks have been identified during this research work.

6.3 Drawbacks of the proposed methods

First and foremost, the computational efficiency remains a problematic aspect. The mathematical framework of marked point processes in which this work is developed comes with an intrinsic computational burden. The use of efficient heuristics we proposed in this work to reduce the search space and to speed up the optimization process has significantly reduced the computation time. Nevertheless, in an era of real-time processing, these methods fail to keep up the pace. In the current settings as presented in this thesis, real-time performance can be obtained only for low-frequency videos with small images and a limited number of objects. Consequently, huge improvements have to be made in this regard to render marked point process models suitable for multiple object tracking in the future.
Another drawback of the proposed models is owed to the weak 'bright ellipse' target model. On the one hand, this target model is simple enough to be efficiently incorporated into the point process model. Moreover, for the applications presented in this thesis such a simple target model suffices to obtain high-quality results. On the other hand, the use of such a simple target model proves to be limiting in terms of applications. Consequently, a more elaborate target model that can integrate appearance information would be desired.

6.4 Perspectives

In this study we modeled the tracking problem at the level of objects. The interactions between these objects are described both in the spatial and temporal domain. Semantic information about object trajectories is then retrieved by grouping the objects based on their labels. An interesting approach would be to design a hierarchical model that integrates both low-level constraints between individual objects and high-level constraints between trajectories.

The geometric mark of a point within the point process depends on the application considered. Simple shapes such as rectangles or ellipses are more desirable to characterize the geometry of objects as they are described by a small number of parameters. In particular, we chose to use only ellipses throughout this work because they fit most of the objects of interest (e.g. cars, boats, vesicles, etc.). Nevertheless, an extension towards the use of multiple shapes would be desired. Lafarge et al. [2010a] propose a multi-marked process to extract objects with different shapes within an image. An extension to video data can be envisioned to distinguish between various object classes. However, care must be taken to preserve the class of an object throughout its trajectory.
We have designed our models to extract individual trajectories of moving objects. However, in some applications such as traffic monitoring, information about the traffic density can take precedence over individual tracks. The notion of traffic here is not restricted to vehicles, as it can also include pedestrian and maritime traffic for satellite data or cell membrane traffic in microscopy images. A model to estimate the density of objects throughout the video rather than individual trajectories can be of interest in such applications.
Finally, we have proposed several ways to improve the performance of the RJMCMC sampler. Even so, the optimization process should be further improved to make such models competitive with existent state of the art tracking algorithms.

6.5 Concluding remarks

This manuscript presents only a part of the work undertaken during the last three years. We have had the opportunity to explore different approaches and theories than those described here, to experiment with a large amount of data and to consistently recover from dead-ends and failed attempts. As such, the last years have been very instructive.
This study has allowed us to evaluate the potential of marked point process models for tracking purposes. We have shown that such models are strong competitors against state of the art methods with respect to the quality of the results. Yet, consistent improvements have to be done to the optimization procedure in order to make marked point process models a leading choice in the future.
Altogether, the vast research work on object detection and tracking in satellite as well as microscopy imagery has a major impact on our daily lives. From increased security levels and wildlife monitoring to sub-cellular activity analysis, object detection and tracking methods in these domains are one of the building blocks of a deeper understanding of our world and the patterns that govern it.

Appendix A

Appendix - Scientific activity

A.1 Journal papers

- P. Crăciun, M. Ortner, and J. Zerubia. A spatio-temporal marked point process model for joint detection and tracking of moving objects. Submitted to *IEEE Transactions on Image Processing*, 2015;
- P. Crăciun and J. Zerubia. Unsupervised marked point process model for boat extraction and counting in harbors from high resolution optical remotely sensed images. In *Revue Francaise de Photogrammétrie et Télédétection*, vol. 207, pp. 33-44, 2014;

A.2 Conference papers

- P. Crăciun, M. Ortner, and J. Zerubia. Joint detection and tracking of moving objects using spatio-temporal marked point processes. In *IEEE Winter Conference on Applications of Computer Vision*, USA, 2015;
- P. Crăciun and J. Zerubia. Towards efficient simulation of marked point process models for boat extraction from high resolution optical remotely sensed images. In *International Geoscience and Remote Sensing Symposium*, Canada, 2014;
- P. Crăciun and J. Zerubia. Unsupervised marked point process model for boat extraction in harbors from high resolution optical remotely sensed image. In *International Conference in Image Processing*, Australia, 2013;
- P. Crăciun, M. Ortner, and J. Zerubia. Integrating RJMCMC and Kalman Filters for multiple object tracking. In *GRETSI - Traitement du Signal et des Images*, France, 2015;
- P. Crăciun and J. Zerubia. Boat extraction in harbors from high resolution satellite images using mathematical morphology and marked point processes. In *GRETSI - Traitement du Signal et des Images*, France, 2013.

A.3 Invited talks

- **University of Lille 1, France**: Multiple object tracking in remotely sensed high resolution image sequences using spatio-temporal marked point processes,

September 2015;

- **Airbus Defense and Space, France**:
  - Multi-target tracking using spatio-temporal marked point process models: Applications to satellite data. May 2015;
  - Boat extraction in Mediterranean harbors using marked point processes. December 2013;
- **University of California Berkeley, Computer Vision Group, USA**: Joint detection and tracking of moving objects using spatio-temporal marked point processes. January 2015;
- **University of California Los Angeles, Center for Vision, Cognition, Learning and Autonomy, USA**: Joint detection and tracking of moving objects using spatio-temporal marked point processes. January 2015;
- **INRIA Rennes, France**: Spatio-temporal marked point process models for multiple target tracking: Applications to microscopy images. December 2014;
- **INRIA Bordeaux, France**: Marked point process models for boat extraction in harbors. May 2013;
- **West University of Timişoara, Faculty of Physics, Romania**: Object detection and counting using marked point process models: Applications in physics. February 2013.

A.4 International Summer School

Summer School on Topics in Space-Time Modeling and Inference, 27-31 May 2013 at Aalborg University, Denmark.

Appendix B

Introduction

Contents

B.1 Motivation

La dernière décennie a été une vitrine pour la guerre asymétrique avec des démonstrations d'attaques hautement organisées et des méthodes de communication et de planification avancées [Kydd and Walter, 2006, Poland, 2010]. En conséquence, la sécurité civile, militaire et économique est menacée par des tentatives d'assassinat bien documentés, des prises d'otages et des attaques terroristes. Des méthodes populaires telles que l'analyse des médias sociaux [Fuchs, 2009] ou l'écoute électronique [Landau, 2011] ne sont pas suffisantes pour prévenir ou détecter des attaques en cours. Mais quand ces méthodes échouent, la surveillance aérienne et spatiale peut aider à détecter les activités criminelles avant ou pendant leur exécution.
Les êtres humains ne sont cependant pas les seuls à pouvoir bénéficier de la surveillance aérienne ou satellitaire. Lors de la derniére décennie, des centaines d'espéces d'animaux sont devenus des espèces menacées par la chasse excessive ou le réchauffement planétaire. Par example, le réchauffement de la planéte a conduit à une perte accélérée de la banquise arctique au cours des derniéres années [Comiso et al., 2008, Stroeve et al., 2012] ce qui a un fort impact sur la faune arctique. La surveillance aérienne peut aider à évaluer les tendances de l'évolution statistique des espéces d l'Arctique [Fretwell et al., 2012, Platonov et al., 2013] sans se préoccuper au sujet des méthodes d'étude intrusives ou de la sécurité humaine.
Les satellites peuvent être équipés d'une variété de capteurs tels que des capteurs acoustiques, radar, ultrasoniques ou optiques. Chaque capteur possède des avantages et des inconvénients et le choix du type de capteur a embarquer à bord d'un

satellite dépend des applications spécifiques pour lesquelles ce satellite a été construit. Les données satellitaires ont déja été utilisées avec succès pour des opérations de recherche et sauvetage [Lukowski and Charbonneau, 2000, Jing and Danping, 2011], d'aide aux catastrophes naturelles [Voigt et al., 2007, Dell'Acqua et al., 2011, Bhangale and Durbha, 2014] ou de surveillance de l'environnement [Caccetta et al., 2011, Singh and Talwar, 2013]. Les progrès technologiques dans l'industrie spatiale permettent désormais d'obtenir des images beaucoup plus précises et fiables qui ont le potentiel de révolutionner les pratiques de surveillance au cours du 21éme siècle. Quel que soit le type de capteurs montés sur le satellite, l'analyse des données acquises est un travail difficile et fastidieux pour les opérateurs humains caractérisé par un niveau élevé de fatigue et d'ennui [Garcia, 2007] en raison de la grande quantité d'informations disponibles dans ces données. Des algorithmes appropriés pour le traitement de données et la recherche d'information automatique et semi-automatique dans des images peuvent aider l'opérateur humain. Néanmoins, il est difficile pour la plupart des applications de garantir un taux d'erreur faible et une grande confiance dans un algorithme, tout en offrant une performance quasi-temps réel.

Cette thèse porte sur l'analyse d'images statiques et dynamiques provenant principalement de capteurs optiques embarqués à très haute résolution. En particulier, nous traitons à deux problèmes principaux: la détection d'objets dans des images statiques et la détection et le suivi d'objets en mouvement dans les vidéos. La détection d'objets et, respectivement, le suivi sont deux étapes intermédiaires pour la compréhension automatique d'une scène. Ces motifs complexes d'information de haut niveau peuvent ensuite être utilisées pour modéliser et détecter des comportements suspects et des valeurs aberrantes peuvent être indicatives de comportements criminels. Des méthodes fondées sur des images et des vidéos fournissent des informations précieuses comme de nombreuses propriétés sur les objets détectés, comme par example la position, la taille, l'apparance ou la forme d'un objet peuvent être directement dérivés à partir de ces données.

Il est important de noter que nous utilisons le terme «vidéo» pour une séquence d'images prises à intervalles de temps relativement courts. La fréquence temporelle des vidéos peut être comprise entre environ $1 - 2$Hz et 30Hz ou plus, en fonction du type de capteur utilisé. Par example, un seul satellite en orbite basse (comme SkySat-1 par exemple) peut produire des images de résoultion sous-mètrique et capturer des vidéos haute résolution jusqu'à 90 seconds à 30 trames par seconde. Cependant, un grand nombre de satellites en orbite basse sont nécessaires pour couvrir une zone terrestre donnée pendant une longue période de temps. Si telle est l'intention, un satellite géostrationnaire pourrait fournir une alternative à une constellation de satellites en orbite basse. Les satellites géostationnaires sont géosynchrones, ce qui signifie qu'ils tournent avec la Terre (par opposition aux satellites en orbite basse qui sont hélio-synchrones) et peuvent couvrir une plus grande surface au sol. Par conséquent, un tel satellite produirait de très grandes images à une fréquence temporelle inférieure. Voilà pourquoi nous considérons une large gamme de fréquences temporelles pour les vidéos considérées dans cette thèse.

L'analyse d'images satellitaires n'est pas le seul domaine où les opérateurs humains sont surchargés sous la quantité de données disponibles. L'imagerie microscopique et la qualité de la vidéo microscopique ont connu une forte augmentation ces dernières années. L'analyse des données de microscopie est désormais monnaie courante dans des domaines tels que la médecine ou la recherche biologique. La haute vitesse d'acquisition des caméras permet l'observation en temps réel de processus cellulaires et sub-cellulaires dynamiques. En tant que telle, cette approche peut générer de vastes quantités de données qui doivent être stockées, traitées et analysées. Des méthodes automatiques et semi-automatiques pour la récuperation de l'information sont d'une grande importance pour l'opérateur humain dans ce domaine. Par conséquent, nous testons également les modèles développés dans cette thèse sur des séquences d'images biologiques synthétiques et réelles. Nous sommes particulièrement intéressés par la détection et le suivi des structures sub-cellulaires appelés vésicules, qui sont responsables du bon fonctionnement d'une cellule vivante. Une compréhension approfondie des modèles dynamiques et géométriques des vésicules et de la cellule en général, peut conduire à une compréhension biologique plus profonde et des traitements médicaux plus adéquats.
Les motifs géométriques complexes nécessitent souvent l'analyse statistique. Des outils statistiques adaptés et des modèles mathématiques appropriés sont nécessaires pour analyser ces données. La géometrie stochastique est l'un des domaines de la recherche mathématique qui vise à fournir de tels modèles et méthodes. La théorie moderne de la géométrie stochastique, mise au point par D.G. Kendall, K. Krickerberg et R.E. Miles, examine des motifs géométriques aléatoires de distributions compléxes [Stoyan et al., 1987, Stoyan and Stoyan, 1994, van Lieshout and Baddeley, 2002]. Les processus ponctuels spatiaux et en particulier les processus ponctuels marqués ont déja été appliquées avec succès à des problèmes de détection d'objets dans des images satellitaires très haute résolution [Ortner, 2004, Perrin et al., 2005, Lacoste et al., 2005, Descamps et al., 2008, Ortner et al., 2008, Descombes et al., 2011].
Cette thèse porte sur l'utilisation de modèles de processus ponctuels spatiaux et spatio-temporelles marqués pour la détection et la suivi d'objets dans différents types d'imagerie optique. L'objectif est d'examiner si les processus ponctuels marqués peuvent produire des résultats compétitifs pour ces tâches. Le fondement théoretique sous-jacent de cette thèse est basée sur la littérature existante, mais plusieurs idées et des approches nouvelles sont introduits afin d'adapter et d'améliorer les méthodes existantes par rapport à la précision de la détection des objets et de proposer une nouvelle approche de suivi d'objets multiples fondée sur les processus ponctuels marqués avec un bon compromis entre la performance de suivi et d'exécution.

B.2 Les défis

L'utilisation des vidéos pour la télédétection par satellite afin de détecter et de suivre des objets est une tâche difficile. Des défis sont recontrés tout au long de la chaîne

de traitement, de l'acquisition d'images à l'analyse de scène. Selon le moment et le lieu de l'apparition, ils peuvent être classées comme suit:

1. **Acquisition d'image / de vidéo**

   - **Le bruit du capteur** représente un écart par rapport aux valeurs radiométriques optimales des pixels de l'image. Selon le type de capteur utilisé, le bruit peut être modélisée soit comme additif, multiplicatif (tavelures) ou impulsif (poivre et sel) [Gonzalez and Woods, 2008];
   - **Les artefacts** sont des structures artificielles qui sont contenues dans l'image et représentent une perturbation du signal. Des exemples d'objets dans cette étape comprennent la saturation du capteur ou des problèmes de conversion A / N (analogique / numérique) [Gonzalez, 2013];
   - **Le flou** est le résultat du mouvement rapide de l'objet par rapport au capteur au faible réglage de l'éclairage qui conduisent à des temps d'exposition longs de l'appareil;
   - **Le contraste faible** apparaît en raison des conditions environnementales. Les conditions météorologiques spécifiques telles que le brouillard ou les nuages sont des situations courantes qui conduisent à un constraste faible dans les caméras optiques;

2. **Transfer d'image / de vidéo**

   - **Des artefacts** lors de cette étape peuvent être causées par une connexion perturbée qui peut conduire à des artefacts puissants ou même à des images manquantes [Gonzalez, 2013];
   - **Les artefacts de compression-décompression** sont des structures comme des blocs qui peuvent dégrader considérablement la qualité des données [Netravali and Haskell, 1995, Antonini, 2003];

3. **Analyse d'image / de vidéo**

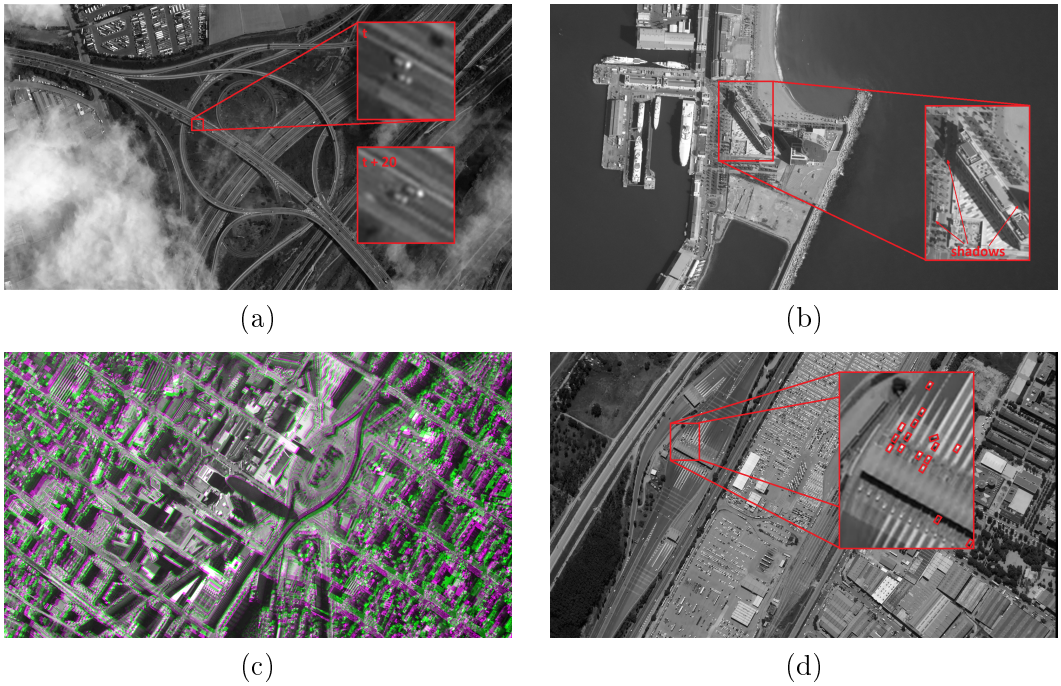
   - **La petite taille de l'objet** est une conséquence de la résolution d'échantillonnage au sol des capteurs considérés. La taille des objets est généralement de quelques pixels (i.e.variant entre 10 et 100 pixels) ce qui rend la détection et la reconnaissance particulièrement difficiles puisque la quantité d'information disponible sur l'apparence et la forme des objets est très limitée. Par example, des scènes urbaines sont caractérisées par un grand nombre de cibles situées près les uns des autres, qui entraîne généralement que deux petits objets voisins soient détectés comme une grande cible. La circulation des véhicules sur une route principale très fréquentée ou des bateaux dans les ports sont des exemples de tels cas;
   - **Les ombres** peuvent être causées soit par les objectifs eux-mêmes ou par les objets environnants. En milieu urbain, les ombres peuvent être

générées par de gros véhicules tels que les camions et les autobus ou par de grands immeubles. Par conséquent, certains objectifs ou cibles potentielles pourraient être manqués ou des ombres pourraient être identifiées comme des cibles;

- **Les mouvements indépendant du capteur et l'objet** ajoute des difficultés supplémentaires pour la détection d'objets. L'enregistrement de l'image et la déformation [Zitovà and Flusser, 2003] sont utilisés pour compenser le mouvement de la caméra. Ensuite, les objets en mouvement peuvent être détectés par leur mouvement relatif par rapport à l'arrière-plan fixe;
- **Les exigences en temps de calcul** posent un grand défi pour les algorithmes de suivi. Dans la plupart des applications, un compromis entre la qualité des résultats et la durée de traitement doit être effectué. Ce problème devient encore plus évident dans les scènes complexes, présentant un grand nombre d'objets.

Les défis décrits ci-dessus donnent une idée de la raison pour laquelle la détection d'objets dans des images statiques, respectivement la détection et le suivi d'objets en mouvement dans des vidéos est très difficile. Cette thèse ne vise pas à relever tous les défis mentionnés ci-dessus. Les développements récents en débruitage d'image [Shao et al., 2014], en restauration d'image [Portilla et al., 2003], en déflouage d'image [Zhang et al., 2013] ou en filtrage temporel [Müller and Müller, 2010] fournissent l'état de l'art des méthodes permettant de faire face à la plupart des défis lors de l'acquisition, du transfert et du traitement initial des données satellitaires. Cependant, cette thèse aborde presque tous les enjeux de l'analyse d'image / du vidéo présentées ci-dessus. La mauvaise qualité des données est prise en compte implicitement gràce à la résistance au bruit du modèle mathématique utilisé.

Une représentation visuelle des défis qui se posent lors de l'analyse d'image / de vidéo est présentée dans la Figure B.1. La figure B.1 (a) montre deux voitures voisins sur la route. La modélisation de l'apparence des véhicules est difficile, car il existe une quantité limitée d'information. En outre, au cours du dépassement, les voitures sont très proches l'une de l'autre ce qui rend la détection individuelle problématique parce que les frontières entre véhicules deviennent floues.
La figure B.1 (b) illustre une difficulté supplémentaire causée par les ombres. L'ombre d'un grand bâtiment pourrait obstruer totalement un objet passant à travers cette zone. L'ombre peut également être générée par de grandes cibles telles que des camions. Cela peut devenir un problème en particulier lorsque plusieurs grands objectifs sont situés ensemble dans un groupe. La méthode de détection et de suivi peut interpréter à tort l'ensemble du groupe comme un unique objet en mouvement.
La figure B.1 (c) montre la difficulté de détecter des objets en mouvement du fait du mouvement de la caméra. Les couleurs de l'image montrent le déplacement de la caméra entre deux trames consécutives. Un détecteur de mouvement qui ne prend pas en compte le mouvement de la caméra pourrait produire un nombre important



(a) (b)

(c) (d)

Figure B.1: Un exemple pour chaque défi de la détection d'objet et le suivi à partir d'un capteur satellitaire donné (©Airbus D&S). (a) La petite taille de l'objet conduit à une modélisation de l'apparence limitée. La reconnaissance d'un objet individuel est difficile. Les bords entre les deux objets sont flous. (b) Les ombres projetées par de grands immeubles augmentent la difficulté de détection d'objets en raison du faible contraste entre les objets et l'arrière-plan. Les ombres dues aux autobus et camions se déplacent avec l'objet et peuvent être identifiées par erreur commes des cibles. (c) Le mouvement de la caméra produit des grandes distorsions dans des trames consécutives. (d) Exemple de zone urbaine dense.

de fausses alarmes puisque les objets statiques semblent également se déplacer en raison du déplacement de la caméra.
Figure B.1 (d) montre un exemple d'une zone urbaine dense. La scène contient près de 100 véhicules. Chaque véhicule dans la petite fenêtre a été étiqueté manuellement avec une carré rouge. L'étiquetage manuel est utilisé comme une information de vérité terrain pour évaluer la performance de l'algorithme automatique proposé de détection et de suivi proposé. Un algorithme efficace devrait diminuer de façon significative le temps de traitement requis par rapport à un opérateur humain, tout en maintenant un haut niveau de précision.
Plusieurs approches ont été dévelopées dans le passé pour relever les défis présentés ci-dessus et sont présentés dans la Chapitre 2. Néanmoins, il ya un grand potentiel pour améliorer la performance par rapport aux méthodes actuelles à la fois en termes de robustesse et de temps de traitement.

B.3 Méthodes et approches proposées

L'objectif de cette thèse est d'étudier l'utilisation de contraintes géométriques dans les méthodes de détection et de suivi d'objets vià des processus ponctuels marqués. Cette approche permet l'inclusion de deux types de contraintes géométriques: la forme de la cible peut être imposée (comme nous traitons principalement d'objets fabriqués par l'homme, leur forme peut généralement être efficacement approximée à l'aide d'une forme paramétrique simple) et la dispersion géometrique des cibles peut être constrainte.
Les processus ponctuels forment une classe de modèles mathématiques au sein du thème général de la géometrie stochastique et ont déjà été appliqés avec succès à des problèmes de détection dans l'imagerie satellitaire. Dans ce qui suit, nous presentons une brève description de la raison pour laquelle cette classe de modèles mathématiques est intéressante pour le traitement d'image et de vidéo.

B.3.1 Probabilités dans l'analyse des images / du vidéo

Dans l'analyse d'image, les processus ponctuels marqués sont une évolution naturelle des champs de Markov en raison de l'augmentation significative de la résolution spatiale des images du fait des récent capteurs. Le lecteur intéressé par une analyse détaillée de cette évolution peut se référer aux manuscrits d'HdR de Pérez [2003] et Descombes [2004].

B.3.1.1 Un modèle stochatisque d'une image

Une image est une ensemble de pixels. Une image, notée I, peut être représente par une fonction qui associe une valeur de niveau de gris à chaque pixel u, $u \in K \subseteq \mathbb{Z}^2$:

$$\begin{aligned} I: \quad & K \to \mathbb{R} \\ & u \to I(u). \end{aligned}$$

Cette définition peut facilement être étendue pour une vidéo en considérant un espace en 3 dimensions tel que pour chaque pixel u, $u \in K \times T \subseteq \mathbb{Z}^3$, où K représente le support d'une image et T est l'axe du temps. Pour plus de simplicité, nous ne présenterons que la justification dans le cas d'une image en 2-D.
Soit $(\Omega, \mathcal{A}, \mathbf{P})$ une espace de probabilité, un modèle stochastique d'une image consiste à considerér que les valeurs de viveau de gris de chaque pixel peuvent être comme une réalisation d'une version stochastique du fonction I. I est maintenant une variable aléatoire:

$$I: \quad \Omega \to \mathbb{R}^K. \tag{B.1}$$

Une loi de probabilité $\mathbf{P}_I$ est associé à la variable aléatoire I telle que $\mathbf{P}_I(A) = \mathbf{P}(I^{-1}(A))$. Le modèle d'image le plus simple consiste à considerer tous les pixels comme conditionnellement indépendant: $\mathbf{P}(I \in A) = \sum_{u \in K} \mathbf{P}(I(u) \in A_u)$.
D'un point de vue physique, une image peut être considérée comme une observation bruitée d'un phénomène sous-jacent. Un moyen efficace de capturer cette relation

physique est d'utiliser un cadre bayésien pour modéliser l'image.
Un modèle bayésien d'une image peut être écrit comme:

$$\mathbf{P}(I = y \in A) = \sum_x \mathbf{P}(I = y \in A | X = x)\mathbf{P}(X = x) \tag{B.2}$$

où $\mathbf{P}(I = y \in A)$ est la loi marginale des observations, $\mathbf{P}(I = y \in A|X = x)$ est la loi de l'observation et $\mathbf{P}(X = x)$ est la loi a priori qui intègre la connaissance préalable sur le phénomène X sous-jacent. En utilisant le théoréme de Bayes, nous pouvons analyser le phénomene X à partir d'observations $y_1, \cdots, y_n$ comme suit:

$$\mathbf{P}(X = x|I = y) \propto \mathbf{P}(I = y|X = x)\mathbf{P}(X = x). \tag{B.3}$$

Le modèle de Bayes est intéressant d'un point de vue statistique, car il fournit une classe d'estimateurs naturels: les estimateurs bayesiens [Marin, 2007, Gelman et al., 2014]. Un exemple d'estimateur bayesien est le critère de Maximum A Posteriori (MAP) qui peut être écrit comme suit:

$$\hat{X}_{MAP} = \arg\max_x \prod_{i=1}^{n} \mathbf{P}(X = x|I_i = y_i) \tag{B.4}$$

quand toutes les observations $y_1, \cdots, y_n$ sont conditionnellement indépendants.

B.3.1.2 Processus ponctuels marqués pour l'analyse d'image

Une description détaillée des processus ponctuels est donnée dans le Chapitre 3. Pour l'instant, il suffit de noter que le processus de point marqué peut être utilisé pour modéliser des configurations aléatoires de formes géométriques simples.
Soit $X : \quad \mathbf{x} = \{x_1, \cdots, x_{n(\mathbf{x})}\}$, où $x_i, \quad i = \overline{1, n(\mathbf{x})}$ est un objet géométrique placé au hasard dans une image et $n(\mathbf{x})$ est le nombre de ces d'objets. Puis, dans un cadre bayésien, un modèle d'observation est donné, par exemple, par:

$$\mathbf{P}(I(u) = y_u|X = \mathbf{x}) = \mathbb{1}(u \in \mathbf{x})\mathbf{P}(I(u) = y_u|u \in \mathbf{x}) + \mathbb{1}(u \notin \mathbf{x})\mathbf{P}(I(u) = y_u|u \notin \mathbf{x}). \tag{B.5}$$

Dans cet exemple, tous les pixels u appartenant à un objet de la configuration forment la silhoutte de la configuration. Perrin et al. [2005] a proposé d'utiliser les valeurs radiométriques des pixels pour déterminer si elles appartiennent à la configuration de l'objet ou à l'arrière-plan. En tant que tel, les pixels appartenant à la configuration ont été modélisés comme une distribution gaussienne avec une valeur moyenne élevée, tandis que les pixels appartenant à l'arrière-plan ont été modélisés comme une distribution gaussienne avec une valeur moyenne inférieure.
Ce modèle bayésien se heurte à deux difficultés dans des images réelles [Ben Hadj et al., 2010b]:

1. La classe de fond n'est en général pas homogéne et assumer une distribution gaussienne pour tout le fond est peu réaliste pour notre problème. Par exemple, considérons une scène urbaine. Si nous voulons détecter des voitures sur

la route, il pourrait être raisonnable de supposer que la zone entourant une seule voiture (la route) suit une distribution gaussiene, tandis que l'ensemble de la zone urbaine ne le fait pas (car cet ensemble contient des ombres, des bâtiments, des arbres, etc.);

2. La probabilité des pixels qui appartiennent à la configuration ne prend pas en compte les propriétés morphologiques des objets. Par exemple, si nous considèrons une image contenant un seul grand rectangle brillant et si nous souhaitons détecter de petites ellipses brillantes, le modèle bayésien choisi conduira à la détection d'autant de petites ellipses que possible dans le grand rectangle, ce qui n'est pas le résultat souhaité.

À l'extérieur de ce cadre bayésien, ces limitations ont été dépassées par une approche détecteur dans laquelle le modèle d'observation est composé par des vraisemblances locales pour chaque objet faisant partie de la configuration. Cette approche a permis l'utilisation de processus ponctuels marqués pour diverses tâches de détection d'objets. Descamps et al. [2008] ont utilisé un processus de point marqué d'ellipses pour compter le nombre de flamants roses dans des grandes colonies contenant des millierrs de spécimens. Perrin et al. [2005] ont développé un processus de point marqué de cercles pour l'extraction des couronnes d'arbres. Leur modèles pourraient également identifier l'ombre de l'arbre, et puis déterminer la position du soleil lorsque l'image a été prise. Ortner [2004] a proposé un processus de point marqué de rectangles pour la détection de la trace au sol des bâtiments. Les interactions d'alignement entre les rectangles ont été définis afin de reproduire la disposition des bâtiments. Plus récemment, Ben Hadj et al. [2010a] ont introduit un processus de point marqué d'ellipses pour compter les bateux dans des ports. Par conséquent, les modèles de processus ponctuels ont montré leur potentiel pour la détection d'objets dans des images haute résolution.

B.3.2 Application à la détection et suivi des objets dans des vidéos

L'objectif de cette thèse est d'étudier la possibilité d'utiliser des modèles de processus de point marqué pour la détection et le suivi d'objets en mouvement dans des vidéos. La détection et le suivi d'objets en mouvement jouent un rôle important dans l'analyse vidéo où il représentent une étape nécessaire dans la réalisation d'analyses de haut niveau, par exemple la surveillance terrestre et maritime du traffic. Le problème de la détection d'objets dans des images statiques est également abordé dans cette thèse, avec un fort accent sur le problème particulièrement difficile de l'extraction des bateaux dans des ports.
Les principales données utilisées dans cette thèse sont des images satellitaires à très haute résolution et des vidéos qui ont été fournies par Airbus Defense and Space. Ces ensembles de données varient à la fois en taille et fréquence temporelle, allant de jeux de données avec peu de trames à des fréquences temporelles basses (14 trames à $1-2$ Hz) jusqu'à de longues vidéos à de hautes fréquences temporelles (plus de 3000 trames à 30 Hz). Les informations de vérité terrain pour ces ensembles de

données ont été construits à la main, et ensuite validés par un expert.
Des ensembles de données secondaires provenant du domaine biologique ont également été analysés dans cette thèse. Ces ensembles de données ont été fournis par l'Institut Pasteur à Paris et l'Institut Curie à Paris. En outre, un logiciel gratuit nommé Icy et développé par l'Unité d'Analyse Quantitative de l'Institut Pasteur à Paris a été utilisé pour créer automatiquement des images biologiques synthétiques et des vidéos avec une vérité terrain associée à chaque séquence pour évaluer la qualité des modèles proposés dans cette thèse.

B.4 Organisation de thèse et contributions

Cette thèse est organisée comme suit:

- **Le chapitre** 1 donne un aperçu de la thèse, décrit les données utilisées, souligne la motivation de ce travail et résume les contributions;
- **Le chapitre** 2 decrit l'état de l'art des approches liées aux problèmes de détection et de suivi d'objets;
- **Le chapitre** 3 présente les processus ponctuels comme un cadre approprié pour la détection d'objets, ainsi que pour le suivi. L'état de l'art pour des techniques d'estimation de paramètres et de simulation spécifiquement adaptéess à ce cadre est aussi décrit;
- **Le chapitre** 4 introduit les premières contributions importantes de cette thèse. Il fournit le modèle développé pour la détection et le comptage des bateaux dans les zones portuaires. Les contributions de ce chapitre peuvent se résumer comme suit:
  - La partie 4.1. présente un modèle amélioré de processus de point marqué mis au point pour la détection des bateaux et le compare au modèle précédemment présenté par Ben Hadj et al. [2010a];
  - La partie 4.2. décrit les techniques d'estimation des paramètres utilisées pour definir les paramètres du modèle;
  - La partie 4.3. propose une nouvelle mise en œvre parallèle du schéma d'optimisation RJMCMC largement connu.
- **Le chapitre** 5 introduit la contribution principale de cette thèse. Dans ce chapitre, nous décrivons un modèle de processus de point marqué spécifiquement développé pour le suivi d'objets multiples. Les contributions de ce chapitre peuvent se résumer comme suit:
  - La partie 5.1. introduit un nouveau modèle et présente une description approfondie des contraintes et des restrictions de ce modèle;

  - La partie 5.2. développe une nouvelle approche de programmation linéaire pour estimer les paramètres du modèle dans le cadre de processus de points;
  - La partie 5.3. propose de nouveaux noyaux de perturbation spécifiquement ajustés pour le problème de suivi des objets multiples à utiliser dans le schéma d'optimisation RJMCMC.

- **Le chapitre** 6 conclut la thèse et décrit les perspectives potentielles de ce travail.

B.5 Publications

Les papiers suivants ont été publiés ou soumis à partir des travaux contenus dans ce manuscrit de thèse:

- **Dans des revues**:
  - P. Crăciun, M. Ortner, et J. Zerubia. A spatio-temporal marked point process model for joint detection and tracking of moving objects. Sousmis à *IEEE Transactions on Image Processing*, 2015;
  - P. Crăciun et J. Zerubia. Unsupervised marked point process model for boat extraction and counting in harbors from high resolution optical remotely sensed images. *Revue Francaise de Photogrammétrie et Télédétection*, vol. 207, pp. 33-44, 2014;
- **Dans des conférences nationales et internationales**:
  - P. Crăciun, M. Ortner, and J. Zerubia. Sousmis à IEEE Computer Vision and Pattern Recognition, USA, 2016;
  - P. Crăciun and J. Zerubia. Sousmis à IEEE International Symposium on Biomedical Imaging, Czech Republic, 2016;
  - P. Crăciun, M. Ortner, et J. Zerubia. Joint detection and tracking of moving objects using spatio-temporal marked point processes. *IEEE Winter Conference on Applications of Computer Vision*, USA, 2015;
  - P. Crăciun et J. Zerubia. Towards efficient simulation of marked point process models for boat extraction from high resolution optical remotely sensed images. *International Geoscience and Remote Sensing Symposium*, Canada, 2014;
  - P. Crăciun et J. Zerubia. Unsupervised marked point process model for boat extraction in harbors from high resolution optical remotely sensed image. *International Conference in Image Processing*, Australie, 2013;
  - P. Crăciun, M. Ortner, et J. Zerubia. Integrating RJMCMC and Kalman Filters for multiple object tracking. *GRETSI - Traitement du Signal et des Images*, France, 2015;

– P. Crăciun et J. Zerubia. Boat extraction in harbors from high resolution satellite images using mathematical morphology and marked point processes. *GRETSI - Traitement du Signal et des Images*, France, 2013.

Appendix C

Résumé étoffé

L'objectif principal de cette thèse est d'analyser l'utilisation des contraintes géometriques dans les méthodes de détection et de suivi d'objets via des processus ponctuels marqués. Cette approche permet l'inclusion de deux types de contraintes géometriques: la forme de la cible (comme nous traintons principalement d'objets fabriqués par l'homme, leur forme peut généralement être efficacement estimée à l'aide d'une forme paramétrique simple) et la dispersion géometrique des cibles.
Nous définissons un processus ponctuel marqué d'ellipses qui se caractérise par la localisation spatiale et temporelle des ellipses, leur demi-grand axe et demi-petit axe, leur angle par rapport à l'horizontale et leur étiquette. Nous utilisons l'étiquette comme un attribut permettant de différencier les trajectoires distinctes (par exemple, toutes les ellipses avec la même étiquette forment une seule trajectoire). Nous utilisons une famille de processus de Gibbs pour définir la fonction de densité de probabilité de nos modèles comme suit:

$$f_\theta(X = \mathbf{X}|\mathbf{Y}) = \frac{1}{c(\theta|\mathbf{Y})} \exp^{-U_\theta(\mathbf{X},\mathbf{Y})} \tag{C.1}$$

où:

- $\mathbf{X} = \{\mathbf{x_1} \cup \mathbf{x_2} \cup \cdots \cup \mathbf{x_t} \cup \cdots \cup \mathbf{x_T}\}$ est la configuration de toutes les ellipses à temps t;
- $\mathbf{Y}$ représente le cube d'images 3D;
- θ est le vecteur de paramètres;
- $c(\theta|\mathbf{Y}) = \int_\Omega \exp^{-U_\theta(\mathbf{X},\mathbf{Y})} \mu(d\mathbf{X})$ est la constante de normalisation, Ω étant l'espace des configurations et $\mu(\cdot)$ étant la mesure de l'intensité du processus de Poisson de référence;
- $U_\theta(\mathbf{X}, \mathbf{Y})$ est le terme d'énergie, qui dépend des parametres.

En utilisant le critère du maximum a posteriori (MAP), la configuration la plus probable des objets correspond au minimum global de l'énergie:

$$X \in arg\max_{\mathbf{X}\in\Omega} f_\theta(X = \mathbf{X}|\mathbf{Y}) = arg\min_{\mathbf{X}\in\Omega}[U_\theta(\mathbf{X}, \mathbf{Y})]. \tag{C.2}$$

La fonction d'énergie est divisée en deux parties: un terme d'énergie externe, $U^{ext}_{\theta_{ext}}(\mathbf{X}, \mathbf{Y})$ qui détermine à quel point la configuration correspond à la séquence d'entrée, et un terme d'énergie interne, $U^{int}_{\theta_{int}}(\mathbf{X})$, qui intègre les connaissances sur

l'interaction entre les objets dans un cadre unique et à travers la totalité de la séquence d'images considérée. L'énergie totale peut être écrite comme la somme de ces deux termes:

$$U_\theta(\mathbf{X}, \mathbf{Y}) = U^{ext}_{\theta_{ext}}(\mathbf{X}, \mathbf{Y}) + U^{int}_{\theta_{int}}(\mathbf{X}). \quad \text{(C.3)}$$

Les vecteurs de paramètres des termes d'énergie externe et interne sont respectivement θ_{ext} et θ_{int} et $\theta = [\theta_{ext}, \theta_{int}]$.

Le terme d'énergie interne. Le terme d'énergie interne est constitué d'un ensemble de contraintes destinées à une détection correcte des objets afin de faciliter le suivi. Ces contraintes ciblent la disposition des objets. Par exemple, la cohérence géometrique et physique doit être maintenue tout au long de la séquence.
Nous définissons trois termes énergétiques internes comme suit:

- **Le modèle dynamique**. Une propriété caractérissant le suivi (par opposition à des détections individuelles par image) est que dans la plupart des cas, les trajectoires des cibles sont lisses. Cela permet de favoriser des configurations où les cibles présentent un mouvement décrit par un modèle dynamique. Pour nos expériences, nous supposons que les objets suivent un modèle dynamique à vitesse constante. Bien que ce modèle de mouvement soit relativement simple, il est assez général pour couvrir une grande variété d'applications de suivi. Ce terme favorise la détection de cibles lorsque le terme d'attache au données à une faible valeur, mais que le modèle dynamique indui l'existence de la cible;

- **Persistance de l'étiquette**. Afin de faire la distinction entre des trajectoires distinctes, une étiquette est ajoutée à la marque de chaque ellipse. Cette étiquette peut être considérée comme un identificateur de trajectoire. Différentes étiquettes signifient différentes trajectoires. Ainsi, le nombre d'étiquettes doit être étroitement lié au nombre de trajectoires dans l'ensemble de données. Idéalement, le grand nombre de cibles dispersées à travers la séquence d'images doit être assingé à un assez petit nombre d'étiquettes. Par conséquent, nous utilisons ce terme pour limiter le nombre d'étiquettes, favorisant ainsi les configurations avec un petit nombre d'étiquettes distinctes;

- **L'exclusion mutuelle**. La manipulation de la collision entre ellipses ou le chevauchement dans une trame donnée est un aspet crucial lors de la détection et le suivi des objects. Dans nos modèles, une pénalité infinie est attribuée à une configuration qui contient des ellipses qui se chevauchent plus qu'un seuil fixé à l'avance. Ainsi, la probabilité de sélection d'une configuration de ce type est égale à zéro. Ce terme d'énergie est utilisé pour limiter le nombre d'ellipses qui couvrent le même emplacement dans une trame donnée.

Une description détaillée de chaque terme peut être trouvée dans la partie 5.1.1.

Terme externe d'énergie. Nous avons développé deux approches alternatives pour relier une configuration aux données: le **Quality model** et le **Statistical model**.

- **Quality model**. Ce modèle combine une preuve de cible qui est calculée en fonction de la différence de trames consécutives pour laquelle un seuil est appliqué en mesurant la distance de contrast entre l'intérieur et le bord de l'ellipse. La différenciation des images est utilisée pour identifier les endroits susceptibles de contenuir des cibles mobiles. Le résultat du cadre de différenciation est une image en niveaux de gris dans lequel l'intensité d'un pixel représente la quantité de changement entre les trames présente à cet endroit. Plus la quantité de changement (resp. plus l'intensité d'un pixel est grande), plus il est probable pour une cible mobile soit présente à cet endroit. Une mesure de distance de contraste est alors utilisée pour affiner la détection et extraire des informations telles que l'orientation et la taille des cibles.

- **Statistical model**. Cette méthode repose sur la théorie de la décision bayésienne et consiste à tester statistiquement deux hypothèses et choisir la plus vraisemblale. Dans notre contexte, l'objectif est de déterminer si une cible est présente ou absente.
  Tout d'abord, nous calculons la différence entre deux trames consécutives. Afin de déterminer la probabilité d'existance d'une cible, nous utilisons une fenêtre glissante que nous translatons d'un pixel à la fois. Nous considérons les deux hypothèses suivantes:

  – $\mathbf{H_0}$: l'ellipse ne couvre que l'arrière-plan aucune cible n'étant présente;
  – $\mathbf{H_1}$: l'ellipse est placé dans le centre d'une cible.

  Nous utilisons la règle de décision de Neyman-Pearson pour choisir entre les deux hypothèses $\mathbf{H_0}$ et $\mathbf{H_1}$.

Une description détaillée de ces deux modèles énergétiques externes peut être trouvée dans la partie 5.1.2.
Terme d'énergie totale. Le terme d'énergie totale peut être écrit comme la somme de tous les termes d'énergie définis ci-dessus:

- **Quality model**:

$$\begin{aligned} U_\theta(\mathbf{X},\mathbf{Y}) &= \gamma_{dyn}U^{int}_{dyn}(\mathbf{X}) + \gamma_{label}U^{int}_{label}(\mathbf{X}) + \gamma_o U^{int}_{overlap}(\mathbf{X}) + \\ &\quad \gamma_{ev}U^{ext}_{ev}(\mathbf{X}|\mathbf{Y}) + \gamma_{cnt}U^{ext}_{cnt}(\mathbf{X}|\mathbf{Y}). \end{aligned} \tag{C.4}$$

- **Statistical model**:

$$\begin{aligned} U_\theta(\mathbf{X},\mathbf{Y}) &= \gamma_{dyn}U^{int}_{dyn}(\mathbf{X}) + \gamma_{label}U^{int}_{label}(\mathbf{X}) + \gamma_o U^{int}_{overlap}(\mathbf{X}) + \\ &\quad \gamma_{stat}U^{ext}_{stat}(\mathbf{X}|\mathbf{Y}). \end{aligned} \tag{C.5}$$

où $\mathbf{X}$ est la configuration d'ellipses, $\mathbf{Y}$ sont les données d'images et $\gamma.$ sont les poids associés à chaque terme de l'énergie totale.
Un total de cinq (resp. quatre) poids $\gamma.$ sont nécessaires pour équilibrer les termes individuels dans l'énergie finale du Quality model (resp. Statistical model).

Ces paramètres ne peuvent pas être réglés de manière empirique. Les méthodes existantes d'estimation des paramètres fondés sur l'algorithme d'Espérance-Maximisation (voir Dempster et al. [1977]) sont très longues et généralement instables dans ce cas. Par conséquent, nous avons développé une technique alternative d'apprentissage des paramètres, afin d'améliorer la stabilité du modèle et de réduire la charge de calcul.

Estimation des paramètres. Une propriété clé de l'énergie décrite dans l'éq. C.4 et l'éq. C.5 est sa linéarité par rapport aux paramètres de poids. Cette propriété est cruciale pour déterminer un algorithme d'apprentissage approprié. La linéarité est exploité dans la Section 5.2. en proposant une procédure d'apprentissage des paramètres fondée sur la programmation linéaire.
D'une part, la fonction de densité a posteriori $\pi(\mathbf{X})$ est une combinaison linéaire des paramètres de poids pour toute configuration donnée, $\mathbf{X}$. D'autre part, si nous utilisons le RJMCMC pour simuler les deux modèles proposés, seul le rapport $\pi(\mathbf{X}')/\pi(\mathbf{X})$ doit être calculé dans une transition de la chaîne de Markov sans connaître la valeur exacte de $\pi(\mathbf{X}')$ ni de $\pi(\mathbf{X})$. A partir de ces propriétés, un ensemble de contraintes $\pi(\mathbf{X}')/\pi(\mathbf{X}) \geq 1$ (or ≤ 1) peut être établi, si une configuration est connue pour être meilleur q'une autre. Ces contraintes peuvent ensuite être transformées en inégalités linéaires des paramètres de poids. Enfin, aprés avoir recueilli un nombre suffisant d'inégalités linéaires, la programmation linéaire peut être utilisée pour trouver une solution réalisable pour les paramètres de poids.
Une fois que les paramètres du modèle ont été trouvés, la fonction de densité peut être simulée pour déterminer la meilleure configuration d'objets qui décrivent un ensemble de données donné.

Simulation. Les énergies décrites dans l'éq. C.4 et l'éq. C.5 ne sont clairement pas convexe. Il est facile de construire des exemples qui ont deux minima pratiquement égaux, séparés par un mur de valeurs élevées de l'énergie. La dépendance causée par les contraintes physiques d'ordre élevé est la principale raison qui pousse l'énergie à être non-convexe.
La distribution ciblée est la distribution postérieure $\mathbf{X}$, i.e. $\pi(\mathbf{X}) = f(\mathbf{X}|\mathbf{Y})$, définie sur une union de sous-espaces de dimensions différantes. La méthode d'optimisation la plus largement connue pour les fonctions d'énergie non-convexes et un nombre inconnu d'objets est le RJMCMC (aussi appelé MCMC à saut réversible) développé par Green [1995]. Le RJMCMC utilise un mélange de noyaux de perturbation pour créer des tunnels à travers les murs de haute énergie.
Nous utilisons le recuit simulé pour trouver un minimiseur de la fonction de l'énergie. La fonction de densité décrite dans l'éq. C.1 peut être réécrite comme:

$$f_{\theta,i}(X = \mathbf{X}|\mathbf{Y}) = \frac{1}{c_{Temp_i}(\theta|\mathbf{Y})} \exp^{-\frac{U_\theta(\mathbf{X},\mathbf{Y})}{Temp_i}} \tag{C.6}$$

où $Temp_i$ est un paramètre de température qui tend vers zéro lorsque i tend vers l'infini. Si $Temp_i$ diminue logarithmiquement, alors X_i tend vers un optimiseur global $f_{\theta,i}$. Dans la pratique, cependant, une loi logarithmique n'est pas acceptable, par conséquent, une loi géométrique est utilisée à la place pour diminuer la temperature. L'échantillonneur RJMCMC a été intégré dans un schéma de recuit simulé pour de meilleures performances. Néanmoins, cela a aussi conduit à un coût de calcul accrue. Pour faire face à ce problème, nous avons proposé deux nouvelles approches:

- **L'intégration des perturbations de Kalman dans le RJMCMC**. Les filtres ont prouvé qu'ils pouvaient fournir des performances de suivi relativement rapides et peu coutante, en particulier pour le suivi d'une seule cible. Les propriétés des filtres séquentiels peuvent être efficacement exploités sur le plan de l'échantillonnage RJMCMC. Le filtre de Kalman est utilisé pour générer des propositions de perturbations plus pertinentes qui sont ensuite évaluées en utilisant un taux d'acceptation de Green approprié. Nous introduisons un noyau de naissance et de mort dédié, à l'aide des mouvements générés par le filtre de Kalman. Des propositions de perturbations meilleures augmentent la probabilité de l'échantillonneur RJMCMC de choisir une telle perturbation qui, à son tour, conduit à une convergence plus rapide. Plus de détails peuvent être trouvés dans la partie 5.3.4.

- **La mise en œuvre paralléle du RJMCMC**. Une mise en œuvre paralléle est proposée. Aujourd'hui, les ordinateurs classiques ont généralement plus de deux unités de traitement et, par conséquent, les implémentations paralléles deviennent ubiquitaires. Une attention particulière doit être accordée afin de veiller à ce que les perturbations paralléles soient conditionnellement indépendante dans les deux dimensions spatiales ainsi que dans le temps. Une description détaillée de l'échantillonneur RJMCMC paralléle se trouve dans la partie 5.3.5.

Les deux approches d'optimisation sont fondamentalement différentes, mais elles ont le même but: une simulation plus rapide de la fonction d'énergie totale décrite précédemment.
Un aspect important de suivi d'objets multiples est non seulement la détection consécutive des cibles dans la séquence d'images, mais aussi l'étiquetage correct des cibles afin que des trajectoires cohérentes puissent être extraites. Dans notre contexte, l'étiquette de l'ellipse détermine la trajectoire à laquelle l'objet appartient. Nous avons analysé deux approches différentes pour réétiqueter les ellipses dans une configuration:

1. **Perturbations de fission et fusion des trajectoires**. Un mouvement de **fission** considère une trajectoire donnée et propose de la scinder en deux trajectoires différentes à un point de partage fixé. Un mouvement de **fusion** propose d'unir deux trajectoires distinctes et de réétiqueter les ellipses de sorte qu'elles forment une seule trajectoire plus longue.

2. **Étiquetage déterministe pendant les mouvements de naissance**. Cette approche de l'étiquetage consiste à attribuer l'étiquette d'une nouvelle ellipse sur la base de son voisinage. Quand une nouvelle ellipse est créé, une recherche est effectuée afin de trouver l'objet le plus proche dans son voisinage, et la valeur de l'étiquette de la nouvelle ellipse est mise égale à celle de l'ellipse representant l'objet le plus proche. La vitesse de calcul obtenue couplée à la bonne performance résultante motivent l'utilisation d'une telle approche.

Une analyse en profondeur de tous les noyaux de perturbation utilisés est présentée dans les parties 5.3.1. et 5.3.2.
Résultats. Les modèles proposés ont été appliqués à une grande variété d'ensembles de données, qui peuvent être classés comme suit:

- **Les bancs d'essai synthétiques utilisés par la communauté de traitement d'images biologiques**. Plusieurs séquences ont été générées avec le logiciel libre Icy, issu de l'Unité d'Analyse Quantitative d'Images Biologiques de l'Institut Pasteur (UMR CNRS 3691), dirigée par J.-C. Olivo-Marin. Les séquences d'images contiennent differents niveaux de bruit et un nombre d'objets variable. différents niveaux de bruit et un nombre différent d'objets. Les objets peuvent entrer ou sortir de la région d'intérêt à tout moment et dans n'importe quel lieu. Ces ensembles de données ont été largement utilisés pour analyser les performances de la méthode, compte tenu de l'existence des informations de vérite terrain précis;

- **Des données biologiques réelles**. Une séquence de données biologiques réelles, fournie gracieusement par J. Salamero, PICT IBiSA, UMR 144 CNRS, Institut Curie ([Basset et al., 2014]), est utilisée pour montrer l'applicabilité de la méthode proposée pour des données réelles dans le contexte biologique;

- **Des données satellitaires simulées**. Ces ensembles de données sont des séquences d'images satellitaires réelles qui ont été modifiées par Airbus D&S pour ressembler à des images produites par un satellite géostationnaire. Une grande variété d'échantillons est explorée, des cameras statiques aux cameras en mouvement, des hautes fréquences temporelles aux des basses fréquences temporelles, d'un seul type d'objet à plusieurs types d'objets qui doivent être détectés et suivis au cours de la séquence d'images.

Les résultats pour chacune des catégories de données sont illustrés à partir d'un petit nombre d'exemples représentatifs. Plus précisément, puisque les bancs d'essai des données biologiques peuvent être générés avec la vérité terrain associée, les limites de l'approche proposée sont testées sur ces séquences. Tout les résultats sont décrits dans la partie 5.4.

Appendix D

Conclusions et perspectives

Contents

D.1 Contributions de la thèse

Au cours de cette thèse de doctorat, nous avons étudié comment la géométrie stochastique peuvent être efficacement utilisée pour l'analyse d'images et de vidéos à hautes résolution, prises par un capteur satellitaire optique ou par un microscope. Le but de ce travail était d'étudier la possibilité d'utiliser des processus ponctuels marqués pour détecter et suivre des objets mobiles multiples, où les cibles apparaisent petites dans l'image avec une taille habituelle de 10 - 100 pixels par objet. Les principales contributions de cette thèse sont:

1. **Un noveau modèle de processus de point marqué spatio-temporel pour la détection et le suivi d'objets en mouvement**. Nous avons développé un modèle intuitif fondé sur un solide bagage théorique qui intègre les contraintes sur la distribution spatiale et temporelle des objets. L'information temporelle nous a permis de restreindre la recherche des objets dans des zones qui présentent des changements significatifs dans des trames consécutives. Nous avons appliqué le modèle pour détecter et suivre divers types d'objets, y compris des voitures, des bateaux et des piétons dans les vidéos satellitaires, ainsi que des vésicules dans les séquences d'images de microscopie. Ces applications ont permis l'utilisation d'un modèle de mouvement de vitesse constante, simple mais suffisamment général. Nous avons utilisé les trajectoires obtenues pour déduire des informations de niveau supérieur telles que l'estimation de la densité du trafic (dans les zones urbaines par exemple);

2. **Un modèle de processus de point marqué spacial amélioré pour l'extraction des bateaux dans les ports**. Le comptage des bateaux dans les ports est une tâche très difficile en raison de la distribution particulière des

objets. Dans des ports, les bateaux sont ancrés très proches les uns des les autres. Par conséquent, deux ou plusieurs bateaux voisins sont généralement détectés comme un objet de grande taille. Ben Hadj et al. [2010a] ont proposé un modèle pour ce problème d'extraction lorsque tous les objets ont la même orientation. Nous construisons sur le modèle présenté par Ben Hadj et al. [2010a] et développons un modèle plus général, et efficace, pour compter les bateaux dans des ports indépendamment de leur orientation. En particulier, nous écartons les contraintes qui rendent le modèle de Ben Hadj et al. [2010a] trop restrictif et introduisons des heuristiques simples pour améliorer la qualité du résultat;

3. **Estimation automatique des paramètres**. La performance des modèles de processus de point marqué pour une application spécifique dépend d'un choix bien adapté et efficace des paramètres. Les modèles présentés dans cette thèse sont une combinaison linéaire de plusieurs termes d'énergie pondérés. Ces poids sont généralement difficiles à definir manuellement car ils manquent une interprétation physique directe. Nous avons développé une technique automatique d'estimation des paramètres fondée sur la programmation linéaire qui peut être efficacement utilisée pour définir les paramètres de ces modèles;

4. **Un échantilonneur RJMCMC intégré avec des mouvements de Kalman**. Les dynamiques de RJMCMC sont utilisées pour simuler les modèles présentés dans cette thèse. Le RJMCMC est un échantillonneur itératif qui utilise un ensemble de noyaux de perturbation pour simuler une chaîne de Markov qui converge vers la distribution souhaitée. Cet échantillonneur a de très bonnes propriétés de convergence et c'est pour cela qu'il est d'un grand intérêt pour la communauté scientifique. Néanmois, ces propriétes sont au prix d'un coût de calcul élevé. Nous avons proposé un nouveau échantillonneur hybride en intégrant l'échantilloneur RJMCMC avec des mouvement de type de Kalman. Les mouvements issus du filtre de Kalman sont utilisés pour inciter le chaîne de Markov à converger plus rapidement vers la distribution souhaitée;

5. **La mise en œuvre parallèle et efficace de l'échantillonneur RJMCMC**. Une tentative réussie pour accroître l'efficacité de calcul de l'échantilonneur RJMCMC a été faite par Verdié and Lafarge [2012]. Its ont proposé une mise en œuvre parallèle de l'échantilonneur sur GPU et obtenu de grandes améliorations en termes de temps de calcul sur la version séquentielle. Ils préconisent un système de partitionnement hiérarchique afin de diviser l'espace de l'image en cellules indépendantes et d'effectuer des perturbations indépendantes dans ces cellules. Par conséquent, des difficultés surgissent pour extraire correctement les objets au bord de la cellule. Nous avons proposé une nouvelle mise en œuvre multi-cœrs du RJMCMC qui maintient une mémoire partagée entre les processeurs de sorte que les perturbations dans une cellule sont réalisées en tenant compte de son viosinage.

Ce travail de recherche montre que les modèles de processus ponctuels marqués peuvent être utilisés à pour le suivi d'objets. Une analyse qualitative des résultats révèle des performances élevées de détection et de suivi pour toutes les vidéos présentées dans cette thèse. Les avantages de cette approche sont discutés dans la section suivante.

D.2 Advantages de méthode proposée

Les modèles présentés dans Chapitre 4 et Chapitre 5 permettent la détection et le suivi d'objets en mouvement et fournissent des informations de haut niveau sur les objets tels que leur trajectoire, leur taille, leur forme et leur orientation. Le terme d'énergie couplé avec le processus d'optimisation en batch permet la détection d'objets faiblement contrastés ou des objets temporairement occlues tout en maintenant des trajectoires cohérentes.
Cette approche est automatique et permet de structurer l'extraction des trajectoires sur la base des interactions entre les objets. Les interactions complexes entre les objets peuvent être efficacement intégrés dans les modèles.
Un avantage important de cette méthode est sa robustesse au type de bruit et qualité des données utilisées. En particulier, l'apprentissage des paramètres est effectuée une seule fois pour chaque ensemble de données. Le modèle peut alors être appliquée avec succès à chaque séquence dans cet ensemble de données. Nous expérimentons également avec l'aide des mêmes paramètres sur plusieurs ensembles de données similaires et montrons que la qualité du résultat n'est pas significativement affectée.
Bien que les avantages d'un tel procédé soient solides, plusieurs inconvénients ont été identifiés au cours de ce travail de thèse.

D.3 Inconvénients des méthodes proposées

Tout d'abord, l'efficacité de calcul demeure un aspect problématique. Le cadre mathématique des processus ponctuels marqués dans lequel ce travail est développé vient avec une charge de calcul intrinsèque. L'utilisation d'heuristiques efficaces que nous avons proposée dans ce travail afin de réduire l'espace de recherche et d'accélérer le processus d'optimisation a considérablement réduit le temps de calcul. Néanmoins, dans une ère de traitement en temps réel, ces méthodes ne parviennent pas à suivre ce rythme. Dans les paramètres actuels tels que présentés dans cette thèse, la performance en temps réel peut être obtenue que pour les vidéos de basse fréquence avec de petites images et un nombre limité d'objets. Par conséquent, des améliorations considérables doivent être réalisées à cet égard pour rendre les modèles de processus de point marqué appropriés pour le suivi d'objets multiples à l'avenir.
Un autre inconvénient des modèles proposés est dû au modèle d'ellipse brillante utilisé pour caract´eriser une cible. Ce modèle de cible est suffisament simple pour être efficacement incorporé dans le modèle de processus de point. De plus, pour les applications présentées dans cette thèse un modèle simple de cible suffit pour

obtenir des résultats de haute qualité. Cependant, l'utilisation d'un tel modèle se révèle être limitatif en termes d'applications. Par conséquent, un modèle de cible plus élaboré qui puisse intégrer plus d'information sur l'apparence des cibles serait souhaitable.

D.4 Perspectives

Dans cette thèse, nous avons modélisé le problème de suivi au niveau des objets. Les interactions entre ces objets sont décrites à la fois dans le domaine spatial et temporel. L'information sémantique à propos de trajectoires d'objets est ensuite récupérée par le regroupement des objets à partir de leurs étiquettes. Une approche intéressante serait de concevoir un modèle hiérarchique qui intègre à la fois les contraintes de bas niveau entre les objets individuels et les contraintes de haut niveau entre les trajectoires.
La marque géométrique d'un point dans le processus ponctuel marqué dépend de l'application considérée. Les formes simples telles que des rectangles ou des ellipses sont plus souhaitables pour caractériser la géométrie des objets car ils sont décrits par un petit nombre de paramétres. En particulier, nous avons choisi d'utiliser uniquement des ellipses au long de ce travail parce qu'elles intègrent la plupart des objets d'intérêt (par exemple, les voitures, les bateaux, les vésicules, etc.). Néanmoins, l'évolution vers l'utilisation de plusieurs formes serait souhaitable. Lafarge et al. [2010a] proposent un processus multi-marqué pour extraire des objets de différentes formes dans une image. Une extension pour les données vidéo peut être envisagée pour faire la distinction entre différentes classes d'objets. Cependant, des précautions doivent être prises pour préserver la classe d'un objet tout au long de sa trajectoire.
Nous avons conçu nos modèles pour extraire des trajectoires individuelles des objets en mouvement. Cependant, dans certaines applications telles que la surveillance de la circulation, des informations sur la densité de la circulation peuvent être plus désirables que les trajectoires elles mêmes. La notion de circulation ici ne se limite pas aux véhicules, elle peut également inclure les piétons et le trafic maritime pour les données satellitaires ou la circulation près de la membrane cellulaire dans les images de microscopie. Un modèle pour estimer la densité des objets tout au long de la vidéo, plutôt que des trajectoires individuelles peut être intéressant dans de telles applications.
Enfin, nous avons proposé plusieurs façons d'améliorer la performance de l'échantilonneur RJMCMC. Même ainsi, le processus d'optimisation devrait être encore amélioré pour rendre ces modèles compétitifs avec l'état de l'art existant sur des algorithmes de suivi en temps réel.

D.5 Conclusions

Ce manuscrit présente seulement une partie des travaux entrepris au cours des ces trois dernières années. Nous avons eu l'occasion d'explorer des approches et théories différentes que celles décrites ici, d'expérimenter avec une grande quantité de données et de proposer systématiquement de nouvelles solutions à partir des impasses et tentatives de modélisation échouées. En tant que telles, ces dernières années ont été très instructives.

Cette étude nous a permis d'évaluer le potentiel des modèles de processus de point marqué pour le suivi d'objets. Nous avons montré que ces modèles sont de puissants concurrents par rapport à l'état de l'art eu égard à la qualité des résultats. Pourtant, des améliorations constantes doivent être faites concernant la procédure d'optimisation afin de faire des modèles de processus de point marqué un choix de premier plan dans l'avenir.

Le vaste travail de recherche actuellement mené dans le monde sur la détection et le suivi d'objets dans des images satellitaires ainsi que dans des images de microscopie a un impact majeur sur notre vie quotidienne. De la sécurité accrue et la surveillance de la faune à l'analyse de l'activité sub-cellulaire, la détection et le suivi d'objets dans ces domaines sont l'un des blocs de construction d'une compréhension plus profonde de notre monde et des motifs qui le régissent.

Bibliography

S. Agarwal and D. Roth. Learning a sparse representation for object detection. *Proc. ECCV*, pages 113–130, 2002. (Cited on page 23.)

G.R. Andrews. *Foundations of multithreaded, parallel and distributed programming.* Addison-Wesley, 1999. (Cited on page 78.)

M. Antonini. *Compression des images et des viéos numériques. Dix années des recherches au CNRS.* Habilitation à diriger des recherches (HDR), Université de Nice - Sophia Antipolis, 2003. (Cited on pages 14 and 156.)

Y. Bar-Shalom and T. Fortmann. *Tracking and Data Association.* Academic Press, San Diego, 1988. (Cited on page 117.)

A. Basset, J. Boulanger, P. Bouthemy, C. Kervrann, and J. Salamero. SLT-LoG: A Vesicle Segmentation Method with Automatic Scale Selection and Local Thresholding Applied to TIRF Microscopy. *ISBI - 2014 IEEE International Symposium on Biomedical Imaging*, pages 533–536, 2014. (Cited on pages 2, 6, 121, 125, 126 and 170.)

M.A. Beaumont. Approximate Bayesian Computation in evolution and ecology. *Annu. Rev. Ecol. Evol. Syst.*, 41:379–406, 2014. (Cited on page 42.)

M.A. Beaumont, A.W. Zhang, and D.J Balding. Approximate Bayesian Computation in population genetics. *Genetics*, 162(4):2025–2035, 2002. (Cited on page 42.)

E. Belikov, P. Deligiannis, P. Totoo, M. Aljabri, and H.-W. Loidl. A survey of high-level parallel programming models. Technical report, Heriot-Watt University, 2013. (Cited on page 78.)

S. Ben Hadj, F. Chatelain, X. Descombes, and J. Zerubia. Parameter estimation for a marked point process within a framework of multidimensional shape extraction from remote sensing images. *Proc. of ISPRS Conference on Photogrammetry Computer Vision and Image Analysis*, XXXVIII, 2010a. (Cited on pages 18, 19, 24, 63, 64, 147, 148, 161, 162 and 172.)

S. Ben Hadj, F. Chatelain, X. Descombes, and J. Zerubia. Estimation des paramètres de modèles de processus ponctuels marqués pour l'extraction d'objets en imagerie spatiale et aérienne haute résolution. Research Report RR-7350, INRIA, 2010b. (Cited on pages 18 and 160.)

J. Berclaz, F. Fleuret, and P. Fua. Multiple object tracking using flow linear programming. *IEEE Workshop on Performance Evaluation of Tracking and Surveillance*, pages 1–8, 2009. (Cited on page 104.)

G. Bertorelle, A. Benazzo, and S. Mona. ABC as a flexible framework to estimate demography over space and time: Some pros, some cons. *Mol. Ecol.*, 19:2609–2625, 2010. (Cited on page 42.)

U.M. Bhangale and S.S. Durbha. High performance SIFT feature classification of VHR satellite imagery for disaster management. *Proc. IGARSS*, pages 3574–3577, 2014. (Cited on pages 11 and 154.)

I. Biederman. Recognition-by-components: a theory of human image understanding. *Psychol. Rev.*, 94:115–147, 1987. (Cited on page 23.)

A. Boisbunon and J. Zerubia. Estimation of the weight parameter with SAEM for marked point processes applied to object detection. *Proc. EUSIPCO*, pages 2185–2189, 2014. (Cited on page 44.)

T. Bouwmans. Recent advanced statistical background modeling for foreground detection: a systematic survey. *Recent patents on computer science*, 4(3):147–176, 2011. (Cited on page 25.)

L. Breiman. Random forests. *Machine learning*, 45(1):5–32, 2001. (Cited on page 26.)

A. Bugeau and P. Perez. Track and cut: simultaneous tracking and segmentation of multiple objects with graph cuts. *EURASIP*, 2008(3):603–619, 2008. (Cited on page 26.)

P. Caccetta, S. Collings, K. Hingee, D. McFarlane, and W. Xiaoliang. Fine-scale monitoring of complex environments using remotely sensed aerial, satellite and other spatial data. *Proc. of ISIDF*, pages 1–5, 2011. (Cited on pages 12 and 154.)

J. Canny. A computational approach to edge detection. *IEEE Transactions of PAMI*, 8(6):679–698, 1986. (Cited on page 68.)

X. Cao, J. Lan, P. Yan, and X. Li. KLT feature based vehicle detection and tracking in airborne videos. *Proc. ICIG*, 2011. (Cited on pages 21 and 25.)

G. Celeux and J. Diebolt. The SEM algorithm: a probabilistic teacher algorithm derived from the EM algorithm for the mixture problem. *Computational statistics quarterly*, 55(4):287–314, 1985. (Cited on pages 43 and 73.)

S. Challa, M.R. Morelande, D. Musicki, and R.J. Evans. *Fundamentals of object tracking*. Cambridge University Press, 2011. (Cited on page 28.)

B. Chapman, G. Jost, and R. van der Pas. *Using OpenMP: portable shared memory parallel programming*. MIT Press, 2007. (Cited on page 78.)

S.A. Cheraghi and U.U. Sheikh. Moving object detection using image registration for a moving camera platform. *Proc. ICCSCE*, 2012. (Cited on pages 21, 25 and 26.)

J.C. Comiso, C.L. Parkinson, R. Gersten, and L. Stock. Accelerated decline in the Arctic sea ice cover. *Geophysical research letters*, 35(1), 2008. (Cited on pages 11 and 153.)

K. Csilléry, M.G.B. Blum, O.E. Gaggiotti, and O. François. Approximate Bayesian Computation ABC in practice. *Trends Ecol. Evol.*, 25:410–418, 2010. (Cited on page 42.)

N. Dalal and B. Triggs. Histograms of oriented gradients for human detection. *Proc. CVPR*, 1:886–893, 2005. (Cited on pages 22 and 127.)

G.B. Dantzig. Linear programming. *Proc. of Symposium on Modern Calculating Machinery and Numerical Methods*, 15:18–21, 1948. (Cited on page 103.)

G.B. Dantzig. *Linear programming and extensions*. Princeton University Press, 1963. (Cited on page 103.)

P. M. Dare. Shadow analysis in high-resolution satellite imagery of urban areas. *Photogrammetric Engineering and Remote Sensing*, 71:169–177, 2005. (Cited on page 81.)

S.J. Davey, M.G. Rutten, and B. Cheung. A comparison of detection performance for several track-before-detect algorithms. *Proc. FUSION*, pages 493–500, 2008. (Cited on page 26.)

F. De Chaumont, S. Dallongeville, N. Chenouard, N. Herve, S. Pop, et al. Icy: an open bioimage informatics platform for extended reproducible research. *Nature Methods*, 9(7):690–696, 2012. (Cited on page 123.)

F. de Chaumont et al. Icy: an open bioimage informatics platform for extended reproducible research. *Nature methods*, 9:690–696, 2012. URL `http://icy.bioimageanalysis.org`. (Cited on pages 1, 2, 122 and 126.)

F. Dell'Acqua, C. Bignami, M. Chini, G. Lisini, D.A. Polli, and S. Stramondo. Earthquake damages rapid monitoring by satellite remote sensing data: L'Aquila, April 6th, 2009 event. *JSTARS*, 4(4):935–943, 2011. (Cited on pages 11 and 154.)

P. DelMoral, A. Boucet, and A. Jasra. An adaptive sequential Monte Carlo method for approximate Baysian computation. *Stat. Comput.*, 22:1009–1020, 2011. (Cited on page 42.)

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*, 39(1):1–38, 1977. (Cited on pages 41 and 168.)

R. Deriche. Using Canny's criteria to derive a recursively implemented optimal edge detector. *IJCV*, pages 167–187, 1987. (Cited on page 68.)

S. Descamps, X. Descombes, A. Bèchet, and J. Zerubia. Automatic flamingo detection using multiple birth and death process. *Proc.of ICASSP*, 2008. (Cited on pages 13, 18, 24, 155 and 161.)

X. Descombes. *Modèles stochastiques en analyse d'image: des champs de Markov aux processus ponctuels marqués.* Habilitation à diriger des recherches (HDR), Université de Nice - Sophia Antipolis, 2004. (Cited on pages 16 and 159.)

X. Descombes, R. Minlos, and E. Zhizhina. Object extraction using a stochastic birth-and-death dynamics in continuum. *JMIV*, 33:347–359, 2009. (Cited on page 58.)

X. Descombes, F. Chatelain, F. Lafarge, C. Lantuejoul, C. Mallet, M. Minlos, M. Schmitt, M. Sigelle, R. Stoica, and E. Zhizhina. *Stochastic Geometry for Image Analysis.* John Wiley and Sons, 2011. (Cited on pages 13, 38, 39, 48, 59, 60, 74 and 155.)

B. Deylon, M. Laveille, and E. Moulines. Convergence of a stochastic approximation version of the EM algorithm. *The Annals of Statistics*, 27(1):94–128, 1999. (Cited on page 44.)

P.J. Diggle and R.J. Gratton. Monte Carlo methods of inference for implicit statistical models. *Journal of the Royal Statistical Society, Series B*, 46:193–227, 1984. (Cited on page 42.)

P. Dollar, Z. Tu, P. Perona, and S. Belongie. Integral channel features. *Proc. BMVC*, 2009. (Cited on page 22.)

A. Doucet, N. de Freitas, and N. Gordon. *Sequential Monte Carlo methods in practice.* Springer Verlag, 2001. (Cited on page 40.)

O. Erdinc, P. Willet, and Y. Bar-Shalom. A physical-shape approach for the probability hypothesis density and cardinalized probability hypothesis density filters. *Proc. SPIE*, 6236, 2006. (Cited on page 31.)

R.M. Feldman and V.-F. Ciriaco. *Applied probabilities and stochastic processes.* Springer Science and Business Media, 2009. (Cited on page 40.)

I. Foster. *Designing and building parallel programs: Concepts and tools for parallel software engineering.* Addison-Wesley, 1995. (Cited on page 78.)

P.T. Fretwell, M.A. LaRue, P. Morin, G.L. Kooyman, B. Wienecke, et al. An emperor penguin population estimate: the first global, synoptic survey of a species from space. *PLOS One*, 2012. (Cited on pages 11 and 153.)

Y. Freund and R.E. Schapire. A decision-theoretic generalization of on-line learning and an application to boosting. *Journal of Computer and System Sciences*, 55(1): 119–139, 1997. (Cited on pages 22 and 26.)

C. Fuchs. Social networking sites and the surveillance society. Research report, University of Salzburg, 2009. (Cited on pages 11 and 153.)

A. Gamal-Eldin, X. Descombes, and J. Zerubia. Multiple birth and cut algorithm for point process optimization. *Proc. SITIS*, pages 35–42, 2010. (Cited on page 58.)

A. Gamal Eldin, X. Descombes, G. Charpiat, and J. Zerubia. A fast multiple birth and cut algorithm using belief propagation. *Proc. of ICIP*, pages 2813–2816, 2011. (Cited on page 58.)

M.L. Garcia. *Design and evaluation of physical protection systems.* Butterworth-Heinemann, 2007. (Cited on pages 12 and 154.)

A. Gaszczak, T. Breckon, and J. Han. Real-time people and vehicle detection from UAV imagery. *Intelligent Robots and Computer Vision: Algorithms and Techniques*, 7878, 2011. (Cited on page 22.)

A. Gelman, W.R. Gilks, and G.O. Roberts. Weak convergence and optimal scaling of random walk metropolis algorithms. *Annals of Applied Probability*, 7(1):110–120, 1997. (Cited on page 143.)

A. Gelman, J. Hwang, and A. Vehtari. Understanding predictive information criteria for Bayesian models. *Statistics and Computing*, 24(6):997–1016, 2014. (Cited on pages 17 and 160.)

S. Geman and C. Huang. Diffusion for global optimization. *SIAM Journal of Control and Optimization*, 24(5):131–143, 1986. (Cited on pages 59 and 60.)

C. J. Geyer. Likelihood inference for spatial point processes. In *Stochastic geometry, likelihood and computation*. CRC Press, 1999. (Cited on page 41.)

A.R. Gonzalez. *Compression based analysis of image artifacts: applications to satellite images.* PhD thesis, Telecom ParisTech, 2013. (Cited on pages 13, 14 and 156.)

R.C. Gonzalez and R.E. Woods. *Digital Image Processing, 3rd edition.* Prentice Hall, 2008. (Cited on pages 13 and 156.)

F. Goudail, P. Réfrégier, and G. Delyon. Bhattacharyya distance as a contrast parameter for statistical processing of noisy optical images. *Journal of Optical Science of America A*, 21(7):1231–1240, 2004. (Cited on page 66.)

J. Goutsias, R. Mahler, and H. Nguyen. *Random sets: theory and applications.* Springer Verlag, 2012. (Cited on page 30.)

P. Green. Reversible jump Markov Chain Monte Carlo computation and Bayesian model determination. *Biometrika*, 82(4):711–732, 1995. (Cited on pages 45, 47, 48, 54, 59, 74, 106, 109 and 168.)

U. Grenander and M. Miller. Representations of knowledge in complex systems. *Journal of Royal Statistical Society*, 56(4):1–33, 1994. (Cited on page 58.)

W. Gropp, E. Lusk, and R. Thakur. *Using MPI: Portable parallel programming with message-passing interface*. MIT Press, 1999. (Cited on page 78.)

F. Han, Z. Tu, and S.-C. Zhu. Range image segmentation by an effective jump-diffusion method. *IEEE Trans. PAMI*, 26(9):1138–1153, 2004. (Cited on page 59.)

W.K. Hastings. Monte Carlo sampling methods using Markov chains and their applications. *Biometrika*, 57:97–109, 1970. (Cited on pages 45 and 59.)

A.J. Haug. *Bayesian estimation and tracking: a practical guide*. Wiley, 2012. (Cited on page 117.)

R. Hoseinnezhad, B.-N. Vo, B.-T. Vo, and D. Sutter. Visual tracking of numerous targets via multi-Bernoulli filtering of image data. *Pattern Recognition*, 45:3625–3635, 2012. (Cited on page 26.)

A.W.N. Ibrahim, P.W. Ching, G. Seet, M. Lau, and W. Czajewski. Moving objects detection and tracking framework for UAV-based surveillance. *Proc. Fourth Pacific-Rim Symposium on Image and Video technology*, 2010. (Cited on pages 28 and 29.)

Y. Iwashita, A. Stoica, and R. Kurazume. People identification using shadow dynamics. *Proc. ICIP*, 2010. (Cited on page 27.)

W. Jank. Implementing and diagnosing the stochastic approximation EM algorithm. *Journal of Computation and Graphical Statistics*, 15(4):803–829, 2006. (Cited on page 44.)

H. Jiang, S. Fels, and J.J. Little. A linear programming approach for multiple object tracking. *Proc. of CVPR*, pages 1–8, 2007. (Cited on page 104.)

P. Jing and Y. Danping. Remote sensing monitoring system for maritime search and rescue. *Proc. SoCPaR*, pages 101–106, 2011. (Cited on pages 11 and 154.)

R. Jones, B. Ristic, N. Redding, and D.M. Booth. Moving target indication and tracking from moving sensors. *Proc. DICTA*, 2005. (Cited on pages 25 and 28.)

O. Kallenberg. *Random measures*. Academic Press London, 1983. (Cited on page 36.)

R.E. Kalman. A new approach to linear filtering and prediction problems. *Trans. ASME - Journal of Basic Engineering*, 82(Series D):35–45, 1960. (Cited on pages 29 and 117.)

J. Kang, I. Cohen, G. Medioni, and C. Yuan. Detection and tracking of moving objects from a moving platform in presence of strong parallax. *Proc. ICCV*, 2005. (Cited on page 28.)

M.A. Keck, L. Galup, and C. Stauffer. Real-time tracking of low-resolution vehicles for wide-area persistent surveillance. *Proc. WACV*, 2013. (Cited on page 25.)

Khronos. OpenCL 1.1 Specification. http://www.khronos.org/registry/cl/specs/opencl-1.1.pdf, 2013. Accessed: Nov. 2013. (Cited on page 79.)

G. Kitagawa. Non-Gaussian state space modeling of non-stationarity. *Journal of American Statistical Association*, 82(400):1032–1041, 1987. (Cited on page 29.)

T.C. Koopmans. *Actively analysis of production and allocation.* John Wiley and Sons, 1951. (Cited on page 103.)

A.H. Kydd and B.F. Walter. The strategies of terrorism. *International security*, 31 (1):49–79, 2006. (Cited on pages 11 and 153.)

C. Lacoste, X. Descombes, and J. Zerubia. Point processes for unsupervised line network extraction in remote sensing. *IEEE Trans. PAMI*, 27(10):1568–1579, 2005. (Cited on pages 13, 24 and 155.)

F. Lafarge, G. Gimel'Farb, and X. Descombes. Geometric feature extraction by a multi-marked point process. *IEEE Trans. PAMI*, 32(9):1597–1609, 2010a. (Cited on pages 59, 150 and 174.)

F. Lafarge, R. Keriven, M. Bredif, and H. Vu. Hybrid multi-view reconstruction by jump-diffusion. *Proc. CVPR*, pages 350–357, 2010b. (Cited on page 59.)

S. Landau. *Surveillance or Security? The risks posed by new wiretapping technologies.* The MIT Press, 2011. (Cited on pages 11 and 153.)

D. Lavigne, S. Sahli, Y. Ouyang, and Y. Sheng. Unsupervised classification and clustering of image features for vehicle detection in large scale aerial images. *Proc. FUSION*, 2010. (Cited on page 21.)

E.L. Lehmann and J. P. Romano, editors. *Testing statistical hypotheses, 3rd edition.* Springer, 2008. (Cited on pages 97 and 98.)

B. Leibe, K. Schindler, N. Cornelis, and L. van Gool. Coupled object detection and tracking from static cameras and moving vehicles. *IEEE Trans. PAMI*, 30(10): 1683–1698, 2008. (Cited on pages 24 and 108.)

Y. Li and R. Briggs. Automatic extraction of roads from high resolution aerial and satellite images with heavy noise. *Proc. of the 6th International Conference on Geographic Information Systems*, 2009. (Cited on pages 68 and 69.)

Y. Li, C. Huang, and R. Nevatia. Learning to associate: HybridBoosted multi-target tracker for crowded scene. *Proc. CVPR*, 2009. (Cited on pages 28 and 29.)

Z. Liu, C. Chen, X. Shen, and X. Zou. Detection of small objects in image data based on the nonlinear principal component analysis neural network. *SPIE Optical Engineering*, 44, 2005. (Cited on page 26.)

N.K. Logothetis and D.L. Sheinberg. Visual object recognition. *Annu. Rev. Neurosci.*, 19:577–621, 1996. (Cited on page 23.)

T.I. Lukowski and F.J. Charbonneau. Synthetic aperture radar and seach and rescue. *Proc. IGARSS*, 5:2374–2376, 2000. (Cited on pages 11 and 154.)

P. Luo, F. Liu, X. Liu, and Y. Yang. Stationary vehicle detection in aerial surveillance with a UAV. *Proc. ICIDT*, 2012. (Cited on pages 21 and 26.)

R. Mahler. Multitarget Bayes filtering via first-order multi target moments. *IEEE Trans. on Aerospace and Electronic Systems*, 39(4):1152–1178, 2003. (Cited on page 30.)

R. Mahler. *Statistical Multisource-Multitarget Information Fusion*. Norwood, MA, USA: Artech House, 2007a. (Cited on page 31.)

R. Mahler. PHD filters of higher order in target number. *IEEE Trans. on Aerospace and Electronic Systems*, 43(4):1523–1543, 2007b. (Cited on page 31.)

R. Mahler. *Advances in Statistical Multisource-Multitarget Information Fusion*. Norwood, MA, USA: Artech House, 2014. (Cited on page 31.)

J.-M. Marin. *Bayesian core: a practical approach to computational Bayesian statistics*. Springer, 2007. (Cited on pages 17 and 160.)

J. Mateu and F. Montes. Pseudo-likelihood inference for Gibbs processes with exponential families through generalized linear models. *Statistical Inference for Stochastic Processes*, 4(2):125–154, 2001. (Cited on page 42.)

T.G. Mattson, B.A. Sanders, and B. Massingill. *Patterns for parallel programming*. Addison-Wesley, 2005. (Cited on pages 77 and 78.)

N. McLaughlin, J. Martinez Del Ricon, and P. Miller. Enhancing linear programming with motion modeling for multi-target tracking. *Proc. of WACV*, pages 71–77, 2015. (Cited on page 104.)

R. Merton. Option pricing when underlying stock returns are discontinuous. *Journal of Financial Economics*, 3:125–144, 1976. (Cited on page 58.)

N. Metropolis, M. Rosenbluth, A. Teller, and E. Teller. Equations of state calculations by fast computing machines. *Journal of Chemical Physics*, 21:1087–1091, 1953. (Cited on pages 45 and 60.)

A. Milan, S. Roth, and K. Schindler. Continuous energy minimization for multitarget tracking. *IEEE Trans. PAMI*, 36(1):58–72, 2014. (Cited on pages 1, 122 and 123.)

T. Müller and M. Müller. CARTIV: Improving camouflage assessment with assistance methods. *Proc. of SPIE*, 7662, 2010. (Cited on pages 14 and 157.)

A.N. Netravali and B.G. Haskell. *Digital pictures: Representation, Compression and Standards*. Springer, 1995. (Cited on pages 14 and 156.)

T.T. Nguyen, H. Grabner, H. Bischof, and B. Gruber. On-line boosting for car detection from aerial images. *Proc. RIVF*, 2007. (Cited on page 22.)

NVidia. CUDA Zone. `http://www.nvidia.com/object/cuda_home_new.html`, 2013. Accessed: Nov. 2013. (Cited on page 78.)

S. Oh, S. Russell, and S. Sastry. Markov Chain Monte Carlo Data Association for general multiple-target tracking problems. *Proc. CDC*, 1:735–742, 2004. (Cited on page 29.)

OpenMP. OpenMP 3.0 Specification. `http://www.openmp.org/mp-documents/spec30.pdf`, 2013. Accessed: June 2013. (Cited on page 78.)

M. Ortner. *Processus ponctuels marqués pour l'extraction automatique de caricatures de batiments à partir de modèles numériques d'élévation.* PhD thesis, Université de Nice - Sophia Antipolis, 2004. (Cited on pages 13, 18, 24, 48, 51, 52, 75, 155 and 161.)

M. Ortner, X. Descombes, and J. Zerubia. An adaptive simulated annealing cooling schedule for object detection in images. Research Report 6336, INRIA, 2007. (Cited on page 61.)

M. Ortner, X. Descombes, and J. Zerubia. A marked point process of rectangles and segments for automatic analysis of Digital Elevation Models. *IEEE Transactions of Pattern Analysis and Machine Intelligence*, 30:105–119, 2008. (Cited on pages 13, 69 and 155.)

N. Otsu. A threshold selection method from gray level histograms. *IEEE Trans. Systems, Man and Cybernetics*, 9:62–66, 1979. (Cited on page 80.)

M. Pace. *Stochastic models and methods for multi-object tracking.* PhD thesis, Université Sciences et Technologies, Bordeaux, 2011. (Cited on page 31.)

P.S. Pacheco. *Parallel programming with MPI.* Morgan Kaufmann, 1996. (Cited on page 78.)

C. Papageorgiou and T. Poggio. A trainable system for object detection. *IJCV*, 38: 15–33, 2000. (Cited on page 22.)

F. Papangelou. The conditional intensity of general point processes and an application to line processes. *Zeitschrift für Wahrscheinlichkeitstheorie und Verwandte Gebiete*, 28(3):207–226, 1974. (Cited on page 36.)

F. Papi, B.T. Vo, M. Bocquel, and B.N. Vo. Generalized labeled multi-Bernoulli approximation of multi-object densities. *IEEE Trans. Signal Processing*, 63(20): 5487–5497, 2015. (Cited on pages 26 and 31.)

M. Pelillo and E.R. Hancock, editors. *Energy minimization methods in computer vision and pattern recognition: International Workshop EMMCVPR 1997.* Springer, 1997. (Cited on page 40.)

A.G.A. Perera, C. Srinivas, A. Hoogs, G. Brooksby, and W. Hu. Multi-object tracking through simultaneous long occlusions and split-merge conditions. *Proc. of CVPR*, 2006. (Cited on pages 21, 25, 28 and 29.)

P. Pérez. *Modèles et algorithmes pour l'analyse probabiliste des images*. Habilitation à diriger des recherches (HDR), Université de Rennes 1, 2003. (Cited on pages 16 and 159.)

G. Perrin, X. Descombes, and J. Zerubia. A marked point process model for tree crown extraction in plantations. *Proc. ICIP*, 1:661–664, 2005. (Cited on pages 13, 17, 18, 24, 155, 160 and 161.)

M. Piccardi. Background subtraction techniques: a review. *Proc. ICSMC*, pages 3099–3104, 2004. (Cited on page 25.)

N.G. Platonov, I.N. Mordvintsev, and V.V. Rozhnov. The possibility of using high resolution satellite imagery for detection of marine mammals. *Bio Bull*, 40:197–205, 2013. (Cited on pages 11 and 153.)

J.M. Poland, editor. *Understanding terrorism: groups, strategies and responses*. Prentice Hall, 2010. (Cited on pages 11 and 153.)

J. Portilla, V. Strela, M.J. Wainwright, and E.P. Simoncelli. Image denoising using scale mixtures of Gaussians in the wavelet domain. *IEEE Trans. PAMI*, 12(11): 1338–1351, 2003. (Cited on pages 14 and 157.)

S. Praveena, G. Karthika, J. Arivukarasi, S. Anbarasan, and P. Rajeshwari. Recognize aircraft using template matching in high resolution satellite images. *IJARSET*, 2(3):518–521, 2015. (Cited on pages 3 and 23.)

N. Proia. *Surveillance maritime par analyse d'images satellitaires optiques panchromatiques*. PhD thesis, Université des Antilles-Guyane, 2010. (Cited on pages 23 and 97.)

J. Prokaj and G. Medioni. Persistent tracking for wide area aerial surveiilance. *Proc. CVPR*, 2014. (Cited on pages 24 and 29.)

V. Reily, H. Idrees, and M. Shah. Detection and tracking of large number of targets in wide area surveillance. *Proc. ECCV*, Lecture Notes in Computer Science (LNCS):186–199, 2010. (Cited on pages 21 and 27.)

B. Ristic, S. Arulampalam, and N. Gordon. *Beyond the Kalman filter: Particle filters for tracking applications*. Artech House Inc., 2004. (Cited on page 26.)

H. Robbins and S. Monro. A stochastic approximation method. *The Annals of Mathematical Statistics*, 22(3):400–407, 1951. (Cited on page 44.)

C. P. Robert and G. Casella. *Monte Carlo Statistical Methods (Springer Texts in Statistics)*. Springer-Verlag New York, Inc., 2005. (Cited on pages 45, 73 and 108.)

G.O. Roberts and J.S. Rosenthal. Optimal scaling for various metropolis-hastings algorithms. *Statistical Science*, 16(4):351–367, 2001. (Cited on page 143.)

I. Saleemi and M. Shah. Multiframe many-many point correspondence for vehicle tracking in high density wide area aerial videos. *IJCV*, 104(2):198–219, 2013. (Cited on pages 21, 25 and 28.)

M. Schmitt and J. Mattioli. *Morphologie mathématique*. Transvalor-Pressess des Mines, 2013. (Cited on page 95.)

L. Shao, R. Yan, X. Li, and Y. Liu. From heuristic optimization to dictionary learning: a review and comprehensive comparison of image denoising algorithms. *IEEE Trans. Cybernetics*, 44(4):1001–1012, 2014. (Cited on pages 14 and 157.)

A.C. Shastry and R.A. Schozengerdt. Airborne video registration and traffic-flow parameter estimation. *IEEE Trans. on Intelligent Transportation Systems*, 6(4): 391–405, 2005. (Cited on page 25.)

H. Shen, S. Li, C. Zhu, H. Chang, and J. Zhang. Moving object detection in aerial video based on spatiotemporal saliency. *Chinese Journal of Aeronautics*, 26(5): 1211–1217, 2013. (Cited on page 29.)

M. Siam and M. ElHelw. Robust autonomous visual detection and tracking of moving targets in UAV imagery. *Proc. of ICSP*, 2012. (Cited on pages 21, 26 and 29.)

S. Singh and R. Talwar. Effects of topographic corrections on MODIS sensor satellite imagery of mountains region. *Proc. of ICSC*, pages 455–460, 2013. (Cited on pages 12 and 154.)

A.W.M. Smeulders, D.M. Chu, R. Cucchiara, S. Cakderara, A. Deghghan, and M. Shah. Visual tracking: an experimental survey. *IEEE Trans. PAMI*, 2013. (Cited on page 28.)

A. Srivastava, M. Miller, and U. Grenander. Multiple target direction of arrival tracking. *IEEE Trans. IP*, 43(5):282–285, 1995. (Cited on page 58.)

A. Srivastava, U. Grenander, G. Jensen, and M. Miller. Jump-diffusion markov processes on orthogonal groups for object pose estimation. *Journal on Statistical Planning and Inference*, 103(1/2):15–37, 2002. (Cited on page 58.)

C. Stauffer and W.E.I. Grimson. Adaptive background mixture models for real-time tracking. *Proc. of CVPR*, pages 246–252, 1999. (Cited on page 25.)

R.S. Stoica, P. Gregori, and J. Mateu. Simulated annealing and object point processes: tools for analysis of spatial patterns. *Stochastic Processes and Their Applications*, 115:1860–1882, 2005. (Cited on page 60.)

D. Stoyan and H. Stoyan. *Fractals, Random Shapes and Point Fields: Methods of Geometrical Statistics.* John Wiley and Sons, 1994. (Cited on pages 13, 74 and 155.)

D. Stoyan, W. S. Kendall, and J. Mecke. *Stochastic Geometry and its Applications.* John Wiley and Sons, 1987. (Cited on pages 13 and 155.)

J.C. Stroeve, V. Kattsov, A. Barrett, M. Serreze, T. Pavlova, M. Holland, and W.N. Meier. Trends in Arctic sea extent from CMIP5, CMIP3 and observations. *Geophysical Research Letters*, 39(16), 2012. (Cited on pages 11 and 153.)

R. Szelisky. *Computer vision: Algorithms and Applications.* Springer, 2011. (Cited on page 21.)

M. Taj and A. Cavallaro. Multi-camera track-before-detect. *Proc. ICSSC*, 2009. (Cited on page 26.)

K. Tanaka and H. Saji. Vehicle extraction from aerial images using voting process and frame matching. *Proc. IV*, 2007. (Cited on page 26.)

M. Teutsch. *Moving object detection and segmentation for remote aerial video surveillance.* PhD thesis, Karlsruher Institut fur Technologie, 2015. (Cited on pages 24, 26 and 28.)

P.W. Trinder, M.I. Cole, K. Hammond, H.-W. Loidl, and G.J. Michaelson. Resource analyses for parallel and distributed coordination. *Concurrency and computation: practice and experience*, pages 309–348, 2013. (Cited on page 78.)

S. Turmer, F. Kurz, P. Reinartz, and U. Stilla. Airborne vehicle detection in dense urban areas using HoG features and disparity maps. *JSTARS*, 6(6):2327–2337, 2013. (Cited on page 22.)

S. Ullman. *High-level vision: Object recognition and visual cognition.* MIT Press, 1996. (Cited on page 23.)

M.N.M. van Lieshout. Stochastic annealing for nearest-neighbor point processes with application to object recognition. *Advances in Applied Probability*, 26:281–300, 1994. (Cited on page 60.)

M.N.M. van Lieshout. *Markov Point Processes and Their Applications.* Imperial College Press, 2000. (Cited on page 41.)

M.N.M. van Lieshout and A.J. Baddeley. Extrapolating and interpolating spatial patterns. In A.B. Lawson and D.G.T. Denison, editors, *Spatial Clustering Modelling.* CRC Press/Chapman & Hall, 2002. (Cited on pages 13, 41 and 155.)

V. Vapnik. *Statistical learning theory.* Wiley, 1998. (Cited on page 26.)

Y. Verdie. *Modelisation de scènes urbaines à partir de données aeriennes.* PhD thesis, Université de Nice - Sophia Antipolis, 2013. (Cited on pages 3, 55 and 57.)

Y. Verdié and F. Lafarge. Efficient Monte Carlo sampler for detecting parametric objects in large scenes. *Proc. ECCV*, 7574:539–552, 2012. (Cited on pages 54, 55, 56, 75, 77, 79, 119, 120, 148 and 172.)

P. Viola and M. Jones. Rapid object detection using a boosted cascade of simple features. *Proc. CVPR*, 2001. (Cited on page 22.)

P. Viola and M. Jones. Robust real-time face detection. *IJCV*, 57(2):137–154, 2004. (Cited on page 22.)

B.-N. Vo and W.-K. Ma. The Gaussian Mixture Probability Density Filter. *IEEE Trans. Signal Processing*, 54(11):4091–4104, 2006. (Cited on page 31.)

B.-N. Vo, S.S. Singh, and A. Doucet. Sequential Monte Carlo implementation of the PHD filter for multi-target tracking. *Proc. Int. Conf. Information Fusion*, pages 792–799, 2003. (Cited on page 31.)

B.-N. Vo, S.S. Singh, and A. Doucet. Sequential Monte Carlo methods for multi-target filtering with random finite sets. *IEEE Trans. Aerospace and Electronic Systems*, 41(4):1224–1245, 2005. (Cited on page 31.)

B.-N. Vo, A. Pasha, and H.D. Tuan. A Gaussian Mixture PHD filter for nonlinear jump Markov models. *IEEE Conf. on Decision and Control*, 1:3162–3167, 2006. (Cited on page 31.)

B.-N. Vo, B.-T. Vo, and D. Phung. Labeled random finite sets and the Bayes multi-target tracking filter. *IEEE Trans. Signal Processing*, 62(24):6554–6567, 2014. (Cited on page 91.)

B.-T. Vo and B.-N. Vo. Labeled random finite sets and multi-object conjugate priors. *IEEE Trans. Signal Processing*, 61(13):3460–3475, 2013. (Cited on page 91.)

B.-T. Vo, B.-N. Vo, and A. Cantoni. The Cardinality Balanced Multi-target Multi-Bernoulli filter and its implementations. *IEEE Trans. Signal Processing*, 57(2): 409–423, 2009. (Cited on page 31.)

B.-T. Vo, B.-N. Vo, N.-T. Pham, and D. Suter. Joint detection and estilation of multiple objects from image observations. *IEEE Trans. Signal Processing*, 58(10): 5129–5141, 2010. (Cited on page 65.)

S. Voigt, T. Kemper, T. Riedlinger, R. Kiefl, K. Scholte, and H. Mehl. Satellite image analysis for disaster and crisis-management support. *IEEE. Trans. GRS*, 45(6):1520–1528, 2007. (Cited on pages 11 and 154.)

T. Vu, B.-N. Vo, and R.J. Evans. A particle marginal Metropolis-Hastings multi-target tracker. *IEEE Trans. Signal Processing*, 62(15):3953–3964, 2014. (Cited on page 91.)

G.C.G. Wei and M.A. Tanner. A Monte Carlo implementation of the EM algorithm and the poor man's data augmentation algorithms. *Journal of the American Statistical Association*, 85(141):699–704, 1990. (Cited on page 43.)

Y. Wei and L. Tao. Efficient histogram-based sliding window. *Proc. CVPR*, 2010. (Cited on page 22.)

G. Welch and G. Bishop. An introduction to the Kalman filter. *Proc. SIGGRAPH*, pages 19–24, 2001. (Cited on page 127.)

C. Wu, X. Cao, R. Lin, and F. Wang. Registration-based moving vehicle detection for low-altitude urban traffic surveillance. *Proc. WCICA*, 2010. (Cited on page 28.)

Y.N. Wu, Z Si, H. Gong, and S.C. Zhu. Learning active basis model for object detection and recognition. *IJCV*, pages 1–38, 2009. (Cited on page 28.)

J. Xiao, H. Cheng, H. Sawhney, and F. Han. Vehicle detection and tracking in wide field-of-view aerial video. *Proc. CVPR*, 2010. (Cited on pages 21, 25, 26 and 29.)

F. Yao, A. Sekmen, and M.J. Malkani. Multiple moving target detection, tracking and recognition from a moving observer. *Proc. ICIA*, 2008. (Cited on page 29.)

A. Yilmaz, O. Javed, and M. Shah. Object tracking: A survey. *ACM Computing Surveys*, 38(4), 2006. (Cited on page 28.)

N. Yokoya and A. Iwasaki. Object detection based on sparse representation and Hough voting for optical remote sensing imagery. *IEEE. JSTARS*, page accepted for publication on 26/01/2015, 2015. (Cited on pages 3, 23 and 24.)

Q. Yu and G. Medioni. Integrated detection and tracking for multiple moving objects using data-driven MCMC data association. *Motion and Video computing*, pages 1–8, 2008. (Cited on pages 28 and 29.)

Q. Yu and G. Medioni. Multiple-target tracking by spatio-temporal Monte Carlo Markov chain data association. *IEEE Trans. PAMI*, 31(12):2196–2210, 2009. (Cited on page 105.)

Q. Yu, G. Medioni, and I. Cohen. Multiple target tracking using spatio-temporal Markov chain Monte Carlo data association. *Proc. of CVPR*, pages 1–8, 2007. (Cited on page 29.)

T. Zajic and R. Mahler. A particle-systems implementation of the PHD multitarget tracking filter. *Signal Processing, Sensor Fusion and Target Recognition*, 5096: 291–299, 2003. (Cited on page 31.)

H. Zhang, D. Wipf, and Y. Zhang. Multi-image blind deblurring using a coupled adaptive sparse prior. *Proc. CVPR*, pages 1051–1058, 2013. (Cited on pages 14 and 157.)

W. Zhang, X. Sun, K. Fu, C. Wang, and H. Wang. Object detection in high-resolution remote sensing images using rotation invariant parts based model. *IEEE Geosci. Remote Sens. Lett.*, 11(1):74–78, 2014. (Cited on page 24.)

Z. Zheng, G. Zhou, Y. Wang, Y. Liu, et al. A novel vehicle detection method with high resolution highway aerial image. *JSTARS*, 6(6):2338–2343, 2013. (Cited on page 26.)

B. Zitovà and J. Flusser. Image registration methods: a survey. *Image and Vision Computing*, 21:977–1000, 2003. (Cited on pages 14 and 157.)

Géométrie stochastique pour la détection et le suivi d'objets multiples dans des séquences d'images haute résolution de télédétection

Abstract: Dans cette thèse, nous combinons les outils de la théorie des probabilités et de la géométrie stochastique pour proposer de nouvelles solutions au problème de la détection et le suivi d'objets multiples dans des séquences d'images haute résolution. Tout d'abord, nous développons un modèle de processus ponctuel marqué spatial pour détecter une classe prédéfinie d'objets (bateaux) en fonction de leurs caractéristiques visuelles et géométriques. Nous concevons une mise en oeuvre de multi-coeur de l'échantillonneur de MCMC à sauts réversibles. Nous montrons des résultats supérieurs à ceux fournis par des methodes proposées dans l'état de l'art et des temps de détection compétitifs.
Les très bons résultats obtenus nous ont motivés pour étendre ces modèles dans le domaine temporel et pour créer un cadre fondé sur des modèles de processus ponctuels marqués spatio-temporels pour détecter et suivre conjointement plusieurs objets dans des séquences d'images. Nous proposons l'utilisation de formes paramétriques simples pour décrire l'apparition de ces objets. Nous construisons de nouveaux modèles fondés sur des énergies dédiées constituées de plusieurs termes qui tiennent compte à la fois l'attache aux données et les contraintes physiques telles que la dynamique de l'objet, la persistance de la trajectoire et de l'exclusion mutuelle. Nous construisons un schéma d'optimisation approprié qui nous permet de trouver des minima locaux de l'énergie hautement non-convexe proposée qui soient proche de l'optimum global.
Comme la simulation de ces modèles requiert un coÃ»t de calcul élevé, nous portons notre attention sur les dernières mises en oeuvre de techniques de filtrage pour le suivi d'objets multiples, qui sont connues pour Ãªtre moins coÃ»teuses en calcul. Nous proposons un échantillonneur hybride combinant le filtre de Kalman avec l'échantillonneur MCMC à sauts réversibles. Des techniques de calcul de haute performance sont également utilisées pour augmenter l'efficacité de calcul de notre méthode. Nous fournissons une analyse en profondeur du cadre proposé, ainsi qu'une comparaison extensive avec des procédés de l'état de l'art sur la base de plusieurs métriques classiques de suivi d'objets et de l'efficacité de calcul. Cette analyse montre une très bonne performance de détection et de suivi ainsi que d'une complexité accrue des modèles. Des tests exhaustifs ont été menés à la fois sur des séquences d'images satellitaires haute résolution et sur des données de microscopie.

Mots-clés: Suivi d'objets multiples, détection d'objets, processus ponctuel marqué, filtre de Kalman, séquences d'images satellitaires, séquences des données de microscopie, haute résolution.

Stochastic geometry for automatic multiple object detection and tracking in remotely sensed high resolution image sequences

Abstract: In this thesis, we combine the methods from probability theory and stochastic geometry to put forward new solutions to the multiple object detection and tracking problem in high resolution remotely sensed image sequences.
First, we develop a spatial marked point process model to detect a pre-defined class of objects (i.e. boats) based on their visual and geometric characteristics. We design a multiple core implementation of the reversible jump MCMC sampler. We show improved results over state of the art methods and competitive detection times.
The very good results obtained motivated us to extend these models to the temporal domain and create a framework based on spatio-temporal marked point process models to jointly detect and track multiple objects in image sequences. We propose the use of simple parametric shapes to describe the appearance of these objects. We build new, dedicated energy based models consisting of several terms that take into account both the image evidence and physical constraints such as object dynamics, track persistence and mutual exclusion. We construct a suitable optimization scheme that allows us to find strong local minima of the proposed highly non-convex energy.
As the simulation of such models comes with a high computational cost, we turn our attention to the recent filter implementations for multiple object tracking, which are known to be less computationally expensive. We propose a hybrid sampler by combining the Kalman filter with the standard Reversible Jump MCMC. High performance computing techniques are also used to increase the computational efficiency of our method. We provide an in-depth analysis of the proposed framework, as well as an extensive comparison with state of the art methods, based on standard multiple object tracking metrics and computational efficiency. This analysis yields a very good detection and tracking performance at the price of an increased complexity of the models. Exhaustive tests have been conducted both on high resolution satellite and microscopy image sequences.

Keywords: Multiple object tracking, object detection, marked point process, Kalman filter, satellite image sequences, microscopy data sequences, high resolution.