

Numéro d'ordre : 2012- Année 2012

Université Lumière Lyon 2

Laboratoire d'InfoRmatique en Image et Systèmes d'information

École Doctorale Informatique et Mathématiques de Lyon

Thèse de l'Université de Lyon

Présentée en vue d'obtenir le grade de Docteur,
spécialité Informatique

par

Imtiaz ALI

Object Detection in Dynamic Background

Thèse soutenue le 05 Mars 2012 devant le jury composé de :

Rapporteur	Vincent Barra	Professeur, Université Blaise Pascal-Clermont-Ferrand
Rapporteur	Jean-Philippe Domenger	Professeur, Université de Bordeaux 1
Examinateur	Thierry Chateau	MCF-HDR, Université Blaise Pascal-Clermont-Ferrand
Examinateur	Patrick Pérez	Distinguished scientist, Technicolor, Rennes
Invité	Hervé Piégay	Directeur de recherche, EVS, ENS, Lyon
Directrice	Laure Tougne	Professeure, Université Lumière Lyon 2
Co-directeur	Julien Mille	Maître de Conférences, Université Claude Bernard Lyon 1

Laboratoire d'InfoRmatique en Image et Systèmes d'information
UMR 5205 CNRS - Université Lumière Lyon 2 - Bât. C
69676, Bron cedex - France
Tel: +33 (0) 4 78 77 43 77 - Fax: +33 (0)4 78 77 23 38

The dedication goes here.

Abstract

Moving object detection is one of the main challenges in many video monitoring applications. In this thesis, we address the difficult problem that consists in object segmentation when background moves permanently. Such situations occur when the background contains water flow, smoke or flames, snowfall, rainfall *etc.* Object detection in moving background was not studied much in the literature so far. Video backgrounds studied in the literature are often composed of static scenes or only contain a small portion of moving regions (for example, fluttering leaves or brightness changes). The main difficulty when we study such situations is to differentiate the objects movements and the background movements that may be almost similar. For example, an object in river moves at the same speed as water. Therefore, motion-based techniques of the literature, relying on displacements vectors in the scene, may fail to discriminate objects from the background, thus generating a lot of false detections. In this complex context, we propose some solutions for object detection.

Object segmentation can be based on different criteria including color, texture, shape and motion. We propose various methods taking into account one or more of these criteria.

We first work on the specific context of wood detection in rivers. It is a part of DADEC project (Détection Automatique de Débris pour l'Aide à l'Etude des Crues) in collaboration with geographers. We propose two approaches for wood detection: a naïve method and the probabilistic image model. The naïve approach is based on binary decisions based on object color and motion, whereas the probabilistic image model uses wood intensity distribution with pixel motion. Such detection methods are used for tracking and counting pieces of wood in rivers.

Secondly, we consider a context in which we suppose *a priori* knowledge about object motion is available. Hence, we propose to model and incorporate this knowledge into the detection process. We show that combining this prior motion knowledge with classical background model improves object detection rate.

Finally, drawing our inspiration from methods used for 2D texture representation, we propose to model moving backgrounds using a frequency-based approach. More precisely, the model takes into account the spatial neighborhoods of pixels but also their temporal neighborhoods. We apply local Fourier transform on the obtained regions in

order to extract spatiotemporal color patterns.

We apply our methods on multiple videos, including river videos under DADEC project, image sequences from the DynTex video database, several synthetic videos and some of our own made videos. We compare our object detection results with the existing methods for real and synthetic videos quantitatively as well as qualitatively.

Keywords: Object detection, segmentation, background model, motion model, dynamic texture, local Fourier transform.

Résumé

La détection et la reconnaissance d'objets dans des vidéos numériques est l'un des principaux challenges dans de nombreuses applications de vidéo surveillance. Dans le cadre de cette thèse, nous nous sommes attaqué au problème difficile de la segmentation d'objets dans des vidéos dont le fond est en mouvement permanent. Il s'agit de situations qui se produisent par exemple lorsque l'on filme des cours d'eau, ou le ciel, ou encore une scène contenant de la fumée, de la pluie, *etc.* Il s'agit d'un sujet assez peu étudié dans la littérature car très souvent les scènes traitées sont plutôt statiques et seules quelques parties bougent, telles que les feuillages par exemple, ou les seuls mouvements sont des changements de luminosité. La principale difficulté dans le cadre des scènes dont le fond est en mouvement est de différencier le mouvement de l'objet du mouvement du fond qui peuvent parfois être très similaires. En effet, par exemple, un objet dans une rivière peut se déplacer à la même allure que l'eau. Les algorithmes de la littérature extrayant des champs de déplacement échouent alors et ceux basés sur des modélisations de fond génèrent de très nombreuses erreurs. C'est donc dans ce cadre compliqué que nous avons tenté d'apporter des solutions.

La segmentation d'objets pouvant se baser sur différents critères : couleur, texture, forme, mouvement, nous avons proposé différentes méthodes prenant en compte un ou plusieurs de ces critères.

Dans un premier temps, nous avons travaillé dans un contexte bien précis qui était celui de la détection des bois morts dans des rivières. Ce problème nous a été apporté par des géographes avec qui nous avons collaboré dans le cadre du projet DADEC (Détection Automatique de Débris pour l'Aide à l'Etude des Crues). Dans ce cadre, nous avons proposé deux méthodes l'une dite " naïve " basée sur la couleur des objets à détecter et sur leur mouvement et l'autre, basée sur une approche probabiliste mettant en oeuvre une modélisation de la couleur de l'objet et également basée sur leur déplacement. Nous avons proposé une méthode pour le comptage des bois morts en utilisant les résultats des segmentations.

Dans un deuxième temps, supposant la connaissance *a priori* du mouvement des objets, dans un contexte quelconque, nous avons proposé un modèle de mouvement de l'objet et avons montré que la prise en compte de cet a priori de mouvement permettait d'améliorer nettement les résultats des segmentations obtenus par les principaux

algorithmes de modélisation de fond que l'on trouve dans la littérature.

Enfin, dans un troisième temps, en s'inspirant de méthodes utilisées pour caractériser des textures 2D, nous avons proposé un modèle de fond basé sur une approche fréquentielle. Plus précisément, le modèle prend en compte non seulement le voisinage spatial d'un pixel mais également le voisinage temporel de ce dernier. Nous avons appliqué la transformée de Fourier locale au voisinage spatiotemporel d'un pixel pour construire un modèle de fond.

Nous avons appliqué nos méthodes sur plusieurs vidéos, notamment les vidéos du projet DADEC, les vidéos de la base DynTex, des vidéos synthétiques et des vidéos que nous avons faites.

Mots clés: Détection d'objets, segmentation, modèle de fond, modèle de mouvement, texture dynamique, transformée de Fourier locale.

Contents

List of Figures

List of Tables

Introduction

Cameras are everywhere these days: public areas are monitored by several cameras in order to increase public order and safety, private property is protected by means of cameras, and shopping malls use cameras to prevent shoplifting. Monitoring systems are still largely operated manually for anomaly detection, in which human operators continuously watch the activities over monitoring screens. Computer vision methods aim at reducing manual efforts involved in the process. In these techniques, image features in the monitored scene are studied to avail object detection, which is one of the first steps of video surveillance. Without a good object detection method, subsequent actions such as event understanding and anomaly detection would be infeasible.

Object detection can be performed by segmenting the monitored scene into foreground (objects of interest) and background (rest of scene). Each image pixel in the current video frame is declared either as a foreground pixel or a background pixel based on its various features. To perform the task with precision and accuracy, one can construct a representation of background and/or foreground. Moreover, an object detection method should be able to overcome obstacles inherent to complex environments. For example, in outdoor scenarios, objects and background may have similar color or motion; objects may pass through shadowed areas; global illumination may change rapidly *etc*.

In our thesis, we focus on videos containing dynamic backgrounds. Specifically, we study videos consisting in continuous motion of backgrounds and objects (*e.g.* floating objects in river, moving escalators, fire, smoke). Moreover, we consider videos acquired in a static camera setting, so that any apparent motion arises from a real motion of an object of interest or a part of the background. From the computer vision point of view, color/texture, shape and motion are amongst the prominent characteristics used in many object detection algorithms. In some cases, we may obtain prior motion knowledge about these features (*e.g.* luggage moving on conveyor belts, pedestrian motion *etc*.) and this information can be used for object detection. For example, regarding

luggage on a conveyor, objects move with speed and towards a direction that may be known, thus the information can be used for their detection. In some other cases, we may have *a priori* on objects appearance or color. The use of skin color model in face detection applications is an example of this category, where each pixel in the image is classified into skin-color and non-skin color. In certain cases, we may have *a priori* on object shapes (*e.g.* vehicle detection, leaf detection *etc.*). Using shape prior, one can avail objects segmentation. In the current work, we aim to detect objects with *a priori* on their motion and appearance.

More specifically, a part of our work is specialized towards floating wood detection and counting in rivers, which is an example of object detection in moving background where strong prior is available. The videos are produced under the DADEC project (Détection Automatique de Débris pour l'aide à l'Etude des Crues). The project aims to study the transport of fallen wood debris (small and large parts of trees) carried with floods by using video analysis. During floods, water flow is largely turbulent and carries a large amount of wood pieces. Remote monitoring of rivers is performed during several years and the obtained videos are manually annotated by geographers, almost frame per frame, in order to count the number of wood pieces passing through the observed scene (examples are given in figure 1). Recording manually each wood passage is extremely time-consuming and thus limits the study to reduced datasets. Therefore, automatic wood detection using computer vision may allow to speed up the process and to broaden the study to larger datasets.

Many detection methods address foreground extraction relying on statistical representations of color/texture, motion or shape. Probabilistic representations are a way to model the expected values of these characteristics, whether the considered pixel belongs either to background or foreground. These representations can also be built in a manner to combine the knowledge of multiple characteristics on each pixel. A frequently used

Figure 1: Two images from videos studied under DADEC project.

method in the literature is background modeling, in which a pixel-wise statistical representation of color characteristics with time is built. Basically, object detection can be obtained by comparing the probability of each pixel in new frame to the corresponding background model and classify it as foreground or background. Background models keep the colors that stay a long time in the scene.

Object detection in outdoor videos in particular is a very challenging task. The backgrounds we study in the context of the DADEC project contain many intensity variations, uneven brightnesses and motions that complicate background modeling based on color. As a matter of fact, classical color-based background models may produce a lot of false detections in such situations. Alternatively, due to the possibility of obtaining prior object intensity distribution, we choose to learn it for wood detection. We propose a probabilistic image model based on wood intensity/color distribution.

To achieve good segmentation in moving backgrounds, we can also include motion information. Motion-based methods rely on the information from motion patterns observed in the monitored scene. Object detection can be based on the dissimilar motion characteristics of objects and backgrounds. However, color and motion are interlinked within moving backgrounds where color patterns are different spatially and repetitive with time. Therefore, neither color nor motion alone can achieve good object detection in moving backgrounds, which consequently leads us to a combined approach based on color and motion. We suppose that *a priori* object motion information is available. We propose a motion model and learn the model parameters in the applications by an offline process. The motion model is designed so that it can be used in conjunction with any background subtraction technique. In this way, we blend object level motion information with pixel level color information using a Bayesian framework for object detection. The advantage of the combination is twofold (i.e. it reduces false detections and number of miss-detection within foreground objects).

Another approach may be to use the periodicity of color to model moving backgrounds, which are often composed of different textures. Unlike color, texture is not a property of a single pixel, but rather of a spatial neighborhood around the pixel. Moving color patterns forming dynamic textures vary spatially and appear repeatedly within time. Thus, they cannot be properly modeled by the existing individual pixel-based background modeling methods. There may be two possible reasons for shortcomings of classic pixel-based background models in such cases. Primarily, they consider each pixel independently and do not take into account spatial neighborhoods of pixels. Secondly, they do not take into account the temporal evolution of pixel values. To model the spatiotemporal textures formed by these evolutions, we propose to use a frequency-

based background model founded on 2D+T region around each pixel. The main idea behind our approach is to model the spatiotemporal color patterns of the scene and use the model for object detection. To our knowledge, no frequency-based approach was previously used for background modeling. We apply our methods to multiple videos. We use river videos in the framework of the DADEC project for wood detection. We devise several synthetic videos as well to test our algorithms. In addition, we apply our frequency-based method on videos containing dynamic textured backgrounds from the DynTex database [Péteri et al., 2010]. Moreover, we make our own videos of floating objects in rivers to test our method. We compare our object detection results with the existing methods on real and synthetic videos both qualitatively and quantitatively.

Thesis organization:
We organize our thesis as follows:

Chapter 1: It presents a state of the art on object detection in video. We explain the existing methods commonly used in fixed camera scenarios. We divide the existing methods on the basis of the three criteria, namely color, shape and motion, which the methods use for object detection. We give an overview of the most frequently used background modeling techniques. The prominent features of these are explained.

Chapter 2: We dedicate this chapter to intensity-based wood detection. We explain the constraints and problems in the application. Due to an outdoor environment, the difficulties arise from many physical phenomena, including brightness variations, cast shadows, apparent similarity between water waves and wood objects *etc.* We propose two methods for wood detection based on the intensity characteristics in the application. In the first approach, each incoming frame is processed by two intensity based segmentation methods separately. An inter-frame difference is applied as well and all resulting segmentations are combined to extract moving objects. In the second approach, we propose a probabilistic image model that uses wood intensity distribution for their detection.

Chapter 3: In this chapter, we develop the motion model that incorporates object-level prior motion knowledge. We use a Bayesian framework for combining object level motion information with pixel-level color information. In moving backgrounds, we explain its relevance to differentiate moving objects from background motions.

Chapter 4: It presents the spectral analysis of dynamic backgrounds. Moving backgrounds exhibit the appearance of spatially varying and time repetitive textures. In this regards, we analyze moving backgrounds and present the possibility to use spatiotemporal color information for background modeling.

Finally, we conclude and present future prospects of our work. In appendix A, we present the application of wood tracking and counting in river videos, which involves the wood detection algorithms described in chapter 2 for this purpose.

Chapter 1

State of the Art

Contents

Object detection in video analysis primarily depends on good image segmentation. Image segmentation is a low level task which is essentially the core part of high level multi tasks *e.g.* object tracking, object recognition and event understanding *etc.* Broadly speaking, image features based on color/texture, shape and motion are the criteria which existing methods use for object detection. The methods which use color/texture criteria for object detection, either rely on *a priori* color information or rely on object and/or background color-based statistical models. Some other methods use objects shapes as criteria for their detection in videos. Similarly, some methods use motion features for object detection. Therefore, we can summarize the existing methods under these criteria. The division is not tight and some methods can be placed in more than one group. We explain the prominent features of object detection techniques that are used in video surveillance systems. We focus on the methods that are frequently used in the fixed camera situations.

Common property of these object detection algorithms is that they produce a resulting binary segmented image with foreground pixels labeled as 1 and background pixels labeled as 0. Before discussing these methods, we present some notations that are used in our thesis work. Vector quantities are represented by small bold letters. In the current image I the triplet of red, green and blue values of pixel $\mathbf{x}$ at time t is represented by

$$I(\mathbf{x},t) = [I_1(\mathbf{x},t) \quad I_2(\mathbf{x},t) \quad I_3(\mathbf{x},t)]^T$$

and $D = 3$ denotes the number of channels. $\mathcal{F}(\mathbf{x},t)$ is the binary label of pixel $(\mathbf{x},t)$. Probability are denoted by P throughout our thesis. The remaining notations are defined when they are used.

1.1 Color and texture based object detection

In video analysis, a large number of methods exist which use color criteria for object detection. In some applications, we may have information about object and/or background colors *a priori* and use them for image segmentation. Let us consider the example of a colored moving object in homogeneous static background. In such case, we may perform image segmentation by thresholding. Image thresholding is a simple object detection method in which a global threshold is used for image segmentation (with assumption on *a priori* color information). [Ritter and Wilson, 2000] explain the image thresholding method to classify pixels as object or background. If a pixel color value is within a given color range then assign the binary value 1 to it, else consider it as a

background pixel and assign value 0 to it. However, the use of image thresholding techniques for object detection in videos is limited. Due to the brightness variations in the scene, the object colors and background colors may vary a lot. Therefore, to accommodate the color variations, statistical models are used to model the object or background color distributions.

For object detection in video, the use of probabilistic color models is wide. These models are built to represent either the color distributions of target objects or background. The models can be used to segregate moving objects from background. In some methods, *a priori* object/background color information are used to compute statistical models. In some other methods, object or background colors are learned to model them. Therefore, these methods can be divided accordingly. First, we present object color-based models and then we explain color-based background models.

1.1.1 Object color model

To perform foreground segmentation, some methods model the object colors. Skin-color detection is an example of such methods. It plays an important role in a wide range of image processing applications ranging from face detection, face tracking, gesture analysis, content-based image retrieval systems and to various human computer interaction domains. Skin-color can also be used as the complementary information to other features such as shape and geometry. It can be considered as a very effective tool for identifying/classifying facial areas provided that the underlying skin-color pixels can be represented, modeled and classified accurately. This is the reason why we have chosen to explain it in detail.

The main goal of skin color models is to build a pixel classification rule that will discriminate between skin and non-skin pixels. Identifying skin colored pixels in videos involves finding the range of values in a color space for which most skin pixels would fall in.

Primary steps for skin detection in an image using color information are

- to represent the image pixels in a suitable color space
- to model the skin and non-skin pixels using a suitable distribution and
- to classify the modeled distributions

Several color spaces are proposed for skin detection in the literature. The choice of color space also determines how effectively one can model the skin-color distribution. Skin-color distribution is modeled either by histogram models or by single/Gaussian

mixture models. [Vezhnevets et al., 2003] and [Kakumanu et al., 2007] present surveys of different color spaces for skin-color representation and skin-pixel detection methods.

Color space for skin detection:
The choice of color space can be considered as the first step in skin-color classification. Several color spaces have been used for skin detection. The RGB color space is a default color space for most available image formats. Any other color space can be obtained from a linear or non-linear transformation from RGB. To reduce the dependence on lighting, the RGB color components can be normalized so that the sum of normalized components is unity. It has been observed that under certain assumptions, differences in skin-color pixels due to lighting conditions and ethnicity can be greatly reduced using normalized RGB space instead of RGB. Also, skin-color clusters in normalized RGB space have relatively lower variance than the corresponding clusters in RGB and hence are shown to be good for skin-color modeling and detection [Yang and Ahuja, 1999, Störring et al., 2003, Sebe et al., 2004].

The perceptual features of color such as hue, saturation and value cannot be described directly by RGB. The HSV color space has been used in skin detection methods by [Wang and Yuan, 2001, Brown et al., 2001]. Similarly, orthogonal color spaces reduce the redundancy present in RGB color channels and represent the color with statistically independent components. The $YCbCr$ space represents color as luminance (Y) computed as a weighted sum of RGB values, and chrominance (Cb and Cr) computed by subtracting the luminance component from B and R values. The $YCbCr$ space is one of the most popular choices for skin detection and has been used by [Chai and Bouzerdoum, 2000, Hsu et al., 2002].

Skin modeling and classification:
Statistical models represent the probability density function (PDF) of skin color. There are various methods in the literature to estimate such PDF. However, we only present, histogram, single gaussian and mixture of gaussian models.

In the following paragraphs, we also present skin detection as a two classes classification problem: skin pixel versus non-skin pixel classification.

A 2D or 3D color histogram can be used to represent the distribution of skin tones in a given color space. The color space is quantized into a number of histogram bins. Each histogram bin stores the count associated with the occurrence of the bin color in the training data set. The histogram bin counts are converted into probability distribution,

by normalizing them. For a given color c, probability $P(c)$ is:

$$P(c) = \frac{C(c)}{T}$$

where $C(c)$ gives the count in the histogram bin associated with color c and T is the total count obtained by summing the counts in all the histogram bins. $P(c)$ corresponds to the likelihood that a given color belongs to the skin. All the pixel values for which the corresponding color likelihood is greater than a predefined threshold are defined as skin pixels. [Zarit et al., 1999, Yoo and Oh, 1999] used a histogram-based approach to classify skin pixels.

[Jones and Rehg, 2002] computed two different histograms, skin and non-skin histograms. Given skin and non-skin histograms, the probability that a given color belongs to skin and non-skin class is defined as

$$P(I(\mathbf{x},t)|skin) = \frac{s(I(\mathbf{x},t))}{T_s}, \quad P(I(\mathbf{x},t)|nonskin) = \frac{n(I(\mathbf{x},t))}{T_n}$$

where $s(I(\mathbf{x},t))$ is the bin value in the skin histogram, $n(I(\mathbf{x},t))$ is the bin value of the non-skin histogram. T_s and T_n represents the total counts in skin and non-skin histogram bins. From generic skin and non-skin histograms, [Jones and Rehg, 2002] demonstrate that there is a reasonable separation between skin and non-skin classes.

Given the class conditional probabilities of skin and non-skin color models, a skin classifier can be built using Bayes maximum likelihood (ML) approach [Duda et al., 2002]. Using this, a given image pixel can be classified as skin, if

$$\frac{P(I(\mathbf{x},t)|skin)}{P(I(\mathbf{x},t)|nonskin)} > \zeta$$

where $0 \leq \zeta \leq 1$ is a threshold value which can be adjusted to trade-off between true and false positives. The histogram-based Bayes classifier (also called as skin probability map) has been widely used for skin segmentation. The method is simple and computationally fast. In [Phung et al., 2005], the authors argue that the Bayesian classifier with histogram technique is found to perform better compared to other tested classifiers.

Another approach for skin-color distribution modeling is to use Gaussian mixtures. The advantage of these parametric models is that they have high capability of generalization with less training data and also that they have small storage requirements. Under controlled brightness conditions, skin colors of different persons cluster in a small region in the normalized RGB color space [Yang and Waibel, 1996]. Skin-color distribution is modeled through a single Gaussian $\mathcal{N}(\mu, \Sigma)$ where μ and Σ are estimated over all the

color samples from the training data using Maximum Likelihood estimation approach. The probability can be used directly as a measure of skin-color likeliness and the classification is normally obtained by comparing it to a certain threshold empirically estimated from the training data.

Though human skin-color samples for people of different races cluster in a small region in the color space, it has been shown that different modes co-exist within this cluster and hence it cannot be modeled effectively by a single Gaussian distribution. Also, under varying brightness conditions, the single mode assumption does not hold. Many researchers, therefore, have used Gaussian mixtures. For example, a Gaussian mixture density function (a sum of individual Gaussians) is used by [Yang and Ahuja, 1999]. Similarly, parameters of the GMM are approximated from the training data through the iterative expectation-maximization (EM) technique. The method has been applied for human face detection and recognition in videos [Hassanpour et al., 2008, Wang et al., 2011].

1.1.2 Background Modeling

In most of real scenarios, objects and background share many colors, which makes it difficult to isolate objects from the background. Similarly, in case of unknown object, the color information can not be modeled due its unavailability. Therefore, colors in the background can be learned. Several methods based on probabilistic background models have been developed [Elhabian et al., 2008]. These methods are also referred as background subtraction techniques, since the methods detect moving objects by subtracting the current image from a background (also called reference) representation. The notion of background subtraction gathers the background modeling step as well as the classification step.

A recent survey of various background modeling techniques is presented in [Brutzer et al., 2011]. We describe frequently used background models in this section, such as: Gaussian mixture model [Stauffer and Grimson, 2000], non-parametric method of kernel density estimation [Elgammal et al., 2002], CodeBook method [Kim et al., 2005] and VuMeter [Goyat et al., 2006].

All these above-mentioned background models are individual pixel based methods. We explain in details in the following paragraphs how color features are used by these methods for background representation because we will compare our proposition to these ones in the following. In particular, we describe the classification and model update in each of them.

1.1.2.1 Parametric method

A simple form of parametric model is to use a single gaussian per pixel for background modeling. If we choose to model the background process of each pixel as a single Gaussian distribution $\mathcal{N}$, the probability of observing a value $I(\mathbf{x})$ at pixel $\mathbf{x}$ can be expressed [Landabaso, 2008]:

$$\mathcal{N}(I(\mathbf{x}), \mu(\mathbf{x}), \Sigma(\mathbf{x})) = \frac{1}{(2\pi)^{\frac{D}{2}} |\Sigma(\mathbf{x})|^{\frac{1}{2}}} e^{-\frac{1}{2}(I(\mathbf{x})-\mu(\mathbf{x}))^T \Sigma^{-1}(I(\mathbf{x})-\mu(\mathbf{x}))} \tag{1.1}$$

where $\mu(\mathbf{x})$ and $\Sigma(\mathbf{x})$ are the mean vector and covariance matrix, respectively, of the Gaussian corresponding to pixel $\mathbf{x}$, and where D denotes the number of features contained in vector $I(\mathbf{x})$.

In single-Gaussian model systems [Wren et al., 1997, Horprasert et al., 1999, Jabri et al., 2000], a pixel is considered as the foreground pixel if it does not fall within 2.5 standard deviations of the mean of the distribution.

Gaussian mixture model:

The pixel-wise Gaussian Mixture Model (GMM) method has been proposed by [Stauffer and Grimson, 2000] for background modeling. Mixture of Gaussians is used in the literature to model the variety of background colors that may appear at each pixel. If a background is modeled by using mixture of Gaussians then the probability $P_{\text{background}}$ of a value $I(\mathbf{x})$ at pixel $\mathbf{x}$ can be written as:

$$P_{\text{background}}(I(\mathbf{x}), \mu(\mathbf{x}), \Sigma(\mathbf{x})) = \sum_{i=1}^{K} \omega_i(\mathbf{x}) * \mathcal{N}(I(\mathbf{x}), \mu_i(\mathbf{x}), \Sigma_i(\mathbf{x}))$$

$$= \sum_{i=1}^{K} \frac{\omega_i(\mathbf{x})}{(2\pi)^{\frac{D}{2}} |\Sigma_i(\mathbf{x})|^{\frac{1}{2}}} e^{-\frac{1}{2}(I(\mathbf{x})-\mu_i(\mathbf{x}))^T \Sigma_i^{-1}(\mathbf{x})(I(\mathbf{x})-\mu_i(\mathbf{x}))} \tag{1.2}$$

where K is the total number of Gaussians and $\omega_i(\mathbf{x})$ is the prior that a background pixel is represented by a certain mode i of the mixture ($\sum_{i=1}^{K} \omega_i(\mathbf{x}) = 1$). These priors are often named as the weights of the Gaussians. $\mu_i(\mathbf{x})$ and $\Sigma_i(\mathbf{x})$ are the mean value and covariance matrix of the i^{th} Gaussian in the mixture for current pixel.

For red, green and blue (RGB) channel values, the covariance matrix can be written as:

$$\Sigma_i(\mathbf{x}) = \begin{pmatrix} \sigma_{i1}^2(\mathbf{x}) & \text{Cov}_{i12}(\mathbf{x}) & \text{Cov}_{i13}(\mathbf{x}) \\ \text{Cov}_{i12}(\mathbf{x}) & \sigma_{i2}^2(\mathbf{x}) & \text{Cov}_{i23}(\mathbf{x}) \\ \text{Cov}_{i12}(\mathbf{x}) & \text{Cov}_{i23}(\mathbf{x}) & \sigma_{i3}^2(\mathbf{x}) \end{pmatrix} \tag{1.3}$$

where σ_i^2 and Cov_i are variances of the values of the corresponding channels and covariances between the values of the two channels that correspond to i^{th} Gaussian of the distribution. The computation of the probability requires the covariance matrix to be inverted. To reduce computation time in covariance matrix inversion for each individual pixel, [Stauffer and Grimson, 2000] assume that all color channels are statistically independent. Thus, the covariance matrix for the i^{th} distribution becomes diagonal:

$$\Sigma_i(\mathbf{x}) = \begin{pmatrix} \sigma_{i1}^2(\mathbf{x}) & 0 & 0 \\ 0 & \sigma_{i2}^2(\mathbf{x}) & 0 \\ 0 & 0 & \sigma_{i3}^2(\mathbf{x}) \end{pmatrix} \tag{1.4}$$

[Stauffer and Grimson, 2000] consider that all the color channels have the same variances which leads to :

$$\Sigma_i(\mathbf{x}) = \sigma_i^2(\mathbf{x}) \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

therefore, Eq. 1.1 in this context becomes

$$\mathcal{N}(I(\mathbf{x}), \mu(\mathbf{x}), \Sigma(\mathbf{x})) = \prod_{d=1}^{D} \frac{1}{\sqrt{2\pi}\sigma_d} e^{-\frac{1}{2\sigma_d^2}(I_d(\mathbf{x})-\mu_d(\mathbf{x}))^2} \tag{1.5}$$

which is the product of D unidimensional Gaussians. Similarly, assuming the independence of channels in the mixture of Gaussians, Eq. 1.2 simplifies to:

$$P_{\text{background}}(I(\mathbf{x}), \mu(\mathbf{x}), \Sigma(\mathbf{x})) = \sum_{i=1}^{K} (\omega_i(\mathbf{x}) \prod_{d=1}^{D} \frac{1}{\sqrt{2\pi\sigma_{id}^2}} e^{-\frac{1}{2\sigma_{id}^2}(I_d(\mathbf{x})-\mu_{id}(\mathbf{x}))^2}) \tag{1.6}$$

In the following paragraphs, we explain the pixel classification and the model update in the GMM algorithm.

Background/Foreground classification:

In [Stauffer and Grimson, 2000], a pixel $\mathbf{x}$ is attributed to the i^{th} Gaussian when the pixel color value $I(\mathbf{x})$ is within 2.5 standard deviations of the distribution mean $\mu_i(\mathbf{x})$. Then, in order to determine whether a Gaussian represents a foreground or a background process, Gaussians of each pixel are reordered according to $\frac{\omega_i(\mathbf{x})}{\sigma_i(\mathbf{x})}$ in the descending order.

The first few Gaussians in the list correspond to the ones with more supporting evidence (*i.e.* more times explaining incoming pixels) at the lowest variance (explained

incoming pixels are always very similar). In other words, the first few Gaussians represent the background process if the background is relatively static (low variance $\sigma_i(\mathbf{x})$) and it is seen most of the time (high weight ($\omega_i(\mathbf{x})$)). On the contrary, unassigned pixel values and the values corresponding to the last Gaussians in the list are classified into the foreground.

In the GMM method of [Stauffer and Grimson, 2000], a pixel is classified as a background pixel if its value matches one of the first B distributions decided by Eq. 1.7, otherwise, it is classified as a foreground pixel.

$$B = \arg\min_{b} \sum_{i=1}^{b} \omega_i(\mathbf{x}) > T \tag{1.7}$$

where i expresses the index of the Gaussians in the reordering $\frac{\omega_i(\mathbf{x})}{\sigma_i(\mathbf{x})}$ as mentioned above.

The process for creating new Gaussian mode is as follows: If none of the distributions matches the current pixel value, the least probable distribution is replaced with a distribution with the current value as its mean value, an initially high variance and low prior weight.

The foreground segmentation task becomes a problem of one-class classification when model for only background class exists [Tax and Duin, 2001, Juszczak and Duin, 2003]. The classification in such case is harder than a standard two-class classification problem. In two-class classification, a decision boundary is supported from both sides by models of each of the classes (foreground and background). In one-class classification, only the background class is available, *i.e.* just one side of the boundary is supported. Based on the model of one class only, it is hard to decide how tight the boundary should fit around the target class. The boundary is set as $2.5\sigma_i(\mathbf{x})$ in the GMM.

Furthermore, the system should adapt to illumination changes in the scene. Therefore, the background model parameters are updated with time. Next, we discuss the update mechanism in the GMM method.

Model update step:

The prime objective of the model update is to keep the background model as accurate as possible. Therefore, the model update tries to accommodate background illumination changes occurring with the passage of time. In the GMM model [Stauffer and Grimson, 2000], when a pixel value is observed, the weight $\omega_i(\mathbf{x}, t)$ of i^{th} Gaussian is updated with

time, as follows:

$$\omega_i(\mathbf{x},t) = \begin{cases} \omega_i(\mathbf{x},t-1) + \alpha(1-\omega_i(\mathbf{x},t-1)) & \text{matched} \\ (1-\alpha)\omega_i(\mathbf{x},t-1) & \text{not matched} \end{cases} \tag{1.8}$$

Thus, the more often a Gaussian explains an incoming pixel, the higher its associated weight is. Note that this is a low-pass filter average of weights, where the last samples (latest in time) have exponentially more relevance than the older ones.

In the same way, variances $\sigma_i^2(\mathbf{x},t)$ and means $\mu_i(\mathbf{x},t)$ of the corresponding Gaussians are updated accordingly:

$$\begin{aligned} \mu_i(\mathbf{x},t) &= (1-\rho_i(\mathbf{x}))\mu_i(\mathbf{x},t-1) + \rho_i(\mathbf{x})I(\mathbf{x},t) \\ \sigma_i^2(\mathbf{x},t) &= (1-\rho_i(\mathbf{x}))\sigma_i^2(\mathbf{x},t-1) + \rho_i(\mathbf{x})(I(\mathbf{x},t)-\mu_i(\mathbf{x},t-1))^T(I(\mathbf{x},t)-\mu_i(\mathbf{x},t-1)) \end{aligned} \tag{1.9}$$

where $\rho_i(\mathbf{x})$ is the adaptation learning rate used in i^{th} Gaussian and pixel $\mathbf{x}$ and it follows:

$$\rho_i(\mathbf{x}) = \alpha\mathcal{N}(I(\mathbf{x},t)|\mu_i(\mathbf{x}),\sigma_i(\mathbf{x})) \tag{1.10}$$

which acts as a low-pass filter. Thus, by updating means and variances, the system is allowed to adapt to slow illumination changes.

In case of fixed camera, Gaussian-based background modeling is the most common approach used in the literature. [KaewTraKulPong and Bowden, 2001] proposed a fast update method for GMM. They argued that adaptation in the GMM method of [Stauffer and Grimson, 2000] was slow. They suggest that ρ is too small due to the likelihood factor. This leads to too slow adaptations in the means and the covariance matrices. Therefore, to improve update method, they propose to use L-recent window update equations, where $L = \frac{1}{\alpha}$ samples. Also, they propose to cut out the likelihood term from ρ in Eq. 1.10. They apply their method for shadows removal in the detected objects.

[Lee, 2005] proposed an effective learning algorithm to improve over the GMM method [Stauffer and Grimson, 2000]. The author suggested some modifications in the adaptation rate ρ_i. A variable is used to count the number of effective observations for i^{th} Gaussian and compute the appropriate learning rate. It is incremented when parameters of a Gaussian are updated. When the Gaussian is reassigned, it is reset to 1 since the old Gaussian has perished and a new one was started with a single data point. He showed that it improves the convergence speed and model accuracy. He also defined the winner-take-all option, where only a single best-matching component is selected for

the parameter update. The method is better when fast brightness changes occur than GMM [Stauffer and Grimson, 2000].

A block based GMM method has also been proposed by [Chen et al., 2007]. They divide the image into blocks and compute a contrast histogram as a descriptor for each block. They further use these contrast histograms as a feature instead of individual pixel color values (*i.e.* GMM algorithm) to build the background model.

Moreover, the GMM method has been combined with image spatial or temporal features to improve pixel classification. [Javed et al., 2002] used, for example, the GMM and intensity gradient simultaneously to remove shadows and compensate the brightness variations. [Izadi and Saeedi, 2008] combined the spatial gradient with the GMM. They isolated objects from the background and also removed shadows by using filtering and morphological operations. Another method by [Huang et al., 2009], proposed a background training parameter into the GMM. The training parameter uses region-based scheme and incorporates both spatial and temporal information. The approach addresses the problem of shadow detection in the foreground. [Cong et al., 2009] detected moving objects by combining the GMM background model with the temporal gradient (computed from successive frames). The combination leads the method to accommodate brightness variations in the background model. Wolf and Jolion [2010] integrated motion information into GMM method on pixel level. They used optical flow method for motion estimation. For each pixel, foreground/background classification is performed on the combine likelihood of motion and color.

1.1.2.2 Non-parametric methods

Parametric models such as the GMM, assume that underlying distribution of colors of each pixel can be represented by K normal distributions. In contrast to parametric models, an approach based on non-parametric kernel density estimation was introduced by [Elgammal et al., 2002]. In their approach, they do not assume any prior distribution for underlying colors. Instead, they estimate the density function directly from colors.

The model keeps a sample color values for each pixel in the image and uses this sample to estimate the density function of the pixel color density distribution. They estimate that an observed pixel color value $I(\mathbf{x})$ is based on L recent samples. It can be expressed as:

$$P_{\text{background}}(I(\mathbf{x},t)) = \frac{1}{L}\sum_{i=1}^{L}\prod_{d=1}^{D} K_{\sigma_d}(I_d(\mathbf{x},t) - I_d(\mathbf{x},t-i)) \tag{1.11}$$

where K_{σ_d} is kernel function with bandwidth σ_d in the d^{th} color channel. [Elgammal et al., 2002] proposed to compute the median absolute deviation over the samples for consecutive color values of the pixel to estimate σ_d^2 for the corresponding color channel of each pixel. Also, the median for each consecutive pair $(I(\mathbf{x}, i), I(\mathbf{x}, i+1))$ is computed independently for each color channel. They used Gaussian kernels for modeling the density at a given pixel, so the Eq. 1.11 in their case becomes:

$$P_{\text{background}}(I(\mathbf{x}, t)) = \frac{1}{L} \sum_{i=1}^{L} \prod_{d=1}^{D} \frac{1}{\sqrt{2\pi\sigma_d^2}} e^{-\frac{1}{2\sigma_d^2}(I_d(\mathbf{x},t) - I_d(\mathbf{x},t-i))^2} \tag{1.12}$$

[Elgammal et al., 2002] argue that kernel density estimation can asymptotically converge to any distribution with sufficient samples. For background/foreground pixel classification, a pixel is considered as a foreground pixel if $P_{\text{background}}(I(\mathbf{x}, t)) < T$, where T is a global threshold for the whole image and can be adjusted to minimize the percentage of false positive detections. Model update is done by adding new samples and ignoring old samples. The entire model is recomputed on the basis of the last n observations. Therefore, the method can adapt to more general and complex background variations.

[Elgammal et al., 2002] used fixed Gaussian kernels for modeling the density at a given pixel. Therefore, it may suffer shortcomings when the background is composed of high and low density areas. In [Mittal and Paragios, 2004], authors addressed the issue and proposed variable bandwidth density estimation for background model (*i.e.* small bandwidths and large bandwidths in high and low density areas respectively). Also, they used motion information for foreground extraction. They extract the information from video by using the optical flow method and combined it with probability density estimation. They applied their method for object detection in non-stationary background.

Similarly, [Ko et al., 2008] proposed to use spatial neighborhood of each pixel in the computation of density estimation. Their method consists in analyzing the temporal variation of intensity distributions, rather than pixel values. They represented the signature of each pixel using a distribution of pixel intensities in a neighborhood, and used the Bhattacharyya distance to compare such distributions over time. This approach can be viewed as a hybrid between pixel− and texture−level comparisons. The distribution computed around a given location at current time could be viewed as a feature vector describing the texture at that location. Thus, the distribution signature is relatively insensitive to small movements of the highly textured background, and at the same time is not tied to individual pixel values for detecting foreground objects. This enables slower

background updates, and therefore minimizes the probability that the foreground object be incorporated into the background. They applied the method to detect birds in the natural environment.

In the following paragraphs, we present the codebook methods frequently applied for background modeling in the literature.

CodeBook method:
[Kim et al., 2005] propose the CodeBook (CB) method to build a background model. It is a quantization technique that uses long scene observations for each pixel. One or several codewords are stored in the codebook for each pixel. The number of codewords for a pixel depends on the background color variation. Therefore, all pixels do not have the same number of codewords.

Each codebook contains some codewords to model a cluster of samples that constructs a part of background, $c_i(\mathbf{x}) \; \forall \; i = 1...\chi$. Each codeword is composed of following information:
$\mathbf{v}_i$: mean (RGB) value of pixel, f: frequency of codeword,
I_{max}: high intensity bound of codeword, I_{min}: low intensity bound of codeword,
p: first occurrence of the codeword and q: last occurrence of the codeword,
λ: MNRL (maximum negative run length), represents the longest number of images where the codeword does not occur in the sequence.

For each new pixel $\mathbf{x}$, intensity is calculated by $I = \sqrt{I_1^2 + I_2^2 + I_3^2}$ and $\mathbf{v}_i = (\bar{I}_{i1}, \bar{I}_{i2}, \bar{I}_{i3})$. The color distortion δ between a given pixel $\mathbf{x}$ and a codeword c_i can be computed by:

$$\begin{aligned} \langle \mathbf{x}, \mathbf{v}_i \rangle^2 &= (\bar{I}_{i1} I_1 + \bar{I}_{i2} I_2 + \bar{I}_{i3} I_3)^2 \\ ||\mathbf{v}_i||^2 &= \bar{I}_{i1}^2 + \bar{I}_{i2}^2 + \bar{I}_{i3}^2 \\ ||\mathbf{x}||^2 &= I_1^2 + I_2^2 + I_3^2 \end{aligned}$$

$$\delta(\mathbf{x}, \mathbf{v}_i) = \sqrt{||\mathbf{x}||^2 - \frac{\langle \mathbf{x}, \mathbf{v}_i \rangle^2}{||\mathbf{v}_i||^2}} \tag{1.13}$$

A pixel $\mathbf{x}$ at time t with an intensity I matches to a codeword c_i, if I is in range [I_{min}, I_{max}] and the color distortion δ is below a threshold ϵ. A new codeword is created for a pixel $\mathbf{x}$ in the following way:

$$\mathbf{v}_i \leftarrow (I_1, I_2, I_3) \quad , \quad I_{min} \leftarrow \max\{0, I - \alpha\} \quad , \quad I_{max} \leftarrow \min\{255, I + \alpha\}$$

$$f \leftarrow 1 \quad , \quad \lambda \leftarrow t - 1 \quad , \quad p \leftarrow t \quad , \quad q \leftarrow t$$

where α is a value which represents a tolerance of intensity. In the CB method, codewords are assigned during the training period and are updated on the observation of the frequency of appearance of these codewords for the same pixel. During the training phase a codeword is updated for a given pixel **x** as follows:

$$\bar{I_1} \leftarrow \frac{\bar{I_1} \times f + I_1}{f+1} \quad \text{idem for } I_2 \text{ and } I_3$$

$$I_{min} \leftarrow \frac{I - \alpha + f \times I_{min}}{f+1} \quad , \quad I_{max} \leftarrow \frac{I + \alpha + f \times I_{max}}{f+1}$$

$$f \leftarrow f + 1 \quad , \quad \lambda \leftarrow \max\{\lambda, t - q\} \quad , \quad p \leftarrow p \quad , \quad q \leftarrow t$$

In the detection phase, the codeword is updated like previously in training phase except for I_{min} and I_{max} which are updated as follow, :

$$I_{min} \leftarrow (1-\gamma)(I - \alpha) + \gamma I_{min}$$
$$I_{max} \leftarrow (1-\gamma)(I + \alpha) + \gamma I_{max}$$

where γ is a coefficient to change adaptation speed. The codebook obtained during the training time represents the training image sequence. It may contain objects information as well, if objects are present in the scene during training time. Therefore, the codewords for objects must be removed from the codebook. The variable λ is used for filtering the codewords which are not updated. It is assumed that the codewords which represent objects colors have higher values of λ, since codewords which represent objects are not updated frequently. The codewords with $\lambda \leq \lambda_{th}$ are either deleted from the codebook or not used in the process of code matching. For this reason, finding an optimal value for λ_{th} is an important task. In [Kim et al., 2005], authors suggested that $\lambda_{th} = \frac{\tau}{2}$ is a good choice. τ is the duration of training time period. The higher value of λ_{th} leads to add object color pixel into codebook. Similarly, smaller value of λ_{th} is unable to model the background movement (like moving leaves).

The original CB method is modified by [Sigari and Fathy, 2008], in which authors propose to use two layered CB models for each pixel. They denote these layers as the main CB for permanent backgrounds and the hidden CB for non-permanent backgrounds. They argue that if a new background appear after the training sequence, then the original CB method cannot model it. They define three thresholds (λ_{H}, λ_{add} and λ_{del}) for λ. The update process of the CB is as follows: Remove all codewords in the hidden CB if $\lambda > \lambda_{\mathrm{H}}$. Move the codewords from the hidden CB to the main CB who stay longer than λ_{add} in the hidden CB. Remove all codewords from the main CB who do not appear longer than λ_{del}. So, this method keeps the data in the hidden CB that

are not important at the current instant but could be more important after. They show that the method is very useful to have a good detection of waving trees for example.

The CB method gives good results in small background movements in comparison to GMM. Especially, the CB methods work well compared to the GMM in case of global brightness changes in the background. Also, shadow removal and object detection rate is better than GMM method due to the model update method. A comparative analysis for the two methods along with other background models has been done by [Dhome et al., 2010], and is shown in Figure 1.1.

VuMeter method:
We present here another color-based background model proposed more recently. It is known as the VuMeter [Goyat et al., 2006]. It is also a non parametric method based on a discrete estimation of the probability distribution. The discretization of the probability density function $P_{\text{background}}$ is carried out with N bins for each parameter. For each pixel, the authors consider the 3 color channels to be statistically independent which leads to $3.N$ elements instead of N^3 elements for each parameter.

Thus, for a pixel color value $I(\mathbf{x}, t)$, the probability density function $P_{\text{background}}(I(\mathbf{x}, t))$ can be approximated by:

$$P_{\text{background}}(I(\mathbf{x}, t)) = \prod_{d=1}^{D} P(I_d(\mathbf{x})) \tag{1.14}$$

If $\mathbf{b}(\mathbf{x}, t)$ denotes the bin index vector associated to $I(\mathbf{x})$ and i is a bin index, then probability density per channel can be expressed as:

$$P_{\text{background}}(I_d(\mathbf{x}, t)) \approx K_d \sum_{i=1}^{N} \pi_{id} \delta(\mathbf{b}_d(\mathbf{x}) - i) \tag{1.15}$$

where δ is the Kronecker delta function, and K_d is a normalization constant to ensure the following condition at each moment

$$\sum_{i=1}^{N} \pi_{id} = 1$$

π_{id} is a discrete mass function which is represented by a bin. At first image ($t = 0$), bins values are set, $\pi_{id} = \frac{1}{N}$ to have a sum to 1 as mentioned above. Similarly, at each new pixel, when its value matches to a bin π_{id}, the level of the bin is updated as follow:

$$\pi_{id}(t+1) = \pi_{id}(t) + \alpha \delta(\mathbf{b}_d(\mathbf{x}, t+1) - i) \tag{1.16}$$

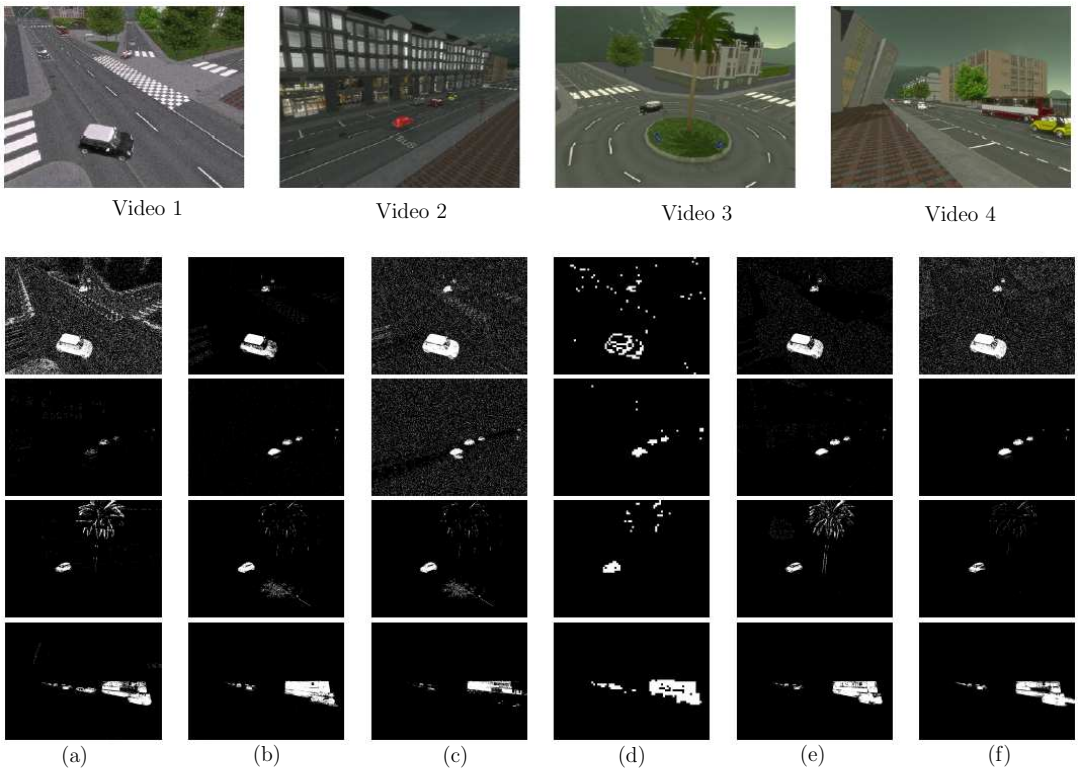


Figure 1.1: Four synthetic videos [Dhome et al., 2010], results of six background modeling algorithms (a) GMM [Stauffer and Grimson, 2000], (b) Fast GMM [KaewTraKulPong and Bowden, 2001], (c) Bayesian [Tuzel et al., 2005], (d) Block-level GMM [Chen et al., 2007], (e) CodeBook [Sigari and Fathy, 2008] and (f) VuMeter [Goyat et al., 2006].

After some time, *i.e.* after a lot of images, the bins which model the background have a high value. Each new pixel with corresponding bins below an empirically set threshold is assigned to the background. In RGB mode, each pixel is modeled by 3 VuMeter (one per color channel). To consider a pixel as background, it must be detected as background with each VuMeter.

To improve background detection and to reduce problems with edges between two bins, the values of classes in the neighborhood of a matched class are also updated, but to a lesser extent. A fixed learning rate is used for model update. To have a good learning and adaptation of algorithm, it is necessary to have a good learning rate α and a good threshold.

A comparative study of background subtraction algorithms is presented in [Dhome et al., 2010]. The authors proposed a benchmark based on artificial videos. 4 different synthetic videos are created with various background conditions. These situations include light intensity variations, moving and stationary backgrounds, random noise, large and small sized vehicles, moving persons *etc.* An advantage is that ground truth data is also

available with the videos. They applied six background modeling algorithms and analyzed their segmentation results. These six algorithms are GMM [Stauffer and Grimson, 2000], a fast GMM [KaewTraKulPong and Bowden, 2001], Bayesian method [Tuzel et al., 2005], block based GMM [Chen et al., 2007], the CodeBook method [Sigari and Fathy, 2008] and the VuMeter [Goyat et al., 2006]. The authors also discussed the evaluation methods of image segmentation techniques and show that the best results are obtained by using the VuMeter method [Goyat et al., 2006]. In Figure 1.1, the segmentation results of the six algorithms are shown.

Texture based background model:
Texture is one of the important properties of visual surfaces which helps us to discriminate one object from another, an object from the background, and to draw inferences about 3D world [Jain and Karu, 1996]. Unlike other image features, such as intensity or color, texture is not a property of a single pixel, but rather of a spatial neighborhood around the pixel.

To detect moving objects in a textured background, an approach has been proposed by [Zhong and Sclaroff, 2003], which uses an auto regressive moving average (ARMA) model for dynamic textures [Doretto et al., 2003]. In this approach at any time t, we observe a noisy version of the image

$$y(t) = I(t) + w(t)$$

where $w(t) \sim P(.)$ is an independent and identically distributed sequence drawn from a known probability density function. Then, the image sequence $\{I(t)\}_{t=1\cdots\tau}$ of τ images is a (linear) dynamic texture if there exists a set of n spatial filters ϕ, such that $I(t) = \phi(x(t))$, where $x(t)$ describes the dynamic texture's state. Thus, the autoregressive model for dynamic texture is expressed as:

$$\begin{aligned} x(t+1) &= Ax(t) + Bv(t), \\ y(t) &= \phi(x(t)) + w(t), \end{aligned} \tag{1.17}$$

with $x(0) = x_0$, $v(t) \sim \mathcal{N}(0, Q)$ is unknown state noise. [Zhong and Sclaroff, 2003] also propose a robust Kalman filter to iteratively update the state of the ARMA model. If the current estimated value for a pixel is different from the predicted value then the pixel is labeled as foreground. A similar approach is proposed by [Monnet et al., 2003]. They use a predictive model to capture the most important variations based on a subspace analysis of the image sequence. The components of this model are used in an

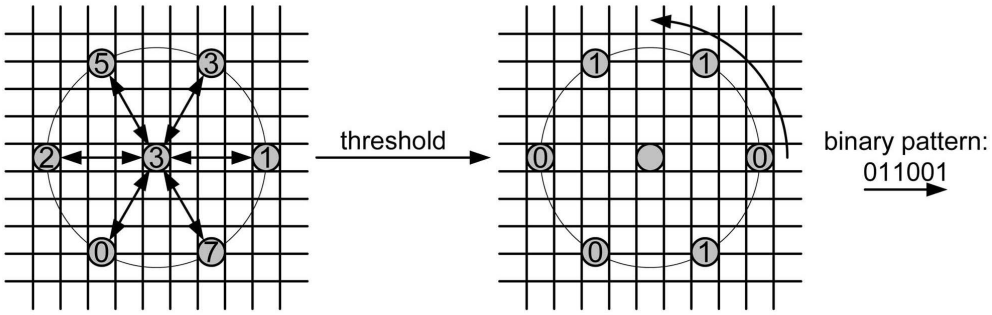


Figure 1.2: Computation of Circular Local Binary Pattern [Heikkilä and Pietikäinen, 2006].

autoregressive form to predict the frame to be observed. Differences in the state space between the prediction and the observation quantify the amount of change and are considered to perform detection.

In spatial domain, a 2D texture-based background model is proposed by [Heikkilä and Pietikäinen, 2006]. The approach is a background model in which only spatial textures (2D) among different zones of images are considered. In their approach, they use the Local Binary Pattern (LBP) as a texture operator, which relates to the earlier work by [Ojala et al., 1996, 2002]. The operator labels the pixels of an image region by thresholding the neighborhood of each pixel with the center value and considering the result as a binary number (binary pattern). The basic version of the LBP operator considers eight neighbors of a pixel. However, [Heikkilä and Pietikäinen, 2006] use a circular LBP operator, which contains user-settable radius in spatial domain around each pixel (see Figure 1.2). The LBP histogram $\mathbf{h}$ is used as a feature vector in background modeling. The background model for the pixel consists of a group of K adaptive LBP histograms per pixel,$\{\mathbf{m}_0, \cdots, \mathbf{m}_{K-1}\}$, where K is a user settable parameter. Each model histogram has a weight between 0 and 1 so that the weights of the K model histograms sum up to one.

The background model is updated by using histogram intersection as a proximity measure. This measure calculates the common part of two histograms and neglects features which only occur in one of the histograms. If the incoming histogram $\mathbf{h}$ matches the model histogram then the best matched model histogram is adapted with the new data updating its bins. Similarly, the weights ω of the model histogram are updated as follows:

$$\begin{aligned} \mathbf{m}_i &= \alpha_b \mathbf{h} + (1 - \alpha_b)\mathbf{m}_i \qquad \alpha_b \in [0, 1] \\ \omega_i &= \alpha_w M_i + (1 - \alpha_w)\omega_i \quad \alpha_w \in [0, 1] \end{aligned} \tag{1.18}$$

where α_b and α_w are two learning rates used for model update. M_i is 1 for the best

matching histogram and 0 for the others. The adaptation speed of the background model is controlled by the learning rates. The bigger the learning rate, the faster is the adaptation. Afterwards, they sort the model histograms in decreasing order according to their weights, and select the first B histograms as the background histograms, such that:

$$\omega_0 + \cdots + \omega_{B-1} > T_B \quad T_B \in [0, 1]$$

For object detection, they compare the incoming histogram (computed from the new frame) with current B background histograms by using the proximity measure. If the proximity is higher than an empirically set threshold for at least one background histogram, then the pixel is classified as background. Otherwise, the pixel is marked as foreground.

They obtain improved object detection over pixel-based methods. However, they apply the background model to relatively static backgrounds. Furthermore, the LBP does not work very robustly on flat image areas, where the gray values of the neighboring pixels are very close to the value of the center pixel. Similarly, these methods assume spatial consistency of the textured background. Therefore, in case of dynamic textures (*i.e.* water ripples, smoke, fire), the LBP-based methods may not produce good object detection results.

1.2 Shape based object detection

Object detection in video can benefit from knowledge about object shapes. Different object types such as persons, vehicles, leaves and flowers *etc.* may require dedicated detection algorithms. Methods which use shape features as criteria for object detection need their appearance information. The information is either known *a priori* or learned during the training period. Also, statistical models based on object shape features are proposed to address the diversity of possible shape appearances. Shape based methods can be divided further into implicit shape models and active shape model.

1.2.1 Implicit shape model

The methods sometimes use object shapes *a priori* to detect them in the image sequence. Typically, an implicit shape is represented with a binary function or a distance function. For example, [Leibe et al., 2004, 2005] use an Implicit Shape Model in which the local structure of object shape appearance is learned. In order to learn the appearance variability of an object category, they build up a codebook of local appearances that are

characteristic for a particular viewpoint of its member objects. They extract local features around interest points and group them with an agglomerative clustering scheme. Based on the codebook, they learn an Implicit Shape Model that specifies where the codebook entries may occur on the object. In this way, they define *allowed* shapes implicitly in terms the local appearances consistency. Implicit Shape Model is formulated in a probabilistic framework that allows to obtain a category-specific segmentation and object recognition.

[Jacobs and Pless, 2007] proposed a shape based background model that includes the expected shape of foreground objects into background model. They attempted to characterize the benefits of scene-specific shape models for object localization. They used an offline approach to learn car and pedestrian shapes. They found pixel-wise likelihood jointly on the background and shape models. They applied their method for anomaly detection. [García-Martin et al., 2011] used implicit shape model for pedestrian detection in crowded environment. They proposed to use an implicit motion model for the pedestrian. They combined an implicit shape model proposed by [Leibe et al., 2005] with an implicit motion model and claim the superiority of their approach over [Leibe et al., 2005]. [Cerutti et al., 2011] used a prior leaf shape model for their detection. Prior shape model improves the detection rate of leaves in natural background. In [Ferrari et al., 2007, 2010], the authors presented an approach for learning and matching shapes with explicit shape models formed by continuous connected curves completely covering the object outlines. [Ferrari et al., 2010], presented a method that can learn complete shape models directly from images. Moreover, it can automatically match the learned model to cluttered test images, thereby localizing novel class instances up to their boundaries. [Caro-Campos et al., 2011] used prior object shape and used it for the detection of stolen objects from the monitored scene. They suppose a rectangular shaped object. An active contour technique is applied to check whether the object contour is present in the current image.

Notice that often it is not possible to use *a priori* shape information because of changing point of view or change of appearance.

1.2.2 Active shape model

Conversely, active shape models represent the prior shape with a set of connected landmark points. The use of prior shape knowledge is extended to statistical shape models which are based on the different object appearances. The constraints which may cause the variability in the shape appearances include brightness changes, view point changes,

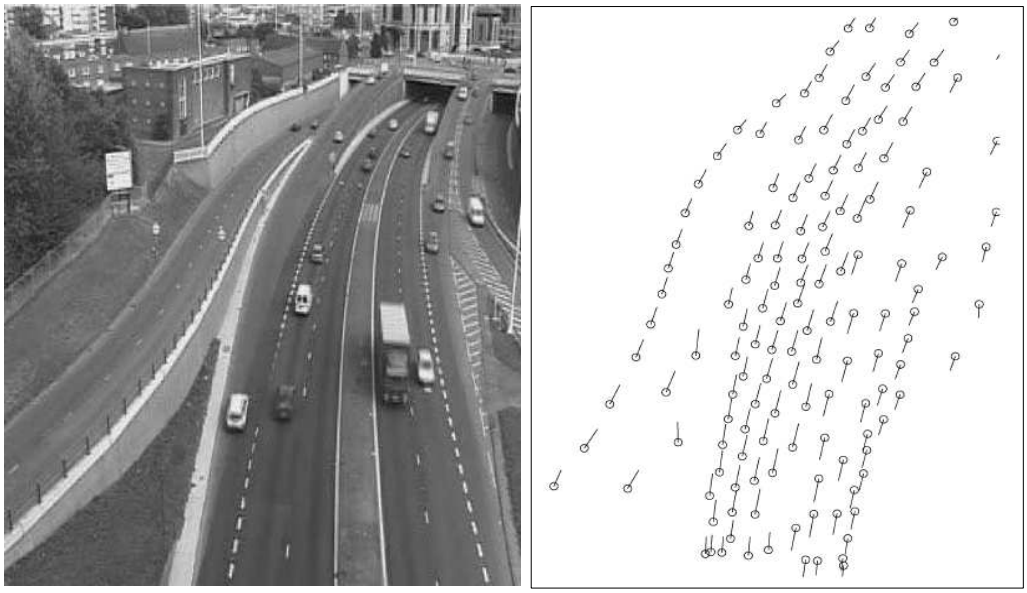
the object-camera distance *etc.* For this reason, a large number of features are used to construct such models that represent an object occurrence in videos. An early method by [Ferryman et al., 1995] present a deformable shape model for vehicle detection in videos. Similarly, [Kuno et al., 1996] describe a robust and reliable method of human detection for visual surveillance system. They first take precise silhouette patterns by detecting and analyzing the change in the brightness between the background image and the current image. They use shape features of the silhouette patterns of humans as the detection parameters.

A popular choice in shape based methods is to use Active Shape Models (ASM) [Cootes et al., 1995]. These are statistical models of objects shapes which iteratively deform to fit to an example of the object in a new image. The shapes are constrained by a statistical shape model to vary only in ways seen in a training set of labeled examples. Active shape models are applied in different object detection algorithms. For example, [Siebel and Maybank, 2002] apply it for multiple people tracking. They use a method which combine object motion with the pedestrian shape model for robust tracking algorithm. In [Jang and Jung, 2008], body-skeleton models are generated for human pose estimation in videos. Skeleton models are produced by a model of body shape variation. Images are labeled with landmark points representing the positions of key features. The coordinates of the skeleton points for each training image are stored. They apply Principal Component Analysis (PCA) on the deviation from the mean values of these points in the training data set. They compare skeleton shape model to the manually obtained human silhouette in the current image for frontal human pose estimation. [Kim et al., 2010] propose hierarchical recovery of human walking poses by the active shape model framework. The information is used with the motion prediction for human gait recognition. Active shape model is also used in face recognition methods. Facial features extracted from the model [García et al., 2010] are used for driver fatigue detection.

Shape appearance is often unknown in object detection. In such case, to help segmentation, we can use object motion instead.

1.3 Motion based object detection

Based on motion information, different motion models can be proposed to facilitate object detection. In some cases, these motion models are built on the *a priori* object motion information. In other cases, some techniques lead to estimate object motion. The two types of methods are presented in the following subsections.



Original image Learned ground-plane velocity map

Figure 1.3: Original road image with learned velocity map [Magee, 2004].

1.3.1 Methods based on prior motion knowledge

In application where *a priori* knowledge about the motion of objects is available, one can use this knowledge to constrain foreground detection. In [Song et al., 2003], for example, human motion is modeled by a joint probability density function of position and velocity of a collection of body parts. They use prior knowledge of human walking in the monitored scene.

In [Magee, 2004], the author used *a priori* road direction and travel information for multiple vehicles detection. The vehicle speed information is used to facilitate vehicle tracking and their position prediction but the information is not used explicitly for the image segmentation. Figure 1.3 contains original road image and the learned ground-plane velocity map from [Magee, 2004]. In the velocity map in Figure 1.3, the circles represent centroids of vehicles and motion directions are marked by lines. The velocity map is used to calculate initial estimate of normalized velocity and direction. The method also represents the possibility of learning vehicle motion. Similarly, in [Sappa et al., 2005], prior knowledge of pedestrian movement was used for 3D model construction from 2D motion silhouettes. They argue that human displacement involves synchronized motion of each body part which need to be modeled. They select head and leg movements in the segmented image sequence for learning motion trajectories.

Motion information can also be used on pixel level for object detection. [Nam and Han, 2006], for example, use per pixel motion prior for moving pixel classification into back-

ground or foreground. In their method, a motion prior distribution is recursively estimated by a particle filter model. The motion prior probability serves as a weight in pixel-wise classification of a pixel into background or foreground.

However, in most of the applications, object motion knowledge is also not available *a priori*. Therefore, the methods use motion estimation for object detection. The algorithms which use motion estimations can further be divided into inter-frame difference and optical flow methods.

1.3.2 Image difference

The most basic form of motion segmentation relies on the inter-frame difference. It consists of absolute difference of intensity of two frames pixel by pixel.

$$\Delta I(\mathbf{x}, t) = |I(\mathbf{x}, t) - I(\mathbf{x}, t-1)|$$

The result is a coarse map of the temporal changes. It is used in the literature due to the simplicity of computation. In order to distinguish between relevant changes due to objects motion or brightness changes and irrelevant temporal changes due to noise, the frame difference is usually compared to a threshold. The reliable decision that a spatial position $\mathbf{x}$ belongs to a moving region is only possible if the frame difference exceeds this threshold value. However, in case of moving background or moving camera, image difference may not provide useful information. Therefore, [Bergen et al., 1992, Kameda and Minoh, 1996] use three video frames for object motion segmentation. They generate two difference images $\Delta I(., t)$ and $\Delta I(., t+1)$ from the successive images at time $t-1, t$ and $t+1$. They binarize the difference and take the intersection of these two difference images to compute a resulting double difference image. Thus, they compute a rough map of the moving pixels and use it for moving object extraction.

Another approach for motion detection is to use Motion History Image (MHI), where successive layering of image silhouettes is used to represent patterns of motion. Whenever a new frame arrives, the existing silhouettes are decreased in value subject to some threshold and the new silhouette (if any) is overlaid at maximal brightness. This layered motion image is termed a Motion History Image (MHI). The representations have an advantage that a range of times from frame to frame to several seconds may be encoded in a single image. It generates a 2D template image for each action in the video. The MHI approach relies on template matching and detect occurrences of a previously learned action. [Bradski and Davis, 2002] compute gradients of the MHI by convolution with separable Sobel filters in the spatial domain. They label motion regions connected to

the current silhouette using a downward stepping floodfill and identify areas of motion directly attached to parts of the object of interest. This representation can be used to determine the object current pose and measure the motions induced by the object in a video scene.

[Viola et al., 2003] propose a method that uses manually extracted patches of image differences and learn a cascade of weak classifiers for pedestrian detection.

Some methods apply probabilistic model to image differences. For example, [Li et al., 2007] partition the image into an array of cells and assume that a cell contains motion if the differences in that cell are approximately uniformly distributed. They use edge-based morphological dilation method to achieve the anisotropic expansion of the detected object regions.

[Verbeke and Vincent, 2007] accumulate n previous frames and use a frame differencing technique to find regions where the motion has occurred. They perform PCA to map data of image differences in a lower-dimensional space where points containing coherent motion are close to each other. The technique is better than simple frame difference techniques, as they accumulate the n previous frames for calculating motion region in the image. However, the method is sensitive to light intensity changes, shadows and sensor noise.

1.3.3 Optical flow based methods

The optical flow method, introduced by [Horn and Schunck, 1981], is based on the apparent motion of the brightness patterns in the image. The method is frequently used to estimate motion in video analysis. Here, we provide an overview of basic terms of optical flow methods and present some of the object detection methods based on optical flow.

1.3.3.1 Optical flow basis

Most of the optical flow algorithms assume *brightness constancy*, *i.e.* when a pixel flows from one image to another (successive in time), its intensity or color does not change. The brightness constancy constraint can be expressed as:

$$I(x, y, t) = I(x + u, y + v, t + 1) \tag{1.19}$$

Applying a first-order Taylor expansion to the right-hand side of Eq. 1.19 yields the following approximation:

$$I(x,y,t) = I(x,y,t) + u\frac{\partial I}{\partial x} + v\frac{\partial I}{\partial y} + 1\frac{\partial I}{\partial t} \tag{1.20}$$

which simplifies to the optical flow constraint equation as:

$$u\frac{\partial I}{\partial x} + v\frac{\partial I}{\partial y} + \frac{\partial I}{\partial t} = 0 \tag{1.21}$$

Both the brightness constancy and the optical flow constraint equations provide one constraint on the two unknowns at each pixel. This is the origin of the *aperture problem* and the reason that the optical flow is an ill-posed problem. Therefore, it must be regularized with a smoothness term. [Horn and Schunck, 1981] optimize a global energy function based on residuals from the brightness constancy constraint and use a particular regularization term that expresses the expected smoothness of the flow field. In comparison, [Lucas and Kanade, 1981] assume that the flow is essentially constant in a local neighborhood of a given pixel, and solve the basic optical flow equations for all the pixels in that neighborhood, by the least squares criterion.

Instead of using the intensity or color values in the images, it is also possible to use features computed from the image sequence. One recently popular choice is to augment or replace Eq. 1.19 with a similar term based on the image gradient:

$$\nabla I(x,y,t) = \nabla I(x+u, y+v, t+1) \tag{1.22}$$

Empirically the gradient is often more robust to (approximately additive) illumination changes than the raw intensities. A comprehensive comparison of optical flow methods is presented in [Baker et al., 2011]. They provide a database and evaluation of most common used optical flow algorithms.

1.3.3.2 Object detection based on optical flow

We present some object detection algorithms that rely on the optical flow. Several methods are proposed in the literature in which optical flow estimation are used with background subtraction for object detection. [Iketani et al., 1998, Wixson, 2000] use the estimation of the consistency of optical flows over a short duration of time. The motion trajectories are used for detection and recognition of the objects. Moreover, in some methods, image texture features are added to the motion features, to form a complete feature set for motion and appearance-based recognition. In [Peteri and Chetverikov,

Figure 1.4: Approximate optical flow: (from left to right) 1^{st} frame, 2^{nd} frame and zoomed head portion to show motion vector field [Wolf and Jolion, 2010].

2005] a method for extracting texture features is based on the normal optical flow and on the texture regularity though the sequence.

[Mittal and Paragios, 2004] propose an adaptive kernel density estimation scheme with a joint pixel-wise model of color (for a normalized color space) and optical flow at each pixel. Similarly, a foreground detection method by [Li et al., 2010] propose an optical flow and background model (OFBM). It is based on Lucas-Kanade optical flow and Gaussian background model methods. They use two successive images to compute LK flow field. The resulting flow field is thresholded to obtain a coarse foreground image. They also apply GMM method to obtain another foreground image and multiply it with foreground image obtained with the optical flow. They apply their method for crowd abnormal behavior detections and crowd density estimations. Another method of similar nature is presented by [Wolf and Jolion, 2010], in which authors propose to take into account spatiotemporal interactions using a global energy function for background subtraction. They modify the pixel classification in GMM method and temporal consistency of foreground/background labeling is enforced by the use of an optical flow-based temporal regularization term (see Figure 1.4).

However, one can notice that the computational complexity of (dense) optical flow techniques is high. Besides the computational complexity, another disadvantage of optical flow is that the flow is not always correct. It is inaccurate at object edges because of the smoothing involved in the computation, which ultimately causes inaccurate object segmentation. Inside objects and in homogeneous regions, the flow tends to be null. Moreover, brightness constancy and consistent flow assumptions are violated in case of moving backgrounds.

1.4 Discussion

In this thesis, we deal with outdoor videos with a particularity: a moving background. We focus on fixed camera situation and we do not make assumptions concerning shape or type of objects we want to extract or recognize. Hence, if we consider again the various criteria we used to classify articles of the literature, we can make the following remarks and explain our approach as follows.

Firstly, regarding to color and texture, two possibilities are offered to us: considering object or background. In chapter 2, as the proposition is driven by application (detecting wood in river), we choose to model the object color. The proposed method intrinsically differs from methods proposed in the literature as it is application oriented and we could not find in the literature such object detection. More precisely, we propose in this chapter a probabilistic image model based on intensity/color information of object. In chapter 4, we study a background color based model and to be more exact, the method is inspired from 2D texture segmentation methods we found in the literature. This is a method that takes into account both temporal and spatial neighbors of a pixel contrary to background modeling methods of the literature. It exploits the fact that in moving background situations, color patterns often appear repeatedly during time. Using coherency of colors leads us to propose a new background model.

Secondly, as we do not make assumptions concerning the appearance of the objects we want to extract, we cannot use shape-based criteria.

Thirdly, in chapter 3, we suppose that it is possible to know a priori the object motion in some situations. Thus, we propose an original method to introduce this knowledge in object detection. We show that this information allows improving results of classical background subtraction methods we can find in the literature.

Chapter 2

Color based object detection

Contents

In this chapter, we study wood detection in rivers as an image segmentation problem and propose methods to achieve it. Floating wood detection in rivers is an example of object detection within moving background. It constitutes the first part of the DADEC project, which is briefly presented in the introduction. The method is designed in order to be applicable according to different weather conditions and be able to detect small and large wood pieces indifferently.

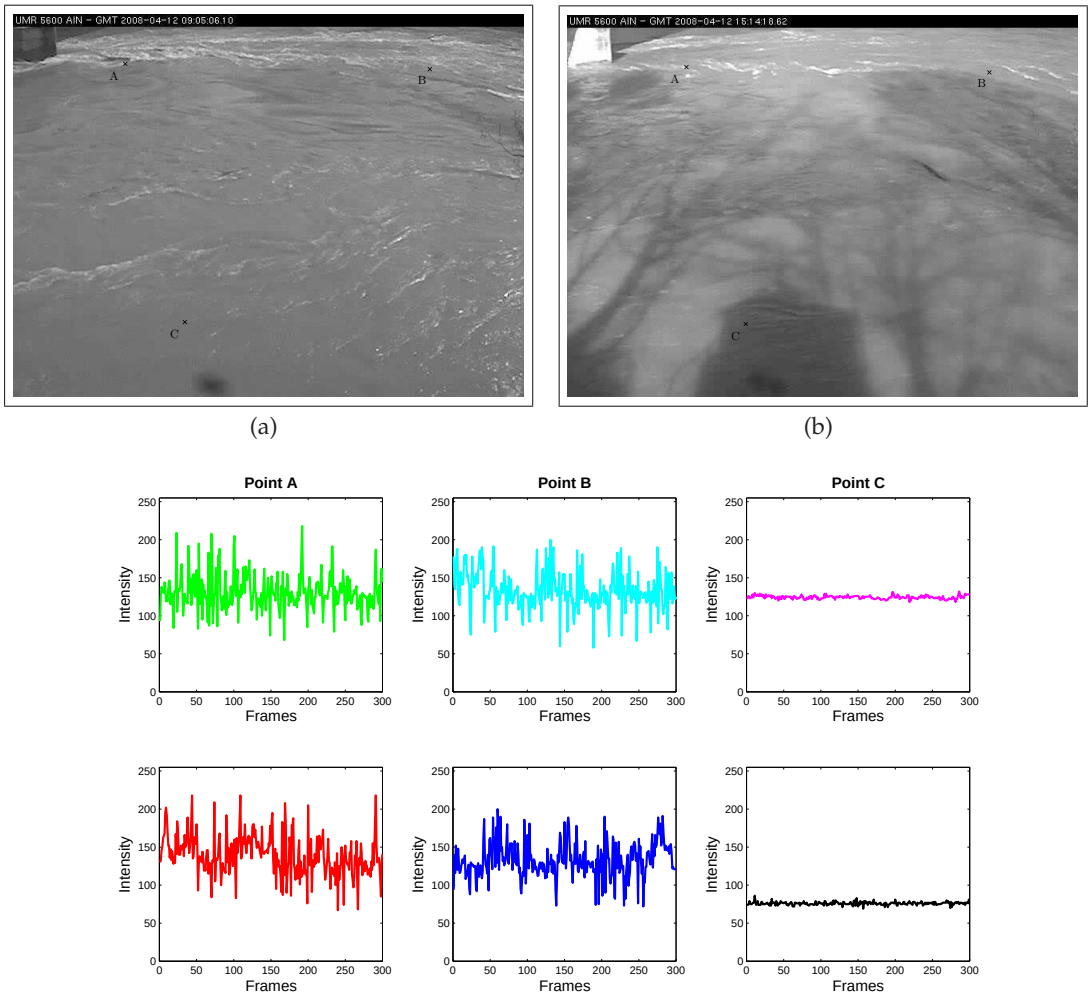
We present two wood detection methods in this chapter, which are based on wood characteristics in the river videos. The first approach involves image intensity and its gradient computed at a pixel level for each frame independently. The information is combined with inter-frame intensity difference between successive frames. In the second approach, we refine the method and propose a probabilistic image model based on wood intensity as well as its temporal variation. Before explaining the methods, we present the constraints related to wood detection in river videos.

2.1 Floating wood detection in river

We present the scenarios of floating wood detection in river. It is an outdoor application and there are many constraints with regards to image analysis. We summarize these constraints to highlight the challenges which are involved in wood detection.

2.1.1 Context

During floods, caused by excessive rains especially in the mountains, the river water flow varies from the normal flow. It carries a lot of fallen trees, bushes, branches of fallen trees and other small pieces of wood from mountains. These fallen trees often get stuck into the pillars of bridges and favor accumulation of small branches, bushes and debris around them and block the water flow. The amount of destruction threat made by trees is directly proportional to their sizes, as larger fallen trees are more dangerous than the smaller parts of fallen trees. Therefore, automatic wood detection using visual monitoring systems can help in the safety of bridges and population living in the vicinity. The statistics of quantities of fallen trees passing through the river can be obtained with such automation. The information can further contribute to the preventive measures for civil infrastructures of bridges and dams. In this context, researchers of the LIRIS Lab (UMR 5200) collaborate with geographer researchers of the EVS Lab (UMR 5600).



(a) (b)

(c)

Figure 2.1: (a),(b) Background images with three points selected, (c) wide intensity variations of background regions with time: in the first row of graphs points A, B and C corresponding to image sequence in (a) and, in the second row of graphs points A, B and C corresponding to image sequence in (b).

2.1.2 Constraints in wood detection in river

We can observe that there are many intensity variations in the background regions in river videos. The Figure 2.1 shows examples of images extracted from a camera positioned by the Ain river (France) at tow different moments. We have marked three points on each image : A, B, C and we have plotted intensity of each of them according to time. As a matter of fact, the dynamic background contains a wide distribution of intensity at these background regions. Also, it demonstrates a high level of complexity and challenge involved in our problem.

Wood detection, primarily, depends on the intensity difference between wood and

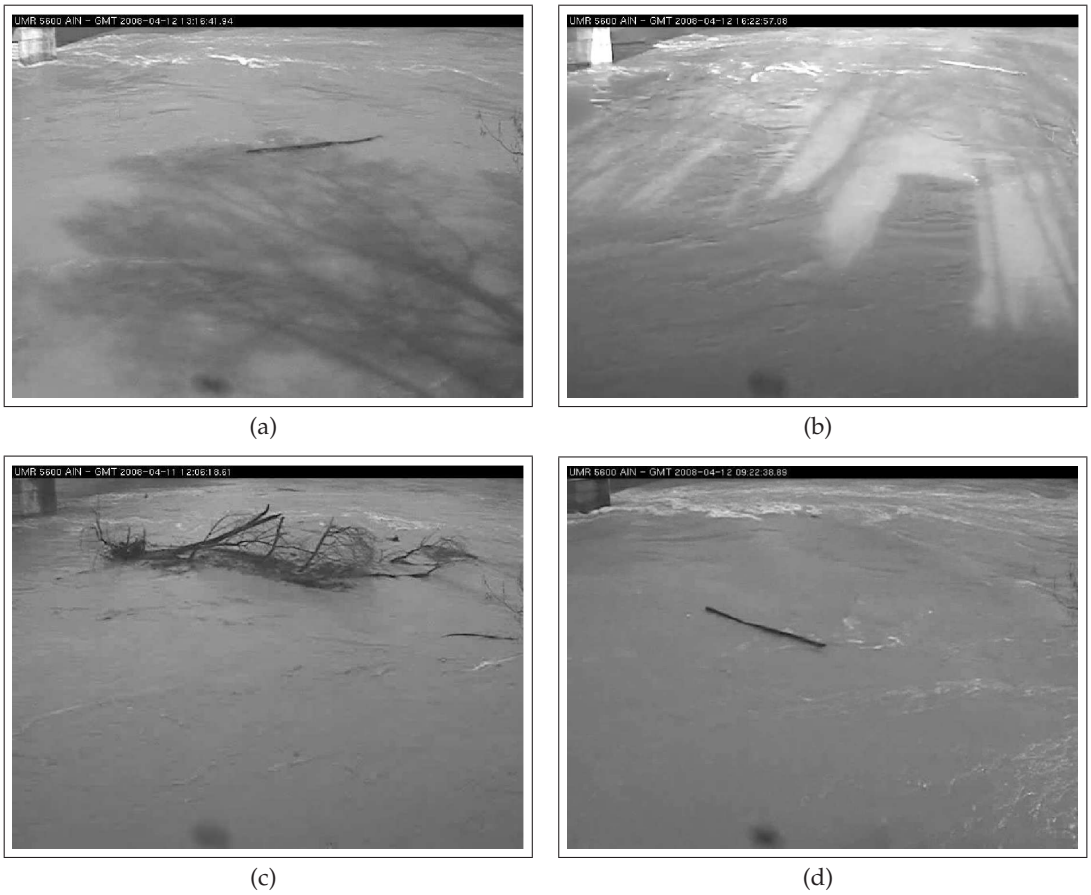


Figure 2.2: A few original images of, (a) a wood piece under the shadows of surrounding trees, (b) cast shadows of surrounding infrastructures, (c) a fallen tree in the cloudy weather and (d) a wood piece having reflection of sun shine from the surface of water.

water. In studied image sequences, wood is darker than water. However, in case of sunshine, the intensity level of water waves resemble the intensity of wood. Moreover, moving clouds may cause rapid brightness changes over the surface of river. So, videos contain sudden uneven brightness, cast shadows from surrounding trees and infrastructures, reflections from water surfaces and dark cloudy conditions. Some representative situations are highlighted in Figure 2.2.

An additional difficulty is caused by the similarity between small wood pieces and waves. In Figure 2.3, a small wood piece and a water wave in the same video frame are shown, both having similar sizes, shapes and brightness levels. Moreover, wood pieces may be partially submerged during motion. This causes occlusion, due to which their apparent size may not remain the same in the image sequence. In these videos, the presence of bridge (top left corner of images) and moving branches of tree before camera (right middle portions of images) can be seen. In the bridge area, there are a lot of turbulences in water waves as shown in Figure 2.2 which makes the background

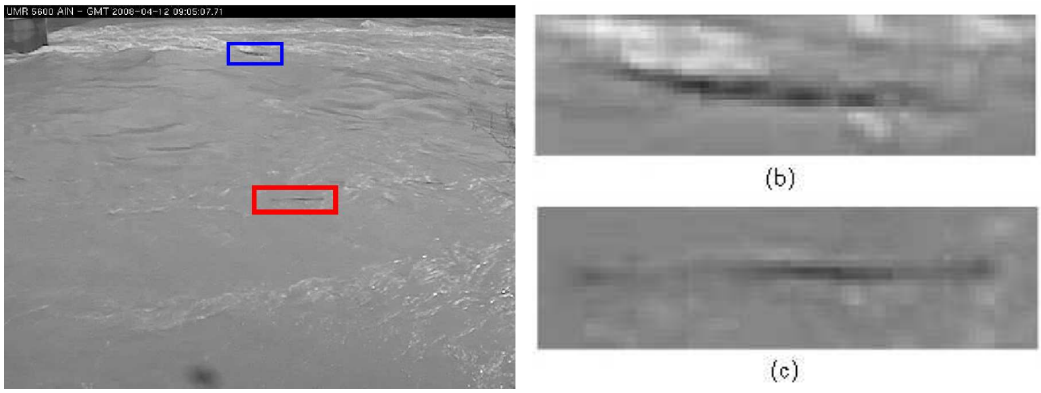


Figure 2.3: An image with a highlighted water wave and a wood piece, (b) upper portion of image is zoomed to show shape and color intensity of wave, (c) lower portion of image is zoomed to show shape and color intensity of wood piece.

highly dynamic. Naturally, the detection algorithm should be robust to these imaging conditions and should detect objects indifferently in any part of river in the videos.

Finally, due to the remote location of monitoring system and the limits of transfer rate of data networks, the frame rate in the video is very low ($\sim$ 4 fps). Consequently, object displacement is large between consecutive frames. Thus, optical flow techniques which are based principally on the *brightness constancy* assumption could not render meaningful results in river videos. To illustrate the inefficiency of optical flow, we applied the optical flow estimation algorithm of [Lucas and Kanade, 1981] on river videos and the results are shown in Figure 2.4. The vector flow field is represented by arrows in the four example images. The motion vectors follow the intensity gradients between consecutive frames. The length of the arrows denotes the magnitude of motion vectors and the flow direction is indicated by arrow heads. As we can notice, the motion vectors are haphazardly distributed in the river video. This shows the difficulty and complex nature of the background in the current videos. Moreover, optical flow results are similar for water waves and wood. In Figure 2.4 (c) and (d), two wood pieces are shown. Optical flow vectors for wood pieces and water waves in the background are similar in size and globally in direction. Therefore, we can argue that the optical flow technique does not seem to be the most appropriate technique to discriminate object (wood) and background (waves) motion in this application.

2.2 Naïve approach for wood detection

The approach is based on color, spatial and temporal features of image. The flow chart of the approach is presented in Figure 2.5. It shows that each frame is treated by two

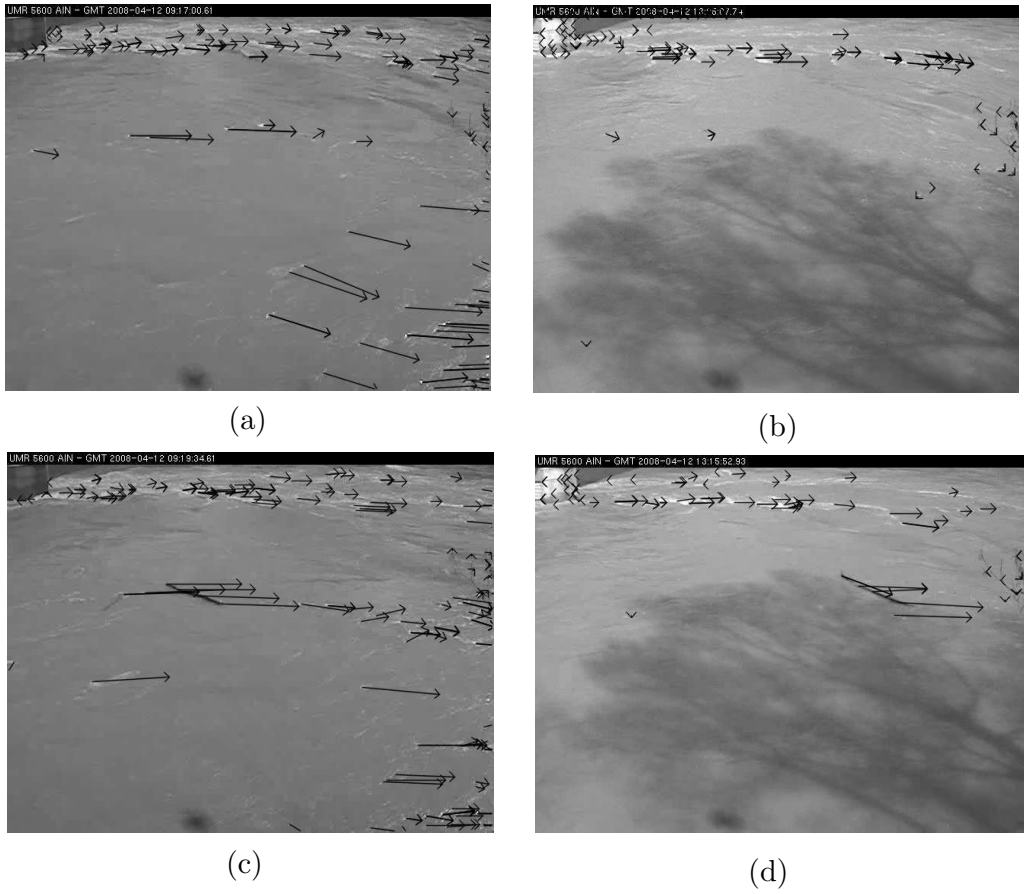


Figure 2.4: Optical flow method of [Lucas and Kanade, 1981] applied on river videos. (a),(b) are background only where (c),(d) are images with wood objects, motion vectors are shown by arrows.

image segmentation processes. The results of these two processes are binary segmented images. One is called the intensity mask (MI) and the other one the gradient mask (MG). These are the results of image segmentation based on intensity histogram thresholding and spatial gradient technique respectively. Furthermore, inter-frame difference (dT) for each image pair is taken to include temporal changes into wood detection. We explain each of these elements in this section.

2.2.1 Intensity mask

Water in the river and wood have rather different intensity levels, and wood is darker than water as shown in Figure 2.2. Histogram thresholding is among the popular techniques for gray-level image segmentation and several strategies have been proposed to implement it in the literature [Otsu, 1979, Pal and Pal, 1993, Jain et al., 1995, LI et al., 1997]. In these methods, peaks and valleys of the 1D brightness histograms of gray-level images can be identified as objects and backgrounds respectively.

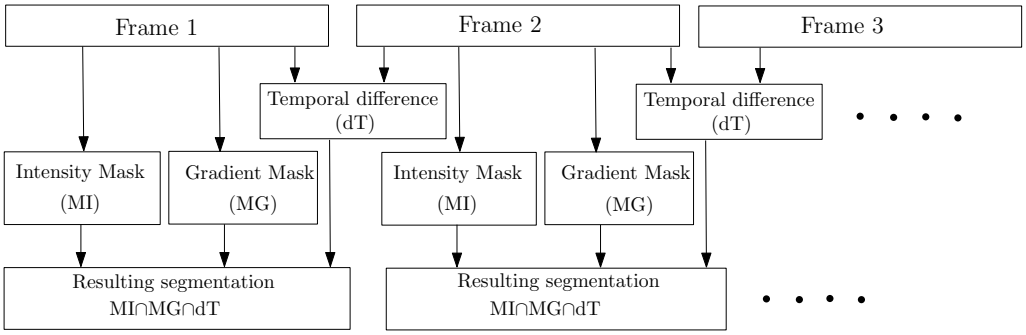


Figure 2.5: Flow chart of naïve approach for wood detection

The Fisher linear discriminant technique is used for automatic histogram thresholding [Otsu, 1979, Wolf and Jolion, 2003]. We calculate an optimal threshold s_i from the gray value histogram $\mathbf{h}$ by assuming two Gaussian distributions in the current image (class 0 for "non-wood" and class 1 for "wood" in our case) and maximizing the inter-class variance. The criterion used in discriminant analysis can be expressed as:

$$k^* = s_i = \arg\max_k(\omega_0\omega_1(\mu_1 - \mu_0)^2) \tag{2.1}$$

where $\omega_0 = \sum_{i=1}^{k} \frac{\mathbf{h}(i)}{N}$ is the normalized mass of the first class, $\omega_1 = \sum_{i=k+1}^{L} \frac{\mathbf{h}(i)}{N}$ is the normalized mass of the second class, μ_0 and μ_1 are the mean gray levels of respective classes and can be expressed as: $\mu_0 = \frac{\sum_{i=1}^{k} i\mathbf{h}(i)}{\sum_{i=1}^{k} \mathbf{h}(i)}$, $\mu_1 = \frac{\sum_{i=k+1}^{L} i\mathbf{h}(i)}{\sum_{i=k+1}^{L} \mathbf{h}(i)}$, where N is the number of pixels and L the number of bins of the histogram.

Each incoming video frame is processed likewise and an intensity mask (MI) is computed (see Figure 2.5), as follows:

$$\mathrm{MI}(\mathbf{x}, t) = \begin{cases} 1 & \text{if} \quad I(\mathbf{x}, t) \leq s_i \\ 0 & \quad\ \text{otherwise} \end{cases} \tag{2.2}$$

The method segregates pixels with dark color from the rest of scene. The technique produces good image segmentation in the absence of sunshine. Intensity masks for the corresponding images are highlighted in blue color in Figure 2.6(b). The left image is an example of intensity mask in the absence of sunshine. It shows that the technique of automatic histogram works well in the absence of sunlight. The right image is the intensity mask obtained in the presence of sunlight and the intensity of water waves and wood resemble one another. It indicates that the method is not adequate alone and produces wrong image segmentation. Therefore, we propose to use spatial feature to overcome the shortcomings of the intensity based segmentation method.

2.2.2 Gradient mask

In the presence of sunlight, cast shadows of surrounding trees and buildings cause the previous method based on the intensity histogram thresholding alone to produce erroneous image segmentation. We propose to use the intensity gradient magnitude. This approach has been extensively investigated for gray-level images [Fu and Mui, 1981, Rosenfeld and Kak, 1982, Pal and Pal, 1993]. Algorithms have also been proposed for the detection of discontinuities within color images [Zhao, 2008]. Each incoming frame is treated by using the Sobel operator. A threshold s_g is empirically set to obtain a binary gradient mask. The method can be expressed as:

$$\mathrm{MG}(\mathbf{x},t) = \begin{cases} 1 & \text{if} \quad \|\nabla I(\mathbf{x},t)\| \geq s_g \\ 0 & \text{otherwise} \end{cases} \tag{2.3}$$

This feature has the advantage, contrary to intensity histogram thresholding, that it operates well in the shadowed region. The gradient masks for the corresponding images are highlighted in green color in Figure 2.6(c). It can be noted that both water waves and moving wood have strong gradients, and therefore, the resulting gradient mask (MG) contains both of them.

2.2.3 Temporal difference

Frame differencing is used for change detection in video sequences in many research works [Elhabian et al., 2008]. Therefore, in order to remove spatially static regions and small water waves, the temporal difference between two consecutive frames is taken into account. Majority of water waves dispersed in two consecutive frames are automatically suppressed by taking such inter-frame normalized differences, as:

$$\Delta_t I(\mathbf{x},t) = \frac{I(\mathbf{x},t) - I(\mathbf{x},t-1)}{255} \tag{2.4}$$

We experimentally set a threshold s_t, and create a temporal binary mask that can be expressed as follows:

$$\mathrm{dT}(\mathbf{x},t) = \begin{cases} 1 & \text{if} \quad |\Delta_t I(\mathbf{x},t)| \geq s_t \\ 0 & \text{otherwise} \end{cases} \tag{2.5}$$

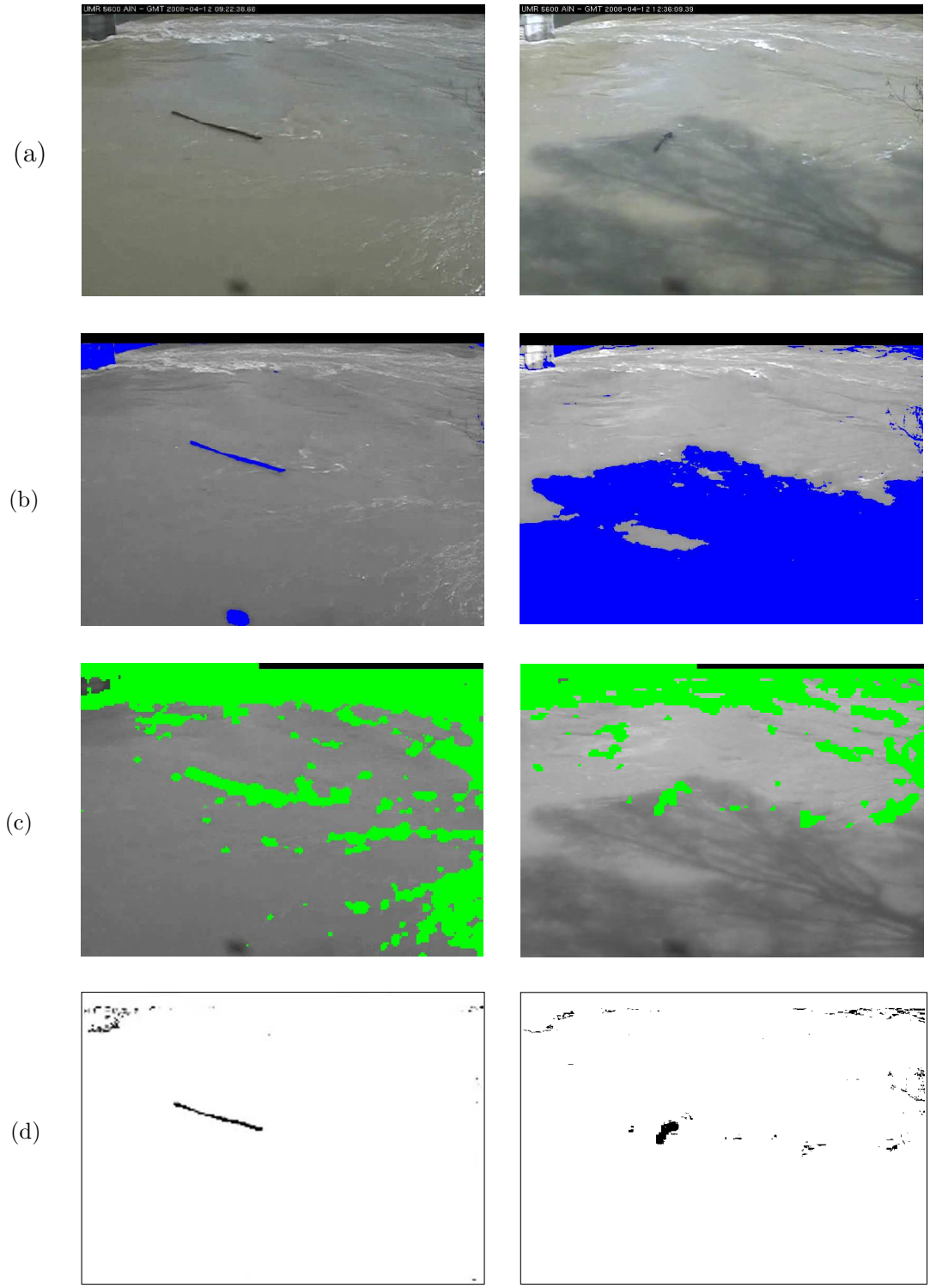


Figure 2.6: The representation of various steps involved in the segmentation, *(a)* original images, *(b)* intensity masks (MI) marked in blue color, *(c)* gradient masks (MG) in green color and *(d)* resulting combinations *i.e.* foreground ($\mathcal{F}$) for the corresponding images.

2.2.4 The resulting combination

In Figure 2.5, we show the histogram thresholding based intensity mask (MI), the edge based gradient mask (MG) and temporal inter-frame difference (dT) are combined to give a resulting image. We take the pixel wise intersection of three binary mask images to compute the final foreground image $\mathcal{F}$. We can write it as:

$$\mathcal{F}(\mathbf{x}, t) = \mathrm{MI}(\mathbf{x}, t)\mathrm{MG}(\mathbf{x}, t)\mathrm{dT}(\mathbf{x}, t) \tag{2.6}$$

The image $\mathcal{F}$ represents the detected wood objects along with some water waves. Two examples of foreground images are shown in Figure 2.6(d). The results obtained with this method are rather good if we keep in mind the dynamic background context. The method is presented in [Ali and Tougne, 2009]. For comparison, Figure 2.17 and 2.18 show the results obtained on the same scene with several background modeling techniques. However, the method have some shortcomings. These are in the form of misdetection within big woods and also false detected water waves. One of the possible reasons of such misdetection may be due to thresholds which are difficult to adjust for changing weather conditions. Secondly, the edge based intensity gradient works well on objects with stick-like apparency (*i.e.* small pieces, long trees), but for pieces with many leaves, the intensity gradient is not the discriminative feature. Thus, it is probable to miss some of inner regions of big wood objects. To overcome these shortcomings, we introduce a more general probabilistic object detection method in the next section, that relies on color and motion features of objects. The approach is not limited to wood detection application, however, for consistency with wood detection, we present the method here.

2.3 Probabilistic approach for object detection

As described in chapter 1, in fixed camera situations, background subtraction techniques are usually applied for object detection in many applications. In this kind of approach, a color based pixel-wise probabilistic representation of the scene is computed and each input frame is compared to this representation. The pixel-wise mismatch is computed between the current image and the background representation, which is thresholded afterwards, and the object pixels in the input image are extracted.

However, it must be noted that our approach has an advantage over existing background subtraction techniques to include the object color distribution. This property allows us to obtain better wood detection. Unlike the previous approach which uses bi-

nary masks, we introduce probability maps (with values ranging from 0 to 1) then can be combined by multiplication. This allows the thresholding step to be pushed back at the end of the process, which is theoretically most robust than applying several thresholds. In the following paragraphs, we present object detection method as Bayesian estimation problem.

Formally, the goal of video segmentation is to create a foreground image $\mathcal{F}(t)$ for each image $I(t)$ at time t. Current image is denoted by $I(t)$, where $I(\mathbf{x}, t)$ denotes the color of a single spatiotemporal pixel $(\mathbf{x}, t)$. A pixel $\mathbf{x}$ in $\mathcal{F}(t)$ is labeled either 0 or 1 according to its belonging to background or to foreground respectively. Notice that at time t, in addition to the current image $I(t)$, the sets of previous images $\mathbf{I}(t-1) = \{I(i)\}_{1 \leq i \leq t-1}$ and segmentations $\mathbf{F}(t-1) = \{\mathcal{F}(i)\}_{1 \leq i \leq t-1}$ are available. General pixel-wise foreground image segmentation at time t can be formulated by thresholding the following *a posteriori* probability at each pixel $\mathbf{x}$ [Li et al., 2004]:

$$P(\mathcal{F}(\mathbf{x}, t) | I(\mathbf{x}, t), \mathbf{I}(t-1), \mathbf{F}(t-1)) > s' \tag{2.7}$$

This probability accounts for the temporal consistency of image and segmentation. In a very general setting, the *a posteriori* probability is conditioned on the entire previous images and segmentations. Using Bayes' rule, we can thus write

$$\begin{aligned} & P(\mathcal{F}(\mathbf{x}, t) | I(\mathbf{x}, t), \mathbf{I}(t-1), \mathbf{F}(t-1)) \\ & = \frac{P(I(\mathbf{x}, t) | \mathbf{I}(t-1), \mathcal{F}(\mathbf{x}, t), \mathbf{F}(t-1)) P(\mathcal{F}(\mathbf{x}, t) | \mathbf{I}(t-1), \mathbf{F}(t-1))}{P(I(\mathbf{x}, t) | \mathbf{I}(t-1), \mathbf{F}(t-1))} \end{aligned}$$

We can ignore denominator as it is independent of $\mathcal{F}(t)$, so the segmentation process can be written as follows:

$$\underbrace{P(I(\mathbf{x}, t) | \mathcal{F}(\mathbf{x}, t), \mathbf{I}(t-1), \mathbf{F}(t-1))}_{\text{Image term}} \underbrace{P(\mathcal{F}(\mathbf{x}, t) | \mathbf{I}(t-1), \mathbf{F}(t-1))}_{\text{Prior term}} > s \tag{2.8}$$

The image term (or image model) is the likelihood that the pixel $\mathbf{x}$ has intensity value in image I conditioned on the fact that it belongs to foreground. This is related to intensity or color distribution inside the objects or background. The prior term in Eq. 2.8 is the probability of a pixel $\mathbf{x}$ to belong to an object, knowing the previous images and segmentations, independently of the current image.

In the following, the image term will be denoted by $P_{\text{image}}(\mathbf{x}, t)$ and the prior term

by $P_{\text{mov}}(\mathbf{x}, t)$. The combination will be denoted by $P_{\text{obj}}(\mathbf{x}, t)$, it can be expressed as:

$$P_{\text{obj}}(\mathbf{x}, t) = P_{\text{image}}(\mathbf{x}, t) \cdot P_{\text{mov}}(\mathbf{x}, t)$$

The general term $P_{\text{image}}(\mathbf{x}, t)$ can be a probabilistic representation of foreground or background. In this chapter, we suppose a probabilistic representation of foreground whereas in chapter 3, we explain a modified GMM method as an image model and integrate it with motion model for object detection. We do not consider $P_{\text{mov}}(\mathbf{x}, t)$, it will be studied in chapter 3.

2.4 Image model for wood

In section 2.2, we have proposed a naïve approach for wood detection [Ali and Tougne, 2009]. It gives satisfactory segmentation results, however, final segmented images contain some false detected water waves. Therefore, to reduce false detections and to improve wood segmentation, we develop a probability based method [Ali et al., 2011].

We refer to Eq. 2.8 in which the image model is expressed as:

$$P_{\text{image}}(\mathbf{x}, t) = P(I(\mathbf{x}, t) | \mathcal{F}(\mathbf{x}, t), \mathbf{I}(t-1), \mathbf{F}(t-1)) \tag{2.9}$$

This is a general expression for the image model. This represents the likelihood that the pixel $\mathbf{x}$ has intensity or color value in current image I conditioned on the fact that it belongs to the foreground. This is related to intensity or color distribution inside the objects or background.

In wood detection application, image model uses wood (*i.e.* foreground) intensity distribution. Thus, image model for wood is a pixel-based probabilistic approach based on intensity and its temporal variation, which can be expressed as:

$$P_{\text{image}}(\mathbf{x}, t) = P_i(\mathbf{x}, t).P_t(\mathbf{x}, t) \tag{2.10}$$

The intensity probability map $P_i(\mathbf{x}, t)$ is the likelihood of the pixel to be wood with respect to its brightness, whereas the temporal probability map $P_t(\mathbf{x}, t)$ contains this information with respect to the brightness temporal variations at each pixel level. Basically, we rely on two observations: wood is darker than water and undergoes permanent motion. We explain the assumptions based on our observations on river videos for image model in the following paragraphs.

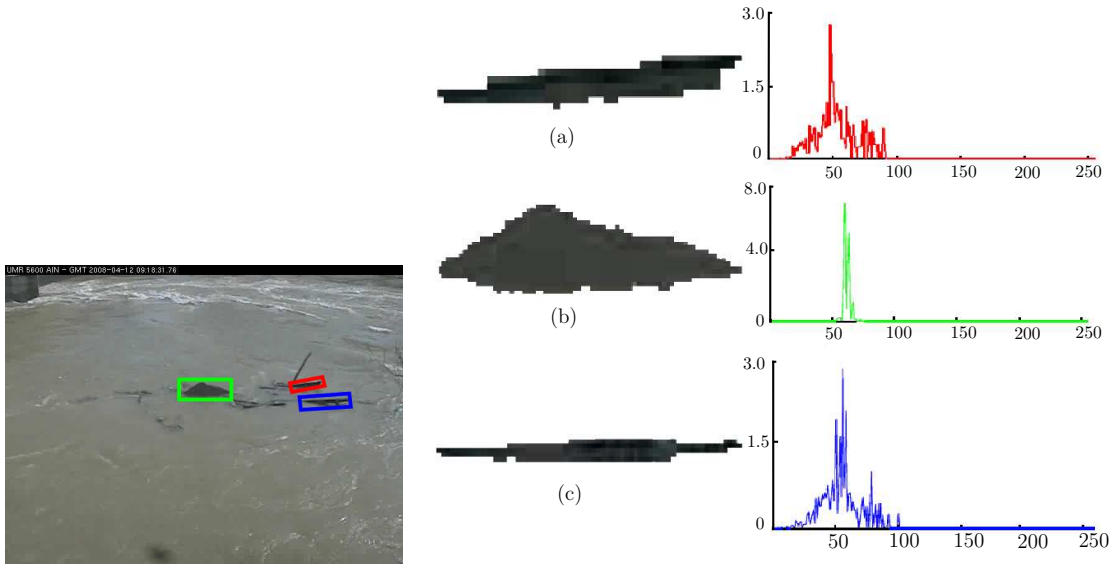


Figure 2.7: Floating wood piece; *(a)* Zoomed portion of wood pixels and corresponding intensity histogram highlighted by red, *(b)* green and *(c)* blue rectangle.

2.4.1 Intensity probability map

In section 2.1.2, we summarized constraints involved in the application. We observe in the videos that the brightness of floating wood pieces is lower than water, even under the shadows of surrounding trees. Moreover, it does not change significantly in the presence of sunlight. Figure 2.7 shows intensity histograms of wood pieces as an example. It seems relevant to approximate the intensity distribution of wood by a Gaussian distribution with a fixed mean and variance (*i.e.* $\mu_{\text{wood}}, \sigma^2_{\text{wood}}$). The probability of the current pixel to belong to wood with respect to its intensity is:

$$P_i(\mathbf{x}, t) = \mathcal{N}(I(\mathbf{x}, t), \mu_{\text{wood}}, \sigma^2_{\text{wood}})$$

$$\mathcal{N}(I(\mathbf{x}, t), \mu_{\text{wood}}, \sigma^2_{\text{wood}}) = \frac{1}{\sqrt{2\pi\sigma^2_{\text{wood}}}} \exp\left(-\frac{(I(\mathbf{x}, t) - \mu_{\text{wood}})^2}{2\sigma^2_{\text{wood}}}\right) \tag{2.11}$$

where $\mathcal{N}$ is a Gaussian probability density function. To find μ_{wood} and σ^2_{wood}, we led experiments on different wood pieces under various lighting conditions, which is discussed in section 2.4.4. Figure 2.8 shows a few examples of intensity probability maps where the $[0, 1]$ range is represented with a color map. In the presence of cast shadows of surrounding trees, the intensity probability map P_i has high values in wood regions but also in undesirable shadowed regions.

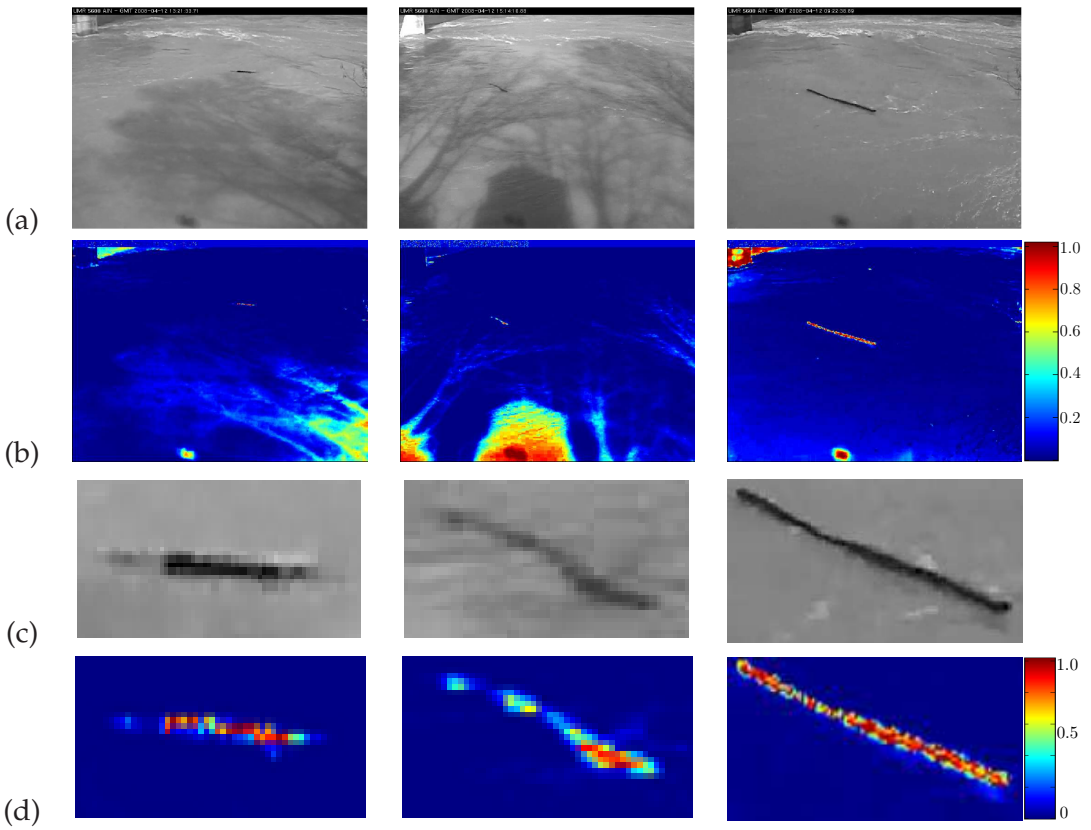


Figure 2.8: *(a)* Few wood images from river videos with different lighting conditions with corresponding: *(b)* Intensity probability map P_i, *(c)* zoomed wood regions, *(d)* Intensity probability map P_i of zoomed wood regions.

2.4.2 Temporal probability map

Wood cannot be extracted relying solely on intensity considerations. Indeed, some objects like bridge pillars or cast shadows of surrounding trees have the same intensity as wood. To remove these static objects, we rely on pixel-wise temporal variations of intensity.

Inter-frame difference was already used in our previous naïve approach discussed in section 2.2. The temporal probability map is an improvement of the difference mask (dT). It is partially based on the normalized inter-frame difference $\Delta_t I$ expressed in Eq.2.4, which takes its values within range $[-1, 1]$. Hard thresholding the absolute inter-frame difference $|\Delta_t I|$ has been extensively tested for object detection. By nature, this technique only detects new object pixels and inevitably removes object areas that overlap in time. This was the case with naïve approach with big wood pieces. Our temporal probability P_t is defined in order to avoid this drawback. We design it according to the observation that, when wood passes through a given pixel, $\Delta_t I$ dips to a negative

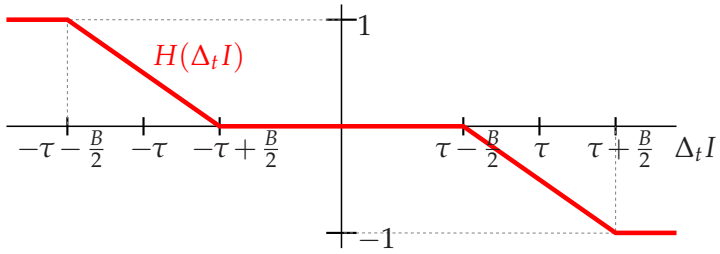


Figure 2.9: Representation of updating function $H(\Delta_t I)$.

value and then to a positive value afterwards. Moreover, P_t should naturally remain constant if $\Delta_t I = 0$. This is achieved using a recursive definition in time:

$$P_t(\mathbf{x}, t) = P_t(\mathbf{x}, t-1) + H(\Delta_t I(\mathbf{x}, t)) \tag{2.12}$$

where $H \in [-1, 1]$ is an updating function, mapping the inter-frame difference to the amount of changes in the temporal probability. We express it in accordance with the considerations previously addressed. To handle noise and ignore insignificant intensity variations due to the non-uniformity of wood or water, $H(\Delta_t I)$ should be null for relatively small values of $|\Delta_t I|$. It allows to handle slow illumination variations as well. Beyond certain threshold value, H should increase or decrease as $\Delta_t I$ gets significantly negative or positive, respectively. Instead of using hard thresholding which would cause H to jump suddenly from 0 to 1 or -1, we use a soft approach less critical with respect to the choice of threshold parameters leading to the following piecewise linear definition:

$$H(\Delta_t I) = \begin{cases} 1 & \text{if } \Delta_t I \in [-1, -\tau - \frac{B}{2}] \\ \alpha \Delta_t I + \beta & \text{if } \Delta_t I \in [-\tau - \frac{B}{2}, -\tau + \frac{B}{2}] \\ 0 & \text{if } \Delta_t I \in [-\tau + \frac{B}{2}, \tau - \frac{B}{2}] \\ \alpha \Delta_t I - \beta & \text{if } \Delta_t I \in [\tau - \frac{B}{2}, \tau + \frac{B}{2}] \\ -1 & \text{if } \Delta_t I \in [\tau + \frac{B}{2}, 1] \end{cases} \tag{2.13}$$

where $\alpha = \frac{-1}{B}$ and $\beta = \frac{1}{2} - \frac{\tau}{B}$. Definition of H in turns requires a threshold τ and transition length B which are illustrated in the plot of Figure 2.9. The choice of τ and B is discussed in section 2.4.4. It should be noted that $P_t(\mathbf{x}, t)$ in Eq. 2.12 is truncated between 0 and 1 afterwards to remain a probability. In the first frame, we set $P_t(\mathbf{x}, 1)$ equals to 0 everywhere, as it is very unlikely that wood pieces appear at initial time. Temporal probability P_t is non-null only if temporal brightness variation is negative enough, *i.e.* if a pixel gets significantly darker or has the same brightness as it had in the

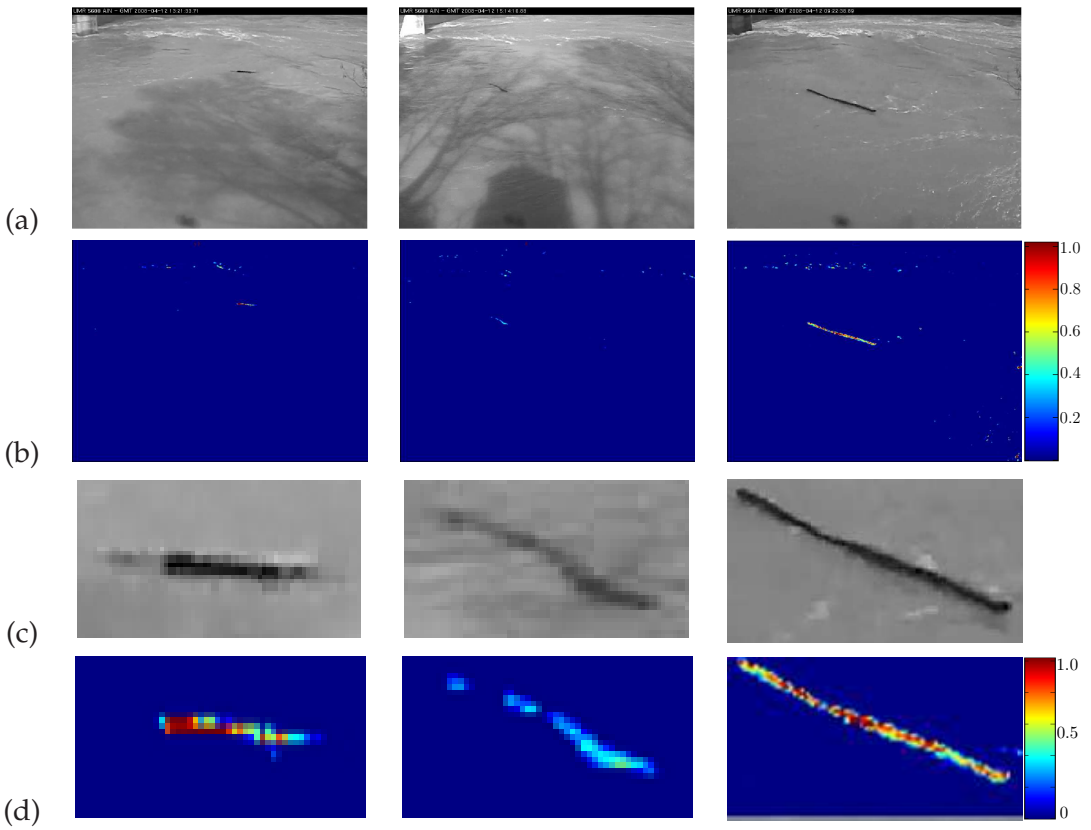


Figure 2.10: *(a)* Few wood images from river videos with different climatic conditions with corresponding: *(b)* Temporal probability map P_t, *(c)* zoomed wood regions, *(d)* Temporal probability map P_t of zoomed wood regions.

previous frame. It helps in removing stationary objects in the scene (*e.g.* pillars of bridge) but preserves big objects. Figure 2.10 shows few examples of temporal probability maps. It highlights the fact that P_t has higher values for the wood pieces than for water or static areas. Hence, it enables to detect moving wood objects and to eliminate many water waves and cast shadows effects.

2.4.3 Combination of intensity and temporal probability maps

Since we expect wood to be simultaneously dark and under motion, wood pixels should have both high intensity and temporal probabilities, hence it is relevant to multiply the two probability maps. According to Eq. 2.10, we ignore prior term here and the equation becomes

$$P_{\text{obj}}(\mathbf{x}, t) = P_{\text{image}}(\mathbf{x}, t) = P_i(\mathbf{x}, t) \cdot P_t(\mathbf{x}, t)$$

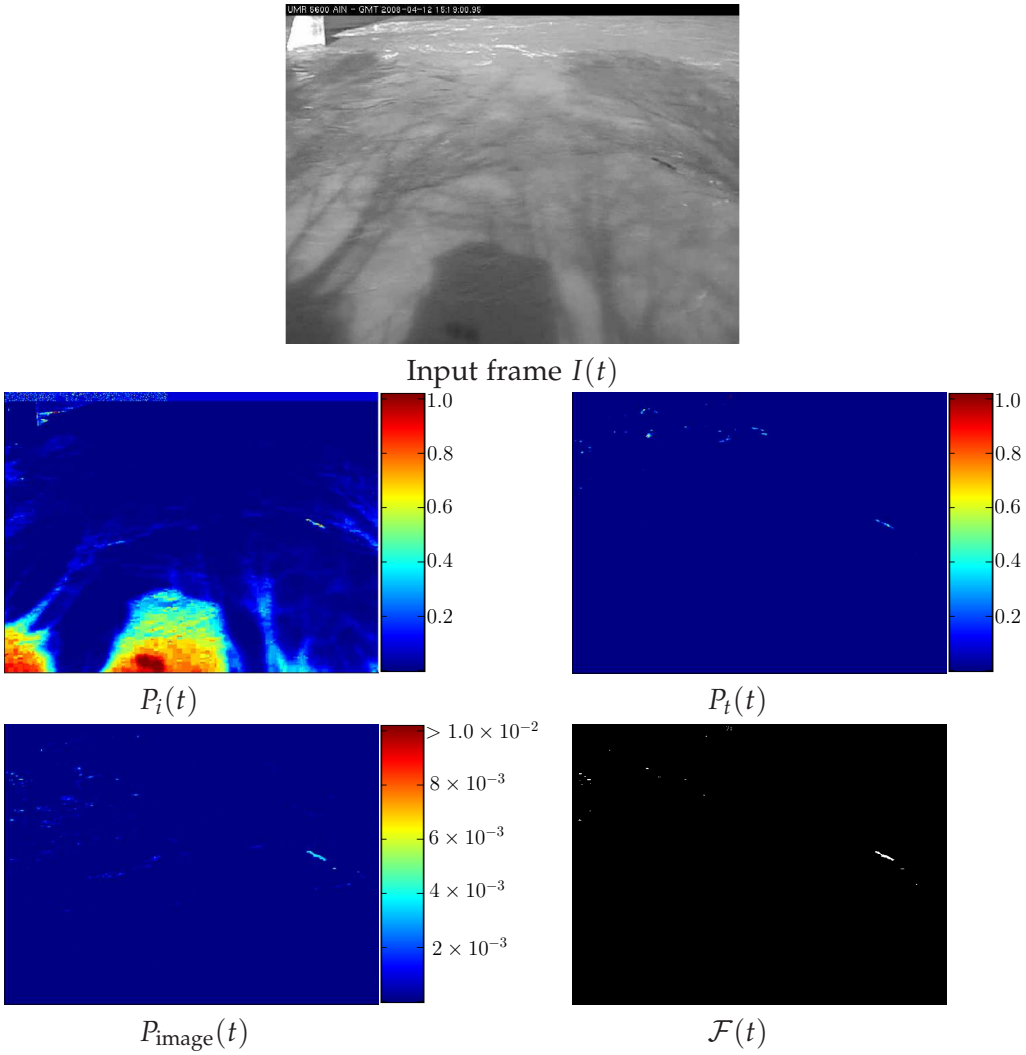


Figure 2.11: An example of wood piece under cast shadows in sunlight with corresponding, intensity probability map $P_i(t)$, temporal probability map $P_t(t)$, (c) image model $P_{\text{image}}(t)$ and resulting foreground image $\mathcal{F}(t)$.

and the foreground image $\mathcal{F}$ is obtained by simple thresholding the joint probability map:

$$\mathcal{F}(\mathbf{x},t) = \begin{cases} 1 & \text{if} \quad P_{\text{image}}(\mathbf{x},t) \geq G_{Th} \\ 0 & \text{otherwise} \end{cases} \tag{2.14}$$

which is illustrated in Figure 2.11. The image probability P_{image} is high for wood pixels but also unfortunately for some pixels located on dark waves. Hence, global threshold G_{Th} should be chosen in order to limit the number of false detections without removing significant parts of real wood pieces (choice of G_{Th} is discussed in section 2.4.4). This example of final foreground image, clearly indicates that the algorithm can detect moving wood pieces even under difficult weather conditions (with a lots of shadows for

example).

2.4.4 Selection of parameters

The computation of the two probability maps (*i.e.* P_i and P_t) involves some parameters and a final threshold value for final foreground extraction. These parameters are selected in a way to obtain best wood segmentation in different weather conditions and at different day time.

First, we explain the parameters for computing intensity probability map P_i. For this purpose, we extract small and large wood pieces under different weather conditions. An example of floating wood divided in three portions highlighted in different colors is shown in Figure 2.7. We experimentally fix $\mu_{\text{wood}} = 55$ and $\sigma^2_{\text{wood}} = 225$ which are optimal values for wood detection.

The computation of the temporal probability involves a threshold τ and the transition length B, and extraction of final foreground image requires a global threshold G_{Th}. Parameter tuning is performed through a brute-force approach, by maximizing the overlap between the foreground image generated with current parameter values on one hand and ground truth segmentations on the other hand, on a training dataset (described in section 2.5.1). The overlap is measured using the Dice similarity measure S, which is a commonly used to evaluate image segmentation quality (see for example [Cardenes et al., 2008, Babalola et al., 2008]). It is expressed as

$$S = \frac{2 \mid X \cap Y \mid}{\mid X \mid + \mid Y \mid} \tag{2.15}$$

where X is the result of image segmentation and Y is the corresponding ground truth image. S is equal to 1 when the segmented region and the ground truth region perfectly overlap, and 0 when they are disjoint.

In the first step, for each parameter, the range of values giving satisfactory results is coarsely located by successive attempts. We vary the parameters (τ, B, G_{Th}) and compute final foreground image $\mathcal{F}$ for each parameter vector value. We determine that the triplet (τ, B, G_{Th}) leading to the best segmentation is located in range $[0.1, 0.4] \times [0.1, 0.4] \times [0.001, 0.02]$. Afterwards, all parameter values within these ranges are tested, with respective steps 0.05, 0.05 and 0.001. The optimal values for these parameters are $\tau = 0.3$, $B = 0.3$ and $G_{Th} = 0.002$. The values are computed for small and big wood pieces extracted from original river video. Also, we make synthetic videos for parameter tuning, these videos are explained in the following section. The average Dice value (sum of individual dice coefficient per image divided by number of images) from entrance to

exit for both small and big wood pieces is maximal. The segmented image based on intensity and temporal features contains floating wood pieces and also a few undesirable remaining waves.

2.5 Results and comparison with other methods

We present image segmentation results which are obtained by applying the image model to synthetic and real videos. In order to show that the image model works equally well under different weather conditions, different wood objects sizes and either on synthetic or real videos. Real videos are extracted from MPEG4 compressed stream and from a monitoring system that is installed on river Ain, France. Video frame size is 640×480 with frame rate of less than 4 frames per second. The algorithm is tested on an Intel Core2 Duo 2.66GHz with 4GB RAM running C code.

Existing background modeling methods are also taken into consideration, for comparison. We apply three background modeling techniques: Mixture of Gaussians (GMM) [Stauffer and Grimson, 2000], the Codebook method [Kim et al., 2005] and the VuMeter method [Goyat et al., 2006].The results of these algorithms are discussed one by one. The final segmentation results of these background models are compared with the results of our image model.

2.5.1 Generation of synthetic videos

The image model is tested on synthetic videos before applying on real videos. The reason for this experimentation is to find out optimal values for the model parameters and final threshold G_{Th}. We choose two wood pieces of small and large sizes extracted from original video. These objects are moved synthetically over real background images. Similarly, the background images for these videos represent two types of scenarios. In the first type, small and large wood are synthetically moved in an image sequence without sunlight. In the second type of scenario, these wood objects are moved in the image sequence with sunlight. Objects are rotated and translated in a way to best approximate the motion encountered in real videos.

Figure 2.12 and 2.13 show scenarios of big and small wood pieces, respectively, in a synthetic video. For each scenario, translation alone and translation plus rotation were considered. So, there are 8 synthetic videos on which we perform our experiments.

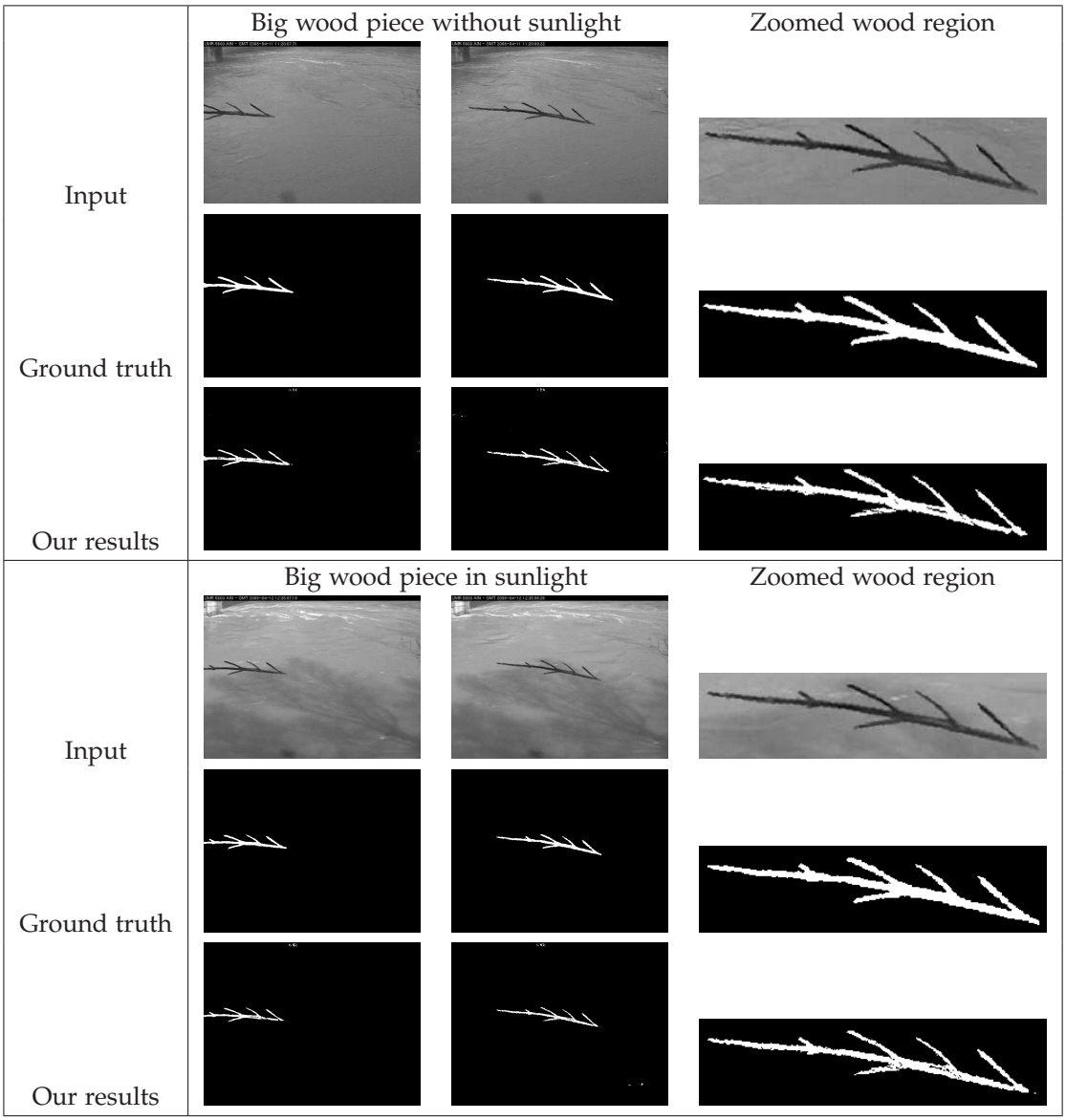


Figure 2.12: Synthetic video images of a big floating wood *Top subfigure:* without sunlight, *Bottom subfigure:* with sunlight, corresponding ground truth images and results of image model. *Third column:* wood regions are zoomed to show wood segmentation.

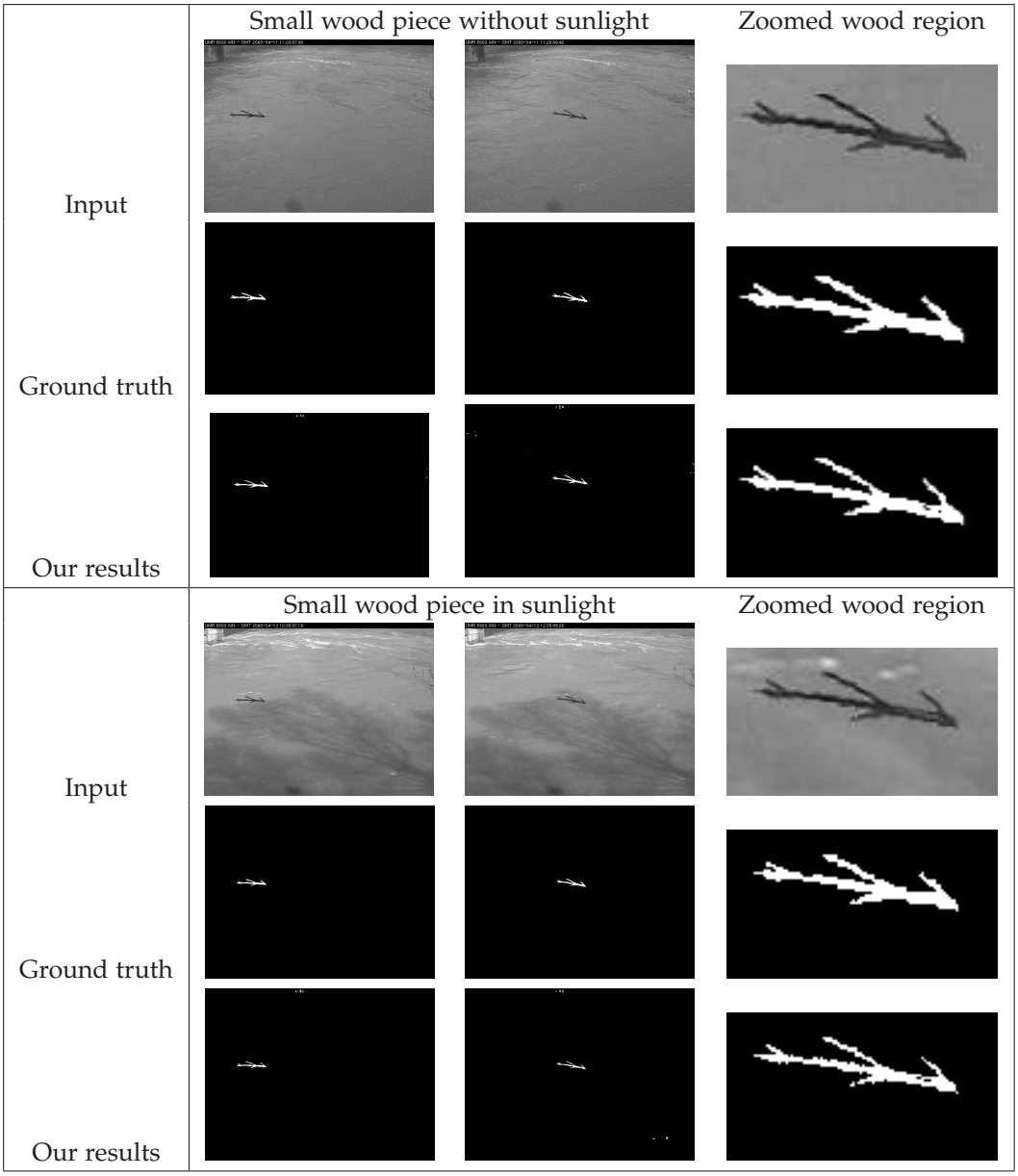


Figure 2.13: Synthetic video images of a small floating wood *Top subfigure:* without sunlight, *Bottom subfigure:* with sunlight, corresponding ground truth images and results of image model. *Third column:* wood region is zoomed to show wood segmentation.

We show here two images from the 15 images in total per object (*i.e.* from entrance to exit of wood object from the scene). We show the results of our image model and corresponding ground truth images. We can notice that the image model is equally applicable for small and big wood objects with different background conditions. Moreover, the wood regions are zoomed in the Figure to highlight the image segmentation results for the corresponding wood objects.

The ground truth images are available for synthetic moving objects. Therefore, these are used to evaluate the image segmentation results. The Dice similarity measure is computed for each image. The parameters are so selected that produce high similarity values.

2.5.2 Comparison with the GMM method

River videos are subjected to classical background modeling techniques for comparison with our image model. The first background model we apply on the river videos is the Gaussian Mixture Models (GMM) algorithm of [Stauffer and Grimson, 2000] presented section 1.1.2. The model parameters are selected in order to obtain less number of water waves in the output images. To achieve this, we select a relatively high learning rate $\alpha = 0.05$ and $K = 5$ which refers to the number of Gaussians per pixel (as explained in section 1.1.2.1). For visual comparison, the resulting foreground images obtained with GMM for few wood objects are shown in the Figures 2.17(b) and 2.18(b).

2.5.3 Comparison with the Codebook method

We also compare with the codebook algorithm developed by [Kim et al., 2005]. The method is explained in section 1.1.2.2. For the learning period, we use the first 100 video frames with no wood objects. Minimum and maximum brightness assigned to the codewords are $I_{min} = 10$ and $I_{max} = 50$ respectively. Maximum negative run length $\lambda = \tau/2$, with $\tau = 500$ for river video experiments. These parameters are tuned heuristically in order to minimize false detection.

A few examples of the foreground images obtained with this method are presented in Figures 2.17(c) and 2.18(c).

2.5.4 Comparison with the VuMeter method

Third background model that we experiment on river videos is the VuMeter (VM) background model proposed by [Goyat et al., 2006] and described in section 1.1.2.2. Foreground images are shown in Figures 2.17(d) and 2.18(d). The parameters are tuned here

as well to minimize the false detection. The learning rate in the Vumeter algorithm is set to $\alpha = 0.005$ and the threshold is $T = 0.1$ for river videos. However, the method also fails to suppress false detection.

2.5.5 Qualitative evaluation

Final foreground results obtained with our image model (IM) are shown in the last rows of Figures 2.17 and 2.18. The image model generates much less false detection than all background models. One of the reasons of good wood segmentation with image model is that we use a fixed wood intensity distribution in the wood image model. This is a strong prior information which is not used in the GMM, the codebook and the VuMeter. Therefore, these background models detect a lot of water waves.

2.5.6 Quantitative evaluation

We have tested the various algorithms both on synthetic and real videos.

Synthetic video: We compare the wood segmentation results of our method with GMM method on synthetic videos. In the synthetic videos, there are two wood pieces, shown in Figure 2.12 and 2.13. We plot the Dice coefficient values for whole image sequences of big and small wood pieces of Figure 2.14 and 2.15, as a function of time.

As can be seen in Figure 2.14 and 2.15, the GMM leads to less accurate segmentation than our method.

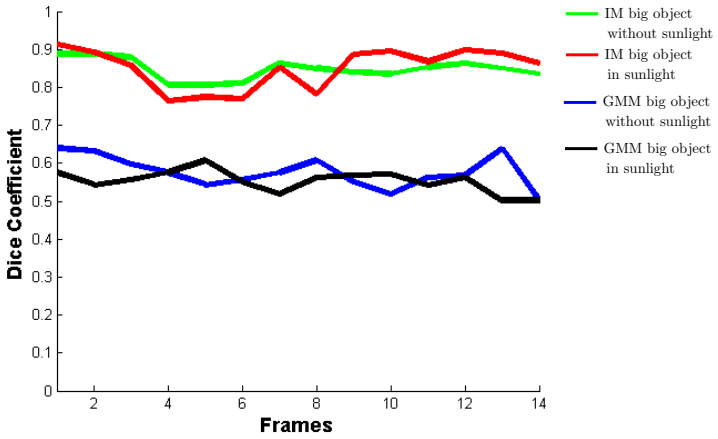


Figure 2.14: Comparison of Dice similarity coefficient per frame for image segmentation results of Mixture of Gaussian (GMM) and our Image Model (IM) for synthetic video of big object shown in Figure 2.12.

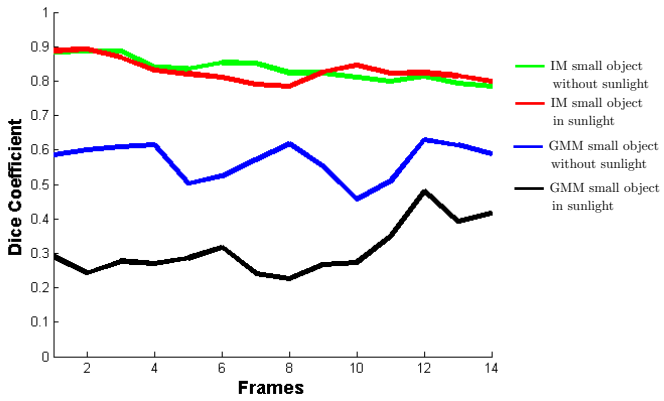


Figure 2.15: Comparison of Dice similarity coefficient per frame for image segmentation results of Mixture of Gaussian (GMM) and our Image Model (IM) for synthetic video of small object shown in Figure 2.13.

On the other hand, the Dice similarity values for our image model (IM) results are high. Average Dices value for big object in all image sequences is 0.87 and for small object, it is 0.83. The results clearly indicate the utility of image model in wood detection.

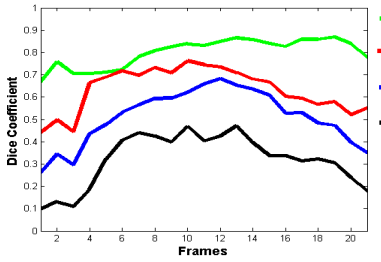
Real videos: The quantitative comparison of results between different existing background methods and our image model is performed by using the Dice similarity measure. We manually generated ground truth images for the image sequence containing the wood objects which are shown in Figure 2.16. The wood object was marked by hand in each of these frames. We show six wood pieces which appear in different number of frames during their passage in videos. Dice coefficient values per frame are also plotted. We also give the minimum, maximum and average Dice values for each the corresponding methods. Higher values of Dice coefficient signify the good segmentation results. We can see that with image model both small and large wood pieces we obtain good segmentation which is reflected in high Dice values. Our image model significantly improves segmentation accuracy over other existing approaches. In the last wood example, image model results are lower than GMM and VueMeter in few frames. It is due to water waves detection with similar intensity values. These results are improved with the use of motion model which is explained in chapter 3 (see Figure 3.11).

In most of the circumstances, wood objects are submerged in water due to their weight. Therefore, most of wood pieces are partially occluded in these videos. Therefore, the proposed method must also detect small wood pieces equally well. Small wood pieces are more prone to be confused with waves. The separation between the two is

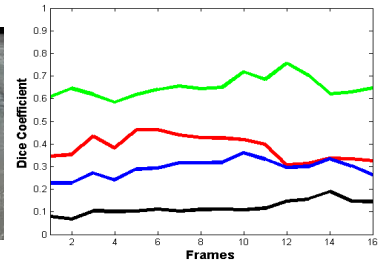
necessary for successful wood tracking and counting (see appendix A). In this way, our image model produces promising results. Even though the results of our Image Model are better than existing background modeling techniques, there are a few mis-detected pixels and a few water waves present in the final foreground image. The water waves appear randomly. This is the reason why we decide to use wood motion information to segregate water and wood movements and improve wood detection.

2.6 Conclusion

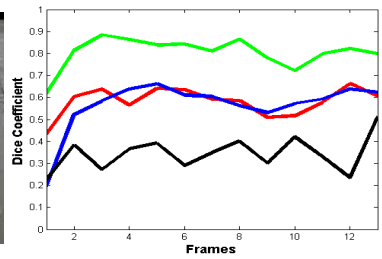
In this chapter, we have presented an image segmentation model dedicated to wood detection. Wood pieces are often submerged into water and the wood above the water surface usually owns small apparent area. These small apparent wood pieces along with large pieces should be detected correctly. We proposed two methods for this purpose, based on the observations that wood is darker than water and in continuous motion. In the first approach, we used two image segmentation techniques applied on each incoming frame and combined them to extract foreground objects. In the second method, we proposed a probabilistic image model for wood segmentation. The results of classical background models are compared with our method. We showed that three background models namely, the Gaussian Mixture Models (GMM), the Codebook method (CB) and the VuMeter method (VM) were not able to produce good object detection rates. The results are evaluated both qualitatively and quantitatively using Dice similarity measure. Despite of better results, there are few mis-detected pixels with image model. Also, few water waves are still present in the results. So, to overcome these difficulties we explore the possibility of incorporating an object motion model based on prior object motion knowledge for their detection.

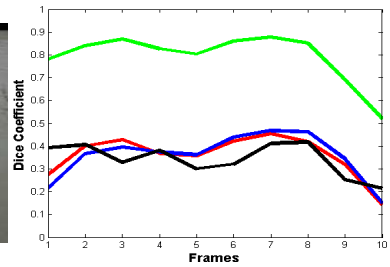
	Dice Coefficient values		
	Max	Min	Average
GMM	0.76	0.44	0.63
VuMeter	0.68	0.26	0.51
CB	0.47	0.10	0.32
IM	0.86	0.66	0.79

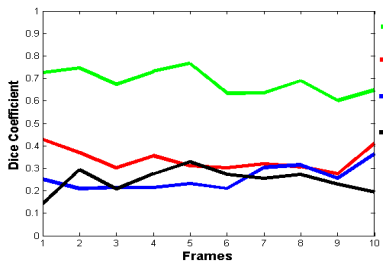
	Dice Coefficient values		
	Max	Min	Average
GMM	0.46	0.30	0.38
VuMeter	0.36	0.22	0.29
CB	0.19	0.07	0.29
IM	0.75	0.58	0.65

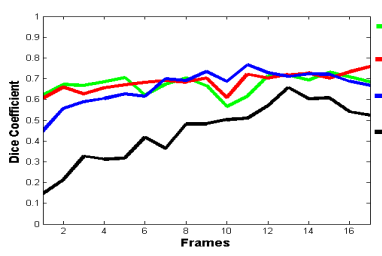
	Dice Coefficient values		
	Max	Min	Average
GMM	0.66	0.43	0.58
VuMeter	0.66	0.19	0.56
CB	0.51	0.22	0.35
IM	0.88	0.61	0.80

	Dice Coefficient values		
	Max	Min	Average
GMM	0.45	0.14	0.35
VuMeter	0.46	0.15	0.36
CB	0.42	0.21	0.34
IM	0.87	0.52	0.79

	Dice Coefficient values		
	Max	Min	Average
GMM	0.43	0.27	0.33
VuMeter	0.36	0.21	0.26
CB	0.33	0.14	0.24
IM	0.76	0.60	0.68

	Dice Coefficient values		
	Max	Min	Average
GMM	0.75	0.60	0.68
VuMeter	0.76	0.44	0.66
CB	0.66	0.16	0.45
IM	0.73	0.56	0.67

Figure 2.16: Comparison of Dice coefficient per frame for image segmentation results obtained with Mixture of Gaussian (GMM), CodeBook (CB), VuMeter (VM) and our Image model (IM) for the corresponding wood objects.

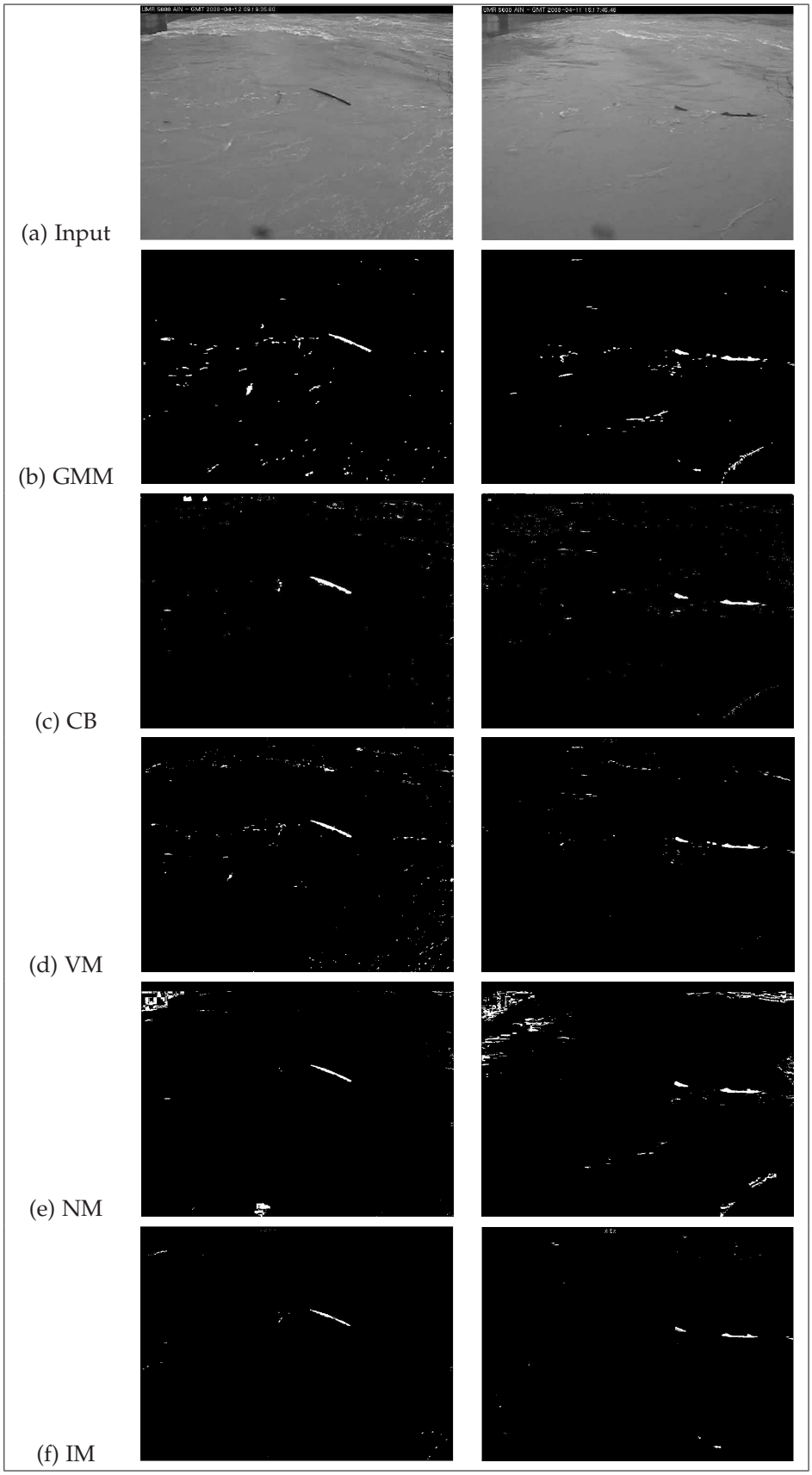


Figure 2.17: *(a)* Original images with corresponding results of, *(b)* Mixture of Gaussian (GMM), *(c)* Codebook (CB), *(d)* VuMeter (VM), *(e)* Naïve Method (NM) and *(f)* our Image Model (IM).

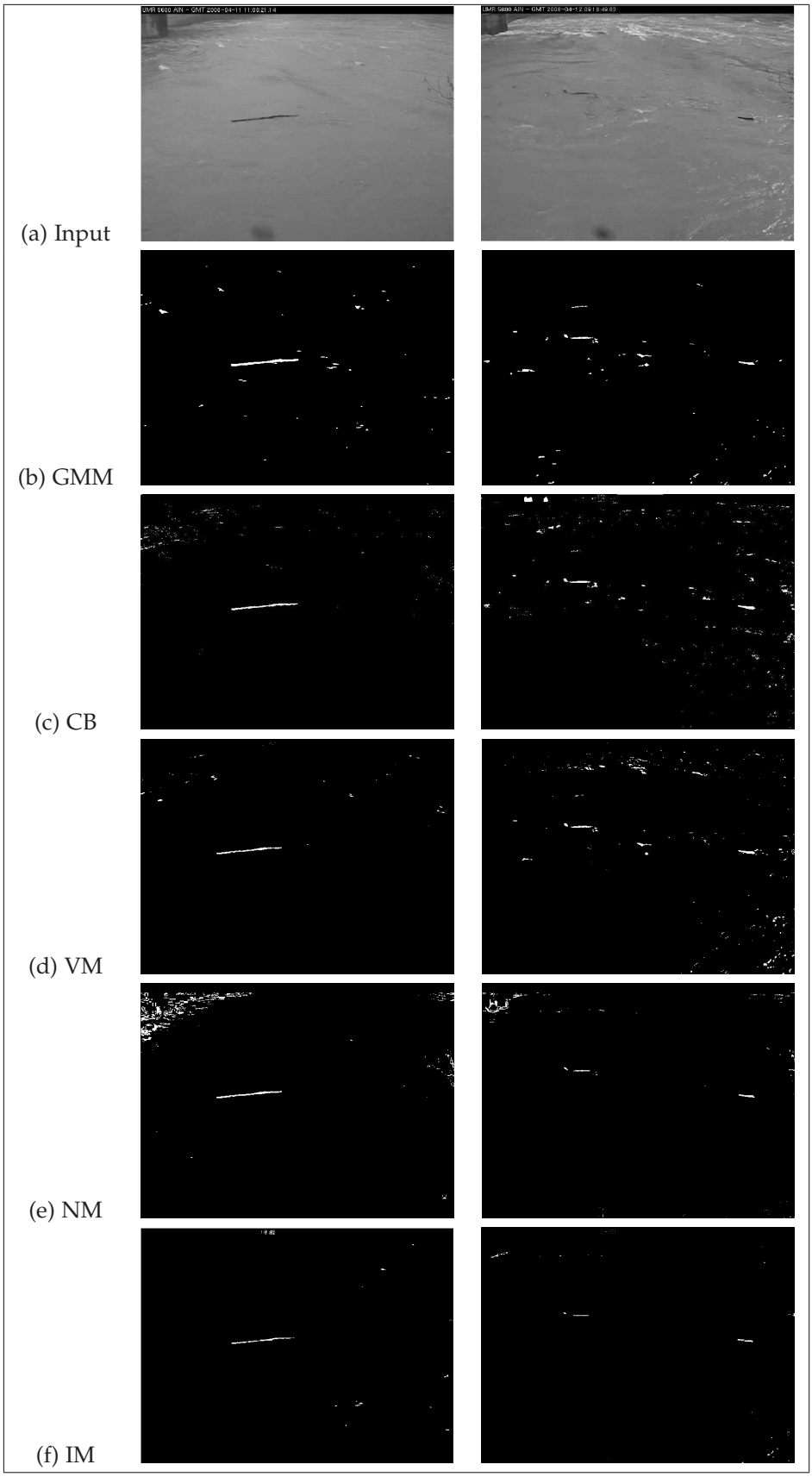


Figure 2.18: *(a)* Original images with corresponding results of, *(b)* Mixture of Gaussian (GMM), *(c)* Codebook (CB), *(d)* VuMeter (VM), *(e)* Naïve Method (NM) and *(f)* our Image Model (IM).

Chapter 3

Object motion model

Contents

As presented in section 1.3, there are many object detection algorithms in the literature that use motion characteristics. Statistical motion models were proposed relying on the motion information. Motion of objects can be either estimated, *e.g.* relying on the optical flow, or constrained, in order to help detection. In some applications, prior knowledge about the expected motion of objects is available. We propose a model that uses such knowledge that can be combined with the image-based object detection algorithm. We use it as an additional information along with color information for object detection. The inclusion of a motion model in the object detection process aims to improve the distinction between moving objects and backgrounds.

3.1 Prior knowledge about object motion

Object motion knowledge is an application-dependent entity. It is related to the way objects move in the monitored scene. More precisely, in a monitoring system where the position of the camera is known and the usual trajectories in the actual $3D$ space are known as well, one may have a strong prior knowledge about the apparent trajectories of expected objects in the image plane. For example, when one wants to detect luggages moving on conveyor belts in airports, luggages move with speed and towards a direction that may be known. Similarly, during normal traffic flow on the road, vehicles motion can be known *a priori*. Floating objects in the river water is another example of object motion where object motion knowledge can be obtained. In Figure 3.1, we show a representative image of each of these situations. It can be noticed that in the floating objects in rivers, there is a global motion in the scene (for example floating objects and water move with same speed and in the same direction, here from the right to the left).

We are interested in global object motion, rather than pixel-wise motions. In section 2.3, we have introduced the general probabilistic object detection method, which is composed of two terms (*i.e.* image and prior terms). The current chapter is devoted to an im-

Figure 3.1: Luggage on conveyor belt, a moving car on road and a floating bottle in river are few examples in which prior object motion can be obtained.

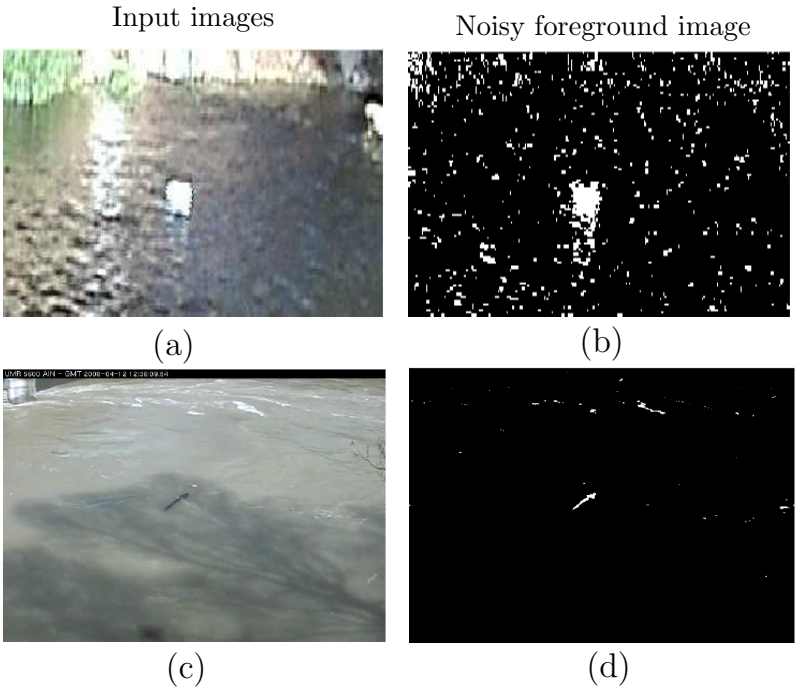


Figure 3.2: *(a)* An image of floating bottle in river, *(b)* the GMM [Stauffer and Grimson, 2000], *(c)* a wood object in river, *(d)* result of wood image model without motion model.

plementation of the prior term. We choose object motion knowledge to be modeled and used as prior term. Along with color features, motion features of objects can also be used for their detection in the videos. In aquatic environment, for example, the perturbation in the background is very large. Due to these perturbations, foreground/background scene segmentation is very difficult. Therefore, in order to improve background subtraction and to decrease the number of false positives due to misclassified pixels, the use of prior knowledge about motion when available, seems relevant. We propose a method to model the prior object motion knowledge. We suppose that motion information is available or can be learned during a training period. The prior motion model is designed so that it can be integrated into various background subtraction methods. In Figure 3.2*(b)* and *(d)*, we present results of classic color-based subtraction methods, which contain noise. We will explain that the results can be improved using prior motion knowledge in the following sections. The detailed implementation of motion model integration into these two image models is explained in the sections 3.4 and 3.5.

Object displacement in the image sequence depends on object speed in real world, but also on video frame rate. Frame rate in standard videos is 25 frames per second (fps), in such videos, object displacement is of a few pixels between two consecutive frames, as reported in many articles, for example in [Elgammal and Davis, 2001]. However, in

low frame rate videos, ($\leq$ 4 fps), object displacements are large.

Learning object motion in training period is one of the methods by which one can acquire object motion knowledge. The variations should be taken into consideration by the motion model that come from the object sizes, speed and video frame rate. We assume that between two consecutive frames, object motion can be approximately considered as rigid motion.

3.2 Prior motion knowledge and motion estimation

We explain the difference between prior motion knowledge and motion estimation in video analysis. To elaborate the difference, we refer to section 2.3, where we have introduced the general point of view of video segmentation. There are two probability terms expressed in Eq. 2.8 (*i.e.* image and prior) for each pixel. On one hand, the image term (or image model) is the likelihood that pixel $\mathbf{x}$ has intensity/color value in image I conditioned on the fact that it belongs to foreground. On the other hand, the prior term $P_{\text{mov}}(\mathbf{x}, t)$ is the probability of pixel $\mathbf{x}$ to belong to an object, knowing the previous images and segmentations, independently of the current image.

Some of previous work, namely optical flow based methods [Wolf and Jolion, 2010, Hosaka et al., 2011], use object motion estimation based on the color gradients of moving objects. The estimation is strictly based on object colors. In this way, motion estimation depends on the current image and does not use explicitly object motion knowledge. Conversely, in our method, we treat the color based image model and object motion separately. In the next section, we define and explain our motion model.

3.3 Rigid motion model

We assume that the objects move continuously and possess a rigid motion. As we consider object motion between two frames which are successive in time, therefore, the rigid motion assumption holds. In the following paragraphs we explain our motion model in detail.

We recall the probabilistic object detection from section 2.3 as:

$$P_{\text{obj}}(\mathbf{x}, t) = P_{\text{image}}(\mathbf{x}, t).P_{\text{mov}}(\mathbf{x}, t) \tag{3.1}$$

where $P_{\text{image}}(\mathbf{x}, t)$ is the image model and $P_{\text{mov}}(\mathbf{x}, t)$ is the prior term. It is expressed in

Eq. 2.8 as:

$$P_{\text{mov}}(\mathbf{x}, t) = P(\mathcal{F}(\mathbf{x}, t)|\mathbf{I}(t-1), \mathbf{F}(t-1))$$

which is the general expression for the prior term. It indicates that the current pixel state in foreground $\mathcal{F}(\mathbf{x}, t)$ is conditioned on the previous history of images $\mathbf{I}(t-1)$ and previous foreground segmentations $\mathbf{F}(t-1)$. We can notice that in Eq. 3.1, if the prior term is equal for all pixels of the current image then final image segmentation would be based only on the local data term (*i.e.* image term). Otherwise, the prior term will influence the final pixel classification.

We propose a model for the prior term, which is based on the available knowledge about possible object displacements, regardless of previous image history. It may be regarded as first-order in time, as we keep the foreground data from the previous frame only. Instead of reasoning at a pixel level, we consider a higher level approach where the state of the current pixel $\mathbf{x}$ is conditioned on the entire previous foreground image $\mathcal{F}(t-1)$. It enables us to model global motion of entire objects rather than local motion of pixels considered independently. Thus, in our case, the prior term becomes:

$$P_{\text{mov}}(\mathbf{x}, t) = P(\mathcal{F}(\mathbf{x}, t)|\mathcal{F}(t-1)) \tag{3.2}$$

It is important to mention that each application possesses its own object motion data sets. These data sets may be obtained from the objects motion characteristics in context.

We consider a probabilistic motion model, assuming that the probability function of object transformation (*i.e.* both rotational and translational motion) is known *a priori*.

3.3.1 Definition of motion model

In our case, an object is a connected component of foreground pixels (using $4-$connexity). Let $\mathcal{C}(t-1)$ be the set of connected components in the foreground image at time $t-1$. Every pixel $\mathbf{x}$ in one of this connected components verifies $\mathcal{F}(\mathbf{x}, t-1) = 1$. Every connected component ψ from the set has a mass center (or center of gravity), $\mathbf{c}_\psi$, which can be expressed as:

$$\mathbf{c}_\psi = \frac{1}{|\psi|} \sum_{\mathbf{x} \in \psi} \mathbf{x}$$

Object ψ undergoes a rigid transformation made up of translation $\mathbf{d}$ and rotation of an angle θ. Let $T_{\theta,\mathbf{c},\mathbf{d}}$ be the rigid transformation of a pixel, with translation vector $\mathbf{d}$ and

rotation defined by center $\mathbf{c}$ and angle θ:

$$T_{\theta,\mathbf{c},\mathbf{d}}(\mathbf{x}) = \mathbf{R}_\theta(\mathbf{x} - \mathbf{c}) + \mathbf{c} + \mathbf{d} \tag{3.3}$$

where $\mathbf{R}_\theta$ is the rotation matrix,

$$\mathbf{R}_\theta = \begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix}$$

The center of gravity is chosen as the center of rotation, hence the object transformation is:

$$\mathcal{T}_{\theta,\mathbf{d}}(\psi) = \left\{ T_{\theta,\mathbf{c}_\psi,\mathbf{d}}(\mathbf{x}) | \mathbf{x} \in \psi \right\} \tag{3.4}$$

Now, we suppose that the object transformation $\mathcal{T}_{\theta,\mathbf{d}}$ follows a probability density function $P_{\text{trans}}(\theta, \mathbf{d})$ which is learned from a training data set (during the learning period). The probability density function adds flexibility to the object detection method.

The objects extracted at time $t-1$ help in foreground object extraction at time t. By this we mean that if an object is at some location in the scene at previous time $t-1$ then this object will transform/move spatially to a new location in the forward direction of motion with time. The set of objects are transformed according to Eq. 3.4. The probability that current pixel belongs to a moving object depends on the object transformation probability $P_{\text{trans}}(\theta, \mathbf{d})$ (*i.e.* object speed and angle of rotation with respect to its previous location in the foreground image at time $t-1$). In other terms, for a given pixel $\mathbf{x}$ there exist many pixels $(\mathbf{x}', t-1)$ that may be transformed into $(\mathbf{x}, t)$ with a given probability. Therefore, the definition of our prior object motion knowledge can be expressed by the following equation:

$$P_{\text{mov}}(\mathbf{x}, t) = \sum_{\psi \in \mathcal{C}(t-1)} \sum_{\mathbf{x}' \in \psi} \sum_{\{(\theta,\mathbf{d}) | T_{\theta,\mathbf{c},\mathbf{d}}(\mathbf{x}') = \mathbf{x}\}} P_{\text{trans}}(\theta, \mathbf{d}) \tag{3.5}$$

This can be applied to moving objects with arbitrary probability function $P_{\text{trans}}(\theta, \mathbf{d})$.

3.3.2 Implementation of motion model

Moving object probability for the current pixel $\mathbf{x}$ is given by Eq. 3.5. As explained earlier, object motion can have an arbitrary probability function $P_{\text{trans}}(\theta, \mathbf{d})$. In the current implementation, we assume object rotation and translation to be statistically independent. Therefore, the object transformation probability can be simplified to:

Algorithm 1 Computation of prior motion probability P_{mov}

Input: $\mathcal{F}(t-1), \mu_\theta, \sigma_\theta, \mu_{\text{d}}, \Sigma_{\text{d}}$
Output: $P_{\text{mov}}(t)$

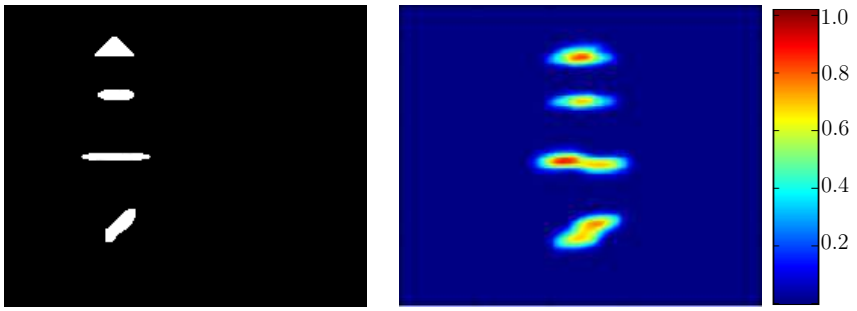
```
Extract set of connected components C(t-1)
for all x ∈ image domain I do
    P_mov(x, t) ← 0
end for
for all ψ ∈ C(t-1) do
    Compute mass center c_ψ
    for all x ∈ ψ do
        for all μ_θ − 3σ_θ ≤ θ ≤ μ_θ + 3σ_θ do
            for all d such that sqrt((d − μ_d)^T Σ_d^{-1} (d − μ_d)) ≤ 3 do
                x' ← R_θ(x − c_ψ) + c_ψ + d
                P_mov(x', t) ← P_mov(x', t) + P_trans(θ, d)
            end for
        end for
    end for
end for
```

$$P_{\text{trans}}(\theta, \mathbf{d}) = P_{\text{rotation}}(\theta).P_{\text{translation}}(\mathbf{d}) \tag{3.6}$$

$P_{\text{translation}}(\mathbf{d})$ is the probability of object translation learned from objects motion and we choose to model it with 2D Gaussian $\mathcal{N}(\mu_{\text{d}}, \Sigma_{\text{d}})$.

$P_{\text{rotation}}(\theta)$ is the object rotational probability with prevision of object rotation in both the directions (*i.e.* clock-wise and counter clock-wise). We also model the rotation probability with a Gaussian $\mathcal{N}(\mu_\theta, \sigma_\theta)$.

Algorithm 1 outlines the process of computing the probability $P_{\text{mov}}(\mathbf{x}, t)$ for all pixels at time t. The algorithm shows that we take the foreground image $\mathcal{F}$ at time $t-1$ as an input and compute the probability $P_{\text{mov}}(\mathbf{x}, t)$ at every pixel using object motion



$\mathcal{F}(t-1)$ $P_{\text{mov}}(t)$

Figure 3.3: Example of $P_{\text{mov}}(t)$ computation with $\mu_\theta = \pi/32$, $\sigma_\theta = \pi/16$, $\mu_{\mathbf{d}} = [48.37\ ;\ 5.6]$ and $\Sigma_{\mathbf{d}} = [511.71\ ,\ -58.21\ ;\ -58.21\ ,\ 35.43]$.

knowledge $P_{\text{trans}}(\theta, \mathbf{d})$. The probabilities are actually accumulated into $P_{\text{mov}}(\mathbf{x}, t)$. The contributions of objects are summed up. In Figure 3.3, we show an example of prior motion probability map, resulting from four objects including a triangle, an ellipse, a horizontal line and a slanted thick line, which represent the foreground image $\mathcal{F}(t-1)$. We can see that $P_{\text{mov}}(\mathbf{x}, t)$ has higher values for the expected object locations in the image. It must be noted that at time $t = 0$, we initialize $P_{\text{mov}}(\mathbf{x}, t) = 0.5$ for all pixels.

Object motion learning is an off-line process and is done once for a given application. Details of the process of learning object motion and all the parameters (*i.e.* μ_{d}, Σ_{d}, μ_θ, σ_θ) involved in the process are discussed for the corresponding applications in section 3.5.

3.3.3 Combination of motion model and image model

In section 2.3, we have presented object detection methodology. We have discussed that we combine motion model and image model for object detection. Eq. 3.1 states that $P_{\text{mov}}(\mathbf{x}, t)$ is combined with image term $P_{\text{image}}(\mathbf{x}, t)$. A pixel of current image is classified as the foreground pixel if its probability is above a threshold s, and we can write:

$$\mathcal{F}(\mathbf{x}, t) = \begin{cases} 1 & \text{if} \quad P_{\text{obj}}(\mathbf{x}, t) \geq s \\ 0 & \quad\;\; \text{otherwise} \end{cases} \tag{3.7}$$

The idea is to reduce the number of missclassified pixels considering only $P_{\text{image}}(\mathbf{x}, t)$. Thus, if an isolated pixel is misclassified in the previous foreground image $\mathcal{F}(t-1)$, then its contribution at time t will not be correlated with its neighboring pixels and will generate a small probability $P_{\text{obj}}(\mathbf{x}, t)$ at time t. Consequently, with low probability, the corresponding pixel would not be considered as an object pixel.

For background subtraction, we can integrate our motion model with different image models. We employ two image models in our work to assess the efficiency of the motion model. First, we use a modified GMM model as an image model. In the second application, we use an image model specialized to wood detection and integrate it with motion model. In following sections, we present the combination of these two image models with motion model and show the contribution of our motion model.

3.4 Modified GMM as image model

The GMM-based background modeling method represents the background likelihood. In order to integrate such model into our framework, the image term $P_{\text{image}}(\mathbf{x}, t)$ of Eq. 2.8 can be expressed as:

$$P_{\text{image}}(\mathbf{x}, t) = 1 - P_{\text{BG}}(\mathbf{x}, t) \tag{3.8}$$

where $P_{\text{BG}}(\mathbf{x}, t)$ is the background likelihood. This corresponds to the probability that pixel $\mathbf{x}$ belongs to background given its color, with respect to the current local background model at $\mathbf{x}$. Basically, this background reference is represented by a Gaussian Mixture Model (GMM). In image processing and computer vision literature, the so-called GMM often refers to the background subtraction algorithm proposed by [Stauffer and Grimson, 2000]. Strictly speaking, the GMM is a parametric statistical model to represent multimodal probability density functions. Stauffer and Grimson's algorithm is actually based on this representation but also holds the on-line update of background model as well as the detection process.

Our use of the GMM differs from [Stauffer and Grimson, 2000] for several reasons. It is important to note that the pixel classification in our object detection method is based not only on an image model but also on the aforementioned motion model. The detection step is performed through the implementation of Eq. 2.8. Secondly, experiments are led on image sequences that do not lend themselves to background on-line updating. Instead, we separate the learning and detection phases. In the latter, the background model is kept fixed. Finally, the computation of background likelihood is more general than in [Stauffer and Grimson, 2000], as we do not assume color components to be mutually independent. The consequences of this generalization will be clarified after the definition of the background likelihood in next paragraphs.

In a very general setting, each spatio-temporal pixel $(\mathbf{x}, t)$ holds a set of $K(\mathbf{x}, t)$ weighted Gaussian functions:

$$\{(\omega_i(\mathbf{x}, t), \mu_i(\mathbf{x}, t), \Sigma_i(\mathbf{x}, t))\}_{1 \leq i \leq K(\mathbf{x}, t)}$$

Each Gaussian is assumed to represent one significant color belonging to the background representation of current pixel. The weight parameters ω_i represent the time proportion that those colors stay in the scene. $\mu_i(\mathbf{x}, t)$ and $\Sigma_i(\mathbf{x}, t)$ are the mean value and covariance matrix of the i^{th} Gaussian in the mixture.

The Gaussian mixture model algorithm of [Stauffer and Grimson, 2000] is a simpli-

fication of the above general pixel-wise GMM. We have explained in section 1.1.2 that they assume independence between red, green and blue color components. Similarly, the variances of the three color components are assumed to possess the same value (σ_i^2) for a given pixel. Therefore, the color covariance matrix is essentially transformed to ($\sigma_i^2\mathbf{I}$, $\mathbf{I}$ is identity matrix), in their approach (refer to Eq. 1.6). Therefore, the background likelihood in their case becomes:

$$P_{\text{background}}(I(\mathbf{x},t),\mu,\Sigma) = \sum_{i=1}^{K(\mathbf{x},t)} \left(\omega_i(\mathbf{x},t) \prod_{d=1}^{D} \frac{1}{\sqrt{2\pi\sigma_i^2}} e^{-\frac{1}{2\sigma_i^2}(I_d(\mathbf{x},t)-\mu_{id}(\mathbf{x},t))} \right)$$

Assuming independence between color components allows not to invert the covariance matrix to compute the probability. Conversely, we do not assume the three color channels to be independent. Similarly, we do not assume same variance values for each color channel per pixel. Notice that, in case of non-updated background, covariance matrices need to be inverted only once. The background likelihood of color $I(\mathbf{x},t)$ is obtained by summation of Gaussian probabilities:

$$P_{\text{BG}}(\mathbf{x},t) = \sum_{i=1}^{K(\mathbf{x},t)} \omega_i(\mathbf{x},t)\mathcal{N}(I(\mathbf{x},t),\mu_i(\mathbf{x},t),\Sigma_i(\mathbf{x},t))$$

In our framework, this general model is simplified to some extent. We keep the same number of Gaussians per pixel, so that K is no longer a function of $(\mathbf{x},t)$. Moreover, the background mixture model is learned offline and is not updated with time. Hence, unlike in [Stauffer and Grimson, 2000], Gaussians functions and associated weights do not vary with respect to time, which gives the following GMM:

$$\{(\omega_i(\mathbf{x}),\mu_i(\mathbf{x}),\Sigma_i(\mathbf{x}))\}_{1\leq i\leq K}$$

The adequation between a tested pixel value and the model is measured for the best matching Gaussian. Thus, the likelihood is actually computed as follows:

$$P_{\text{BG}}(\mathbf{x},t) = \max_{1\leq i\leq K} \omega_i(\mathbf{x})\mathcal{N}(I(\mathbf{x},t),\mu_i(\mathbf{x}),\Sigma_i(\mathbf{x})) \tag{3.9}$$

Using Eq. 3.9, the image term is computed according to Eq. 3.8 and multiplied by $P_{\text{mov}}(\mathbf{x},t)$, so that the modified GMM is integrated with our motion model.

Computation of P_{image}:
We perform background learning and detection in two steps. We summarize background learning step modifications in view of [Stauffer and Grimson, 2000] method as follows:

- We take the current history of the values of red, green and blue channels of each pixel for n consecutive frames containing only background. We consider $p \times p$ neighborhoods for each pixel, where p is a small number. In the neighborhoods, we compute the parameters μ and Σ of K gaussian distributions per pixel by running the Expectation-Maximization algorithm.
- When background learning is completed, we perform object detection. During detection phase, every new pixel value is checked against existing model components. For a given pixel, the best matching gaussian is determined and the background likelihood is computed using Eq. 3.9. Then, we compute P_{image} by using Eq. 3.8.

3.5 Results of motion model combined with image models

For each application, we obtain object motion knowledge from a training data set, in which motion model parameters are estimated. We combine the motion model based on object knowledge *a priori* with two image models. In the first subsection, we present the combination with the modified GMM method as an image model. In the second subsection, we use the image model specialized to wood, described in section 2.4. We integrate the image model with corresponding motion model for wood detection. Image segmentation results are evaluated both qualitatively and quantitatively.

3.5.1 Combination with modified GMM

We apply modified GMM as an image model to illustrate the integration of our motion model into a background subtraction technique. We apply our method to detect object in color videos.

Application to synthetic data:
A synthetic video is made to illustrate the method of incorporation of motion model into background subtraction algorithm. In this video, a yellow square block moves from left to right in the image plane as shown in Figure 3.4(a). The background image is divided

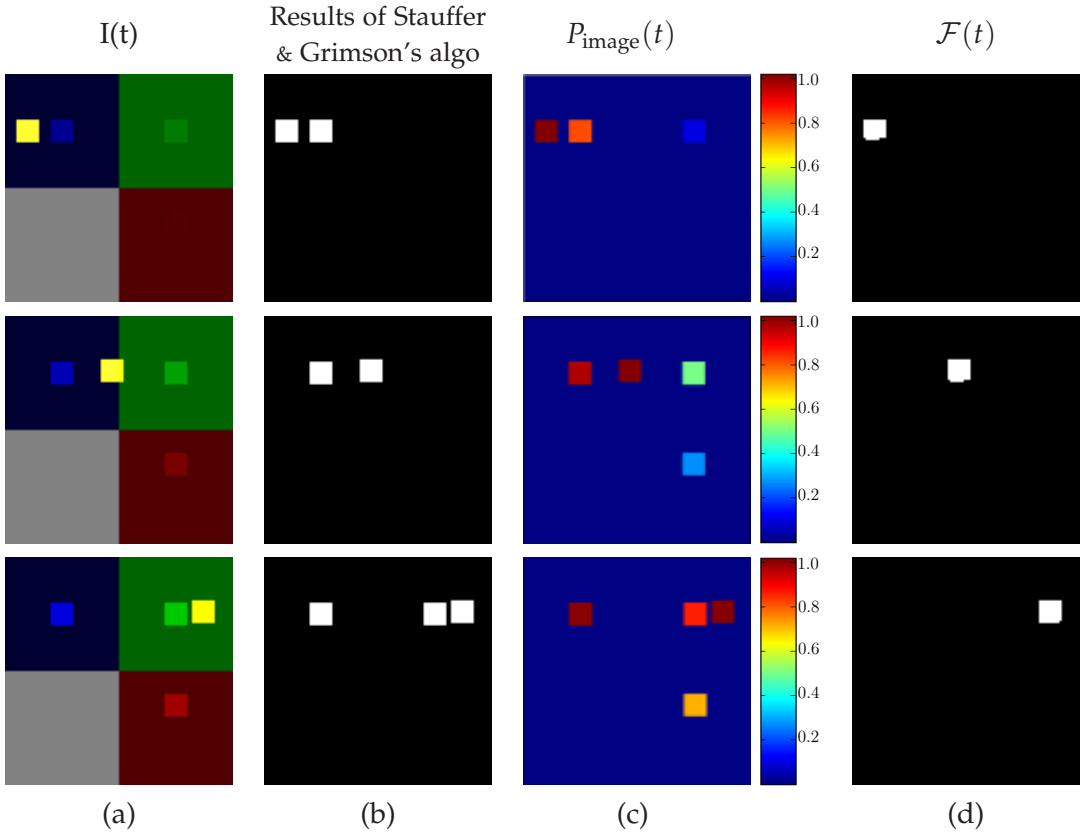


Figure 3.4: (a) Synthetic test images with a moving yellow square from left to right, background colors change with time in three areas with corresponding: (b) results of Stuaffer and Grimson's algorithm, (c) probability $P_{\text{image}}(t)$ and (d) $\mathcal{F}(t)$ with (modified GMM + motion model).

into four regions. Three parts of these regions undergo progressive change of color with time. The sizes of these subregions are equal to the size of the moving block. Moreover, the square block moves over the regions of changing background in the upper part of the image sequence. The number of Gaussians is set to $K = 1$ and $p = 1$ in the modified GMM. Object motion information $P_{\text{trans}}(\theta, \mathbf{d})$ *a priori* is available.

As we explained above $P_{\text{image}}(\mathbf{x}, t)$ is the probability of a pixel $\mathbf{x}$ to be a foreground pixel with respect to color. In the case of modified GMM, it is an increasing function of the dissimilarity between $I(\mathbf{x}, t)$ and the background colors $\mu_i(\mathbf{x}, t)$. For the moving block, it has a high value. The probability P_{image} has also a high value for background regions as shown in Figure 3.4(c). Therefore, if we only consider $P_{\text{image}}(t)$ for object detection, we may also detect the changing background regions as foreground objects. This happens with background subtraction techniques, based on color only, such as Stauffer and Grimson's GMM. In Figure 3.4(b), we show the results obtained with Stauffer and Grimson's GMM algorithm also classify changing background regions as foreground.

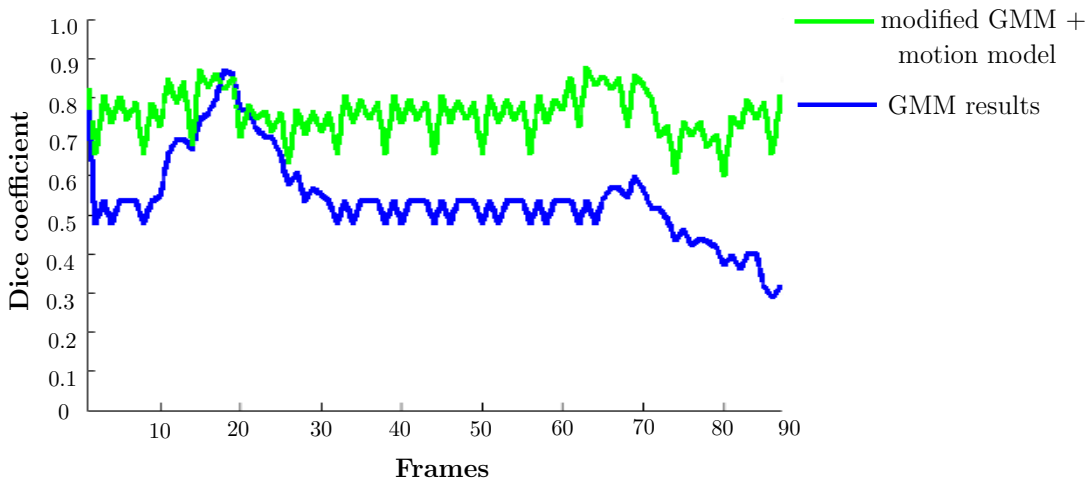


Figure 3.5: Dice coefficient per frame of image segmentation results with GMM [Stauffer and Grimson, 2000] and with (modified GMM + motion model).

In our approach, we compute $P_{\text{mov}}(t)$ and $P_{\text{image}}(t)$ for each frame. We compute foreground image $\mathcal{F}(t)$ by using Eq. 3.7. The segmentation results in Figure 3.4(d) show that we remove the false detections arising with non-constrained background subtraction. To give the quantitative analysis for the entire image sequence, we evaluate the image segmentation results by using the Dice similarity measure[1], using available ground truth. The Dice coefficient is computed for GMM segmentation results and our segmentation results per frame. The comparison is shown in Figure 3.5. The two curves meet at the point when the moving object exactly overlap the changing background region in the blue region of synthetic image sequence. The segmentation results computed with incorporation of $P_{\text{mov}}(t)$ have higher Dice coefficient measures. We can notice that the Dice coefficient is below 0.9, which means that we miss a few object pixels, which may be due to the fact that likelihood values $P_{\text{obj}}(\mathbf{x}, t)$ at the edges of the square block are low, and therefore, truncated when we apply a threshold. The comparison shows that the incorporation of motion model into background subtraction improves the object segmentation.

[1]For definition, see section 2.4.4.

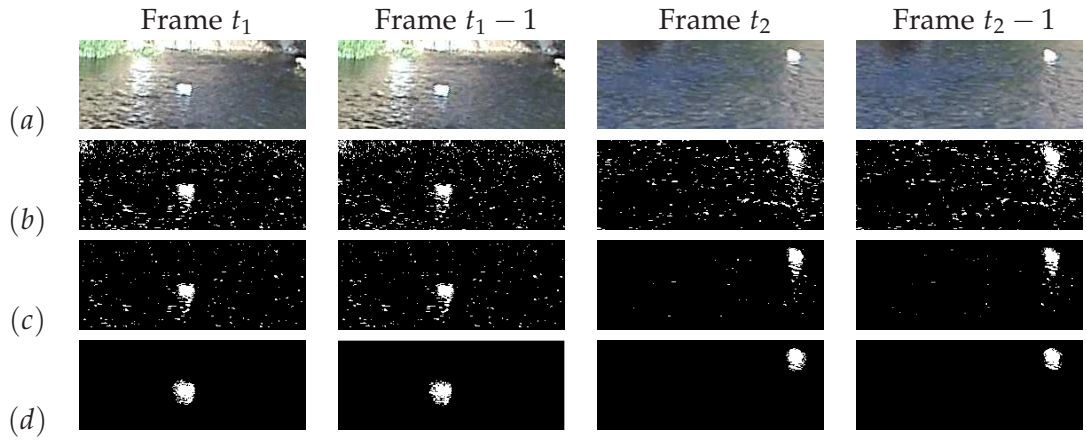


Figure 3.6: (a) Two floating objects in river with two consecutive images at different time instances, (b) foreground images $\mathcal{F}$ with $p = 1$, (c) $\mathcal{F}$ with $p = 5$ pixels, (d) $\mathcal{F}$ obtained with P_{mov}.

Application to real data:
We perform experiments on river videos which contain floating objects. The background contains moving vegetation and water ripples. There are also sunlight reflections from the surface of water which cause uneven brightness. We take 100 consecutive images of the background only with no floating object in the training period to build background model.

Learning object motion:
Object motion knowledge is obtained in the application as explained in section 3.1. It is an off-line process in which, we compute mass centers of moving objects from manually segmented sequences. Object motion from entrance to exit from the scene is noted in the training period. In river videos, water flow is from right to left of the image plane. The objects move in the different areas of the scene. The frame rate in these videos is 25 fps and therefore object displacement between two frames is relatively small. During learning the motion model parameters are computed. The parameter values for this application are: $\mu_\theta = 0$, $\sigma_\theta = 0.150$, $\mu_{\mathbf{d}} = [-2.37\ ;\ 0.05]$ and $\Sigma_{\mathbf{d}} = [1.37\ ,\ 0.14\ ;\ 0.14\ ,\ 1.43]$. We compute $P_{\text{mov}}(t)$ for current image by the method as explained in section 3.3. Then, $P_{\text{image}}(t)$ and $P_{\text{mov}}(t)$ for current image are multiplied by using Eq. 3.1. In this way we get the joint probability $P_{\text{obj}}(t)$, which is higher for moving object pixels. To get a final foreground image $\mathcal{F}(t)$ the joint probability $P_{\text{obj}}(t)$ is thresholded as given by Eq. 3.7.

$\mathbf{P_{image}}$ *for floating objects:*
The results obtained from Stauffer and Grimson's method, which is pixel-based, contain

a lot of noise. To overcome this problem, our modified GMM model considers a spatial neighborhood around pixels to compute color distribution, rather than pixel alone (recall that the size of this neighborhood is controlled by parameter p). The number of Gaussians is set to $K = 4$ to model background color probability density. In this neighborhood, the probable colors are the ones which stay longer and more static. This means that static single colored objects in the scene form tight clusters, whereas moving ones form wide clusters, due to different reflecting surfaces during motions. We make p vary from 1 to 9. The results are evaluated by visual inspection. We kept ($p = 5$) in our experiments, which leads to the best results. For comparison we show the results in Figure 3.6. The second row of images in Figure 3.6(b) shows the foreground images obtained when no spatial neighborhood is taken into account ($p = 1$). We can notice the number of false positives due to the dynamic nature of background is high if no spatial neighborhood is considered. The third row of images in Figure 3.6(c) shows the foreground images obtained with a spatial neighborhood ($p = 5$). The number of false positives is reduced when we take into consideration the neighborhood of each pixel. We further use this approach with motion model for background subtraction.

$P_{\text{mov}}(t)$ and $P_{\text{image}}(t)$ are computed for each incoming video frame. In Figure 3.6(d) we show the final foreground images $\mathcal{F}(t)$. The incorporation of P_{mov} in the object detection improves significantly the foreground detection. In the process, we do not use any morphological operations (*e.g.* dilation or erosion) to minimize noise. The foreground segmentation is evaluated by visual inspection. In the next section, we show motion model applied to wood detection.

3.5.2 Combination of motion model with image model for wood

For wood detection, we have presented the image model in section 2.4. We recall the image model for wood detection:

$$P_{\text{image}}(\mathbf{x}, t) = P_i(\mathbf{x}, t).P_t(\mathbf{x}, t)$$

It consists of intensity and temporal information of moving pixels in the current image in terms of probabilities. P_i contains likeliness of pixels to be wood with respect to their brightness. P_t contains likeliness to be wood with respect to brightness variations at each pixel level.

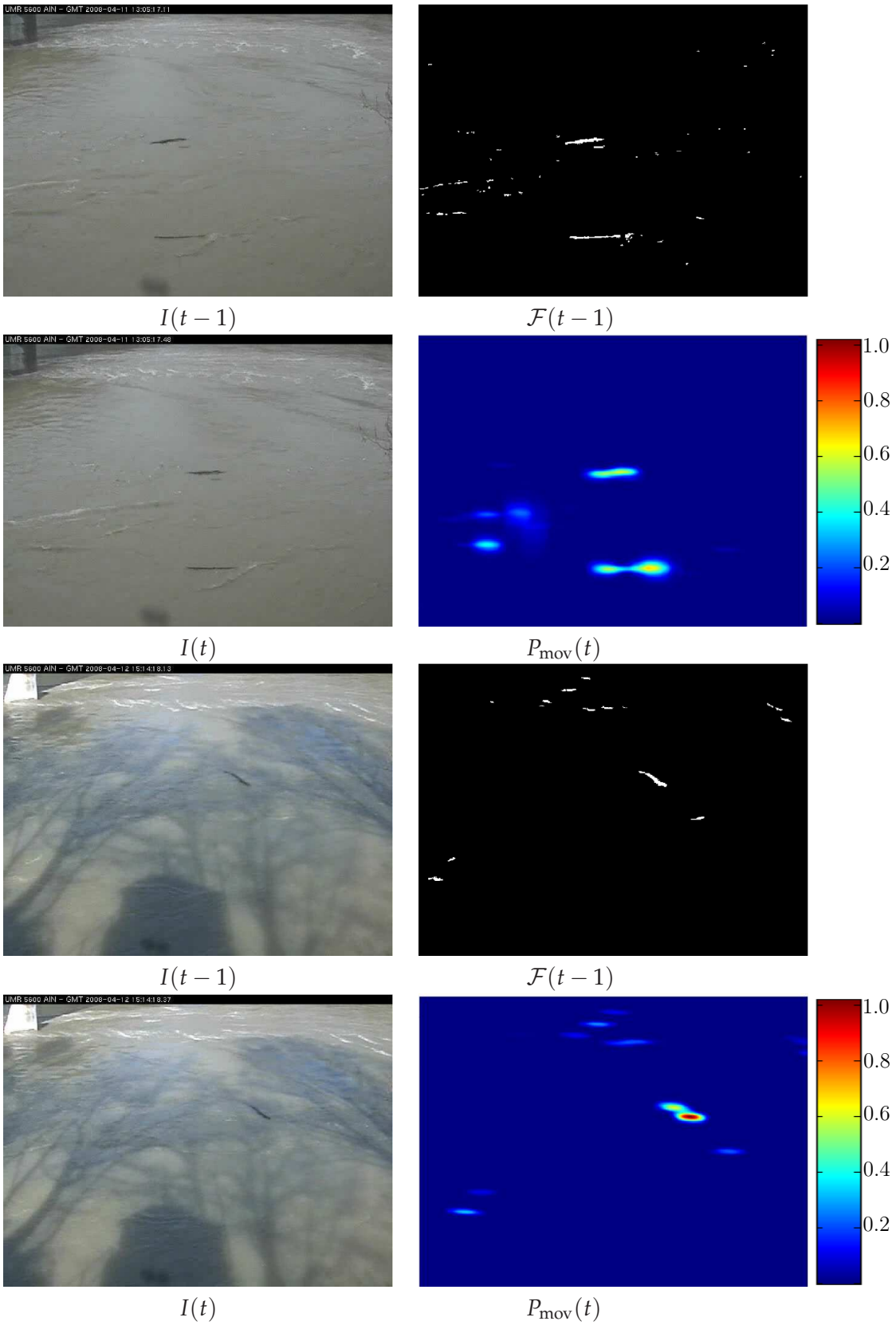


Figure 3.7: Two floating wood objects, $I(t-1)$ and $I(t)$ two consecutive images of wood objects, corresponding previous foreground images $\mathcal{F}(t-1)$, $P_{\text{mov}}(t)$ have higher values for corresponding wood pieces with respect to water waves.

Learning wood motion:
To model the motion of wood pieces, we manually label the mass centers for every wood piece in each frame in the learning period of river videos. Wood displacements in successive video frames are noted for every wood piece in the learning period. The wood objects do not come one by one in the scene and most of the time an image contains more than one wood piece. We take worst case scenarios of multiple wood pieces in motion simultaneously.

Wood pieces and water move at the same speed and water flow is largely turbulent during floods, due to which wood pieces often rotate with water waves. Smaller wood pieces rotate and translate more in comparison to larger trees. However, the motion model should be equally applicable in all circumstances. To compute the parameters of the motion model, we take the image sequences of 200 wood objects with an average of 18 images per wood object (*i.e.* from entrance to exit of a wood piece). Image sequences of 3600 frames in total that contain manually labeled wood objects are used for learning wood motion. These wood objects are from multiple videos and during different times of the day. After learning, we obtain the parameter values that are $\mu_\theta = 0$, $\sigma_\theta = 0.196$, $\mu_{\mathbf{d}} = [48.37\ ;\ 2.95]$ and $\Sigma_{\mathbf{d}} = [511.71\ ,\ 23.79\ ;\ 23.79\ ,\ 14.96]$. Learning data suggests that wood objects have strong translational motion but also have rotational motion. Another reason for strong translational motion is the low frame rate of river videos, which is $\leq$ 4fps.

As explained in section 3.3, $P_{\text{mov}}(t)$ is the probability of moving objects in the current image based on previous foreground $\mathcal{F}(t-1)$ and wood motion knowledge *a priori*. Therefore, the expected object pixels must have higher probability values. In the first example in Figure 3.7, there are two small moving wood objects and water waves that are present at $\mathcal{F}(t-1)$. The second example in Figure 3.7 shows a small floating wood piece under the cast shadows of surrounding trees. The probability values $P_{\text{mov}}(t)$ for all wood objects are higher than the water waves. These examples indicate the fact that prior wood motion data can help in the distinction of wood objects from water waves due to their motion characteristics.

Experimental results:

Figure 3.8 and Figure 3.9 show the comparison of results which are computed without and with incorporation of P_{mov} with the image model. Two consecutive images of wood objects are shown with a portion of image zoomed near the wood object. The second row contains probability P_{obj} obtained without prior motion model P_{mov}, for each object. The third row of images represents the resulting probability P_{obj} computed with incorporation of motion model P_{mov} for each wood object. It is important to mention that our method works at a low computational cost. Video frames have size 640×480 and are extracted from MPEG4-compressed streams tested on an Intel Core2 Duo 2.66GHz with 4GB RAM running C code. Average execution time for computing all probabilities in a single frame is 0.35 second.

Results of our method show two major advantages of prior motion knowledge. First, the probability P_{mov} has higher values for wood object regions in comparison to water waves regions, which is clear from the results shown in Figure 3.8 and Figure 3.9. Top portions of the second row of each wood example show that, when motion prior is not used, water waves are as prominent as wood objects, in terms of probability. This increases the difficulty of computation of a global threshold value that should work in all cases. The drawback is removed with the integration of P_{mov} which helps in minimizing false detection of water waves as objects. The second advantage is that our method improves the foreground object segmentation. Miss rate of wood pixels inside wood objects is high without P_{mov}. The wood object regions are zoomed to show this advantage for all wood objects.

The method improves distinction between water waves and wood objects. The accuracy of wood counting, which is based on the image segmentation results, is improved in this way.

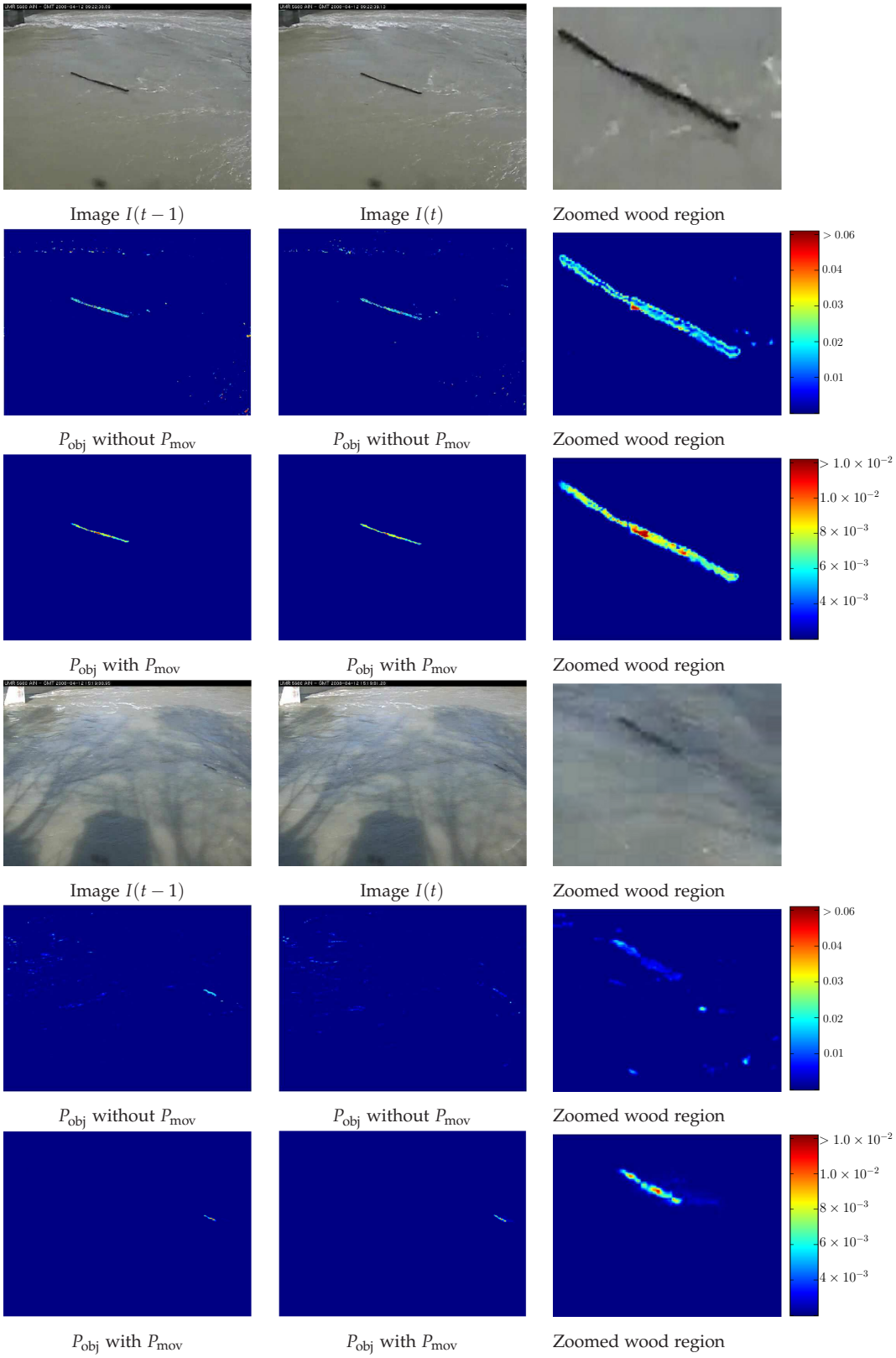


Figure 3.8: Two floating wood objects, $I(t-1)$ and $I(t)$ two consecutive images of wood objects with zoomed wood region, P_{obj} without P_{mov} and P_{obj} with P_{mov} for corresponding wood pieces.

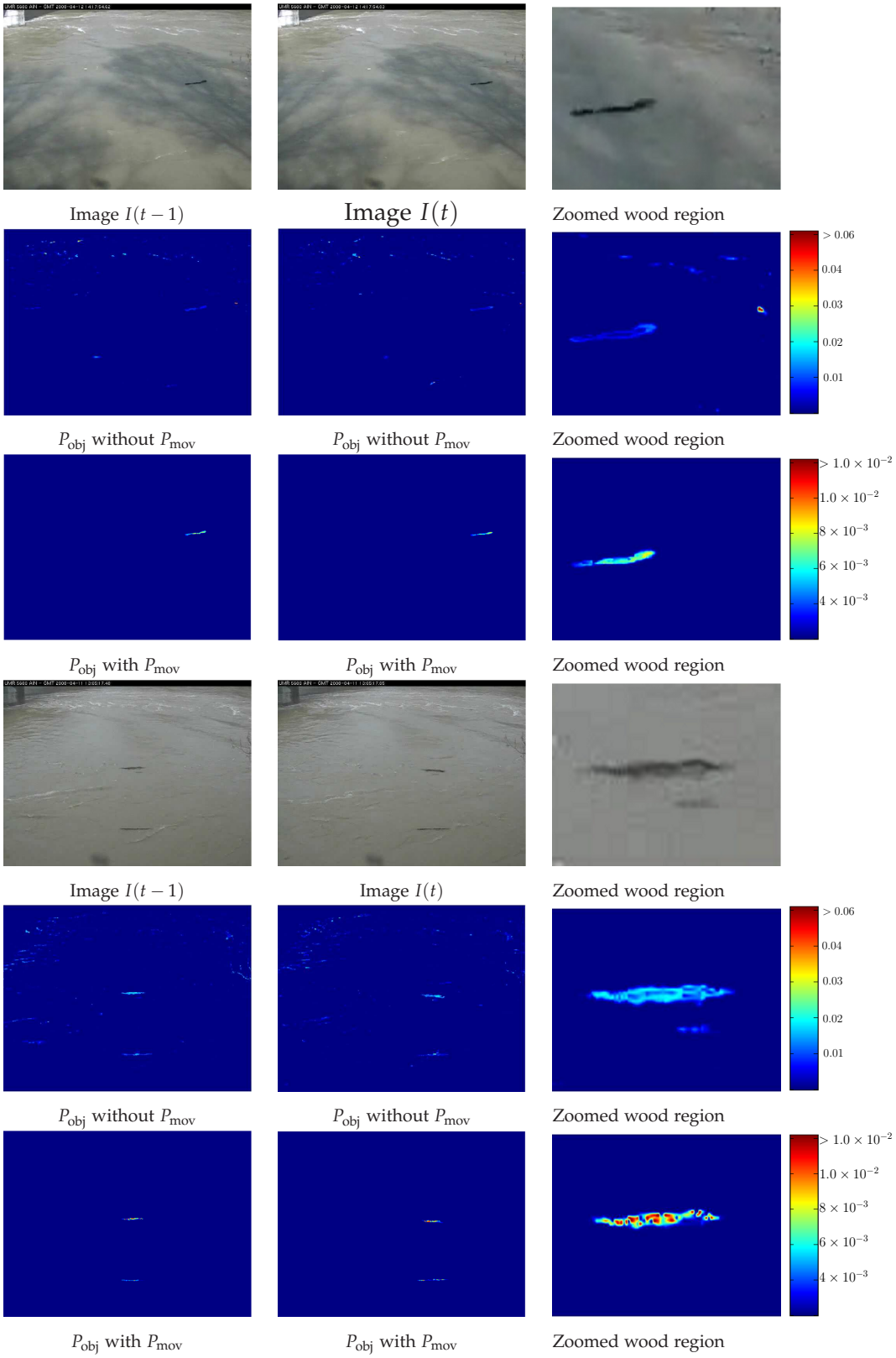


Figure 3.9: Two floating wood objects, $I(t-1)$ and $I(t)$ two consecutive images of wood objects with zoomed wood region, P_{obj} without P_{mov} and P_{obj} with P_{mov} for corresponding wood pieces.

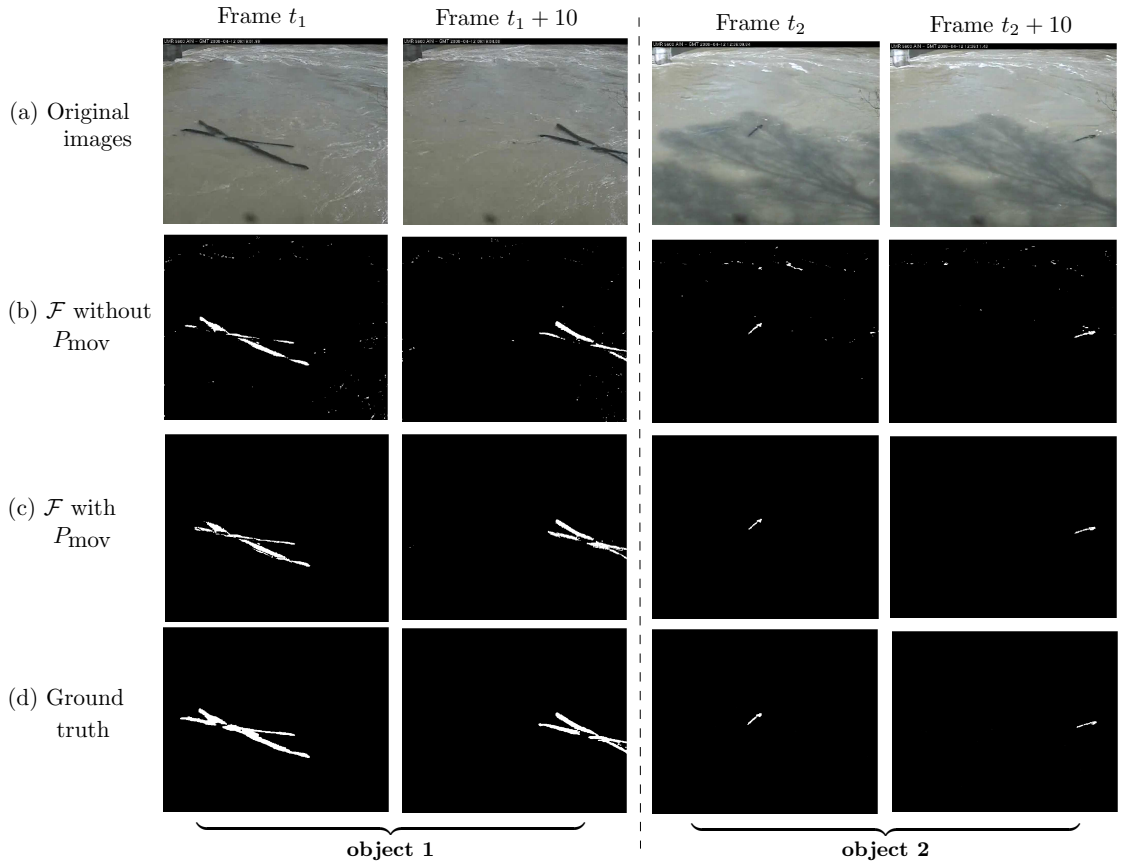


Figure 3.10: Two floating wood objects; **object 1** a big trunk of tree and **object 2** a small wood piece. *(a)* original images with corresponding *(b)* foreground images $\mathcal{F}$ without P_{mov}, *(c)* foreground images $\mathcal{F}$ with P_{mov} and *(d)* ground truth images.

To evaluate our object segmentation results, we use the Dice coefficient. Furthermore, to make comparative evaluations, we decide to evaluate our final segmentation results with and without prior motion P_{mov} in order to show the improvements made by the incorporation of P_{mov}. In Figure 3.10, we show two wood pieces with the results of final segmentation obtained without P_{mov} and with P_{mov}. The ground truth images are obtained manually for each of the wood pieces. Object 1 is made up of two large wood pieces and object 2 is a small wood piece. Object 2 is rotated with water flow. We show two instances of images which are 10 frames apart from each other so that object rotation is clearly visible. Pixel-wise comparison is carried out between ground truth images and our segmentation results. The Dice coefficient value is computed for each frame. Figure 3.11 shows Dice coefficient values per frame for object 1 and object 2. Average Dice coefficient values for both objects are high with P_{mov} in all cases which shows the superiority of the method.

We have successfully tested our method on several river videos. After wood detection, we track them in the consecutive frames and count them, which is detailed in appendix

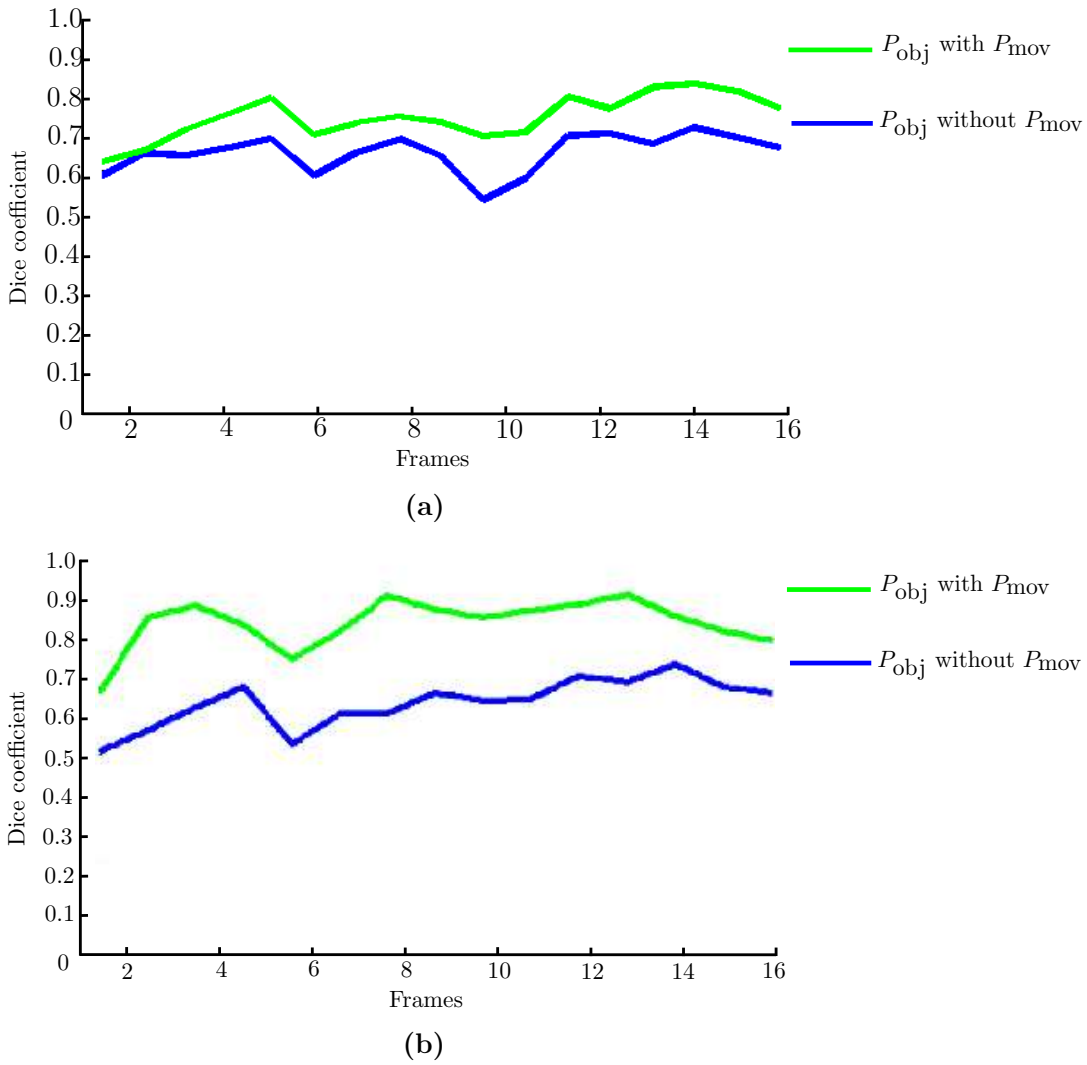


Figure 3.11: Dice coefficient per frame *(a)* for **object 1** and *(b)* for **object 2** of Figure 3.10.

A).

3.6 Conclusion

In this chapter, we presented object motion based on prior motion knowledge. The object motion parameters are learned by an off-line process. A parametric motion model is proposed, which can be integrated with any background subtraction techniques. We proposed to use motion model jointly with color based image model for object detection. We used two different image models in this chapter. The first one is a modified GMM method. The fusion of modified GMM with motion model produces very good object detection rate in moving background. Similarly, another image model specialized to

wood is used with motion model combinedly to extract wood in the river videos. The results are compared before and after the integration of motion model. We evaluate the segmentation results by using Dice similarity measure. The results indicate the improvements in the object segmentation are twofolds *i.e.* not only false detection of background pixels is reduced but also mis-detection rate within the object is minimized.

Chapter 4

Background modeling using frequency based approach

Contents

Moving backgrounds can be composed of time repetitive textures, for example, water ripples, moving vegetation in the wind, fire, moving escalators *etc.* In such conditions, individual pixel-based background models are not able to represent these regional changes. As a matter of fact, pixel based background models consider each pixel independently. These methods neither take into account spatial neighborhoods of pixels for background modeling nor the frequency of colors. For example, the GMM [Stauffer and Grimson, 2000] use model update to include per pixel temporal evolution of background. However, the temporal variation is not considered. New colors are added into the pixel-wise background representation thanks to the updating step, but the temporal organization of these colors is ignored. Thus, these algorithms produce a lot of false detections when they are applied in repetitively moving backgrounds.

In section 1.1.2.2, we presented texture-based background models. In these approaches, spatial texture (2D) are considered and a background model proposed by [Heikkilä and Pietikäinen, 2006], uses local binary pattern as texture operator. This approach gives better background representation compared to pixel-based approaches. However, it does not work very robustly on flat image areas where the gray values of the neighboring pixels are very close to the value of the center pixel. Similarly, LBP is strictly in spatial domain and does not take into account temporal evolution of background region which may change local texture temporally, therefore, may produce poor object detection results in case of spatially varying and time repetitive moving backgrounds.

In this chapter, we present a frequency-based approach, which is a novel method for background representation. To our knowledge, spatiotemporal frequency analysis has not been explored for background modeling in the literature. In the next section, we present spatiotemporal and spectral methods applied by several authors in dynamic textured backgrounds. These methods motivate us to explore frequency based approach for unknown object detection in moving textured backgrounds.

4.1 Spatiotemporal and spectral methods

For spatial textures with time extent, one of the methods is to consider their patterns as time series, which is referred to as *dynamic texture* [Doretto et al., 2003] in the literature. However, working with videos that contain textures of unknown spatiotemporal extents is different from working with static textured images.

Spatiotemporal approaches are applied to address the problem of dynamic textures. An earlier work by Szummer *et al.* [Szummer and Rosalind, 1996] focused on temporal

texture modeling. They proposed a spatiotemporal auto regressive model (STAR) for temporal textures recognition, which is a (2D+T) extension of 2D autoregressive models. The method has been modified by incorporating spatial correlation for modeling temporal textures by [Doretto et al., 2003, Doretto and Soatto, 2006].

Similarly, to detect foreground objects in a dynamic textured background, an approach was developed by [Zhong and Sclaroff, 2003], that uses an auto regressive moving average (ARMA) model. They proposed a robust Kalman filter to iteratively update the state of the dynamic texture ARMA model. If the estimated value for a pixel is different from the predicted value, then the pixel is labeled as foreground.

The main idea of our approach is to model the spatiotemporal color patterns of background for object detection. In moving backgrounds, these color patterns often appear repeatedly with time. So, a background model can be built on the frequency analysis of such patterns. The use of frequency analysis for texture segmentation is common in image processing. For example, the Gabor transform was used for texture segmentation [Bovik et al., 1990]. It is essentially a Fourier transform windowed by a Gaussian envelope. To select appropriate Gabor filters, the power spectrum analysis of the Fourier transform of the textured image was performed [Manjunath and Ma, 1996, Puzicha et al., 1997, Wang et al., 2006]. Local Fourier transform in spatial domain was applied by [Zhou et al., 2001] for texture classification and content based image retrieval. In [Abraham et al., 2005], dynamic texture synthesis was carried out by using Fourier descriptors. They apply 2D Fourier transform on the whole image and the most significant frequencies contributed by all pixels are retained. In their approach, they assume temporal stationarity of spatial 2D textures and do not consider temporal evolution of spatial textures.

Our method is inspired from frequency based 2D texture segmentation approaches. Variations in the background are both spatial and temporal in case of moving background. Therefore, the background model should be constructed by using spatiotemporal data in the region around each pixel by applying frequency analysis. We propose to use local Fourier transform on the neighborhood of each pixel (both spatial and temporal). We perform local spectral analysis, that captures the frequency components of the spatiotemporal region around each pixel. We then construct a background model based on the observations of the background process during a training period. Once the background model is constructed, new frames are subjected to object detection phase. The following section describes our proposed background model.

4.2 Multidimensional Fourier transform

Multidimensional Fourier transform is used in data retrieval, multi spectral analysis *etc*. We use tri-dimensional Fourier transform in our analysis. The Fourier transform for a signal $f(x, y, t)$ can be written as:

$$\mathcal{FT}\{f(x, y, t)\} = F(u, v, w) \tag{4.1}$$

which is defined by:

$$F(u, v, w) = \int_{-\infty}^{+\infty} \int_{-\infty}^{+\infty} \int_{-\infty}^{+\infty} f(x, y, t) e^{-i2\pi(xu+yv+tw)} dxdydt \tag{4.2}$$

Similarly, the discrete Fourier transform of a signal sampled on a $N_x \times N_y \times N_t$ grid is:

$$F(u, v, w) = \sum_{x=0}^{N_x-1} \sum_{y=0}^{N_y-1} \sum_{t=0}^{N_t-1} f(x, y, t) e^{-i2\pi(\frac{xu}{N_x}+\frac{yv}{N_y}+\frac{tw}{N_t})} \tag{4.3}$$

Due to the separability property of Fourier transform, the tri-dimensional discrete Fourier transform (3D-DFT) can be computed in three steps. This can be achieved by computing mono-dimensional Fourier transform operation on $f(x, y, t)$ with respect to x, y and t successively.

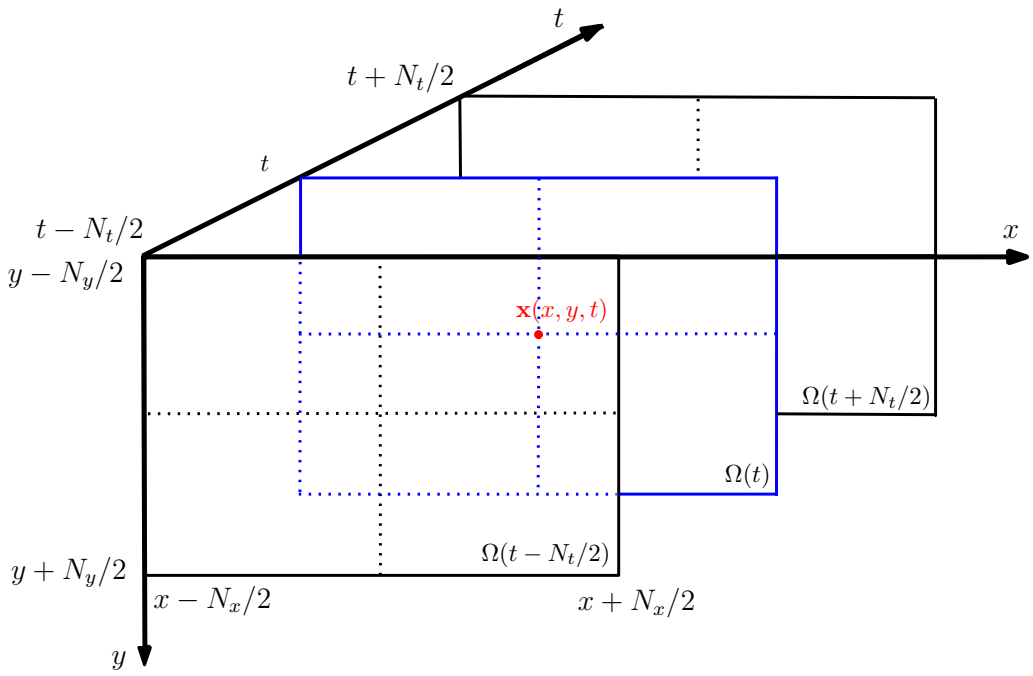


Figure 4.1: Spatiotemporal region Ω around a pixel $\mathbf{x}(x, y, t)$ with size $N_x \times N_y \times N_t$.

4.3 Space-time local Fourier transform

First, we describe some notations used for a pixel representation in space-time and frequency domain. Let a pixel in space-time be represented by $\mathbf{p} = (\mathbf{x}, t)$ and $\mathbf{u} = (u, v, w)$ be a space-time frequency vector. A spatiotemporal cuboid centered at a pixel (see Figure 4.1) is denoted as:

$$\Omega(\mathbf{p}) = \Omega(x, y, t) = \left[x - \frac{N_x}{2}, \cdot\cdot, x + \frac{N_x}{2}\right] \times \left[y - \frac{N_y}{2}, \cdot\cdot, y + \frac{N_y}{2}\right] \times \left[t - \frac{N_t}{2}, \cdot\cdot, t + \frac{N_t}{2}\right]$$

It is important to note that N_x, N_y and N_t should be chosen according to the maximal period (spatial and temporal respectively) which is expected in the data. Let us consider a gray scale image sequence as a real-valued function $f(\mathbf{p})$ defined for each pixel $\mathbf{p}$. Let us introduce a complex-valued function $\hat{F}(\mathbf{u}, \mathbf{p})$, corresponding to the space-time local Fourier transform for a pixel $\mathbf{p}$ given frequency $\mathbf{u}$. It is expressed as:

$$\hat{F}(\mathbf{u}, \mathbf{p}) = \sum_{\mathbf{p}' \in \Omega(\mathbf{p})} f(\mathbf{p}')\omega(\mathbf{p} - \mathbf{p}')e^{-i2\pi((\mathbf{p}-\mathbf{p}')\cdot\mathbf{u})} \tag{4.4}$$

where $\cdot$ denotes the dot product such that $\mathbf{p} \cdot \mathbf{u} = ux + vy + wt$

$$\omega(x, y, t) = \frac{1}{\sqrt{2\pi\sigma_x^2\sigma_y^2\sigma_t^2}} \; e^{(-\frac{x^2}{2\sigma_x^2} - \frac{y^2}{2\sigma_y^2} - \frac{t^2}{2\sigma_t^2})}$$

ω is the gaussian window function which is truncated beyond 3 times the standard deviation in each dimension. We chose $\sigma_x = \frac{N_x}{6}$, such that ω is negligible when $x = \pm\frac{N_x}{2}$ and similarly for σ_y and σ_t. In our method, we take the magnitude of Fourier coefficients which have information of the quantity of each frequency component present inside spatiotemporal cuboid Ω around the pixel $\mathbf{p}$. We denote it as a spectrum $\mathcal{S}(\mathbf{u}, \mathbf{p})$. This can be expressed as the complex modulus of the Fourier coefficient:

$$\mathcal{S}(\mathbf{u}, \mathbf{p}) = |\hat{F}(\mathbf{u}, \mathbf{p})| \tag{4.5}$$

The space-time local Fourier transform produces $N_x \times N_y \times N_t$ frequency components. A spectrum feature vector is constructed for the pixel $\mathbf{p}$, by concatenating the Fourier coefficient values in a 1D vector as:

$$\mathbf{v}(\mathbf{p}) = [\mathcal{S}(\mathbf{u}_1, \mathbf{p}), \mathcal{S}(\mathbf{u}_2, \mathbf{p}) \cdots \mathcal{S}(\mathbf{u}_M, \mathbf{p})] \tag{4.6}$$

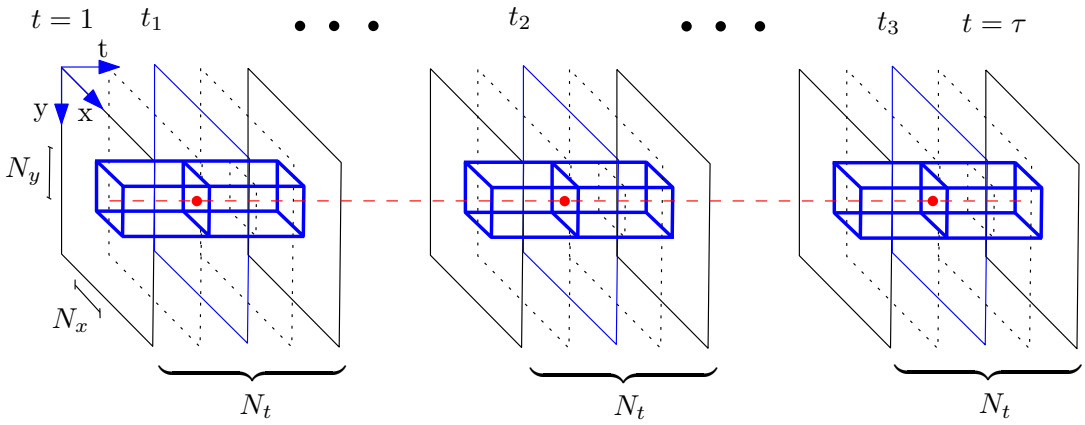


Figure 4.2: An example of sequence containing τ images for learning background. Three spectrum feature vectors $n = 3$ are learned at time instants t_1, t_2 and t_3 during the training period (*i.e.* $t = 1 \cdots \tau$). The location of pixel is represented by red dots in spatiotemporal window Ω of size $N_x \times N_y \times N_t$.

where

$$M = N_x \times N_y \times N_t$$

For color images, we compute the local Fourier transform independently on each channel values. For a given pixel, the three spectra are concatenated in $\mathbf{v}(\mathbf{p})$ (in this case, $M = 3N_x \times N_y \times N_t$).

4.4 Scene modeling based on space-time local Fourier transform

The background learning process is as follows. We take the spatiotemporal input data from τ learning images to compute local Fourier transform. We learn n spectra per pixel during training period. The i^{th} learned spectrum vector is:

$$\mathbf{v}^i_{\text{background}}(\mathbf{x}) = \mathbf{v}(\mathbf{x}, t_i) \quad \forall \quad i = 1 \cdots n$$

The frequency background model at a given spatial location $\mathbf{x}$ can be expressed as the set of learned spectrum vectors:

$$\mathcal{M}(\mathbf{x}) = \left\{ \mathbf{v}^i_{\text{background}}(\mathbf{x}) \right\}_{i=1 \cdots n}$$

Figure 4.2 shows the space-time neighborhoods over which training spectra are computed (in this example, $n = 3$).

To learn the dynamic temporal texture in the background, the parameter N_t is crucial.

If we have small period of temporal texture repetition (*i.e.* fast background motion) in an application then we can limit ourselves to use a small value for N_t. Otherwise, repetitive motions with large periods (*i.e.* slow background motion) need a high value of N_t. The remarks are also valid for N_x and N_y (*i.e.* slow and fast varying background in space can be modeled with large and small values of these parameters respectively).

4.5 Object detection

For object detection, we buffer a set of N_t incoming frames in the memory. We take spatiotemporal data around each pixel of this set of incoming frames as explained in section 4.4. We compute a spectrum vector for each pixel, by applying Eq. 4.5 and Eq. 4.6, on current image data of N_t frames.

For each pixel $\mathbf{p}$, the current spectrum feature vector $\mathbf{v}(\mathbf{x}, t)$ is compared with the set of n background learned spectrum feature vectors. Let d be the dissimilarity function between $\mathbf{v}(\mathbf{x}, t)$ and the model associated to pixel $\mathbf{x}$, namely $\mathcal{M}(\mathbf{x})$. We can write it mathematically as:

$$d((\mathbf{x}, t), \mathcal{M}(\mathbf{x})) = \min_{i=1\cdots n} \mathcal{D}(\mathbf{v}(\mathbf{x}, t), \mathbf{v}^i_{\text{background}}(\mathbf{x})) \tag{4.7}$$

where $\mathcal{D}$ is a distance function between two spectrum feature vectors. We choose to use the frequency-wise squared Euclidean distance:

$$\mathcal{D}(\mathbf{v}, \mathbf{v}_{\text{background}}) = \left\|\mathbf{v} - \mathbf{v}_{\text{background}}\right\|^2$$

High value of the distance measure d leads to the following interpretation: the current spectrum vector $\mathbf{v}(\mathbf{x})$ is not close to any of the n learned spectrum vectors of training sequence, and may corresponds to an object pixel in the scene. In other words, current spectrum vector is composed of frequencies which do not exist in the background. We can consider a pixel $\mathbf{x}$ as a moving object pixel if d is greater than a threshold ϵ. Therefore, a foreground image $\mathcal{F}(\mathbf{x}, t)$ is produced by using the following equation:

$$\mathcal{F}(\mathbf{x}, t) = \begin{cases} 1 & \text{if} \quad d((\mathbf{x}, t), \mathcal{M}(\mathbf{x})) \geq \epsilon \\ 0 & \quad\ \text{otherwise} \end{cases} \tag{4.8}$$

In this way the perturbations in the scene, apart from the spatially varying and time repetitive textures, are identified and used for object detection.

In the next section, we analyze the proposed frequency based model and show the

relevance of the model in the particular case of rivers.

4.6 Background spectral analysis and object detection results

The background representation using frequency analysis requires some further explanation and needs to be clarified with examples. We use a video containing a floating object in a river to illustrate our method. We show that using frequency analysis, the discrimination between different background regions and moving objects can be obtained. Two background pixels $\mathbf{x}_1$ and $\mathbf{x}_2$ are marked in an image from the video (see Figure 4.4). An object passes through the pixel $\mathbf{x}_2$ in the water region. We take spatiotemporal region $N_x \times N_y \times N_t = 5 \times 5 \times 3$ and $n = 8$ for respective points in the video.

4.6.1 Background spectral analysis

We expect the spectra $\mathbf{v}(\mathbf{x}_1, .)$ and $\mathbf{v}(\mathbf{x}_2, .)$ to be different enough (high interclass variance) and spectra generated around a given pixel to be similar (low intraclass variance). Moreover, we expect spectra generated during the passage of object at $\mathbf{x}_2$ to be different than the one without object. We represent the spectra for visual comparison in Figure 4.3(a). The magnitude values of local Fourier transform of the two pixel locations ($\mathbf{x}_1$ and $\mathbf{x}_2$ in Figure 4.4) are shown in the frequency domain.

For this data, we found that the spectra of the three RGB components were similar (color saturation is relatively low, causing colors to be located near the black-to-white axis as shown in the histogram of Figure 4.8).

Therefore, we show only one spectrum at multiple time instances. For $\mathbf{p}_{1t} = (\mathbf{x}_1, t)$ and $\mathbf{p}_{2t} = (\mathbf{x}_2, t)$, we show the three spectra at time $t = t_1, t_2$ and t_3. We use logarithmic transformation on the Fourier coefficient values and then normalize them so they remain in the range from 0 to 255. In Figure 4.3(a), the first three columns represent spectra $\mathcal{S}(., \mathbf{p}_{1t})$ and last three columns represent spectra $\mathcal{S}(., \mathbf{p}_{2t})$. It must be noted that the time instances (*i.e.* t_1, t_2 and t_3) are not consecutive. Two prominent properties are highlighted here. The first one is that local Fourier coefficient values of the corresponding frequencies within a spatiotemporal region are rather similar at different time instances. This can be seen observing the values of $\mathcal{S}(., \mathbf{p}_{11}), \mathcal{S}(., \mathbf{p}_{12})$ and $\mathcal{S}(., \mathbf{p}_{13})$. Therefore, it implies that the values of local Fourier transform can be used as a feature for background modeling. The second property is that the values of local Fourier transform are dissimilar for two different regions. This fact can be seen by observing, for example, the values of the corresponding frequencies of $\mathcal{S}(., \mathbf{p}_{11})$ and $\mathcal{S}(., \mathbf{p}_{21})$.

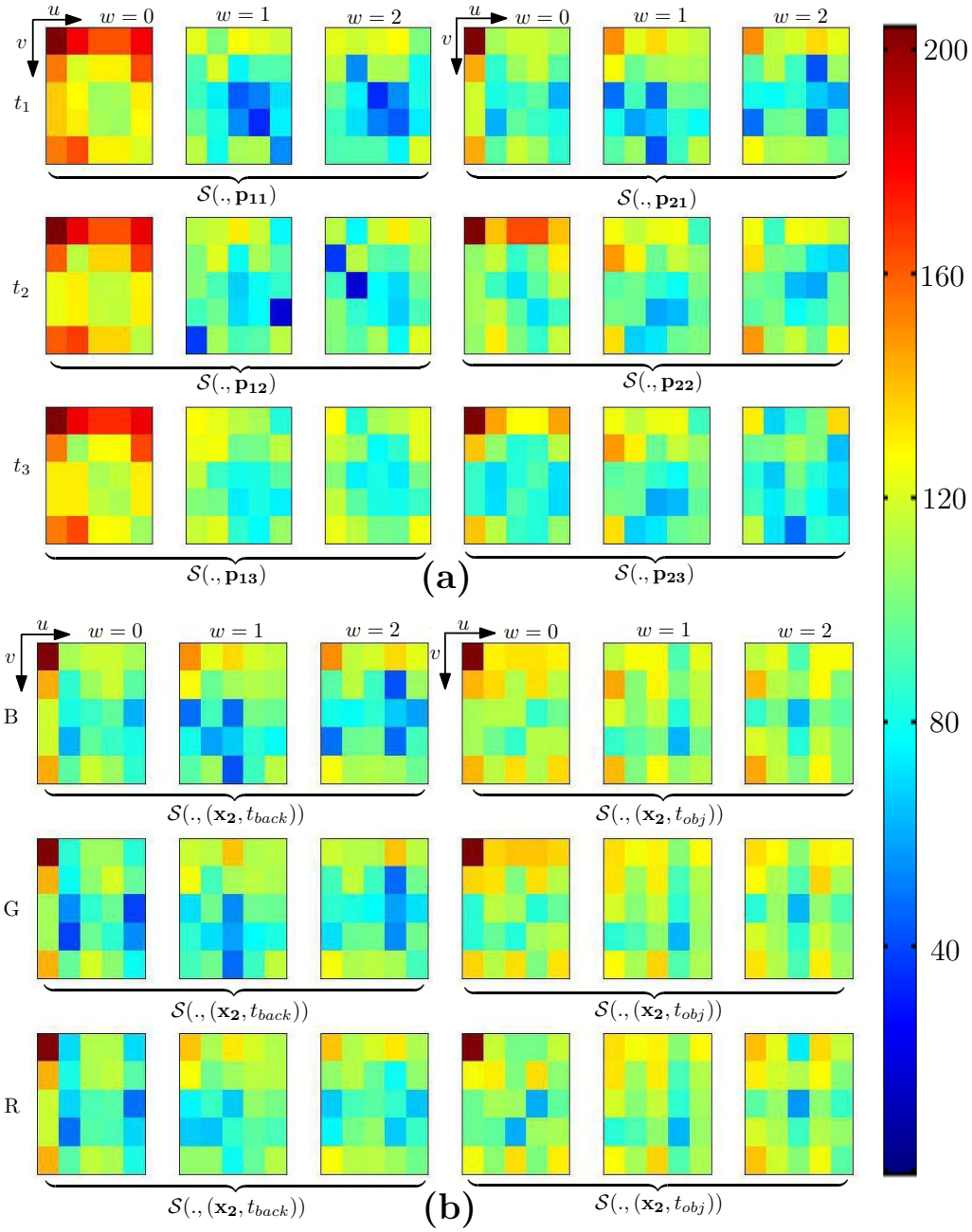


Figure 4.3: Graphical representation of values of local Fourier transform coefficients: *(a)* $\mathcal{S}(., \mathbf{p}_{1t})$ and $\mathcal{S}(., \mathbf{p}_{2t})$ represent spectra for the B channel at spatiotemporal locations $(\mathbf{x}_1, t)$ and $(\mathbf{x}_2, t)$ at time $t = t_1, t_2, t_3$, *(b)* Two spectra $\mathcal{S}(., (\mathbf{x}_2, t_{\text{back}}))$ and $\mathcal{S}(., (\mathbf{x}_2, t_{\text{obj}}))$ for Blue (B), Green (G) and Red (R) color channels of pixel $\mathbf{x}_2$ background only and object passage times respectively.

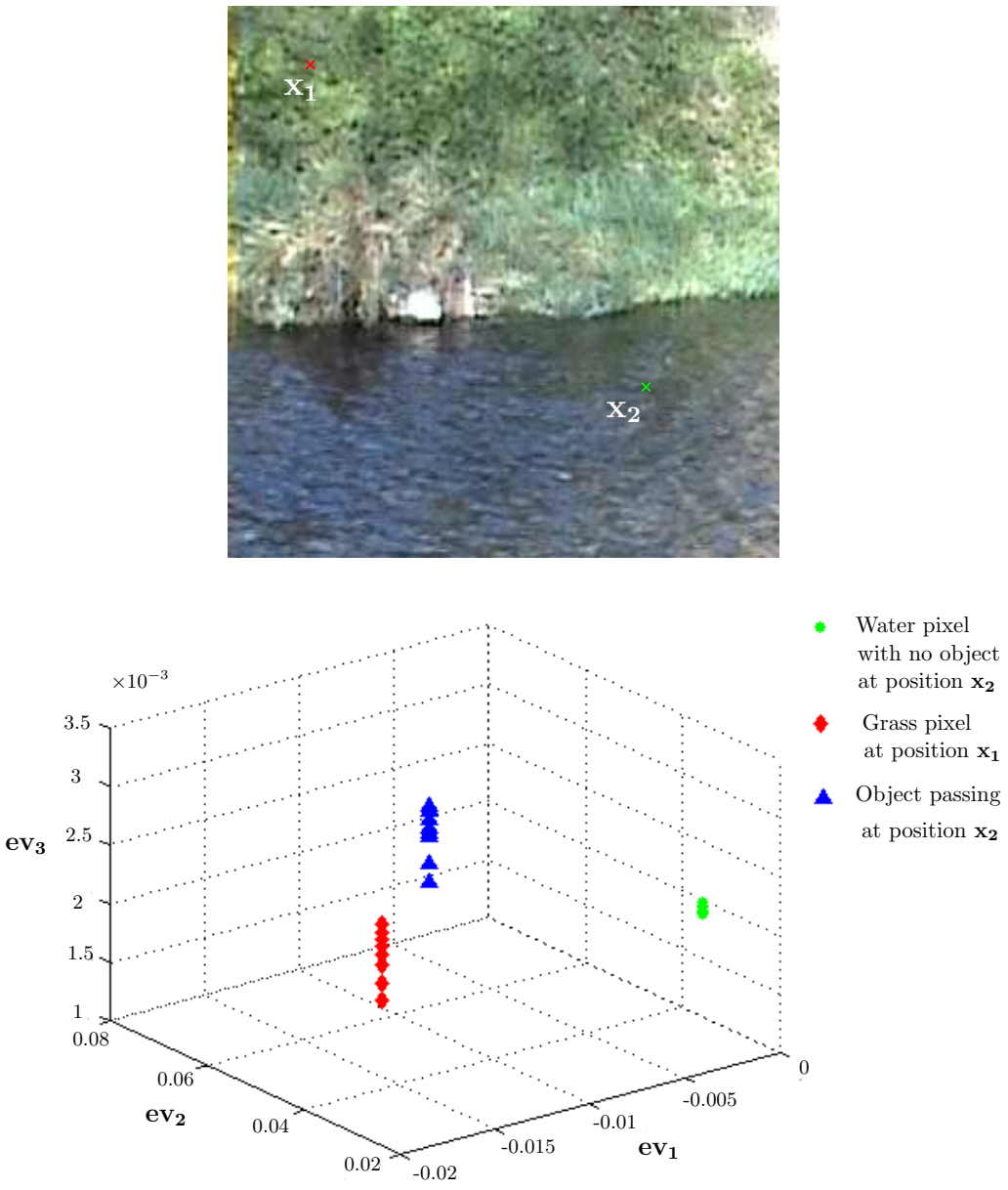


Figure 4.4: An image of background with two points $\mathbf{x}_1$ marked in red and $\mathbf{x}_2$ marked in green color. A subspace linear discriminant analysis (LDA) for the two background pixels $\mathbf{x}_1$, $\mathbf{x}_2$ and an object pixel with spatiotemporal region $N_x \times N_y \times N_t = 5 \times 5 \times 3$ with $n = 8$, data is projected onto first 3 eigenvectors.

We also present the analysis of spatiotemporal frequency components in case of object motion. In the river video, a floating object passes through the pixel $\mathbf{x}_2$. To illustrate the effects of object passage on the spatiotemporal frequencies, we show 3 spectra $\mathcal{S}(.,(\mathbf{x},.))$ for RGB color channels at $\mathbf{x}_2$ (Figure 4.3(b)). Let t_{back} and t_{obj} be respectively the times when $\mathbf{x}_2$ is a background pixel and when $\mathbf{x}_2$ is an object pixel. The first three columns in Figure 4.3(b) represent the spectrum at $\mathbf{x}_2$ with only background. The last three columns in Figure 4.3(b) show the spectra at the same spatial position during the object passage through the point. For these spatiotemporal positions, the coefficient values of the two respective spectra are different. We can observe an increase of the corresponding spatiotemporal frequencies values. This difference is used for object detection.

4.6.2 Projection into discriminative subspace

Since the magnitudes of neighboring frequencies are highly related, we expect to have high correlation between several components of the feature vectors. This leads us to study the relevance of our feature space using dimensionality reduction technique. We use Fisher linear discriminant analysis (LDA), in order to project the feature points onto a subspace that maximizes interclass variance while minimizing intraclass variance.

We present the results of discriminant analysis applied to the set of spectrum feature vectors of $\mathbf{x}_1$ and $\mathbf{x}_2$. For $\mathbf{x}_2$, both background and object spectra are generated, which makes a total of three groups of spectra. We took $n = 8$ spectra per group. We apply LDA on the data matrix and projected the resulting data onto the first three eigenvectors ($\mathbf{ev_1}, \mathbf{ev_2}$ and $\mathbf{ev_3}$) as seen in Figure 4.4. We can observe that the projected data is clustered into distinct areas. This Figure shows that it will be possible, using these features, to distinguish the different color patterns of the respective pixels.

4.6.3 Object detection results

We present our experiments on both synthetic and real natural videos. We use three videos from DynTex database [Péteri et al., 2010], which contains multiple videos with dynamic textures. We also test our algorithm on two other videos. One video containing floating objects and one a speedboat, which we have made.

We compare our results with the GMM [Stauffer and Grimson, 2000] and modified GMM (explained in section 3.4). For the GMM, we use three components (RGB) mixture model, $K = 5$ number of Gaussians per pixel and a relatively high learning rate of 0.05. Similarly, for modified GMM we use RGB mixture model with $K = 5$ and τ

Figure 4.5: A synthetic square of 30×30 pixel moving from left to right with downward moving escalator in the background.

images during learning period. In modified GMM method, recall that pixel-wise color distributions are extracted on a $p \times p$ spatial neighborhood around each pixel. The value of p is mentioned for each application.

In the following paragraphs, we explain the result of our method on synthetically moved and real objects. As explained in section 4.4 and 4.5, our method is composed of two steps, the background learning and object detection. Thus, we use τ images from the videos for the background learning.

Before giving the results of various experiments, let us explain the effects of changing various model parameters on the escalator video from the Dyntex database [Péteri et al., 2010] in detail. An escalator moves downwards in the video. The motion is an example of dynamic texture with large temporal extent. In this experiment, we change both spatial and temporal sizes of neighborhoods per pixel in order to show the effects of these parameters.

Synthetic moving block in the escalator video:
The original video does not contain any object to detect. Therefore, we synthetically move a square portion of escalator as an object.

Motion of this square block is from left to right in the image plane. Few images from the sequence are shown in Figure 4.5. The block is simultaneously translated and rotated with an angle of 5 degrees clockwise per image in 100 consecutive images. It is important to note that we use different sets of images for training and detection.

We show the final foreground image, that is obtained by using Eq. 4.8, with the corresponding parameters in Figure 4.6. We show one foreground image of the image sequence. The effects of changing size of the spatiotemporal neighborhood can be observed. We use odd values from 1×1 to 9×9 for $N_x \times N_y$. When $N_x \times N_y = 1 \times 1$, no spatial neighborhood per pixel is considered, which boils down to extracting purely temporal patterns.

The results obtained with these different parameter settings are shown in the first

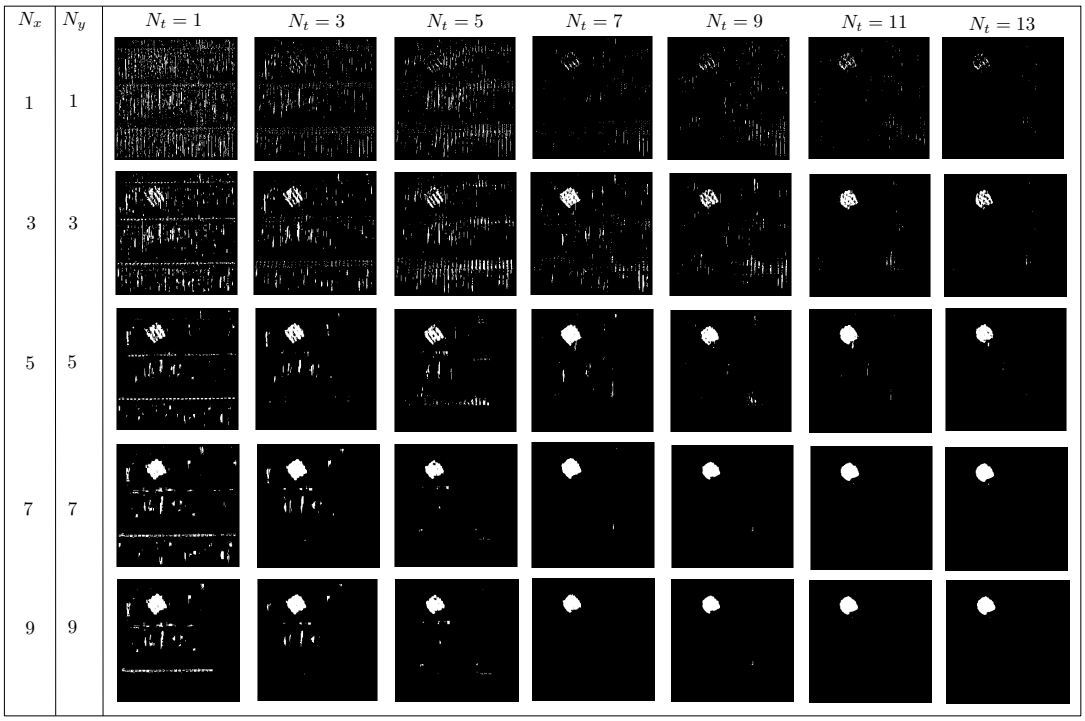


Figure 4.6: Foreground image results with different model parameters of spatiotemporal region $N_x \times N_y \times N_t$ per pixel for a moving square block with moving escalator in the background.

row of Figure 4.6. Similarly, we vary the value of N_t per pixel from 1 to 13. We show the results of our method when only spatial neighborhoods are used (*i.e.* $N_t = 1$) in the first column of Figure 4.6. The escalator motion in the background is slow and increasing the value of N_t enables the background model to capture the periodicity of moving escalator.

We made vary the spatiotemporal region $N_x \times N_y \times N_t$ per pixel from $1 \times 1 \times 1$ to $9 \times 9 \times 13$. We can notice that the object detection results is not improved much above $7 \times 7 \times 11$. Therefore we take the small value, with the point of view of computation time taken during learning and detection. The computation time in training and detection for the video are given in Table 4.2.

We show a few examples of the foreground image from the sequence in Figure 4.7. We show that the frequency-based background model can be used to detect an object even if it has similar colors as the background. For comparison, we also show the results of the GMM and modified GMM. As these two are pixel based approaches, they cannot detect the moving square. Conversely, our method is designed to address the repetitive color structures, therefore, it gives good detection rate.

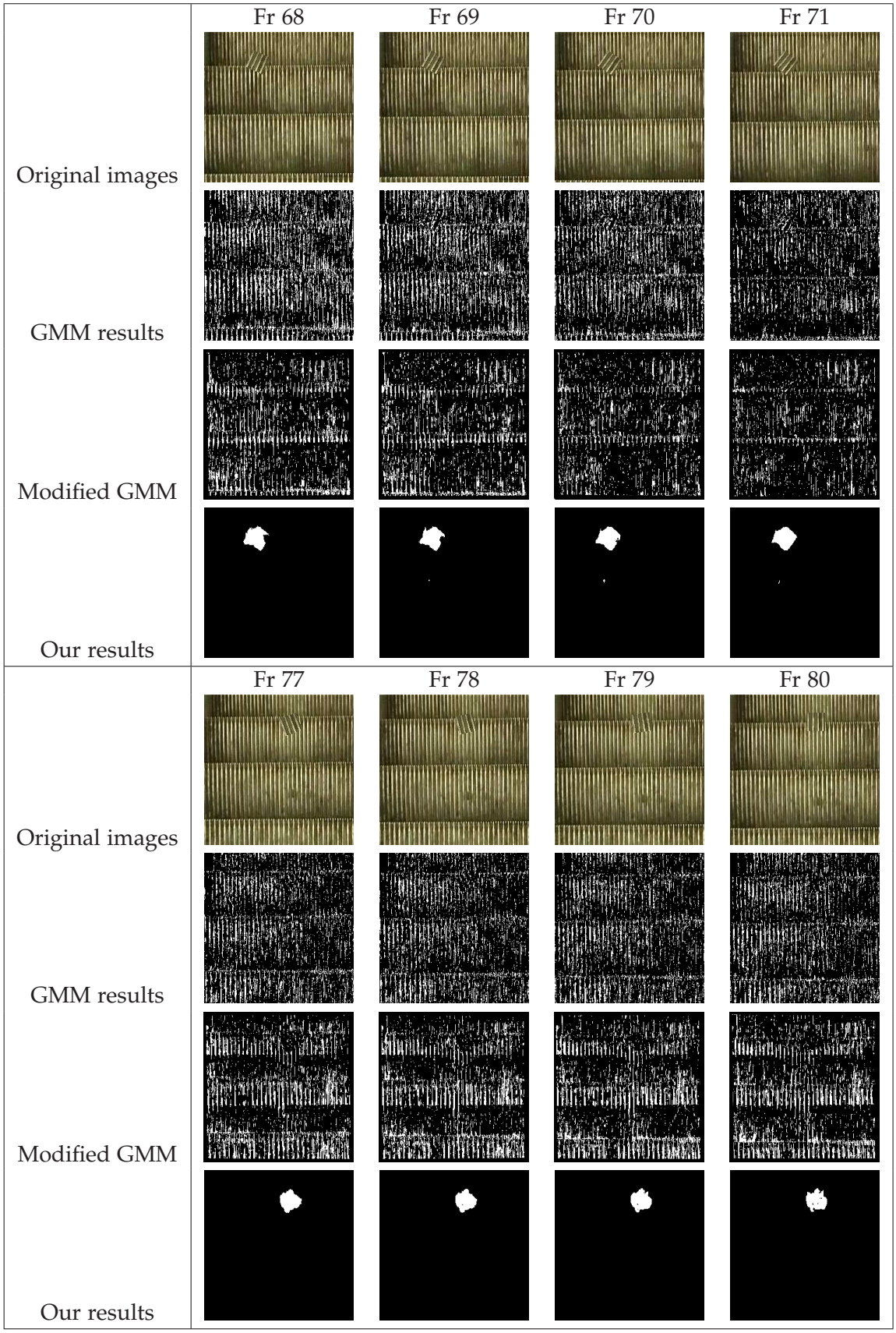
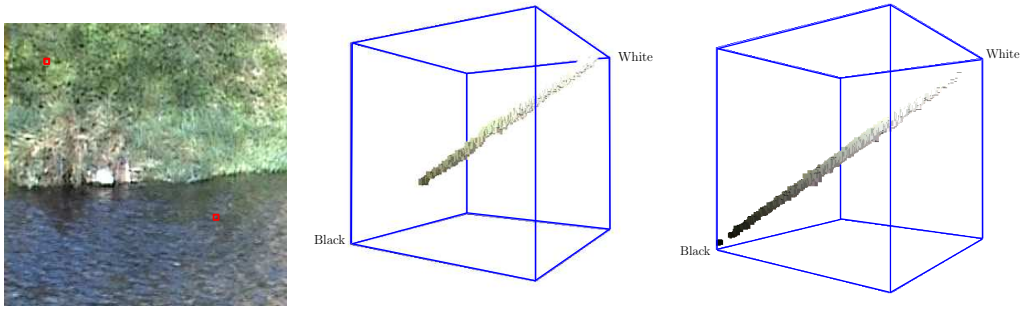


Figure 4.7: A synthetic square of 30×30 pixels moving from left to right with downward moving escalator in the background with corresponding results of: the GMM [Stauffer and Grimson, 2000], modified GMM ($p = 5$) and our method with $N_x \times N_y \times N_t = 7 \times 7 \times 11$.



(a) (b) (c)

Figure 4.8: *(a)* An image from river videos with aquatic environment and reflections from water surface with color histogram of histogram 100 successive frames of 4×4 pixels *(b)* of green leaves and *(c)* in water region.

Floating bottle video:

We apply our algorithm to a river video, in which water ripples form local temporal textures (see Figure 4.8). The color histograms show 100 consecutive color values of two regions of 4×4 pixels size that are highlighted by squares in Figure 4.8(a). The green leaves in the surroundings of river contain repetitive textures from green to light green Figure 4.8(b). The pixels in the water region contain almost all intermediate values between black and white Figure 4.8(c). We can see that pixel values have a wide distributions especially in the water region. The color histograms reveal the fact that pixel-based background models such as the GMM [Stauffer and Grimson, 2000] will not be able to build a relevant background representation in such conditions. The distributions tell us about the color diversity, however, the temporal variations of pixel values in successive frames are not evident from the histograms.

In section 4.4, we introduced n, which is a user-defined parameter. This denotes the number of spectra considered per pixel in the learning period. When there are limited repetitive motions in the background, we expect that a small number of learning spectra would be sufficient. Therefore, we test different values of n on the river video, so that we can observe its effects on the object detection. In the video, a bottle floats from right to left with water flow. In Figure 4.9, we show one foreground image and the effects of different values of n on the object detection results. We show an image from the floating bottle video and the corresponding output results with value of n varying from 1 to 10. We can see that object detection is not improved above $n = 8$. Therefore, we select $n = 8$ in our experimentation.

The quasi-periodic changes which occur in background regions are learned during

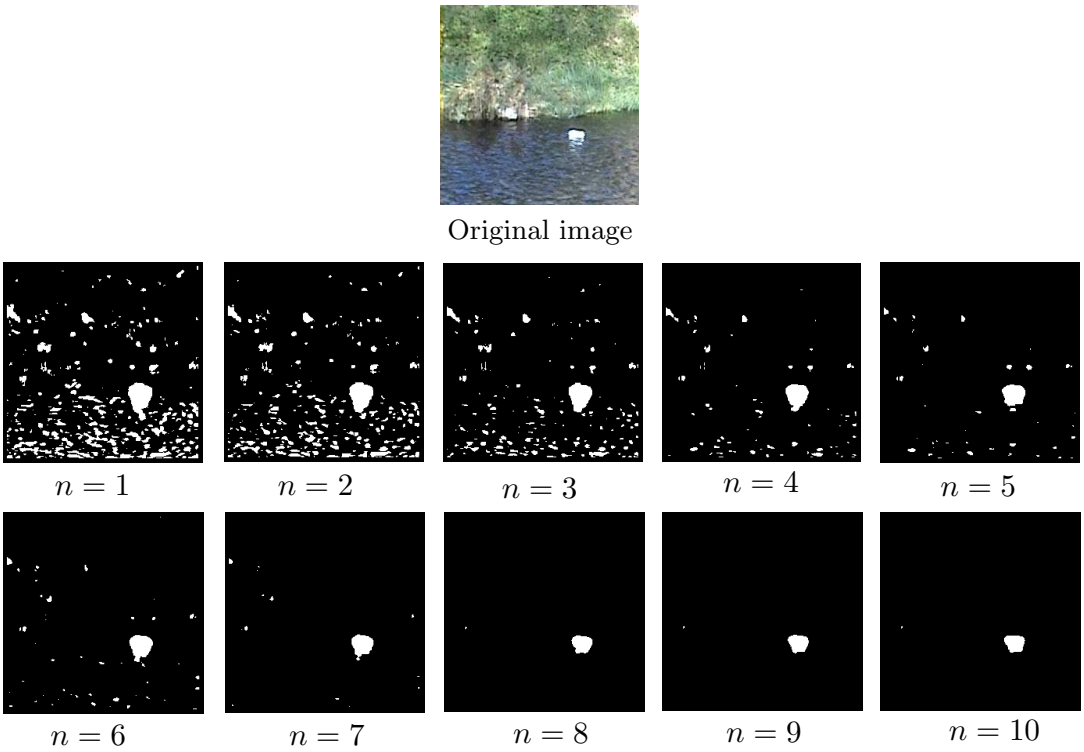


Figure 4.9: Results with variable n, number of spectra per pixel, for floating object sequence and $N_x \times N_y \times N_t = 5 \times 5 \times 5$.

the first τ frames. In this application the spatiotemporal region $N_x \times N_y \times N_t = 5 \times 5 \times 5$ per pixel gives optimal results. Also, we show the GMM and modified GMM results for the video for comparison in Figure 4.10. We can notice that the GMM results contain many false detected background pixels. Therefore, we apply our modified GMM algorithm on the video to see its effects. In the modified GMM, for each pixel, we use spatial neighborhood of width $p = 5$. By applying the modified GMM, we can see that false detections are reduced. However, there are still false detected pixels in the foreground images.

Similarly, we show object detection results obtained by using frequency based background model. The results also indicate that dynamic changes in the aquatic region, waving grass and leaves in the background are well handled by our method.

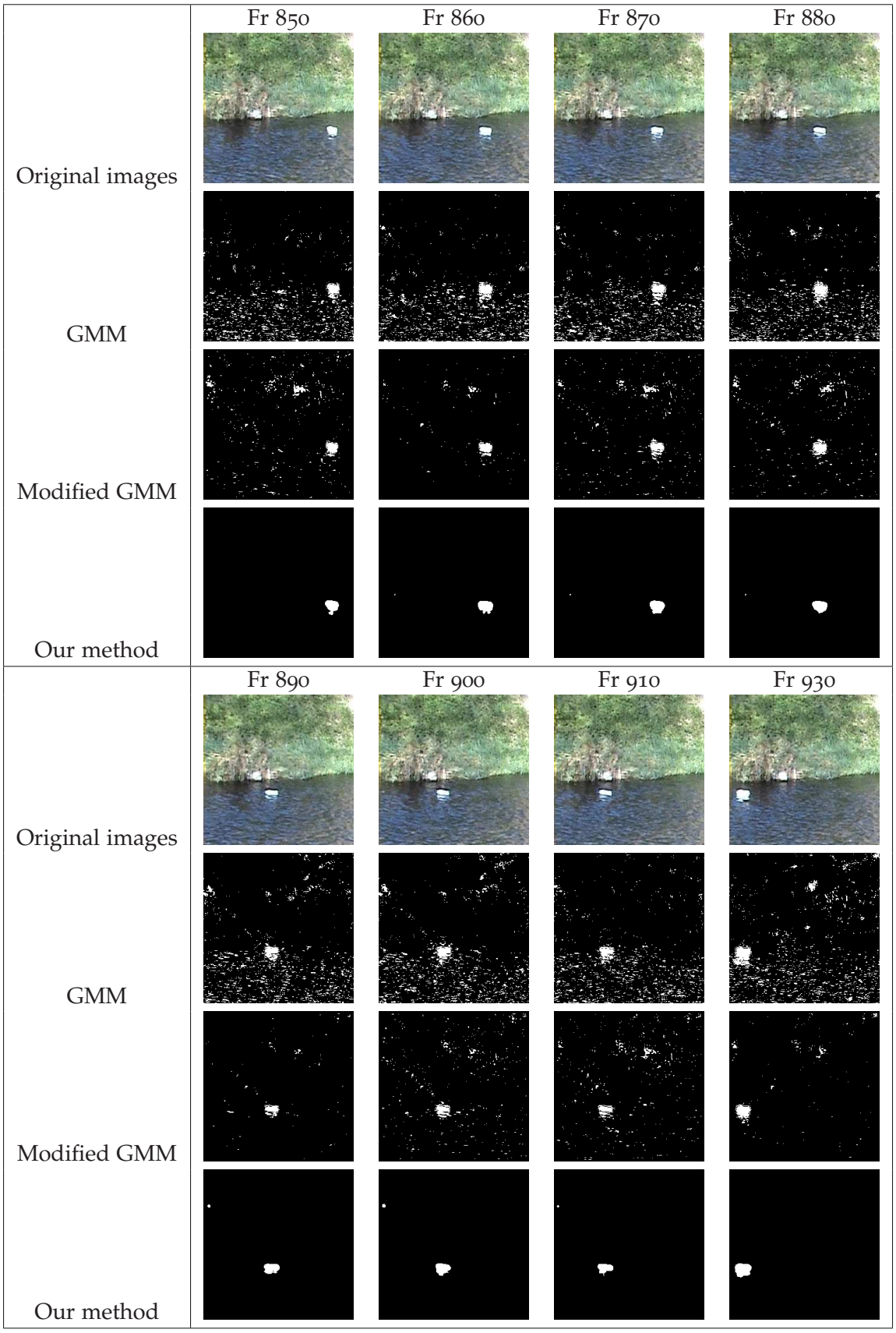


Figure 4.10: Images from a river video with corresponding results of: the GMM [Stauffer and Grimson, 2000], modified GMM ($p = 5$) and our frequency-based background model with $N_x \times N_y \times N_t = 5 \times 5 \times 5$.

Synthetic moving block in the wheat field:
We show that the frequency-based background model can be used to detect an object in dynamic textured natural background. The video is from the DynTex database [Péteri et al., 2010], containing a continuously moving wheat field. Some images from the sequence are shown in Figure 4.11. We can notice the spatially varying and time repetitive textures in the images. The original video does not contain any object to detect. Therefore, we move synthetically a square block as an object. Motion of the square block is from left to right in the image plane (similar to the previous experiment on the escalator video).

We show some resulting images obtained with the GMM, modified GMM and our model for the video. The results with the GMM method are very noisy and are improved to some extent by using the modified GMM. One possible reason is that in the modified GMM method, we take spatial neighborhoods into account to estimate the color distribution, so that pixel-wise statistics are more confident. However, the results of modified GMM have many mis-detected pixels in the square block. The results obtained with the frequency based model contain less mis-detected pixels in square block and few false detections. The optimal dimension of spatiotemporal cuboid is $5 \times 5 \times 5$ for the video. We can see in the results that our method produces equally good detection in such example.

A duck video:
We use a video containing a moving duck in water from the Dyntex database [Péteri et al., 2010]. The background contains water ripples and dark cast shadows of surroundings as seen in Figure 4.12. A duck enters in the scene from the top-right corner and moves across the scene to the middle of the image plane. When the duck moves under the cast shadows, it shares many colors with the background. Figure 4.12 shows results of the GMM, modified GMM and our frequency based background models.

The results obtained with GMM contain a lot of false detection, as the GMM cannot model strong background oscillations. Therefore, many parts of the foreground object (duck in this case) are mis-classified.

Similarly, we apply modified GMM to the video and use $p = 5$ per pixel. The results obtained with the modified GMM are less noisy but still contain many false detections. Whereas the results of our frequency based background model show that not only we have minimum false background pixels but also the foreground object has a few mis-detected pixels. There is a slight over segmentation in some results. However, the error is negligible.

A speedboat video:

Finally, we test our method to a video containing a moving speedboat in a river. The image sequence is gray scale. A few images from the sequence are shown in Figure 4.13. It contains original image sequence with corresponding results of GMM background model, modified GMM and our frequency based background model. The results obtained with GMM contain a lot of false detection of water ripples. We can notice that neither GMM nor modified GMM could model this dynamic background well. Therefore, many false detected pixels are present in these results. For this video, we use $p = 3$ for each pixel in the modified GMM method.

However, the results of our frequency based background model show that not only we have minimum false detections but also good foreground object rate. During speedboat motion, waves are generated around it and these water waves could not be separated from moving boat regions. These waves are detected by the GMM, modified GMM and our method. The waves produced due to motion do not appear during the learning. They produce high frequency coefficient values in the respective pixels, therefore, they are detected as well. However, keeping in view the camera object distance, the error can be neglected. Let us remember that we do not use any morphological (erosion or dilation) operations in any of our results.

Quantitative comparison:

Results of image segmentation of the three background models are evaluated with the Dice similarity measure. Ground truth images are available for the synthetic object motion in the escalator and wheat videos. For the rest of the videos, we manually obtained ground truth images. For this purpose, we randomly selected ($\sim 15\%$) images per video. Quantitative comparison of image segmentation for the applications are summarized in Table 4.1. We present average Dice value which is computed by summing all the Dice values per image divided by the number of images. Average Dice coefficient values for all videos are very low in case of the GMM and modified GMM, where high values

Table 4.1: Quantitative comparison of Dice similarity measure of GMM, modified GMM and our proposed background model

	Escalator	Duck	Bottle	Boat	Wheat
GMM	0.03	0.11	0.14	0.10	0.15
modified GMM	0.08	0.30	0.28	0.18	0.43
Frequency based method	0.87	0.81	0.80	0.78	0.96

Table 4.2: Computation time by our method during training period and object detection

Video	$N_x \times N_y \times N_t$	Total training time (s)	Detection time per frame (s)
Speedboat	$3 \times 3 \times 3$	35.50	6.12
Bottle Wheat	$5 \times 5 \times 5$	46.82	6.55
Duck	$5 \times 5 \times 11$	76.35	8.78
Escalator	$7 \times 7 \times 11$	197.70	15.40

are obtained with our frequency based background model. We can remark that in the bottle video, the average Dice value is smaller than in other cases due to the reflection in the water that creates some false detections. Image segmentation results indicate the superiority of frequency based background model for object detection over the GMM in dynamic textured and moving background.

Finally, we give the computation time taken by our method during the training periods and object detection. The image size is 256×256 and $n = 8$ in all videos. The method is tested on an Intel Core2 Duo 2.66GHz with 4GB RAM, running a C code. We summarize the computation time in the Table 4.2. We provide the two computation times for each video with the corresponding spatiotemporal window sizes considered in the respective videos. As expected, the computation time increases with the size of spatiotemporal neighborhoods per pixel. With a view to compare, one may note that time taken by the GMM is in the order of 100 ms per frame.

Limitations of our method:

Our proposed method has several advantages over classic background models. Object detection rate is satisfactory in difficult spatially varying and time repetitive textured backgrounds. However, some limitations of the proposed method may appear. First, in our method, we separate learning and detection phases. The background learning is carried out first and object detection is applied afterwards. Time interval between the training period and the detection phase should not be too long (*i.e.* global brightness conditions should remain similar). Secondly, due to lack of model parameters update with time, the effects local brightness change, *e.g.* due to self shadows, in some situations may results in false detections and reduce the fine object boundary details.

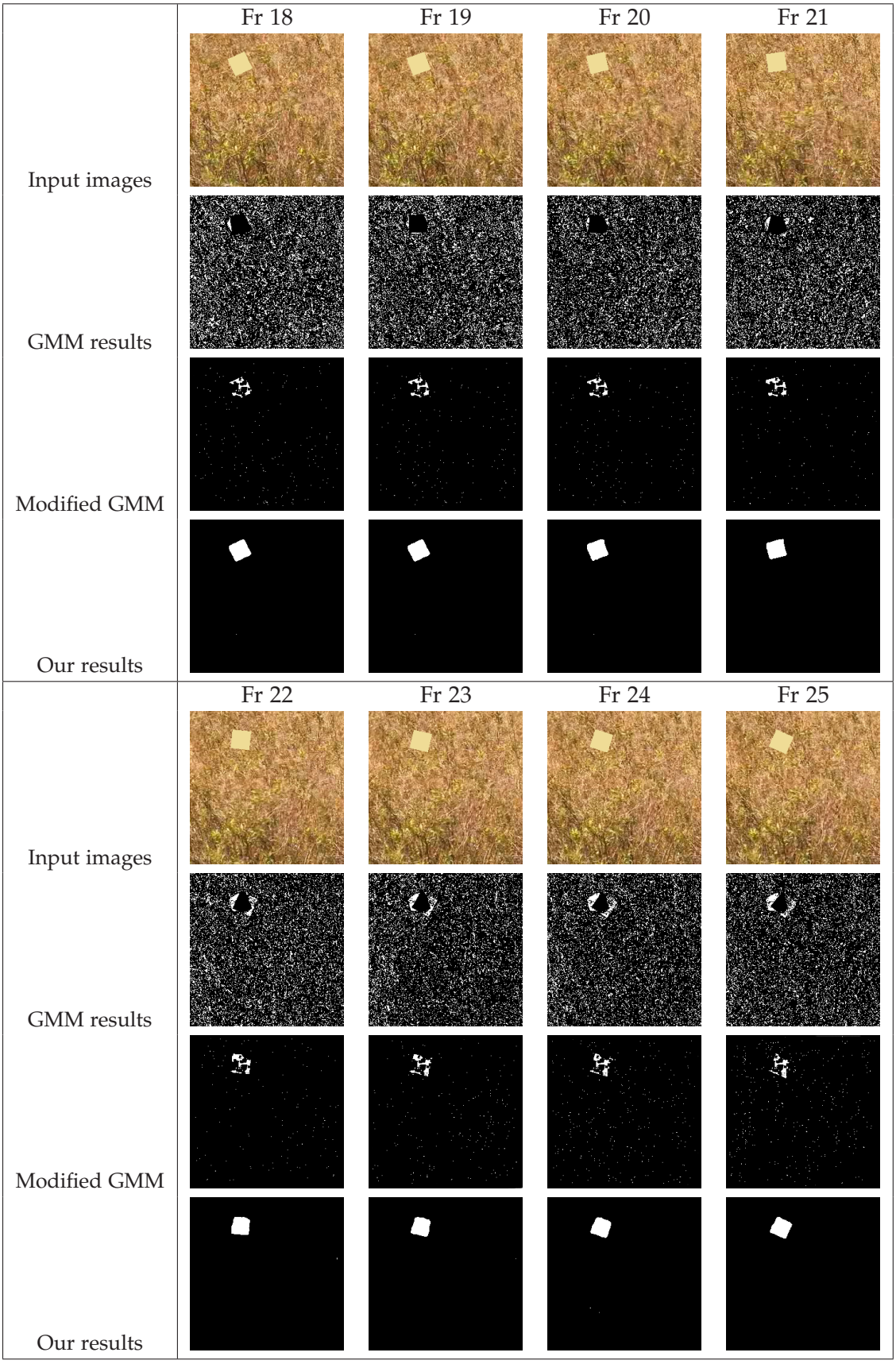


Figure 4.11: A synthetic block in the moving wheat field in the background with corresponding results of: the GMM [Stauffer and Grimson, 2000], modified GMM ($p = 5$) and frequency-based method with $N_x \times N_y \times N_t = 5 \times 5 \times 5$.

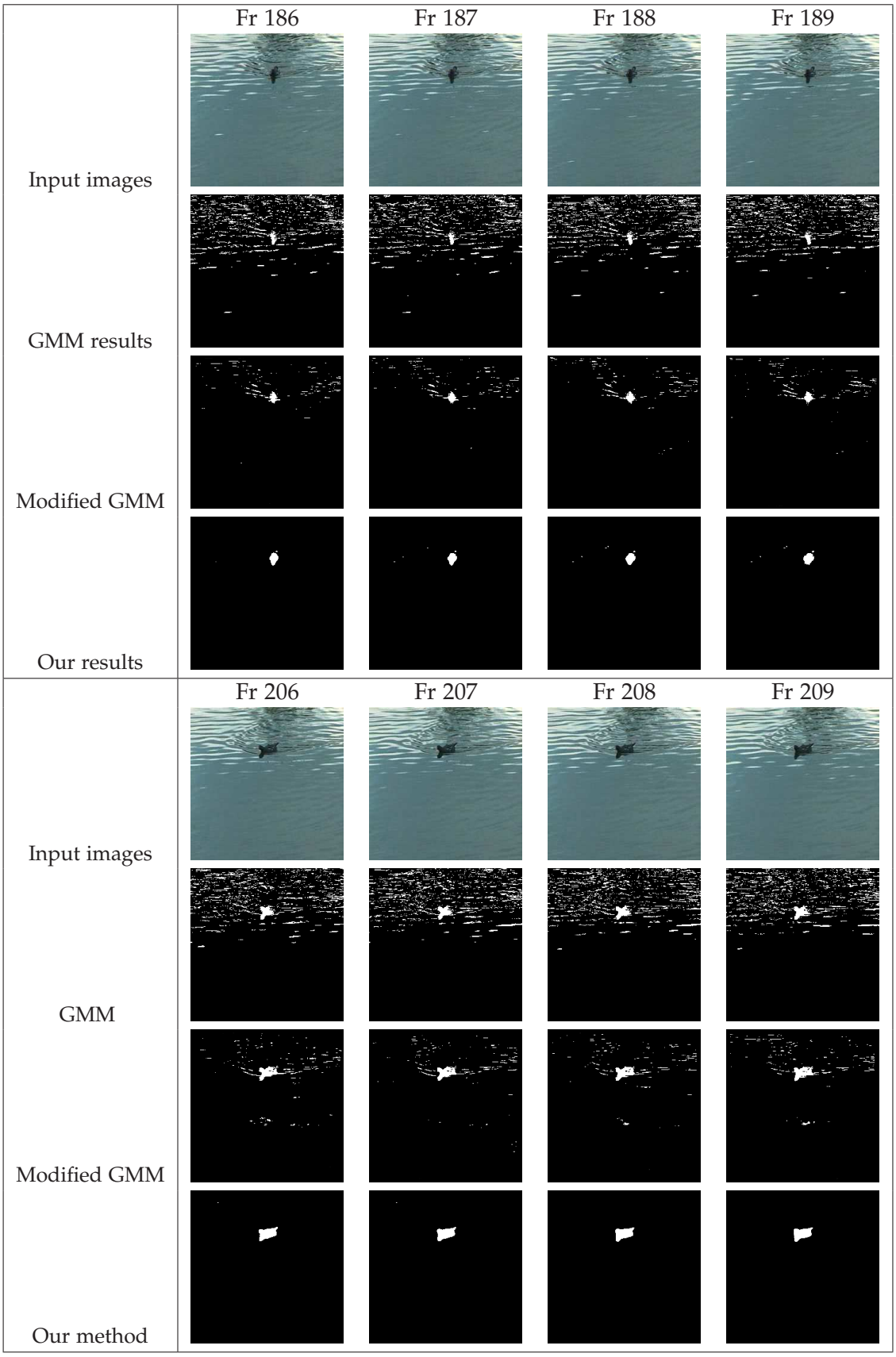


Figure 4.12: A moving duck in rippling water in the background with corresponding results of: the GMM [Stauffer and Grimson, 2000], modified GMM ($p = 5$) and frequency based method with $N_x \times N_y \times N_t = 5 \times 5 \times 11$.

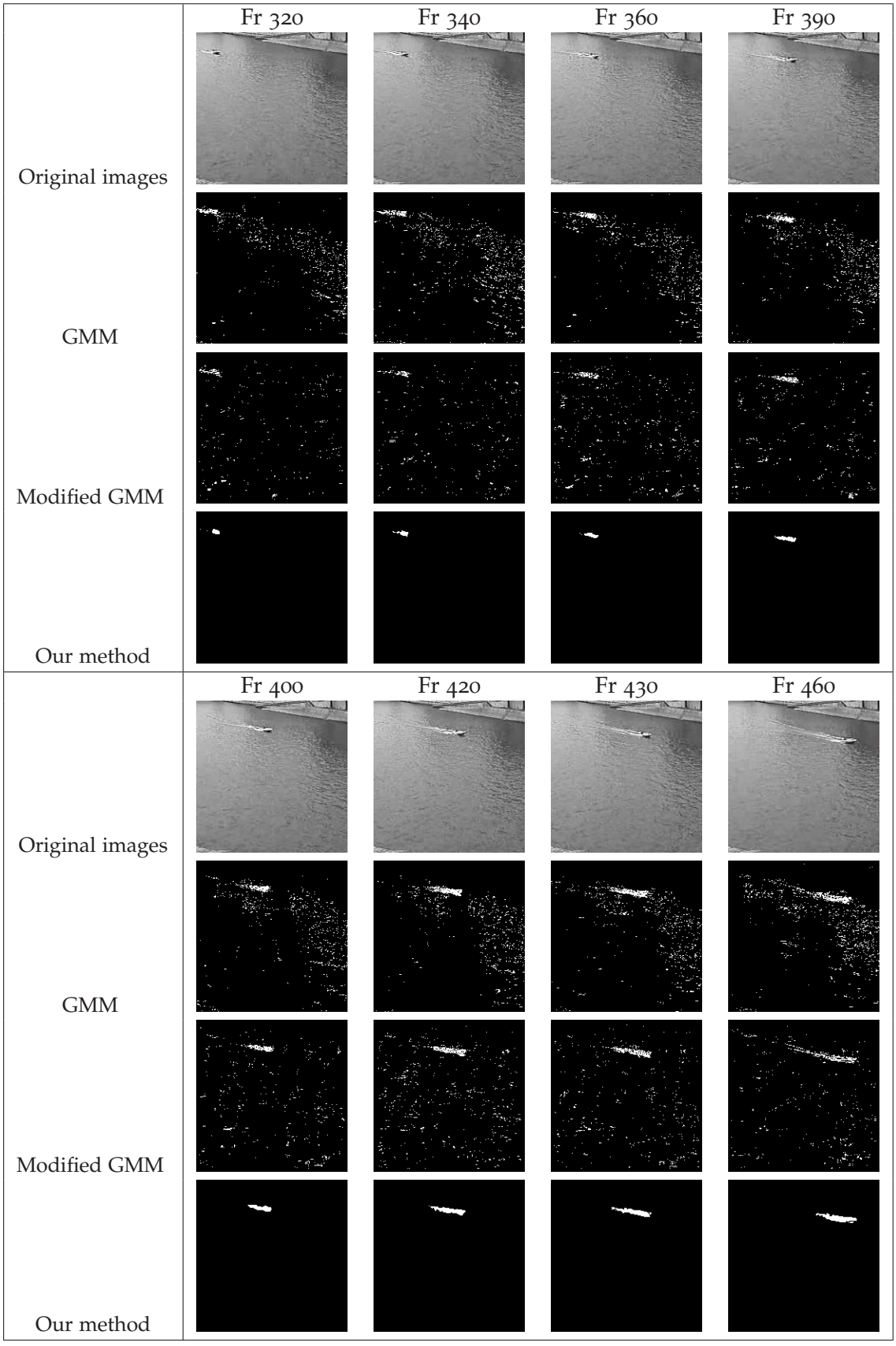


Figure 4.13: Images from a river video with moving boat with corresponding results of: the GMM [Stauffer and Grimson, 2000], modified GMM ($p = 3$) and frequency-based method with $\Omega = 3 \times 3 \times 3$.

4.7 Conclusion

In this chapter, we presented a novel frequency-based model dedicated to moving backgrounds. Spatially varying and time repetitive textures in the background regions are very efficiently modeled by our frequency based method which relies on spectral feature vectors. For example, we demonstrated the effects of changing spatiotemporal neighborhoods around each pixel which change the extent of the spatiotemporal textures that can be properly captured. Also, we apply our method of object detection to both synthetic moving objects and the real ones in the videos. We obtain high accuracy in foreground/background segmentation and outperform a commonly used background model, namely GMM. In outdoor scenarios, our background model leads to better detection and segmentation than the GMM method which fails to capture the time repetitive background motions. As future work, we plan to include an update mechanism in the background model. Among other image phenomena, this could handle global brightness change through the image sequence.

Conclusion and future work

In our thesis, we focused on the videos obtained using fixed cameras with dynamic background. Specifically, we studied videos containing continuous motion of backgrounds and objects. We addressed the fundamental issue of object segmentation in moving backgrounds. The chosen approach consists in background subtraction.

In chapter 1, we presented the existing object detection methods. We explained that color, texture, shape and motion are criteria which the existing methods use for objects detection. Each image pixel in a video frame is classified either as a foreground pixel or a background pixel based on a combination of such criteria. An important part of the literature in this field is related to background modeling that consists in building a representation per pixel of the monitored scene. In some cases, we may acquire *a priori* information on these features, which can be used to improve object segmentation.

In our work, we aimed to detect objects with *a priori* information on their motion and appearance. We may know *a priori* the color information of foreground object or background.

As a matter of fact, in some applications we may have *a priori* information on objects color when we know the type of searched objects. It can be beneficial to use this information in object detection (a famous example of using *a priori* foreground object color is skin-color model). We developed a similar approach where we use wood intensity distribution for its extraction in chapter 2. We have modeled wood intensity distribution with a Gaussian. The obtained model is used jointly with temporal information partially based on inter-frame differences. The model has been tested for wood but we believe that it may be applied on other kind of applications where color of searched objects is known. We compared the results obtained with our image model with the results given by existing background models of the literature.

It is worth noting that the image model we developed has been applied for wood detection, which is a restricted application. It is dependent on wood intensity distribution

in the studied environment. As future work, a non-parametric estimation of probability could be considered in order to deal with larger classes of objects. Another extension could be to consider texture distribution instead of only color one.

Furthermore, foreground objects may be detected using their motion characteristics. Object motion information can be estimated from the data or available *a priori*.

In the former case, optical flow techniques are commonly used to estimate object motion in image sequence. These methods mostly rely on the brightness constancy in the image sequence for object motion detection. Each pixel contributes to motion field computation and the classification of moving and non-moving pixels is obtained thanks to the likelihood of the motion characteristics. However, we studied videos in which objects displacements similar to background motion. Therefore the motion fields obtained by optical flow techniques would not allow discriminating objects from background.

When motion information is available *a priori*, we can use it to improve background subtraction. With a view to studying permanently moving backgrounds, we proposed a rigid motion model. We used prior motion knowledge learned from an image sequence by an offline method.

Our proposed motion model, based on Bayesian framework, can be combined object level motion information with pixel level color information. Moreover, it is designed so that it can be used with any background subtraction method. In this context, we proposed a modified GMM method and combined it with our motion model. We also applied it to wood detection and compared the results obtained with to the ones without motion model.

We have shown that object detection is improved using prior motion knowledge. In such way, false detections in the foreground are reduced.

One can remark that our approach is based on a really simple type of motion (only translation and rotation). As a future work, we could consider more general type of displacements in which object could deform for example. Another extension could be to constrained object detection on its 3D prior motion. It would be more general and allow the object to undergo displacements of any motion in any direction while constraining object detection.

Some moving backgrounds are composed of locally and periodically moving regions. Moving color patterns forming dynamic textures vary spatially and appear repeatedly with time. Thus, they may not be modeled by the existing individual pixel-based or region-based background modeling methods. We have presented a spectral background analysis in moving background context. We have explained that using frequency anal-

ysis we could achieve discrimination in background regions and also between background and moving objects. Our method draws its inspiration from 2D texture segmentation. The main idea behind our approach is to model the spatiotemporal color patterns of the scene and use the model for object detection. We proposed a frequency-based background model founded on the spatiotemporal region around each pixel. In this method, a spectrum is associated to each pixel. A pixel is classified as foreground pixel if its spectrum is different enough to the background spectra, which are extracted during the learning phase.

We have applied the frequency-based model to several videos from DynTex database. Our object detection method has produced good object segmentation in the presence of repetitive color motion in the background. We have compared our results with the GMM method and modified GMM method. The comparative analysis indicates that with the frequency-based background model we can obtain better object detection in apparently complex and moving backgrounds.

However, our frequency-based background model has some limitations. The time interval between the training and detection phases should not be too long. As a matter of fact, if brightness conditions or color patterns are not similar to the learned ones, the detection may fail. Consequently, as future work, an adaptive model could be developed to overcome these problems. Moreover, the spatio-temporal textures may be modeled by an adaptive mechanism with respect to slow and fast moving areas in the background. A method can be based on the temporal extent of texture per region. This can be coupled with small and large number of spectra for slow and fast moving areas in the background.

Appendix A

Application: wood tracking and counting in rivers

Contents

In the introduction of our thesis, we briefly presented the DADEC project, which is constituted of two major parts. The first one is wood detection and the second one is wood tracking and counting in river videos. We addressed the problem of wood detection in chapter 2 and 3 in detail. We notice that wood counting primarily depends on good wood segmentation algorithm in place.

In this appendix, we present the second part of the DADEC project, which consists in counting the wood objects passage through a point of the river at different times of day. The aim is to quantify wood transport within river systems in order to understand the relevant processes and develop wood budgets. For this purpose, a monitoring system was installed at a gauging station on the Ain River, a 3500 km^2 piedmont river (France), in early 2007. Videos are obtained during 12 floods, the duration of which is two to three consecutive days. We study the flood videos of April, 2008 for which manual counting data was available from the geographers. In the manual counting, operators need to watch the videos frame per frame and note the location and time of wood passage. The manual method is time-consuming and limits the amount of data that can be processed. Automatic wood detection and counting by computer vision techniques would allow to accelerate counting and to increase the number of processed datasets. The proposed wood detection and counting software has been delivered to geographers. They have been using it for wood counting purposes since July, 2010.

It must be noted that the number of wood objects in any video frame is variable *i.e.* they do not come one by one in the video and may appear in any part of the scene randomly (for example see Figure A.2). However, these objects move with water, therefore, their direction of motion is known *a priori*. Similarly, there is continuous global motion in the scene. The difficulties which we face in terms of wood tracking and counting are summarized in the following section.

A.1 Problems and constraints in wood tracking and counting

Wood tracking in the globally moving background is difficult for several reasons. The constraints related to wood tracking are complemented to wood detection problems which we discussed in chapter 2. These can be summarized as follows:

- A major difficulty arises due to the turbulences in water during floods. Water waves are stronger and more persistent in floods than in normal river flow. Moreover, during day light, water waves resemble small wood objects. Therefore, it is difficult to distinguishing water waves from wood in such conditions. So, the

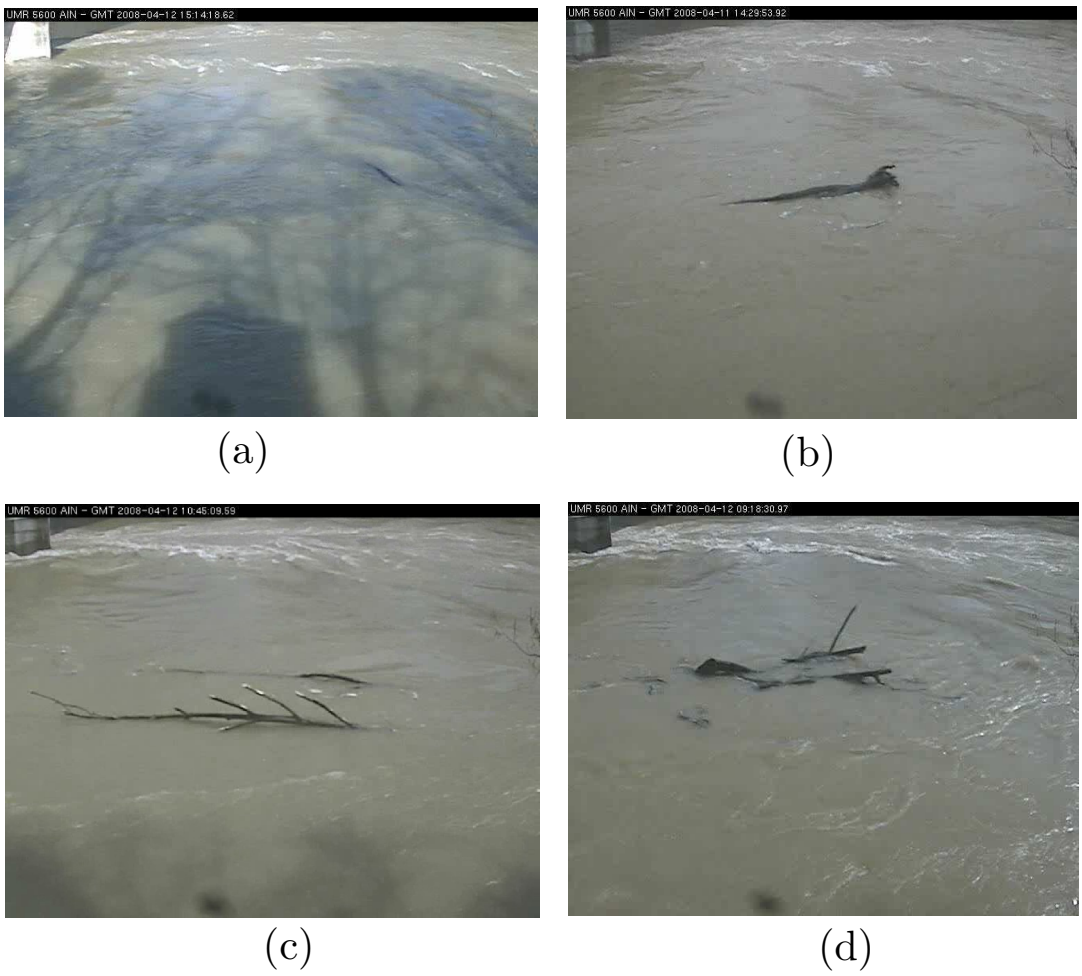


Figure A.1: Some examples of different wood shapes, *(a)* a small wood object, *(b)* a tree trunk, *(c)* a tree with multiple branches and *(d)* a complex wood shape.

proposed method is composed of two steps: In the first one, the object detection method developed in chapter 3 (image model combined with motion model) to extract objects. In the second step, the remaining false detected water waves are eliminated by a tracking algorithm, so that they may not affect wood counting.

- For wood tracking, wood objects should be visible during their passage in videos as a single entity. However, this is not true in most of the cases, as wood objects are partially submerged in the water (see Figure A.4). In addition, some small objects are submerged totally during some frames and reappear afterwards.

- Another problem related to partial occlusion is that these wood objects split into multiple parts and appear as separate wood objects (see for example Figure A.1(d)).

- Floating wood may appear in a variety of shapes. These include small parts of trees, large branches of trees with leaves, wood debris and roots of trees *etc.* There-

Figure A.2: An example image of multiple small wood objects in a single video frame.

fore, apparent sizes of wood objects vary from few pixels for small objects to hundreds of pixels for large objects. Some representative wood objects are shown in Figure A.1.

- Small wood objects rotate and translate more than large wood objects, which should be taken into account when learning the parameters of the motion model. We show a small wood piece rotating during motion in Figure A.3.
- The number of frames in which a wood object appears is variable in the videos we studied. The number of frames is from one to a maximum of 13 consecutive frames.
- Similarly, wood motion trajectories vary with the water speed and wood sizes. During floods, the water speed is high and turbulent from normal flow. An image exhibiting multiple wood passage trajectories in a small video are shown in Figure A.5.
- Finally, due to the remote location of the monitoring system, the frame rate has been adjusted very low by the geographers. They use Internet connections for transferring videos from the monitoring location to the database center. Therefore, the choice is justified from the point of view of geographers, however, it has consequences for our developments and in particular on objects displacements.

Figure A.3: An example of small rotating wood object.

The proposed algorithm should cope with the above-mentioned difficulties with minimum counting errors. A "good" wood counting algorithm should have the following properties:

- False counting of water waves as wood objects should be minimum.
- Wood objects do not come one by one and may appear randomly in any part of the river. Therefore, the counting method should be able to count them with minimal scene related assumptions. The only assumption we will do here is that water flows from left to right.
- A wood object should be counted once even if it is split into many parts during its passage in the video.

We present our wood tracking and counting method in the following sections.

A.2 Wood tracking in video

In object tracking methods, the trajectory of an object over time is basically generated by locating its position in every frame of the video [Yilmaz et al., 2006]. The tasks of detecting the object and establishing correspondence between the object instances across frames can either be performed separately or jointly. In the first case, possible object regions in every frame are obtained by means of an object detection algorithm, and then the tracker makes correspondence of objects across frames. In the latter case, the object region and correspondence is jointly estimated by iteratively updating object location and region information obtained from previous frames. In our wood detection method described in section 3.5.2, a joint intensity and motion based approach was proposed. In this way, it is related to the two conventional tracking methods, as segmentation result at time t is used to extract foreground objects at time $t+1$. We use a method in which the corresponding extracted foreground objects are related within successive frames. Therefore, the tracking method we use for wood counting can be regarded as a temporal linking method.

The methodology of wood tracking can be subdivided into temporal linking and counting. Before explaining the two parts, we present the extraction of the representative points of objects in the foreground image.

A.2.1 Extraction of representative points

The foreground image $\mathcal{F}(.,t)$ is composed of several connected components. Due to the partial occlusions of wood objects, a given piece of wood may appear in several small pieces. Therefore even if the two frames contain the same number of objects, they may contain a different number of connected components. An example of which is shown in Figure A.4. In order to group several connected components that may correspond to the same object, we first rely on centroids. Such approach was already used in object tracking, see for example [Veenman et al., 2001]. The centroid c_{ψ_i} of a given component ψ_i is taken as the representative of ψ_i. Note that it is the centroid of region pixels:

$$\boldsymbol{c}_{\psi_i} = \frac{1}{|\psi_i|} \sum_{\mathbf{x} \in \psi_i} \mathbf{x}$$

To evaluate the closeness between two connected components ψ_1 and ψ_2, we choose to consider the euclidean distance between $\boldsymbol{c}_{\psi_1}$ and $\boldsymbol{c}_{\psi_2}$. This distance allows us to label the connected components of the same object even if their size vary from one frame to another due to occlusion. We perform hierarchical grouping of connected components as long as the distance between their centroids is below a threshold λ. At each step, the two closest connected components ψ_a and ψ_b are merged in a new region whose representative center is assigned to the average $(\boldsymbol{c}_{\psi_a} + \boldsymbol{c}_{\psi_b})/2$, until $\left\|\boldsymbol{c}_{\psi_a} - \boldsymbol{c}_{\psi_b}\right\| < \lambda$. Let $\psi = \{\psi_i\}_{i=1...n}$ be a set of gathered connected components. Its representative center $\boldsymbol{c}_\psi$ is the average of centroids $\boldsymbol{c}_{\psi_i}$, and the following relation is verified:

$$\psi = \{\psi_1, \cdots, \psi_i, \cdots \psi_n\} \Rightarrow \boldsymbol{c}_\psi = \frac{1}{n} \sum_{i=1}^{n} \boldsymbol{c}_{\psi_i} \quad \text{and} \quad \left\|\boldsymbol{c}_{\psi_i} - \boldsymbol{c}_\psi\right\| < \lambda \quad \forall i \in 1...n$$

This method is robust to partial occlusion of wood in water. Hence, every object in the frame is localized by a representative point, which is linked to its corresponding point in the next frame.

A.2.2 Temporal linking of floating wood

We build a list of barycenters of all objects that are linked temporally in consecutive frames.

As we mentioned earlier, wood objects can appear in any part of river, therefore, each object is considered as a potential piece of wood, regardless of its location in the image plane.

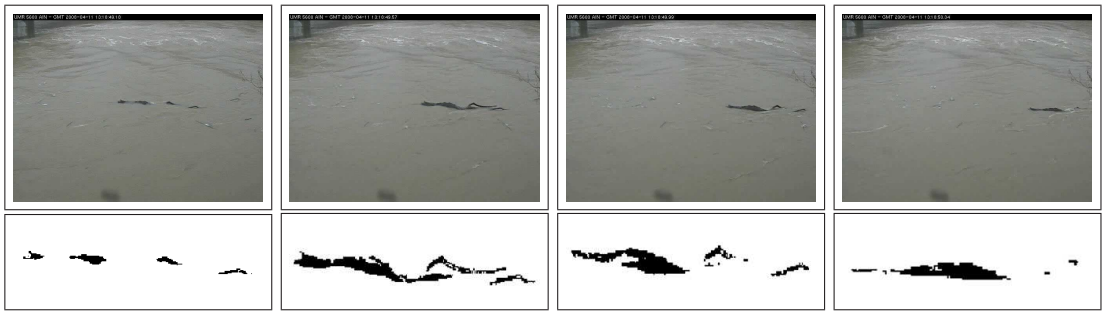


Figure A.4: Four consecutive frames of a moving wood object, zoomed resulting segmented object regions show the appearance of floating wood in the images.

Let $\boldsymbol{c}_\psi(t)$ and $\boldsymbol{c}_\psi(t+1)$ be the representative points of object ψ matched in two consecutive frames. This is actually verified if their distance is below δ:

$$\left\|\boldsymbol{c}_\psi(t) - \boldsymbol{c}_\psi(t+1)\right\| \leq \delta \tag{A.1}$$

where $\delta = 100$ pixels, which is the maximal displacement of wood pieces we learned after experimentally testing on different videos. Incidentally, it allows us to give a lower bound of threshold λ. If λ is lower than δ, objects may be mismatched in consecutive frames. A component of an object may be mistakenly matched with another component of the same object, which does not happen if $\lambda > \delta$. Conversely, two objects moving simultaneously can be counted separately.

The positions of the representative point of an object in consecutive frames can be linked by line segments, which yields a trace in the *summary image*. Such an image is a graphical representation of trajectories during a given duration, as shown in Fig. A.5. It represents wood and water waves trajectories in a small portion of a video. We can notice from the summary image that wood objects make longer traces than waves. The summary image also exhibits that water waves disappear after some frames. This property is used in the following section, to distinguish wood pieces from waves.

A.2.3 Counting wood pieces

We assume that wood objects are more persistent than water waves. Therefore, wood objects are present in more consecutive frames compared to water waves. We attribute an object as a wood object when it is present in sufficiently high number of frames. Let K be this minimal number of frames. Hence, we can easily eliminate water waves by using this method. Thus, it provides wood counting results with less number of false detected water waves.

The method of counting is devised in a manner that if a wood object undergoes total

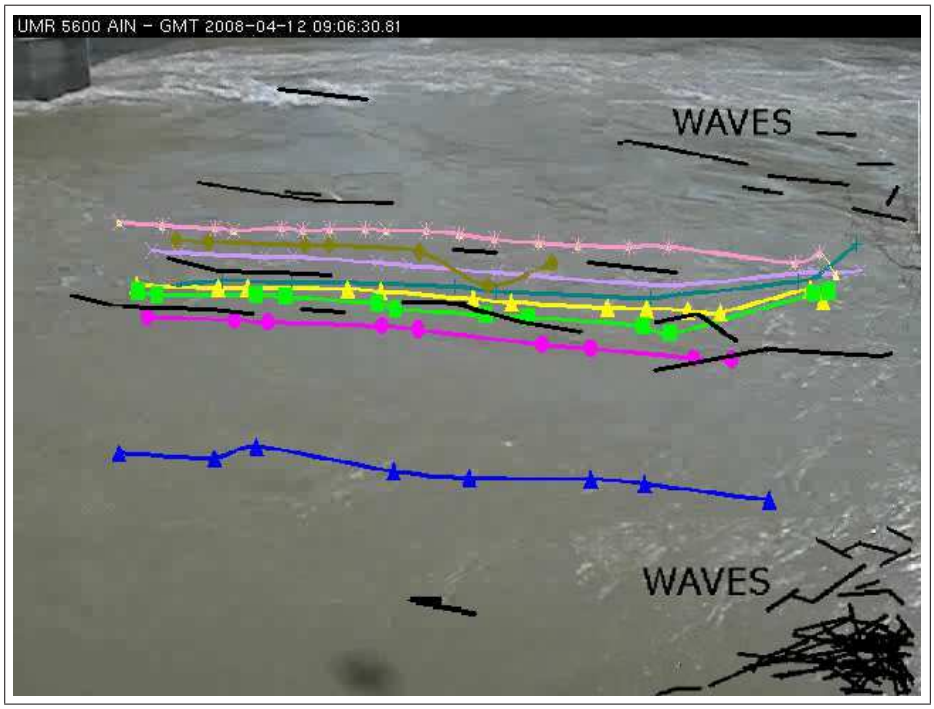


Figure A.5: A summary of a small video representing wood (in different colors) and water waves (black color) trajectories.

submergence and reappear after few frames, it is counted only once.

For each object ψ, we determine the number of consecutive frames in which it appears. We obtain a sequence of n representative points $\{\boldsymbol{c}_\psi(t), \boldsymbol{c}_\psi(t+1), ..., \boldsymbol{c}_\psi(t+n-1)\}$ in which each couple $(\boldsymbol{c}_\psi(t+i), \boldsymbol{c}_\psi(t+i+1))$ verifies Eq. A.1. Wood pieces and water waves are separated from one another according to their persistence in the consecutive frames. Hence, the chosen criterion to consider ψ as a wood piece is $n \geq K$. If a wood piece is not totally submerged, its representative point at different times should all be linked two by two. Unlike wood pieces, waves generally disappear after three or four frames. Hence, floating wood is counted on this basis. The relevancy of this limit is evaluated in section A.3.

A.3 Experimental test

In chapter 2, we explained the two wood detection methods, namely, naive and probabilistic model. We use the wood counting method on the segmentations provided by these two methods. As explained earlier, no assumption is made about the location of wood pieces in the water or according to static parts such as bridges. Consequently, our method could be used with another camera in a likewise scene. We test our wood counting algorithm on five videos of 1500 frames each, for which we have ground truth

Table A.1: Quantitative evaluation of wood counting with naive and probabilistic methods

	Naive method				Probabilistic method		
Video	N_t	N_d	N_{pd}	N_w	N_d	N_{pd}	N_w
1	45	40	5	13	41	4	1
2	87	75	12	10	80	7	3
3	52	43	9	7	47	5	0
4	41	38	3	6	37	4	0
5	85	76	9	19	79	6	4
Total	**3**10	**2**72	**3**8	**5**5	**2**84	**2**6	**8**
		87.7%	**12.3**%		**91.6**%	**8.4**%	

counting data for validation. Frames have size 640×480 and are extracted from MPEG4-compressed streams. Tests are executed on an Intel Core2 Duo 2.66GHz with 4GB RAM running C code.

In section A.2.3, we introduced the minimal number of consecutive frames K during which objects should appear to be counted as wood. This number is evaluated in Figure A.6. Qualitative evaluation is given by *Precision* and *Recall* measures, defined as:

$$Precision = \frac{N_d}{N_d + N_w} \quad ; \quad Recall = \frac{N_d}{N_t}$$

where N_d is the number of detected wood pieces, N_w is the number of waves detected as wood, N_t is the total number of wood objects and N_{pd} is the number of non detected wood pieces *i.e.* ($N_t = N_d + N_{pd}$) .

We plot the values of N_t, N_d and N_w in Figure A.6(a) against values of K. We also show the precision and recall curves obtained with the probabilistic model, for one video (Video 1) in Figure A.6(b). We can see that the best trade-off between true and false wood counting is obtained with $K = 4$. Successful wood counting is validated manually by visual inspection frame per frame.

Table A.1 shows the quantitative evaluation in terms of wood pieces actually present and counted as wood, the number of wood pieces that are not counted by our algorithm and the number of waves that are detected as wood pieces. The number of wood detected is clearly higher percentage than the number of non detected wood. In some situations, the brightness of waves are very close to wood pieces and they may rest for more than four frames in some cases. If the water waves are continuously present in four frames false detection occurs but the percentage of such false detection is not very important. Moreover, wood pieces sometimes appear in one or two frames, such type

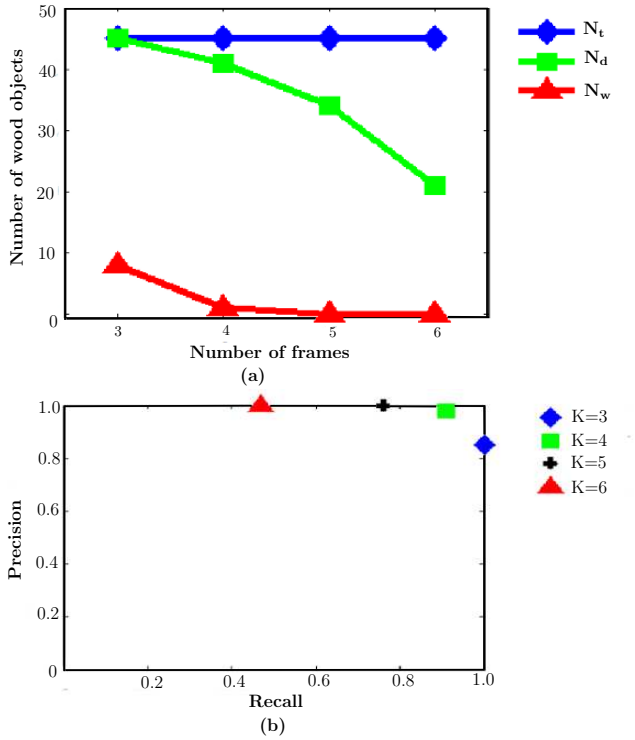


Figure A.6: (*a*) Quantitative comparison, for probabilistic wood detection method, of selection of the number of consecutive frames for wood attribution for video 1, (*b*) precision and recall for the video evaluated for $K = 3, 4, 5$ and 6 consecutive frames.

of wood pieces cannot be detected. Wood detection rate is nearly 98% while successful counting rate above 90%. Wood detection rate by naive and probabilistic methods are similar. But, the difference lies between the two results is the number of detected water waves as wood. We can clearly see this in Table that number of water waves with the probablistic image model is reduced drastically, which indicates that wood counting results depends on good wood segmentation results.

After visual inspection, it turns out that undetected wood pieces correspond to very small parts, which are not critical with respect to the application. These small pieces are often totally submerged in some frames.

Bibliography

B. Abraham, O. I. Camps, and M. Sznaier. Dynamic texture with Fourier descriptors. In *International workshop on Texture Analysis and Synthesis*, pages 53–58, 2005. 92

I. Ali and L. Tougne. Unsupervised video analysis for counting of wood in river during floods. In *International Symposium on Visual Computing*, volume 5876 of *LNCS*, pages 578 – 587. Springer, 2009. 45, 47

I. Ali, J. Mille, and L. Tougne. Wood detection and tracking in videos of river. In Springer Verlag, editor, *Scandinavian Conference on Image Analysis*, Lecture Notes in Computer Science, pages 646–655, 2011. 47

K. Babalola, B. Patenaude, P. Aljabar, J. Schnabe, D. Kennedy, W. Crum, S. Smith, T. Cootes, M. Jenkinson, and D. Rueckert. Comparison and evaluation of segmentation techniques for subcortical structures in brain MRI. In *International Conference on Medical Image Computing and Computer-Assisted Intervention (MICCAI)*, volume 5241 of *LNCS*, pages 409–416, New York USA, 2008. Springer. 53

S. Baker, D. Scharstein, J. P. Lewis, S. Roth, M. J. Black, and R. Seitz. A database and evaluation methodology for optical flow. *International Journal Computer Vision*, 92(1): 1–31, 2011. 32

J. R. Bergen, P. Burt, R. Hingorani, and S. Peleg. A three frame algorithm for estimating two-component image motion. *IEEE Transactions Pattern Recognition Machine Intelligence*, 14(9):886–896, 1992. 30

A. C. Bovik, M. Clark, and W. S. Geisler. Multichannel texture analysis using localized spatial filters. *IEEE Transactions Pattern Analysis Machine Intelligence*, 12(1):55–73, 1990. 92

G. Bradski and J. Davis. Motion segmentation and pose recognition with Motion History Gradients. *Machine Vision and Applications*, 13(3):74–184, 2002. 30

D. Brown, I. Craw, and J. Lewthwaite. A SOM based approach to skin detection with

application in real time systems. In *British Machine Vision Conference*, pages 491–500, 2001. 11

S. Brutzer, B. Höferlin, and G. Heidemann. Evaluation of background subtraction techniques for video surveillance. In *IEEE Computer Vision and Pattern Recognition*, pages 1937–1944, 2011. 13

R. Cardenes, M. Bach, Y. Chi, I. Marras, R. de Luis García, M. Anderson, P. Cashman, and M. Bultelle. Multimodal evaluation for medical image segmentation. In *International Conference on Computer Analysis of Images and Patterns (CAIP)*, volume 4673 of *LNCS*, pages 229–236, Vienna, Austria, 2008. Springer. 53

L. Caro-Campos, J. Carlos San M. Avedillo, and J. M. Martinez. Discrimination of abandoned and stolen object based on active contours. In *IEEE Conference on Advanced Video and Signal-Based Surveillance (AVSS)*, pages 101–106, 2011. 27

G. Cerutti, L. Tougne, A. Vacavant, and D. Coquin. A Parametric Active Polygon for Leaf Segmentation and Shape Estimation. In *International Symposium on Visual Computing*, 2011. 27

D. Chai and A. Bouzerdoum. A Bayesian approach to skin color classification in YCbCr color space. In *IEEE TENCONoo*, volume 2, pages 421–424, 2000. 11

Y. T. Chen, C.S. Chen, C. R. Huang, and Y. P. Hung. Efficient hierarchical method for background subtraction. *Pattern Recognition*, 40(10):2706–2715, 2007. 18, 23, 24

D. N. T. Cong, L. Khoudour, C. Achard, and P. Phothisane. People re-identification by means of a camera network using a graph-based approach. In *IAPR Conference on Machine Vision and Application*, volume 90, pages 152–155, 2009. 18

T. F. Cootes, C. J. Taylor, D. H. Cooper, and J. Graham. Active shape models - their training and application. *Computer Vision and Image Understanding*, 61(1):38–59, 1995. 28

Y. Dhome, N. Tronson, A. Vacavant, T. Chateau, C. Gabard, Y. Goyat, and D. Gruyer. A benchmark for background subtraction algorithms in monocular vision: A comparative study. In *International Conference Image Process. Theory, Tool and Applications*, pages 66–71, 2010. 22, 23

G. Doretto and S. Soatto. Dynamic shape and appearance models. *IEEE Transactions Pattern Analysis Machine Intelligence*, 28(12):2006–2019, 2006. 92

G. Doretto, A. Chiuso, Y. N. Wu, and S. Soatto. Dynamic textures. *International Journal Computer Vision*, 51(2):91–109, 2003. 24, 91, 92

R.O. Duda, P.E. Hart, and D.G. Stork. *Pattern Classification*. Wiley InterScience, New York, 2002. 12

A. Elgammal and L. Davis. Probabilistic framework for segmenting people under occlusion. In *IEEE International Conference on Computer Vision*, pages 145–152, 2001. 68

A. Elgammal, R. Duraiswami, D. Harwood, and L. S. Davis. Background and foreground modeling using nonparametric kernel density for visual surveillance. In *Proceedings of the IEEE*, volume 90, pages 1151–1163, 2002. 13, 18, 19

S. Y. Elhabian, K. M. El-Sayed, and S. H. Ahmed. Moving object detection in spatial domain using background removal techniques - state-of-art. *Recent Patents on Computer Science*, 1:32–54, 2008. 13, 43

V. Ferrari, F. Jurie, and C. Schmid. Accurate object detection with deformable shape models learnt from images. In *IEEE Computer Vision and Pattern Recognition*, pages 1–8, 2007. 27

V. Ferrari, F. Jurie, and C. Schmid. From images to shape models for object detection. *International Journal of Computer Vision*, 87(3):284–303, 2010. 27

J. M. Ferryman, A. D. Worrall, G. D. Sullivan, and K. D. Baker. A generic deformable model for vehicle recognition. In *British Conference on Machine Vision*, pages 127–136, 1995. 28

K.S. Fu and J.K. Mui. A survey on image segmentation. *Pattern Recognition*, 13:3–16, 1981. 43

H. García, A. Salazar, D. Alvarez, and Á. Orozco. Driving fatigue detection using active shape models. In *International Conference on Advances in Visual Computing*, ISVC'10, pages 171–180, 2010. 28

Á. García-Martin, A. Hauptmann, and J. M. Martinez. People Detection Based on Appearance and Motion Models. In *IEEE International Conference on Advanced Video and Signal-Based Surveillance*, pages 256–260, 2011. 27

Y. Goyat, T. Chateau, L. Malaterre, and L. Trassoudaine. Vehicle trajectories evaluation by static video sensors. In *IEEE International Conference on Intelligent Transportation Systems*, pages 864–869, 2006. 13, 22, 23, 24, 54, 57

R. Hassanpour, A. Shahbahrami, and S. Wong. Adaptive Gaussian mixture model for skin color segmentation. In *World Academy of Science Engineering and Technology*, volume 31, pages 1–6, 2008. 13

M. Heikkilä and M. Pietikäinen. A texture-based method for modeling the background and detecting moving objects. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 28(4):657–662, 2006. 25, 91

B. K. P. Horn and B. G. Schunck. Determining optical flow. *Artificial Intelligence*, 17: 185–203, 1981. 31, 32

T. Horprasert, D. Harwood, and L.S. Davis. A statistical approach for real-time robust background subtraction and shadow detection. In *IEEE International Conference on Computer Vision*, 1999. 14

T. Hosaka, T. Kobayashi, and N. Otsu. Object detection using background subtraction and foreground motion estimation. *IPSJ Transactions on Computer Vision and Applications*, 3:9–20, 2011. 69

R. Hsu, M. Abdel-mottaleb, and A. K. Jain. Face detection in color images. *IEEE Transactions Pattern Analysis Machine Intelligence*, 24(5):696–706, 2002. 11

T. Huang, J. Qiu, T. Sakayori, S. Goto, and T. Ikenaga. Motion detection based on background modeling and performance analysis for outdoor surveillance. In *International Conference on Computer Modeling and Simulation*, pages 38–42, 2009. 18

A. Iketani, A. Nagai, Y. Kuno, and Y. Shirai. Detecting persons on changing background. In *International Conference on Pattern Recognition*, volume 1, pages 74–77, 1998. 32

M. Izadi and P. Saeedi. Robust region-based background subtraction and shadow removing using color and gradient information. In *International Conference on Pattern Recognition*, pages 1–5, 2008. 18

S. Jabri, Z. Duric, H. Wechsler, and A. Rosenfeld. Detection and location of people in video images using adaptive fusion of color and edge information. In *International Conference Pattern Recognition*, volume 4, pages 627–630, 2000. 14

N. Jacobs and R. Pless. Shape background modeling : The shape of things that came. In *IEEE Workshop on Motion and Video Computing (WMVC)*, 2007. 27

A.K. Jain and K. Karu. Learning texture discrimination masks. *IEEE Transactions Pattern Analysis Machine Intelligence*, 18(2):195–205, 1996. 24

R. Jain, R. Kasturi, and B.G. Schunck. *Machine vision*. McGraw-Hill, New York, NY, 1995. 41

C. Jang and K. Jung. Human pose estimation using active shape models. *World Academy of Science, Engineering and Technology*, 44(1):312–316, 2008. 28

O. Javed, K. Shafique, and M. Shah. A hierarchical approach to robust background subtraction using color and gradient information. In *Workshop on Motion and Video Computing*, pages 22–27, 2002. 18

M.J. Jones and J.M. Rehg. Statistical color models with application to skin detection. *International Journal Computer Vision*, 46(1):81–96, 2002. 12

P. Juszczak and R. P. W. Duin. Uncertainty sampling methods for one-class classifiers. In *In Proceedings of the ICMLŠ03 Workshop on Learning from Imbalanced Data Sets*, volume 6, page 5, 2003. 16

P. KaewTraKulPong and R. Bowden. An improved adaptive background mixture model for real-time tracking with shadow detection. In *European Workshop on Advanced Video Based Surveillance Systems*, 2001. 17, 23, 24

P. Kakumanu, S. Makrogiannis, and N. Bourbakis. A survey of skin-color modeling and detection methods. *Pattern Recognition*, 40(3):1106–1122, 2007. 11

Y. Kameda and M. Minoh. A human motion estimation method using 3-successive video frames. In *International Conference on Virtual Systems*, pages 135–140, 1996. 30

D. Kim, D. Kim, and J. Paik. Gait recognition using active shape model and motion prediction. *IET Computer Vision*, 4(1):25–36, 2010. 28

K. Kim, T. Thanarat, H. Chalidabbhognse, D. Harwood, and L. Davis. Real time foreground-background segmentation using codebook model. *Real-Time Imaging*, 11 (3):172–185, 2005. 13, 20, 21, 54, 57

T. Ko, S. Soatto, and D. Estrin. Background subtraction on distributions. In *European Conference on Computer Vision*, pages 276–289, 2008. 19

Y. Kuno, T. Watanabe, Y. Shimosakoda, and S. Nakagawa. Automated detection of human for visual surveillance system. In *IEEE International Conference on Pattern Recognition*, pages 865–869, 1996. 28

J-L. Landabaso. *A unified framework for consistent 2D/3D foreground object detection*. PhD thesis, Image Processing Department, Technical University of Catalunya, 2008. 14

D.S. Lee. Effective gaussian mixture learning for video background subtraction. *IEEE Transactions Pattern Analysis Machine Intelligence*, 27(5):827–832, 2005. 17

B. Leibe, A. Leonardis, and B. Schiele. Combined object categorization and segmentation with an implicit shape model. In *ECCV workshop on statistical learning in computer vision*, pages 17–32, 2004. 26

B. Leibe, E. Seemann, and B. Schiele. Pedestrian detection in crowded scenes. In *IEEE Computer Vision and Pattern Recognition*, pages 878–885, 2005. 26, 27

L. LI, J. GONG, and W. CHEN. Gray-level image thresholding based on Fisher linear projection of two dimensional histogram. *Pattern Recognition*, 30(5):743–749, 1997. 41

L. Li, W. M. Huang, I.Y. H. Gu, and Q. Tian. Statistical modeling of complex background for foreground object detection. *IEEE Transactions on Image Processing*, 13(11):1459–1472, 2004. 46

R. Li, S. Yu, and X. Yang. Efficient spatio-temporal segmentation for extracting moving objects in video sequences. *IEEE Transactions on Consumer Electronics*, 53(3):1161–1167, 2007. 31

W. Li, X. Wu, K. Matsumoto, and H. Zhao. Foreground detection based on optical flow and background subtract. In *IEEE Conference on Communications, Circuits and Systems*, pages 359–362, 2010. 33

B. D. Lucas and T. Kanade. An iterative image registration technique with an application to stereo vision. In *International Joint Conference on Artificial Intelligence*, pages 674–679, 1981. 32, 40, 41

D. R. Magee. Tracking multiple vehicles using foreground, background and motion models. *Image and Vision Computing*, 22(2):143–155, 2004. 29

B. S. Manjunath and W. Y. Ma. Texture features for browsing and retrieval of image data. *IEEE Transactions Pattern Analysis and Machine Intelligence*, 18(8):837–842, 1996. 92

A. Mittal and N. Paragios. Motion-based background subtraction using adaptive kernel density estimation. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 302–309, 2004. 19, 33

A. Monnet, A. Mittal, N. Paragios, and R. Visvanathan. Background modeling and subtraction of dynamic scenes. In *IEEE International Conference on Computer Vision*, volume 2, pages 1305–1312, 2003. 24

W. Nam and J. Han. Motion-based background modeling for foreground segmentation. In *ACM international workshop on Video surveillance and sensor networks*, pages 35–44. ACM, 2006. 29

T. Ojala, M. Pietikäinen, and D. Harwood. A comparative study of texture measures with classification based on feature distributions. *Pattern Recognition*, 29(1):51–59, 1996. 25

T. Ojala, M. Pietikäinen, and T. Mäenpää. Multiresolution gray-scale and rotation invariant texture classification with local binary patterns. *IEEE Transactions Pattern Analysis Machine Intelligence*, 24(7):971–987, July 2002. 25

N. Otsu. A threshold selection method from gray-level histograms. *IEEE Trans Systems Man Cybernet*, 9:62–66, 1979. 41, 42

N.R. Pal and S.K. Pal. A review on image segmentation techniques. *Pattern Recognition*, 26(9):1277–1294, 1993. 41, 43

R. Peteri and D. Chetverikov. Dynamic texture recognition using normal flow and texture regularity. In *Iberian Conference on Pattern Recognition and Image Analysis*, pages 223–230, 2005. 32

R. Péteri, S.r Fazekas, and M. J. Huiskes. DynTex : a Comprehensive Database of Dynamic Textures. *Pattern Recognition Letters*, 31:1627–1632, 2010. 4, 100, 101, 107

S. L. Phung, A. Bouzerdoum, and D. Chai. Skin segmentation using color pixel classification: Analysis and comparison. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 27(1):148–154, 2005. 12

J. Puzicha, T. Hofmann, and J. M. Buhmann. Non-parametric similarity measures for unsupervised texture segmentation and image retrieval. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 267–272, 1997. 92

G. X. Ritter and J. N. Wilson. *Handbook of Computer Vision Algorithms in Image Algebra*. CRC Press, USA, 2nd edition, 2000. 9

A. Rosenfeld and A. Kak. *Digital picture processing*, volume 2. Academic Press, New York, NY,, 1982. 2nd Ed. 43

A. D. Sappa, N. Aifanti, S. Malassiotis, and M. G. Strintzis. Prior knowledge based motion model representation. *Electronic Letters on Computer Vision and Image Analysis*, 5(3):55–67, 2005. 29

N. Sebe, T. Cohen, T.S. Huang, and T. Gevers. Skin detection, a Bayesian network approach. In *International Conference on Pattern Recognition*, pages 903–906, 2004. 11

N. T. Siebel and S. J. Maybank. Fusion of multiple tracking algorithms for robust people tracking. In *European Conference on Computer Vision*, ECCV '02, pages 373–387, 2002. 28

M. H. Sigari and M. Fathy. Real-time background modeling/subtraction using two-layer codebook model. In *International MultiConference of Engineers and Computer Scientists*, 2008. 21, 23, 24

Y. Song, L. Goncalves, and P. Perona. Unsupervised learning of human motion. *IEEE Transactions Pattern Analysis and Machine Intelligence*, 25(7):814–827, 2003. 29

C. Stauffer and W. Grimson. Learning patterns of activity using real-time tracking. *IEEE Transactions Pattern Analysis Machine Intelligence*, 22(8):747–757, 2000. 13, 14, 15, 16, 17, 18, 23, 24, 54, 57, 68, 74, 75, 76, 78, 91, 100, 103, 104, 106, 110, 111, 112

M. Störring, T. Koéka, H.J. Anderson, and E. Granum. Tracking regions of human skin through illumination changes. *Pattern Recognition Letters*, 24(11), 2003. 11

M. Szummer and W. P. Rosalind. Temporal texture modeling. In *IEEE International Conference on Image Processing*, pages 823–826, 1996. 91

D. M. J. Tax and R. P. W. Duin. Combining one-class classifiers. In *Multiple Classifier Systems*, volume 2096 of *Lecture Notes in Computer Science*, pages 299–308. Springer, 2001. 16

O. Tuzel, F. Porikli, and P. Meer. A Bayesian approach to background modeling. In *IEEE Computer Vision and Pattern Recognition - Workshops*, pages 58–, 2005. 23, 24

C. Veenman, M. Reinders, and E. Backer. Resolving motion correspondence for densely moving points. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 23(1): 54–72, 2001. 125

N. Verbeke and N. Vincent. A PCA-based technique to detect moving objects. In *Scandanvian Conference on Image Analysis*, pages 641–650, 2007. 31

V. Vezhnevets, V. Sazonov, and A. Andreeva. A survey on pixel-based skin color detection techniques. In *GRAPHICON*, pages 85–92, 2003. 11

P. A. Viola, M. J. Jones, and D. Snow. Detecting pedestrians using patterns of motion and appearance. In *International Conference on Computer Vision*, pages 734–741, 2003. 31

C. Wang, Y. Jeung, L. Luo, J. Wang, and J. Chong. Real-time face recognition using adaptive skin-color model. In *IEEE Conference on Information Science and Applications*, pages 1–6, 2011. 13

H. Wang, X.H. Wang, Y. Zhou, and J. Yang. Colour texture segmentation using quaternion-Gabor filters. In *IEEE International Conference on Image Processing*, pages 745–748, 2006. 92

Y. Wang and B. Yuan. A novel approach for human face detection from color images under complex background. *Pattern Recognition*, 34(10):1983–1992, 2001. 11

L. Wixson. Detecting salient motion by accumulating directionally-consistent flow. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(8):774–780, 2000. 32

C. Wolf and J-M. Jolion. Integrating a discrete motion model into GMM based background subtraction. In *IEEE International Conference on Pattern Recognition*, pages 9–12, 2010. 18, 33, 69

C. Wolf and J.M. Jolion. Extraction and recognition of artificial text in multimedia documents. *Pattern Analysis and Applications*, 6(4):309–326, 2003. 42

C. Wren, A. Azarbayejani, T. Darrell, and A. Pentland. Pfinder: realtime tracking of the human body. *IEEE Transactions Pattern Analysis Machine Intelligence*, 19(7):780–785, 1997. 14

J. Yang and A. Waibel. A real-time face tracker. In *IEEE Workshop on Applications of Computer Vision*, 1996. 12

M.H. Yang and N. Ahuja. Gaussian mixture model for human skin color and its application in image and video databases. In *Conference on Storage and Retrieval for Image and Video Databases*, volume 3656, pages 458–466, 1999. 11, 13

A. Yilmaz, O. Javed, and M. Shah. Object tracking: A survey. *ACM Computing Surveys*, 38(4), 2006. 124

T.W. Yoo and I.S. Oh. A fast algorithm for tracking human faces based on chromatic histograms. *Pattern Recognition Letters*, 20(10):967–978, 1999. 12

B.D. Zarit, J.B. Super, and F.K.H. Quek. Comparison of five color models in skin pixel classification. In *International Workshop on Recognition, Analysis, and Tracking of Faces and Gestures in Real-Time Systems*, 1999. 12

An Zhao. Robust histogram-based object tracking in image sequences. In *Digital Image Computing Techniques and Applications*, pages 45–52, 2008. 43

J. Zhong and S. Sclaroff. Segmenting foreground objects from a dynamic textured background via a robust Kalman filter. In *International Conference Computer Vision*, pages 44–50, 2003. 24, 92

F. Zhou, J. F. Feng, and Q. Y. Shi. Texture feature based on local Fourier transform. In *IEEE International Conference on Image Processing*, volume 2, pages 610–613, 2001. 92