Universidad Autónoma de Madrid
Escuela politécnica superior

Grado en Ingeniería de Tecnologías y Servicios de Telecomunicación

Trabajo Fin de Grado

GENERACIÓN AUTOMÁTICA DE CONJUNTOS DE EVALUACIÓN DE CAMUFLAJE

Autor: Sergio Alvarez Balanya
Tutor: Marcos Escudero Viñolo
Ponente: Jesús Bescós Cano

Julio 2018

GENERACIÓN AUTOMÁTICA DE CONJUNTOS DE EVALUACIÓN DE CAMUFLAJE

Autor: Sergio Alvarez Balanya
Tutor: Marcos Escudero Viñolo
Ponente: Jesús Bescós Cano

Video Processing and Understanding Lab
Departamento de Tecnología Electrónica y de las Comunicaciones
Escuela Politécnica Superior
Universidad Autónoma de Madrid
Julio 2018

Trabajo parcialmente financiado por el Ministerio de Economía y Competitividad del Gobierno de España bajo el proyecto TEC2014-53176-R (HAVideo) (2015-2017)

Abstract

Resumen

La sustracción de fondo se ha convertido en una etapa principal en muchos algoritmos de visión por ordenador, como consecuencia, en los últimos años se han publicado numerosos estudios presentando diferentes enfoques. Sin embargo, la sustracción de fondo se sigue considerando un problema no resuelto. Esto puede deberse en parte a que los diferentes métodos se desarrollan en contextos distintos, como por ejemplo la video vigilancia y la captura de movimiento. La reciente aparición de conjuntos de datos completos proporciona un marco general de evaluación para algoritmos de sustracción de fondo. Estos conjuntos de datos constan de secuencias variadas en las que se pueden encontrar los problemas típicos que se dan en sustracción de fondo. Con esto, se consiguen resultados representativos que dan información general sobre el rendimiento de un algoritmo. Sin embargo, esta medida puede no ser adecuada cuando se quiere obtener información acerca de lo robusto que es un algoritmo frente a un problema determinado. Este trabajo se centra en una de estas problemas, el camuflaje o alta similitud entre las muestras de fondo y de frente. En la literatura se encuentran pocos estudios que investiguen directamente el camuflaje y no existe ningún método comúnmente aceptado para superarlo.

En este trabajo, proponemos una solución novedosa para modelar el camuflaje basado en el teorema de Jung. Con base en esta solución, generamos probabilidades de camuflaje para cada píxel de frente en una secuencia usando anotaciones disponibles para discriminar entre frente y fondo.

La evaluación de la solución propuesta se realiza en términos de discrepancia, umbralizando las probabilidades de camuflaje para obtener una máscara binaria sobre la que aplicamos medidas de clasificación clásica. De este modo, podemos analizar más a fondo el efecto de las características seleccionadas por diferentes algoritmos de sustracción de fondo en el manejo del camuflaje. Además, la solución propuesta también permite la clasificación de un conjunto de secuencias en términos de camuflaje.

Los experimentos llevados a cabo en el popular conjunto de datos CDNET2014 sugieren que el uso de ciertas características alternativas al color—por ejemplo, el movimiento—es beneficioso en términos de robustez al camuflaje.

Palabras Clave

Teorema de Jung, sustracción de fondo, camuflaje

Abstract

Background subtraction has become a key step in several computer vision algorithms. There are plenty of studies proposing different and varied approaches. However, the problem of background subtraction is not yet fully addressed. One reason might be the fact that each method has been developed for different tasks, e.g. video surveillance or optical motion capture. The recent appearance of comprehensive datasets provides a common framework for evaluating background subtraction algorithms. These datasets present a balanced repertoire of sequences in which common challenges are present. This leads to extensive overall scores in which robustness against different challenges is considered, but not particularized to these challenges. A particularly barely studied challenge, and the focus of our work, is camouflage: the resemblance between background and foreground samples. The research community agrees that there isn't yet a commonly accepted approach to handle camouflage.

In this work, we propose a novel solution for modeling camouflage based on the Jung's theorem. Based on this solution, we generate camouflage likelihoods for every foreground pixel in a sequence using available ground-truth information to discriminate the background from the foreground.

The evaluation of the proposed solution is performed in discrepancy terms by thresholding the camouflage likelihoods to obtain a binary mask on which we apply classical classification metrics. Thereby, we are able to further analyze the effect of the features selected by different background subtraction algorithms in handling camouflage. Furthermore, the proposed solution also permits the ranking of a set of sequences in terms of camouflage.

The experiments carried out on the popular CDNET2014 dataset suggest that the use of certain alternative features to color—e,g, motion—is beneficial to robustly handle camouflage.

Key words

Jung's theorem, background subtraction, camouflage

Agradecimientos

En primer lugar quiero dar las gracias a mi tutor Marcos, por toda la ayuda que me ha prestado en el desarrollo de este trabajo y el esfuerzo que ha invertido en mi formación.

Tambien quiero dar las gracias a la gente del VPULab, que entre todos forman un entorno de trabajo muy agradable. A mis compañeros de clase, especialemente a Borja, Quike, Cecilia, Juancho, Los Sergios, Fonso, Jorge, Dani, Manu, Pablo, Diego, Jamil.

Y por último a mi familia, que siempre me han dado su apoyo. Sin ellos se que no hubiera podido dedicar tanto tiempo a este trabajo.

Índice general

List of figures

List of tables

1
Introduction

In this chapter we first introduce the motivation of this work and follow by enumerating partial objectives that may lead the accomplishment of our main goal. The chapter concludes by presenting the structure of this document.

1.1. Motivation

Background subtraction has become a key step in several computer vision algorithms, therefore it has been extensively studied. A plethora of works, e.g. [1], [2], and [3], have been published in the last years proposing different approaches. Likewise, extensive reviews and evaluations have been conducted, e.g. [4], and [5]. In their review, [4], authors point out that, despite the numerous studies, the problem of background subtraction is not yet fully addressed. Authors arise to the conclusion that this is due, along other reasons, to the lack of a common framework as each method is developed for different contexts and is focused on different challenges.

A major challenge in background subtraction is camouflage. It happens when spatially correlated foreground and background share similar appearances. Throughout this work we have studied color camouflage, but it is worth remarking that camouflage can appear in spite of the features used. In spite of all the research carried out in background subtraction, camouflage remains barely studied as there is little work published on this matter. Few researches, [6], [7], have addressed this problem but there isn't yet a commonly accepted approach to handle it.

With this in mind, we present a novel evaluation system that conveys a quantitative measure of the performance of an algorithm against camouflage. Our system generates camouflage likelihoods at pixel level from existing ground-truth annotations providing benchmark annotations to enhance and particularize the quantitative evaluation of mechanisms designed to handle camouflage. As it generates these masks automatically from available annotations, it does not require additional labeling, which ensures easy integration to existing and future datasets. This work addresses the problem of providing a general framework to evaluate the performance of background subtraction techniques against camouflage. Moreover it presents a novel method to model camouflage, contributing to its study.

1.2. Objectives

In order to obtain camouflage likelihoods, we first need to model camouflage in a quantitative way—i.e. to measure camouflage in terms of real numbers. Therefore, part of this work is focused on the design of a quantitative measure of camouflage. We believe that we have found an innovative solution based on the Jung's theorem [8] that is able to characterize camouflage, as we show in Chapter 4.

The aim of this work is to provide a system to automatically generate camouflage datasets. We propose to use widely used datasets exploiting their ground-truth annotations to build camouflage likelihood maps. Then, we use these annotations to evaluate the performance of background subtraction algorithms against camouflage. In this work we obtain camouflage likelihoods for the CDNET2014 dataset and rank several state-of-the-art algorithms According to the performance of these algorithms, we finish by presenting a selection of the most challenging sequences in terms of camouflage.

These objectives can be arranged as follows:

- To design a quantitative measure of camouflage assuming background-foreground segmentation is available.
- To obtain camouflage likelihoods for the widely used CDNET2014 dataset.
- To rank different state-of-the-art algorithms using these likelihoods.
- To rank top challenging sequences in the CDNET2014 attending to the average score obtained by all the analyzed algorithms.

1.3. Document Organization

The rest of this document is organized as follows: The second chapter provides a brief overview of the background subtraction task and reviews exiting literature on camouflage. The design of the proposed camouflage scoring method is described in the third chapter, followed by the experiments description in Chapter 4. Our conclusions and future work are drawn in the fifth chapter.

2

Related Work

2.1. Background subtraction

In the scope of computer vision and automatic video analysis, background subtraction is the action of discriminating the foreground from the background. It is worth noting that this categorization is not fixed and depends on the final application. Background subtraction is a key step in many computer vision tasks including intelligent visual surveillance and human-machine interaction. The performance of background subtraction algorithms is key for the good performance of these algorithms; hence, the design of robust background subtraction algorithms is of high relevance.

2.1.1. Overview

In [4] authors review different approaches in background modeling for foreground detection, including a detailed description of a generic background subtraction process. Following the scheme there provided, we can define three main steps in a background subtraction process: background initialization, background maintenance, and foreground detection. Additional key tasks are needed in order to design a background subtraction algorithm. These include, definition of a background model, selection of the analysis unit, and choice of the image features (color, motion, etc..).

Background modeling

There are different strategies to model the background, the choice of one over another depends on the particular scenario and task, as the modeling is subjected to the statistical behavior of the background. The model mainly determines the ability to deal with uni-modal and multi-modal backgrounds. Background models can be classified into parametric or non-parametric. Parametric models approximate the background representation to a parametric function that is estimated during the background initialization and background maintenance stages. Non-parametric models store past background samples and define the background representation with a set of rules on this stored set.

Background initialization

Background initialization refers to the selection of a background model. A common approach is to use the first frame of a video. This is done under the assumption that at the beginning of a sequence there are no foreground objects. However, a more realistic approach is to use a set of training frames which may potentially include foreground objects. This turns into a big challenge when more than half of the training frames contain foreground objects. The selection of the initialization algorithm depends on the number of background modes—in dynamic backgrounds—and of the complexity of the background model. In [4] authors distinguish between three types of initialization algorithms: batch, incremental and progressive algorithms. Batch algorithms use N frames (consecutive or not) to compute the first background model in a *one-shot* way. Incremental ones, in which N is known, and progressive algorithms, with unknown N, that generate partial backgrounds and continue until a complete background is obtained.

Background maintenance

As the background of a sequence may change with time, it is necessary to use a mechanism to ensure that the model is adapted to the background changes. This process of adapting the background representation is called background maintenance. It can be done at each frame, but it is commonly accepted that updating the model just when necessary is a better approach. We can distinguish between three types of schemes regarding how the foreground detection impact the background maintenance: blind, selective and fuzzy. A blind algorithm does not rely on the foreground detection stage. Although this keep the maintenance scheme simple, foreground objects may pollute the background image spoiling the process. Selective models solve this problem by applying different update rules based on the foreground detection. This system relies on the stability of the foreground detector, but erroneous classifications tend to mess up the background. This problem is addressed by taking into account the uncertainty in the classification. These are the fuzzy schemes, in which the update rule is tuned using both the result of the foreground detection and its uncertainty or score.

Foreground detection

Foreground detection is a classification task in which pixels are labeled as foreground or background. The decision is made comparing the current image with the background model. The labeling is done at pixel level, but the picture's element used through this stage doesn't need to be the pixel. Decisions can be made at pixel level, at block level, or at cluster level. Block level and cluster level methods are more robust against noise but they achieve worse precision. As stated in [4], there are five features commonly used in the literature. Color, edge, stereo, motion, and texture features. These features have distinctive properties that can be used to handle challenging situations. Color, for instance, offers great performance discriminating the foreground when no notable challenge is present, but in presence of camouflage or shadows it tends to fail. In this case it is better to rely on alternative features, such as motion. In the next subsection we provide a description of the different challenges that commonly appear in background subtraction algorithms.

2.1.2. Challenges

According to [4] there are 13 challenging situations where background subtraction algorithms are likely to fail. These can be grouped in: camera related challenges, background related challenges, and foreground related challenges, following the association used by [9].

Camera related challenges

- Noisy image. Embraces all types of noise that can be included during the video capturing path: from the acquisition process, through the compression, to the final composition of each frame.
- Camera jitter. Static cameras can swing due to wind, or other external factor, when placed in non-stable supports, leading to nominal motion. If uncompensated, this motion can induce false foreground detections.
- Camera automatic adjustments. Some cameras carry out automatic processes to adapt to scene changes. These adjustments may have a great impact in the modeled background with respect to the one used in previous frames.

Background related challenges

- Illumination changes. They can affect the whole scene (global) or localized areas (local). Global changes can be divided into gradual changes such as daylight in outdoor scenes, and abrupt changes such as switching the light on in indoor scenarios.
- Removed background objects. Inanimate background objects can be moved at a certain time uncovering part of the background. This sudden change can lead into false foreground detection. These blobs due to erroneous detections are also known as *ghosts*.
- Inserted background objects. Inanimate background objects can be introduced in the scene at some point in the sequence. These new objects should not be considered foreground. It is the opposite situation to the previous one, being both very common in video surveillance scenarios.
- Dynamic backgrounds. Some parts of the background can move, as a consequence, the pixels representing the background may be images of more than one background object. This is handled with multi-modal background models. Typical situations of dynamic backgrounds are waving trees and moving water.

Foreground related challenges

- Bootstrapping. When a certain part of the background is occluded for a long time, it may be impossible to model its evolution, or even its appearance, as a result of not having enough samples.
- Shadows. Foreground or moving shadows, also known as cast-shadows, can be detected as foreground. They move as a foreground object, and are represented by similar, but lower-intense, modes than those in the background model. Shadow detection is a research field by itself.
- Beginning moving object. It presents a very similar challenge to removed background objects. The main difference relies on the object nature, in this case an animated object begins its movement. It leads to the same problems as removed background objects.

- Sleeping foreground object. As in the previous case, this is the counterpart of inserted background objects, but with animated objects such as people. The decision of incorporating these objects to the background is task dependent. When the foreground object is there since initialization, and no management of this situations is performed, the challenge is also known as $hot - start$.

- Camouflage. It occurs when the foreground and the background share similar appearances, this may lead to inaccurate foreground segmentation. It is one of the least studied challenges and the focus of this work.

- Foreground aperture. When a foreground object is included in the background model and it contains uniform colored regions, changes may not be fully detected when it moves. Changes are only detected on boundaries and not on uniform regions. Therefore, the foreground mask may present holes in the homogeneous regions. The foreground aperture cause may be explained by the sequence: removed background, beginning moving object, or camouflage.

Several studies, argue that, regardless of the fact that background subtraction has been widely studied, there is still no system capable of dealing with all these challenges at the same time. Different approaches have been proposed to deal with one or some of these challenges but none is able to handle all of them by proposing a trade-off solution between having a flexible background model and an accurate foreground detector.

Moreover, these challenges need to be managed at different stages of the background subtraction process. For example, illumination changes need to be addressed in the modeling and the maintenance of the background, but bootstrapping requires specific solutions in the initialization step. In our opinion camouflage represents a barely addressed challenge that is usually ignored or managed in post-processing stages.

2.1.3. Evaluation

Several studies, e.g. [10], [11], and [5], have proposed different evaluation techniques for background subtraction. There are subjective measures, where performance is evaluated by human consideration, and objective measures. Following the taxonomy proposed in [10], the last ones can be cataloged into empirical and analytical. Analytical measures consider the theoretical description, requirements, and complexity of an algorithm, whereas empirical measures rely on the algorithm results and video properties. Among empirical methods we can find stand-alone evaluation techniques, that provide an automatic evaluation process, and discrepancy measures, where the results of the algorithm are compared against ground-truth masks. These masks are usually hand labeled, therefore, traditionally, there were only available a few small datasets. In [4] it is stated that this issue has been addressed in the last years with the appearance of new datasets, like the CDNET2014 dataset [12], that provide sequences covering all types of challenges in background subtraction with accurate human annotated ground-truth.

When comparing with ground-truth, background subtraction can be considered as a classification problem where each pixel is labeled either as a foreground or as a background one. Consequently, the performance of an algorithm can be measured in terms of true positives (TP), false positives (FP), true negatives (TN), and false negatives (FN). Combinations of these metrics, such as precision or recall, are used to give an overall score of the algorithm's performance.

We aim to take advantage of the ground-truth masks to develop a quantitative metric of the performance of a given algorithm against camouflage. Typically, this is addressed applying classical discrepancy measures over sequences that contain camouflage. This categorization is done subjectively and is categorical, so it does not provide any information of how much camouflage is in the sequence. This may lead to biased evaluations.

2.2. Existing features, metrics and methods to handle camouflage

Few studies have been published regarding the modeling and handling of camouflage. Nevertheless some approaches have been proposed in this matter. There are some studies that propose to use a *camouflage dataset* to evaluate the performance of an algorithm against camouflage. In [6], authors provide a new dataset including sequences expressly recorded to contain camouflage. Likewise, in [5] authors evaluate several background techniques against challenging situations, including camouflage. These *camouflage sequences* are temporally cropped from longer videos selecting fragments where camouflage is present.

In both studies camouflage is annotated at sequence level. One of the major drawbacks of this approach is that it relies on human intervention to decide whether a sequence contains camouflage. In our opinion, the main limitation is that there is still considerable ambiguity with regard to the evaluation against camouflage. Evaluation is performed on a traditional way using these sequences; hence, it is impossible to determine how the algorithm robustness against camouflage affects its performance. This is due to the fact that no sequence contains only camouflaged pixels. Our annotation method overcomes these limitations as it does not rely on human intervention because annotations are automatically generated, and it provides an objective evaluation system that reports results based only on camouflage samples.

An alternative approach is proposed in [7] where camouflage is modeled as a quantitative feature. Authors generate a predicted camouflage mask that combined with a preliminary foreground detection provides a better segmentation. Authors suggest that a camouflaged pixel will generate similar likelihood values for the foreground classification and the background classification, and that this value will be very much alike the obtained for neighboring camouflaged pixels. This is an interesting study based on heuristic considerations.

In the literature there are also some studies proposing methods to handle camouflage. These exploit non-color features as camouflage is a feature-related challenge. For example, the use of a wavelet domain is proposed in [6]. Results show that this method is able to outperform alternative methods for most of the sequences. The use of the Hue histogram has also been proposed [3]. Authors claim that the presence of camouflage can be detected in the color distribution of the image. Although not expressly targeting camouflage, other algorithms that exploit alternative features, e.g. motion in [13], show a robust performance against camouflage. A discussion of the effectiveness of some of these methods is given in next section.

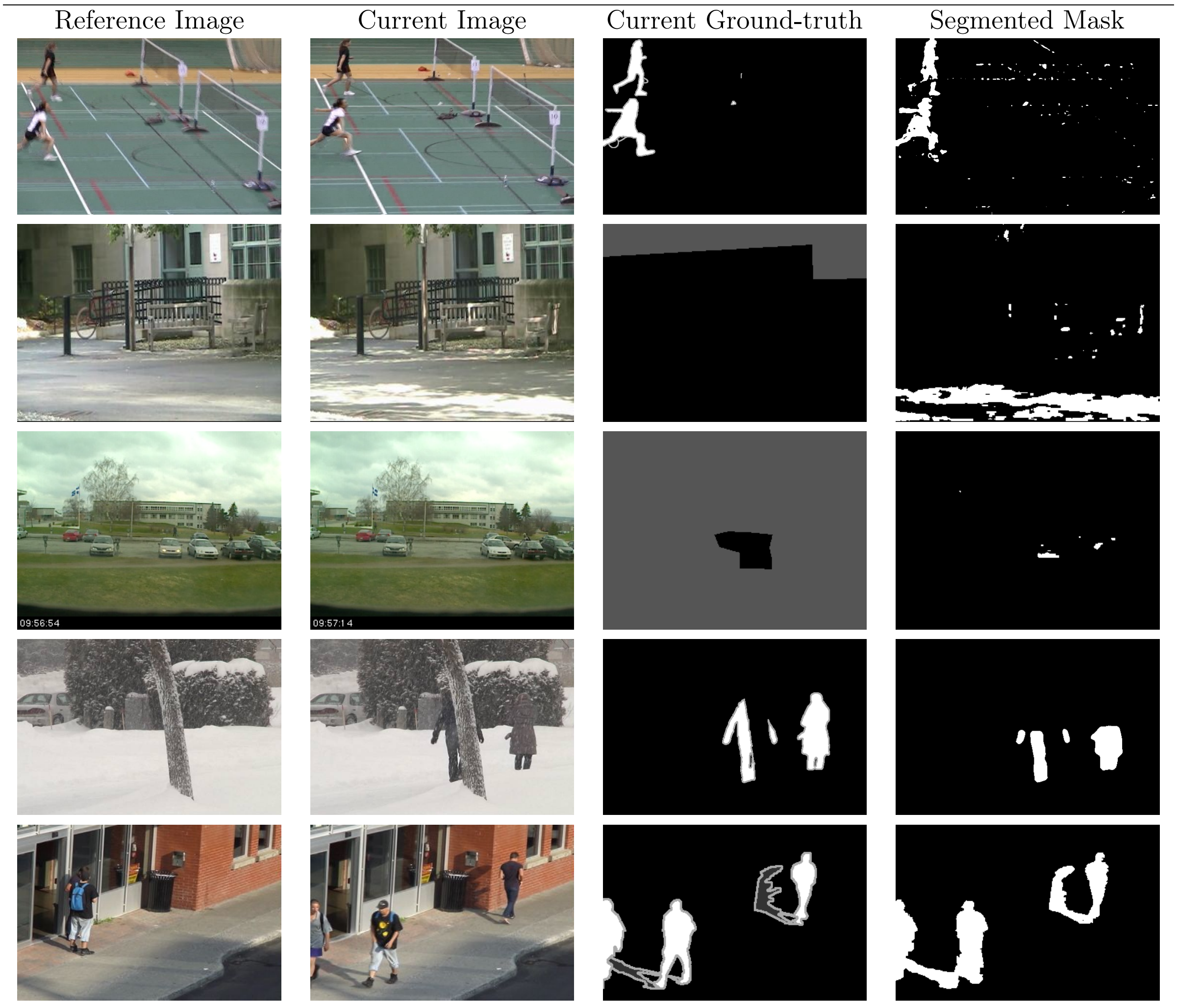

Figure 2.1: Examples of background subtraction challenges. The reference image shows a previous image in the sequence that may be, either the previous frame, or an earlier frame. The first, second, and fifth rows show false detections due to camera jitter, illumination changes, and shadows respectively. In the third row we can see the appearance of *ghosts*. The fourth row presents an example of camouflage where the foreground is not detected.

2.3. Ranked Algorithms

In this section we provide a brief description of the algorithms ranked in Chapter 4. We have evaluated 13 algorithms which are divided in four categories based on their performance in the CDNET2014, additional details on this organization are given in subsection 4.1.3. Moreover, Table 4.1 presents a concise summary of the evaluated algorithms.

GMM [14] is a method presented 20 years ago, but it remains a popular algorithm today. Many modern techniques use some modifications of this algorithm in some part of their scheme. It extents the idea of using an adaptive gaussian to deal with multi-modal backgrounds. Due to its high importance in the field of background subtraction, this model presents a reference when ranking different algorithms.

KDE [15] methods are the counterpart of GMM in non-parametric models. They appeared as an alternative to GMM. Mainly motivated by the fact that the parameter estimation in parametric models is prone to spoil the foreground detection when estimation errors occur. These errors are avoided by comparing new pixels with a non-parametric kernel built from previous background samples. Along with GMM, KDE is a reference baseline when ranking background subtraction algorithms.

IUTIS_5 [16] is the top performing algorithm in terms of average precision in the CDNET2014 challenge. This method emerges from the combination of results from five algorithms: SuBSENSE[17], FTSG[13], CwisarDH[18], AMBER[19], and Spectral-360[20]. All of these except the last one, Spectral-360, are also individually evaluated in this work. In [16], authors propose to rely on a combination of masks using genetic programming to overcome the limitations of current state-of-the-art algorithms. The method is based on the combination of foreground masks obtained by each of these algorithms and on a set of operators to perform post-processing. Authors select the top-9 performing algorithms in the CDNET2014 dataset and trained different setups on a subset of videos of the CDNET2014 dataset. This subset is assembled by one video of each category in the dataset (see subsection 4.1.1). They point out that the use of 5 algorithms results in a better trade-off as it produces the same F-score than more populated combinations while reducing the extra complexity.

PAWCS [21] builds a pixel-level model using persistence-based characterizations to adequately model the background for long periods of time. In [21], authors claim that the method is highly adaptive to different challenging scenarios without the need of manual readjustment. Occurrence of pixels is registered in the background model using a counter. The algorithm uses a combination of color and Local Binary Similarity Patterns (LBSP) [22] features to characterize each pixel. Moreover, global dictionaries are used to represent 2D occurrence maps which are applied, along segmentation results, to dynamically tune the updating rules and the foreground detection mechanism.

CwisarDH [18] makes use of a weightless neural network to model the background. It is an extension of the WiSARD [23] architecture expressly tuned to achieve state-of-the-art performance in the CDNET2014 dataset. The main improvement is the introduction of a pixel classification history support, i.e. a buffer for each pixel storing its foreground classifications. Whenever this buffer is filled, the network is retrained using the stored values, and the buffer is reinitialized. Otherwise, if the pixel is classified as background, the buffer is rebooted but no retraining is performed.

FTSG [13] combines two foreground detection mechanisms, one using *flux tensors* (temporal variations of the optical-flow)[24] and another using split gaussians. Besides, another split gaussian method is used to model the background. Split gaussian models use a different and variable number of gaussians depending on the scene changes. Authors claim that the fusion foreground detector is robust enough to ensure good performance even by using a selective maintenance scheme (see subsection 2.1.1). The algorithm includes a module, based on edge detection, to discriminate between stopped moving objects and uncovered background.

SuBSENSE [17] presents a pixel-level method that relies on spatio-temporal binary features, LBSP [22], which combined with color provides robust foreground detection. This method defines a non-parametric background model. Foreground detection is carried out by computing the distance of a new sample to its background model and by thresholding the result to classify the new sample. The update rule is dynamically tuned using pixel-level feedback loops that analyze the background blinking as well as the dynamism of the scene to optimize the updating rules. Authors point out that the combination of features also contributes to better detection of camouflaged objects.

EFIC [1] presents an edge based foreground segmentation with inner region classification. Foreground edges are extracted using Local Ternary Patterns (LTP) features [25]. To fill the foreground segmentations, the rest of the pixels are classified into interior or exterior pixels thresholding distances in a graph formed by the foreground edges. The resulting masks are enhanced using a variant of the watershed algorithm. Two camera motion compensation modules are also included in this algorithm, one for constant motion away from the original position, i.e. PTZ, and another one for motion around the same position, i.e. jitter. Moreover it describes a ghost removal module based on the Chamfer distance [26] between the contour of the apparent foreground object and the edge image computed at the beginning. The ghost removal module only works on static regions of the image previously detected by using optical-flow.

AMBER [19] presents a pixel-level background subtraction method that prioritizes low computational cost maintaining state-of-the-art performance. It uses a small non-parametric background model. The main innovation is the introduction of a method that guaranties robust foreground detection even with few background samples. An on-line counter of background samples occurrence is used as an efficacy measure. If the efficacy of a sample decreases, the sample is replaced with a new value; the model keeps track of potential background samples. To increase robustness against high frequency noise in the scene, the classification is also computed by down-sampling the frame and the background model. Pixels that are classified as foreground in the full resolution process but are reclassified as background in the down-sampled frame, are classified as noisy-pixels. An additional variable keeps track of the blinking activity of these pixels. This variable is used to discard intermittent foreground pixels and to tune a threshold which is used for foreground detection.

RMoG [27] extends the classic mixture of gaussians method (GMM) to model regions instead of individual pixels. To this end, the updating rule is based on the adaptation of the Expectation Maximization (EM) [28] algorithm used to find the maximum likelihood function given a set of data. This set is conformed by the membership of any data sample to all the possible clusters. Consequently, a sample can be modeled as a mixture distribution which is a combination of color and space. The EM algorithm is applied on-line by making use of the gradient descent technique.

AAPSA [3] is a background subtraction algorithm designed to achieve state-of-the-art performance in different scenarios without parameter tuning. The method analyzes the scene and apply different techniques based on the dynamism of the scenario. It models the background with two self organized maps, one for static scenarios and another for dynamic backgrounds. The background model is initialized with a batch method that uses 40 frames. The initialization is done at the beginning and whenever a big change is detected. The foreground detector presents a complex scheme; first the Hue histogram of the frame is analyzed in order to detect camouflage, if camouflage is detected the updating rules are adapted in consequence. Then, regarding the estimated dynamism of the scene, the algorithm can apply one of four different modules, each one with different updating rules. When a region is continuously classified as foreground, it is considered a suspicious foreground region. If these regions are detected, interest points are extracted using SURF [29] over these regions. The quantity of SURF points within a region is used as an indicator of a foreground region: the more points, the higher the likelihood of it being a foreground.

GraphCutDiff [2] presents a change detection algorithm based on a mixture of gaussians combined with an optical-flow foreground detector. First, an initial background representation is obtained as the fusion of both foreground detectors. The rule used to define the fusion is variable depending on the noise level of the image. Then, the method refines the initial background representation with an energy minimization technique based on graph cuts which is applied to the absolute difference between the constructed background and the original image.

SC-SOBS [30] is an improved version the SOBS algorithm presented in the same paper. Both algorithms rely on a neural background model built by learning image sequence variations in a self-organizing way. Each pixel is modeled using a matrix of weight vectors organized in neural maps. The obtained neural maps of all the pixels are stacked together into a 2D neural map of the whole image. The SOBS method compares the value of a new pixel with its own neural map in order to decide whether the new pixel is foreground or background. The SC-SOBS algorithm enhances the foreground detection process by taking into account the number of detections in its neighboring maps.

3
Design

3.1. Preliminaries

In order to evaluate the performance of an algorithm in camouflage situations, we propose a discrepancy measure. In this section we explain how we automatically generate camouflage likelihoods exploiting ground-truth annotations.

Our system generates on-line annotations for each frame of a video. A frame and its ground-truth feed the model and a camouflage likelihood map is computed. For each pixel we keep N background samples as a simple non-parametric model. Whenever a pixel is labeled as background we add its value to the background model. On the other hand, whenever a pixel is labeled as foreground we estimate a camouflage likelihood. Figure 3.1 overviews the system process.

We understand that, when the value of a pixel in a new frame is similar to its background, the pixel is prone to be camouflaged. If the value of the pixel differs from the background, then it is unlikely camouflaged. Given this assumption, we propose to use the Jung's theorem [8] to measure the difference between the value of a foreground pixel and its background model.

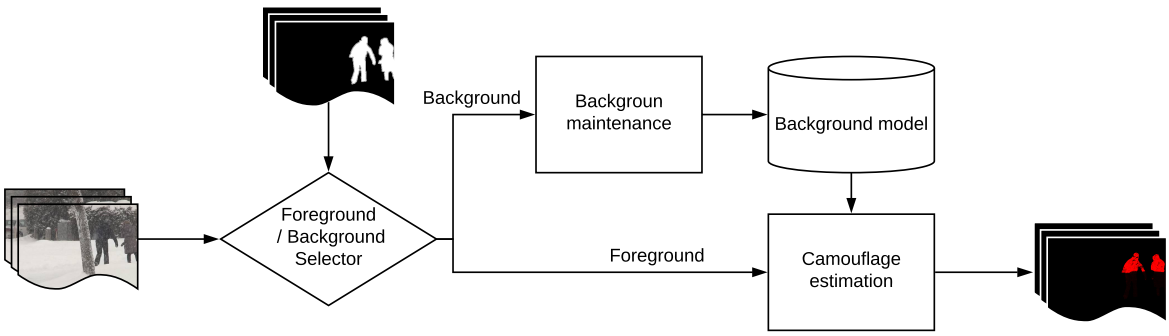

Figure 3.1: Camouflage likelihood estimator process.

3.2. Background model

3.2.1. Structure

We propose a simple non-parametric background model which takes advantage of ground-truth annotations. As the background model operates at pixel-level and it performs independently for every pixel, throughout this section we describe the background model for a given pixel p. This behavior can be extended to every other pixel.

The model keeps a maximum of N background samples, we call sample (s) to the characterization of a pixel p at a certain frame. A pixel is characterized by its RGB color vector; hence, the background model can be defined as a set $B \subset \mathbb{R}^3$ with $|B| \leq N$. The set is initialized empty and it is filled following the updating rule described in the next subsection.

3.2.2. Updating

Let $c(.)$ be a function that maps a pixel into one of three possible classes: *foreground*, *background*, and *others*; $c : p \rightarrow \{foreground, background, others\}$ The mapping function c is defined for every frame by its corresponding ground-truth annotation (see subsection 4.1.1). As stated before, the model is initialized empty, and, sequentially for each frame t, the pixel is mapped by $c(p)$. Whenever the pixel is classified as *background* the characterization of the pixel is added to B. If B already contains N samples, the oldest sample in the model is removed before the insertion of the new sample.

Whenever the pixel is categorized as *foreground*, we compute a camouflage likelihood as we describe in section 3.4. The rest of the pixels—i.e. pixels mapped as *others*—are ignored.

3.3. Jung's theorem

The Jung's theorem states that, given a compact set $K \subset \mathbb{R}^n$, there is a compact ball that contains K with radius l such that

$$l \leq d\sqrt{\frac{n}{2(n+1)}}, \tag{3.1}$$

where $d = \max_{\mathbf{a},\mathbf{b} \in K} \|\mathbf{a} - \mathbf{b}\|_2$,i.e. d is the maximum euclidean distance between all the samples contained in K.

Applying Jung's Theorem we obtain the minimum radius, hereinafter the Jung's Radius, that fit all the samples in a set K under the same sphere. This radius is the biggest possible value of l defined in 3.1.

We believe that the magnitude of the Jung's Radius is a good indicator of the typical fluctuations in the value of a background pixel. A high value of Jung's Radius entails high distance between background samples. We presume that the inclusion of a non-camouflaged foreground pixel characterization into its background model has a big impact in the Jung's Radius of this new set, which is prone to be increased with respect to that of the background set. This effect is less noticeable when the pixel is camouflaged as its value may be closer to that of the background set. Our aim is to take advantage of this behavior and use the Jung's Radius variation to provide a quantitative measure of camouflage.

3.4. Camouflage estimation

In an attempt to quantitatively describe camouflage we propose to use Jung's theorem. Specifically, our aim is to use Jung's Radius as an inner distance descriptor. We aim to measure how far the value of a foreground pixel falls from its background set. Let p be a foreground pixel, and let B be its background model. We can measure the increment of the Jung's Radius by computing the following ratio:

$$r_p = \frac{l_{B \cup s}}{l_B}, \tag{3.2}$$

where l_B is the Jung's Radius of B and $l_{B \cup s}$ is the Jung's Radius of the set built by the inclusion of the current characterization of p, s, into B.

We chose this ratio, henceforth named Jung's Ratio, because it provides a quantitative measure of how much a foreground sample differs from its background, taking into account the dispersiveness of its background model.

Another way to understand this process is that we are estimating the size of the ball C that contains all the possible values of the background for a given pixel p at frame t, and, given the new characterization of p, we aim to measure how much bigger this ball should be in order to contain also this new sample. A Jung's Ratio, $r_p > 1$ means that, either the estimation of C is inaccurate, or that the new sample does not come from a background pixel. As we have explained in section 3.2, our background model relies on labeled annotations, therefore we assume that our estimation of C is accurate. Furthermore, because of the annotations, we already know that this sample comes from a foreground pixel as we only compute this ratio for foreground pixels. This allows a formal solution to be found of how salient a foreground pixel is with respect to the background in terms of the features used, in this case color. Under the same argument, it is easy to verify that a Jung's ratio, $r_p = 1$ indicates that p is camouflaged, and as the value of r_p increases, the likelihood of p being camouflaged decreases. An example of this behavior is shown in Figure 3.2. In the figure there are two indicated frames which correspond to those represented at the top. See how the Jung's Ratio increases when a non-camouflaged foreground object, the mans face, appears but remains low when the object is camouflaged, the mans hair.

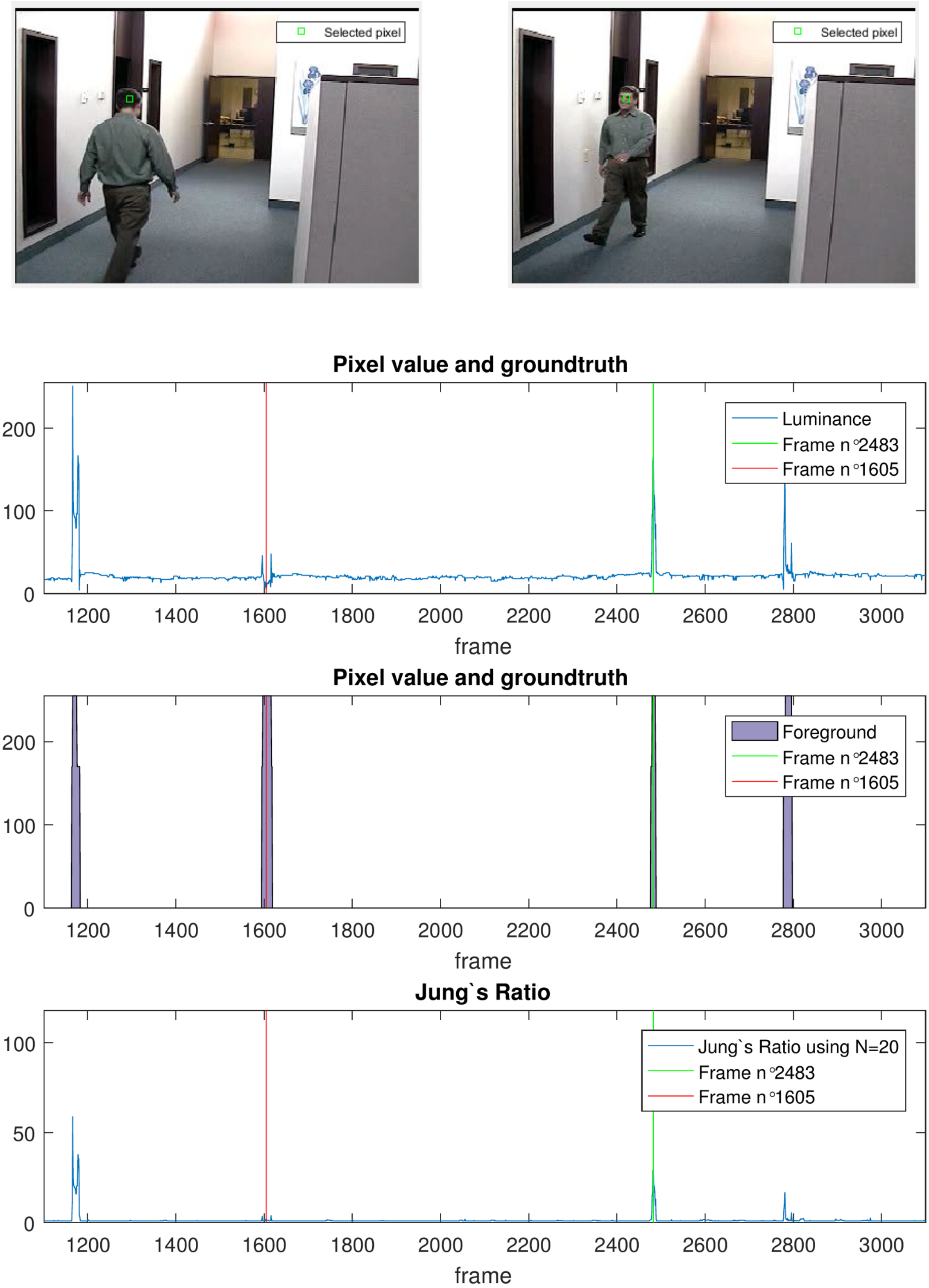

Figure 3.2: Jung's Ratio evolution in the "cubicle" sequence. The image at the top left corner represents the frame n○ 2483, the frame n○ 2788 is shown at the top right corner. The figure shows in each row from top to bottom: the luminance, the mask classification, and the Jung's Ratio evolution for the selected pixel shown in both frames.

4

Experiments

4.1. Experimental context

4.1.1. Dataset

We have generated camouflage likelihood maps for the CDNET2014 dataset [12]. We have chosen this dataset due to several reasons. First of all, it provides multiple sequences covering all types of challenges with accurate ground-truth annotations, which turns it into one of the most popular datasets for background subtraction. Besides, in the official web-page [1] authors provide comprehensive results of the different algorithms with multiple metrics, including: Precision, Recall, and False-Negative Rate. Furthermore, along with a detailed ranking, it is possible to download the foreground masks generated by each one of the rated algorithms, which eases the task of evaluating some algorithms as one can make straight use of their results.

The change detection dataset arranges a total of 53 sequences into 11 different categories encompassing common challenging situations.

- Bad Weather: This category contains four videos. They all have in common the presence of blizzard or snow that produces low visibility. This low visibility often leads to the appearance of camouflage as foreground objects can be hardly segregated from the background.
- Low Frame Rate: This category includes four videos captured at a low frame rate. In this case, the presence of camouflage will depend on the particular sequence. The low frame rate may spoil the use of motion as an alternative feature.
- Night Videos: There are six videos in this category. All of them are recorded during night and show roadway scenarios. We can expect camouflage situations in this category as everything looks the same in the dark. Moreover, vehicles lights might occlude the already low intensity characterizations of pixels belonging to cars and other vehicles.
- PTZ: This category contains four videos recorded using a PTZ camera. Our background model is designed for static background, for this reason we have decided not to evaluate on PTZ sequences as the results for this category might spoil our analysis.

[1] `http://changedetection.net/`

- Thermal: This category contains five videos recorded using a thermal camera. Sequences recorded with this type of camera usually present camouflage as foreground objects that do not exhibit higher temperatures than the background are hardly distinguishable from it: they are camouflaged in terms of temperature.

- Shadow: There are six videos in this category. In these videos prevail hard and soft shadows and intermittent shades. This challenge lead to false foreground detection but the miss-classification of foreground objects is not usually a main issue in this category. As camouflage is evaluated for true foreground pixels we do not expect sequences in this category to present a big challenge regarding camouflage.

- Intermittent Object Motion: This category contains six videos presenting background objects moving away, abandoned objects, and objects stopping for a short while and then moving again.

- Camera Jitter: This category contains four videos with heavy camera jitter. The multi-modal background due to jitter may lead to a big Jung's Radius of the background model; hence, we expect to obtain non-zero likelihood for a lot of camouflage pixels.

- Dynamic Background: This category contains six videos with dynamic background motion such as water or trees swinging in the wind. As in the previous category, multi-modal backgrounds are expected to convey large Jung's Radius, leading to the same situation.

- Baseline: This category contains four videos with no notable issues. As no remarkable challenge is present in videos within this category we do not expect to find challenging camouflage situations.

- Turbulence: There are four videos in this category, they are recorded in air turbulence situations. Foreground objects are small as the viewpoint is far away from the recorded scene. Besides, the sequences are recorded in gray-scale. Challenging sequences are expected within this category as these conditions often lead to the appearance of camouflage.

The ground-truth annotations accompanying the videos provide a exhaustive pixel-level classification that distinguish between 5 labels according to the following values:

- 0: Static, represents the absolute background.

- 50: Hard shadow.

- 85: Outside region of interest.

- 170: Unknown motion; usually around moving objects, due to semi-transparency and motion blur.

- 255: Motion, represents the certain foreground.

Our background model is designed assuming static camera; hence we have discarded the PTZ category as we know in advance that our background model will not be able to accurately represent the background. Therefore, we have built camouflage likelihood maps for the 49 sequences conforming the other 10 categories.

Algorithm	Model	Features Used	Average Precision	Quality Category
IUTIS-5 [16]	Combination of algorithms using Genetic Programming	Color, Motion, LBSP	0.8087	HQ
PAWCS [21]	Codebook, local and global dictionaries	Color, LBSP	0.7857	
CwisarDH [18]	Weightless neural network	Color	0.7725	
FTSG [13]	Flux tensor and mixture of Gaussians	Color, Motion	0.7696	
SuBSENSE [17]	Non-parametric sample-based	Color, LBSP	0.7509	HMQ
EFIC [1]	Mixture of Gaussians	Color, Edges, Motion	0.7221	
AMBER [19]	Non-parametric sample-based	Color	0.7163	
RMoG [27]	Mixture of Gaussians	Color, Texture	0.6965	LMQ
AAPSA [3]	Two SOM networks	Color, SURF	0.6916	
GraphCutDiff [2]	Mixture of Gaussians and graph cut	Color, Motion	0.6666	
SC-SOBS [30]	Self-organized neural network	Color	0.6091	LQ
GMM [14]	Mixture of Gaussians	Color	0.6025	
KDE [15]	Non-parametric kernel	Color	0.5811	

Table 4.1: Description of ranked algorithms. Average Precision is obtained from the results reported in [12].

4.1.2. Performance measures

These algorithms are evaluated in the CDNET2014 dataset using discrepancy measures. As explained in subsection 2.1.3 performance is typically defined in terms of true positives, false negatives, true negatives, false positives, and by combinations of these metrics.

To evaluate performance on camouflage using similar metrics, the problem can be defined as a classification task by thresholding the camouflage likelihood maps in order to obtain a binary classification of camouflaged and non-camouflaged pixels. Then, we can define true positives (TP) and false negatives (FN) taking in consideration only camouflaged pixels. We propose to use the Recall measure (TP / (TP + FN)) on camouflaged pixels, i.e. the accuracy of the camouflage classification. Throughout this document we will use the term **score** or $R(\tau)$ indistinctively to refer to the Recall computed on camouflaged pixels, where τ is the value used to threshold the likelihood maps.

In the results detailed in the next section we have used a threshold of 1 when reporting overall scores to target fully camouflaged pixels. Nevertheless, it is interesting to compute the Recall for different values of threshold to account for flexibility of background models. To this aim, we compute $R(\tau)$ for τ in the range $[0, 1]$. The decrease of **score** consequence of an increase in the threshold provides information about the robustness of an algorithm against camouflage independently of its overall performance; hence, the evolution of $R(\tau)$ will allow us to observe the impact of the selected features in camouflage situations of different *intensity*. Notice that, for a threshold, $\tau = 0$, the **score** matches the Recall.

4.1.3. Algorithms

Our metric of performance against camouflage only accounts for camouflaged pixels, providing an objective characterization of the methods behavior. However, it may produce controversial results as algorithms with poor performance, regarding Precision and False-Positive Rate in foreground detection, may rank higher than algorithms with better overall performance. For instance, an algorithm classifying all the pixels of a frame as foreground has a camouflage accuracy of 100%. To handle this situation, we have decided to group algorithms according to their average Precision, as it reports a complementary evaluation to the Recall. We have established four quality categories: HQ (high quality), HMQ (high-medium quality), LMQ (low-medium quality), and LQ(low quality); in decreasing order of their average Precision reported as in [12]. See arrangement in Table 4.1.

We presume that algorithms that exploit other features than color will present a robuster performance against camouflage. AAPSA [3] presents a camouflage detector in order to adapt its updating rule, thus we can expect a better management of camouflaged sequences by AAPSA. Differently, edge detectors rely on the change of color to determine where an edge is present, therefore they may not be a good alternative. This limitation can be overcome by the inclusion of another feature as in EFIC [1]. EFIC relies on motion besides edges and color, their combination may provide a robust behavior against camouflage. Poor overall performance methods that rely only on color are not expected to present a good performance against color-camouflage, which is the one modeled in this work. However, it may be interesting to observe the behavior of CwisarDH [18], the only algorithm in the top quality category that relies only on color.

4.2. Experimental setup

The first two experiments target the design of the likelihood measure. Specifically, we study how the proposed camouflage estimation scheme describes empiric results. Our first challenge is to map the Jung's Ratio to a reliable probability estimator. In the following experiment, we fine-tune the N parameter of our model, which refers to the number of samples used in the background model. Examples of the generated camouflage likelihood maps are shown in Figure 4.6.

4.2.1. Likelihood of camouflage

Given the proposed measure of camouflage (see Chapter 3), we want to study how it describes the performance of selected algorithms and define a likelihood function. Let r_p be the Jung's Ratio of a pixel p. We want to find a function f such that $f(r_p)$ returns accurately the probability of p being camouflaged.

As r_p is defined in the range $[1, \infty)$ we propose its inverse as a first approach. Our camouflage likelihood function can be then defined as:

$$f(r_p) = \frac{1}{r_p} \tag{4.1}$$

We have studied the distribution of false negatives and true positives with respect to $f(r_p)$ over its range, $(0, 1]$, for some algorithms. We chose to use one algorithm of each quality category to keep the analysis simple yet representative.

As seen in Figure 4.1, most of the pixels present low camouflage estimation, but this trend is more notable for true positives (correctly classified camouflaged pixels), hence, there exists a relation between the proposed likelihood measure and the probability of miss-classifying a foreground pixel. By computing the False Negative Rate against the inverted Jung's Ratio we can infer the probability of miss-classify a pixel due to camouflage in terms of this measure. The False Negative Rate is defined as:

$$FNR = \frac{FN}{FN + TP} \tag{4.2}$$

It is easy to verify that, taking into account only pixels with a certain computed likelihood $y = f(r_p)$, the FNR computed over these pixels returns the ratio of miss-classified foreground pixels with estimated likelihood y, which can be generalized as the estimated probability; hence, the FNR computed for enough evenly sampled values of the likelihood function along the (0, 1] range returns the probability of miss-classifying foreground pixels in terms of the proposed likelihood function.

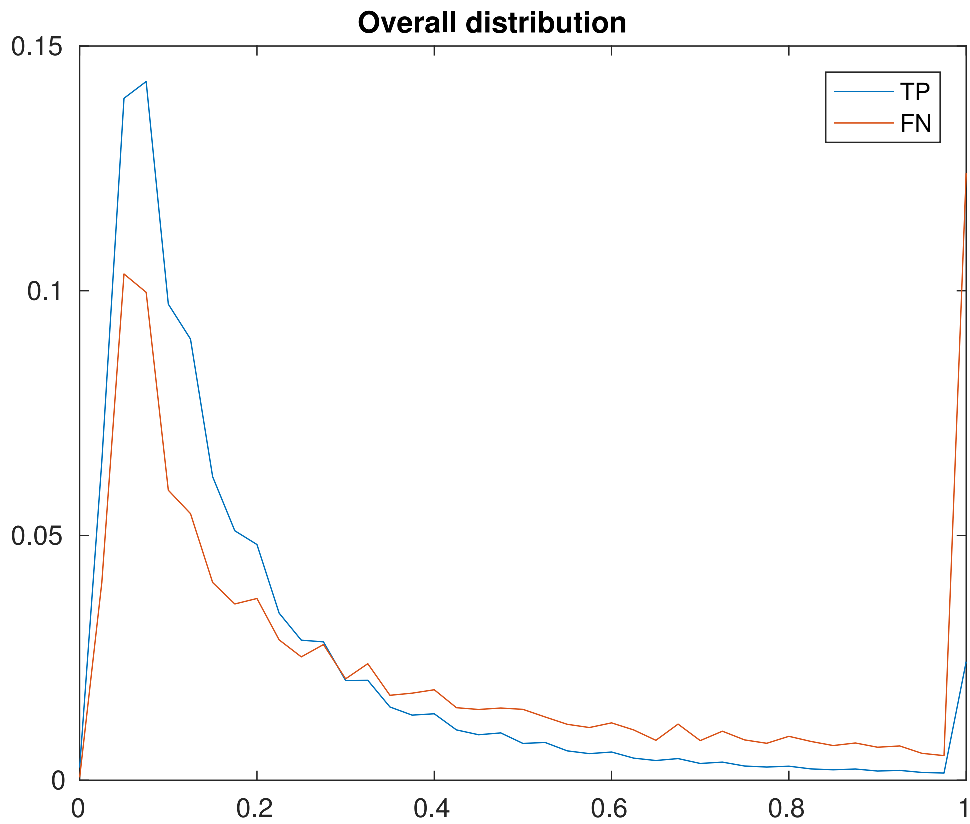

Figure 4.1: Average distribution of the camouflage likelihood for FN and TP pixels along all the sequences in the dataset.

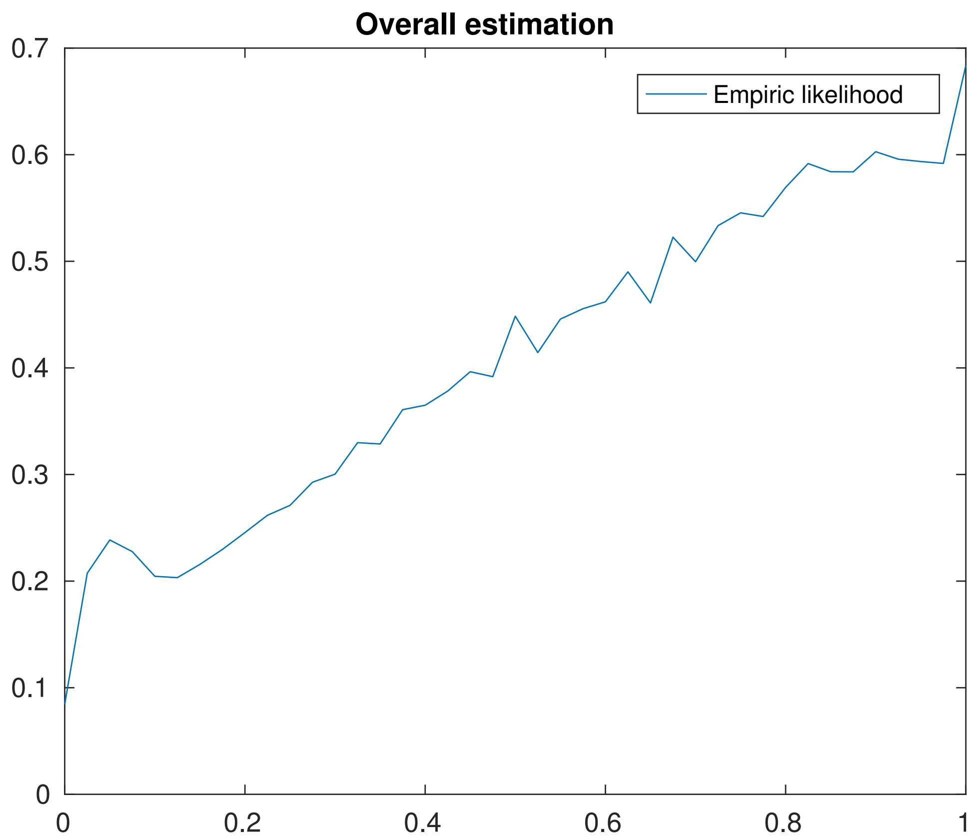

Figure 4.2: Average estimated probability across all the sequences in the dataset.

On average, the proposed measure is a good estimator of the probability of miss-classifying a foreground pixel. As seen in Figure 4.2, the empirical probability shows a linear behavior against the inverted Jung's Ratio with a nearly linear relation, implying direct proportionality. Thus, as this function tracks satisfactorily the empirical probability, we set definitively the measure defined by Equation 4.1 as our likelihood function.

4.2.2. Sensibility study, setting the value of N

The design of our background model is intentionally basic in order to keep a simple yet robust scheme. Although, some configuration is required, we have to set the parameter N (see section 3.2). This parameter indicates the number of background samples stored for every pixel. This value cannot be set randomly, as a high value of N will result in a high usage of memory and in an increase of the computational cost due to the requirement of calculation of the maximum distance between background samples, which is compulsory to obtain the Jung's Radius. Moreover, a background model composed of a high number N of samples may not adapt to fast changes of the background, resulting in obsolete representations of it. On the other hand, a low value of N may limit the models certainty as a certain number of samples are needed in order to store a proper background representation, e.g. to keep track of typical background fluctuations, we need a subset of background samples big enough to observe its general distribution.

To determine the sensibility of out likelihood measure with N we first observe the impact of N in the Jung's Ratio on a single pixel. As the Jung's Ratio is function of the pixel p and discrete time t, we can select a pixel in advance and represent its Jung's Ratio in time. This can be performed for different values of N. Results in Figure 4.3 suggest that there is not a high difference in the computed Jung's Ratio regarding the choice of the N value for the analyzed pixel.

A more general approach to determine N is to study its impact in $R(\tau)$ for a subset of algorithms. Figures 4.4 and 4.5 depict the obtained average score for different values of N regarding different categories and different thresholds, respectively. The study has been conducted using one algorithm of each category, the same way as in the previous experiment (see subsection 4.2.1). Again, it can be easily verified that the selected N value has a low impact on the obtained scores. Nevertheless, low values of N produce more stable results with the threshold, i.e. flatter curves in Figure 4.7, hindering the study of the relation between the score and the threshold. The lack of background samples may induce a volatile behavior of the Jung's Ratio, as a small background model may not be able to model highly-dynamic background.

Given these results, the N value is set to $N = 20$ for the next experiments. This value is big enough to offer a robust response of the Jung's Ratio. Moreover, Figure 4.5 shows that the score obtained for different thresholds settles around the value obtained for $N = 20$ and does not remarkably change for higher values of N. In our opinion, a bigger value will result in a unnecessary increase in computational cost.

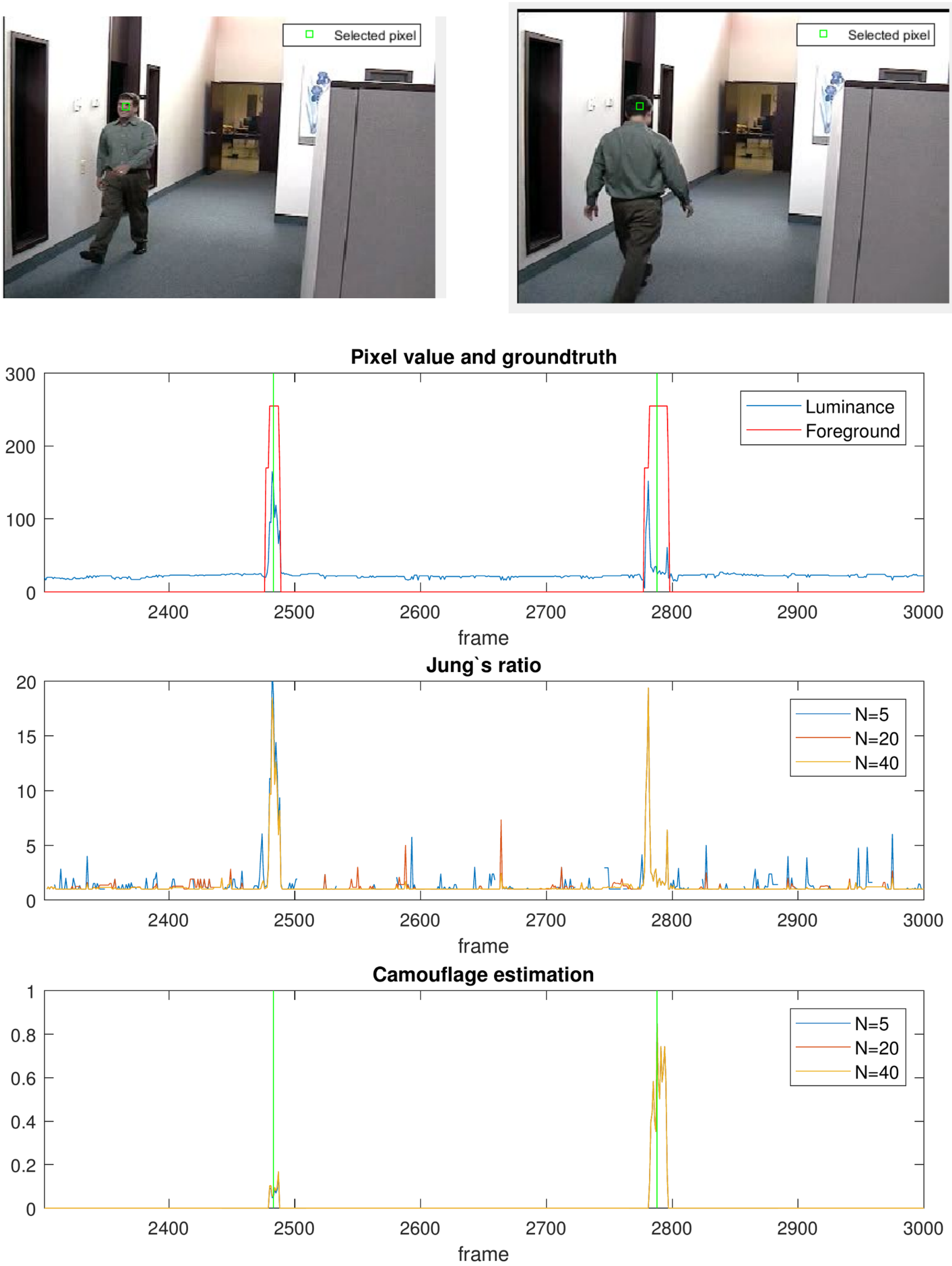

Figure 4.3: Evolution of a pixel in the shadow/cubicle sequence. This figure shows the same sequence as Figure 3.2.

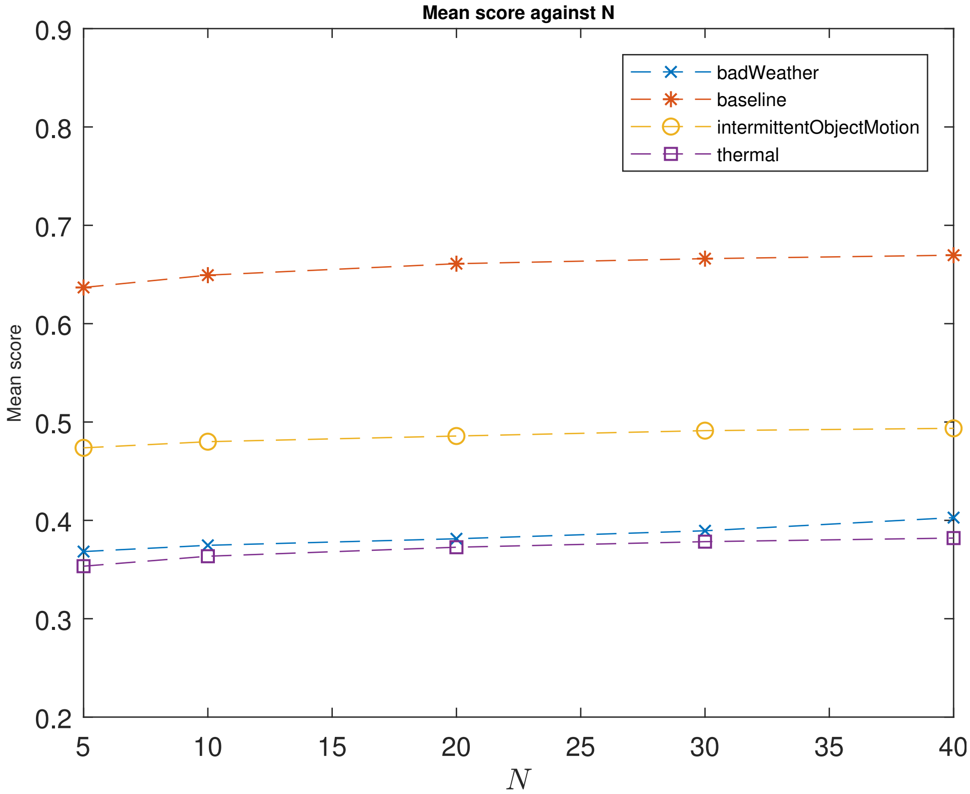

Figure 4.4: Score against N for different categories.

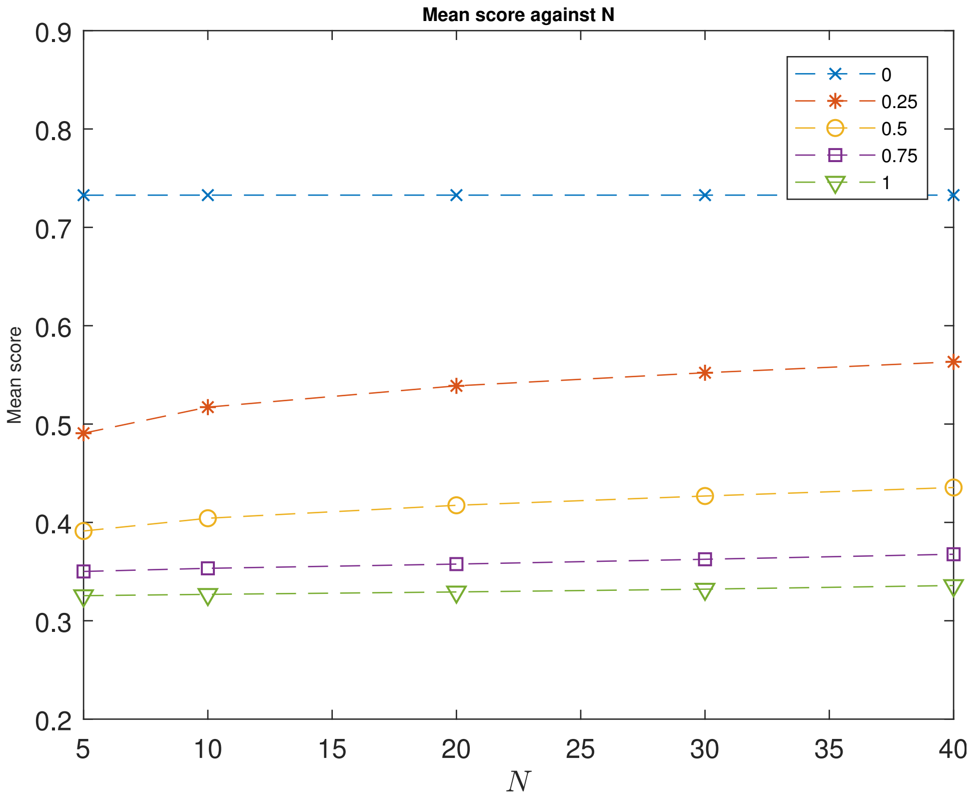

Figure 4.5: Score against N for different values of threshold.

Figure 4.6: Examples of some camouflage likelihoods. The brighter the higher the likelihood.

4.3. Experimental results

These experiments deal with the evaluation of background subtraction algorithms. We first rank the algorithms using the whole dataset with the intention of providing a more exhaustive study. Then, we observe which categories and videos are the most challenging regarding camouflage and rank the algorithms using these sequences.

4.3.1. Ranking BS algorithms

The camouflage maps obtained for the sequences in the CDNET2014 dataset, allows us to perform a camouflage-oriented ranking of different background subtraction algorithms. We have ranked the algorithms according to their performance on the whole dataset. As stated before, to report scores in Table 4.3 we have thresholded the camouflage maps using a value of 1 to target only fully camouflaged pixels. In any case, the robustness against camouflage as a function of τ can be observed in Figure 4.7. Regarding the interpretations of the curves there provided, we expect that a method robust to camouflage will maintain a similar score despite the camouflage likelihood of the targeted pixels, i.e. the method will present a flat curve in the graph. Likewise, a steeper curve towards the right bottom of the chart represents a decrease in performance for highly camouflage pixels.

Category name	Assigned Code
badWeather	1
baseline	2
cameraJitter	3
dynamicBackground	4
intermittentObjectMotion	5
lowFramerate	6
nightVideos	7
shadow	8
thermal	9
turbulente	10

Table 4.2: Categories in the CDNET2014 dataset.

Quality category	Algorithm	Categories										Overall
		1	2	3	4	5	6	7	8	9	10	
HQ	PAWCS [21]	**0.3790**	0.6665	**0.5226**	**0.6897**	**0.5377**	0.6300	0.2591	0.6261	**0.4721**	**0.4834**	**0.5258**
	IUTIS_5 [16]	0.2864	0.6909	0.4298	0.5236	0.4815	**0.6727**	0.4543	**0.6563**	0.2764	0.3232	0.4834
	FTSG [13]	0.2807	**0.6958**	0.3961	0.5859	0.5071	0.5512	**0.5562**	0.5847	0.2072	0.3077	0.4768
	CwisarDH [18]	0.2247	0.3171	0.2706	0.3129	0.2716	0.3894	0.2416	0.2186	0.0713	0.1379	0.2446
HMQ	EFIC [1]	**0.5110**	**0.6937**	**0.7041**	0.5392	**0.6621**	0.6787	**0.5558**	**0.7346**	0.4754	0.3439	**0.5929**
	SuBSENSE [17]	0.4231	0.6707	0.4889	0.4303	0.4920	**0.7401**	0.4573	0.7056	**0.4756**	**0.5209**	0.5360
	AMBER [19]	0.2756	0.6401	0.4219	**0.7566**	0.5318	0.3567	0.5412	0.5543	0.1780	0.2670	0.4702
LMQ	GraphCutDiff [2]	**0.6647**	**0.4895**	**0.5226**	0.5026	**0.2783**	0.3640	**0.5687**	**0.4662**	**0.2757**	**0.2900**	**0.4407**
	RMoG [27]	0.2274	0.3949	0.5062	**0.5090**	0.2630	**0.4084**	0.4188	0.2106	0.0209	0.1953	0.3151
	AAPSA [3]	0.2720	0.4844	0.3034	0.3074	0.2516	0.3959	0.2614	0.2656	0.0704	0.2747	0.2814
LQ	SC-SOBS [30]	0.1892	**0.5400**	**0.5150**	**0.6212**	**0.4261**	**0.5472**	**0.4369**	**0.3071**	0.0370	0.1847	**0.3844**
	KDE [15]	0.2213	0.4798	0.3192	0.3076	0.3370	0.5000	0.3452	0.2821	0.0744	**0.4285**	0.3224
	GMM [14]	**0.2437**	0.5213	0.2975	0.3377	0.3750	0.3786	0.3716	0.2441	**0.0985**	0.2839	0.3135

Table 4.3: Overall and per category score of the different algorithms. See Table 4.2 for categories correspondence.

We provide detailed report of the score obtained in each category for the different algorithms in Table 4.3. Besides, robustness against camouflage can be analyzed in Figure 4.7. These results need to be interpreted with caution. As we are evaluating the Recall only for camouflaged

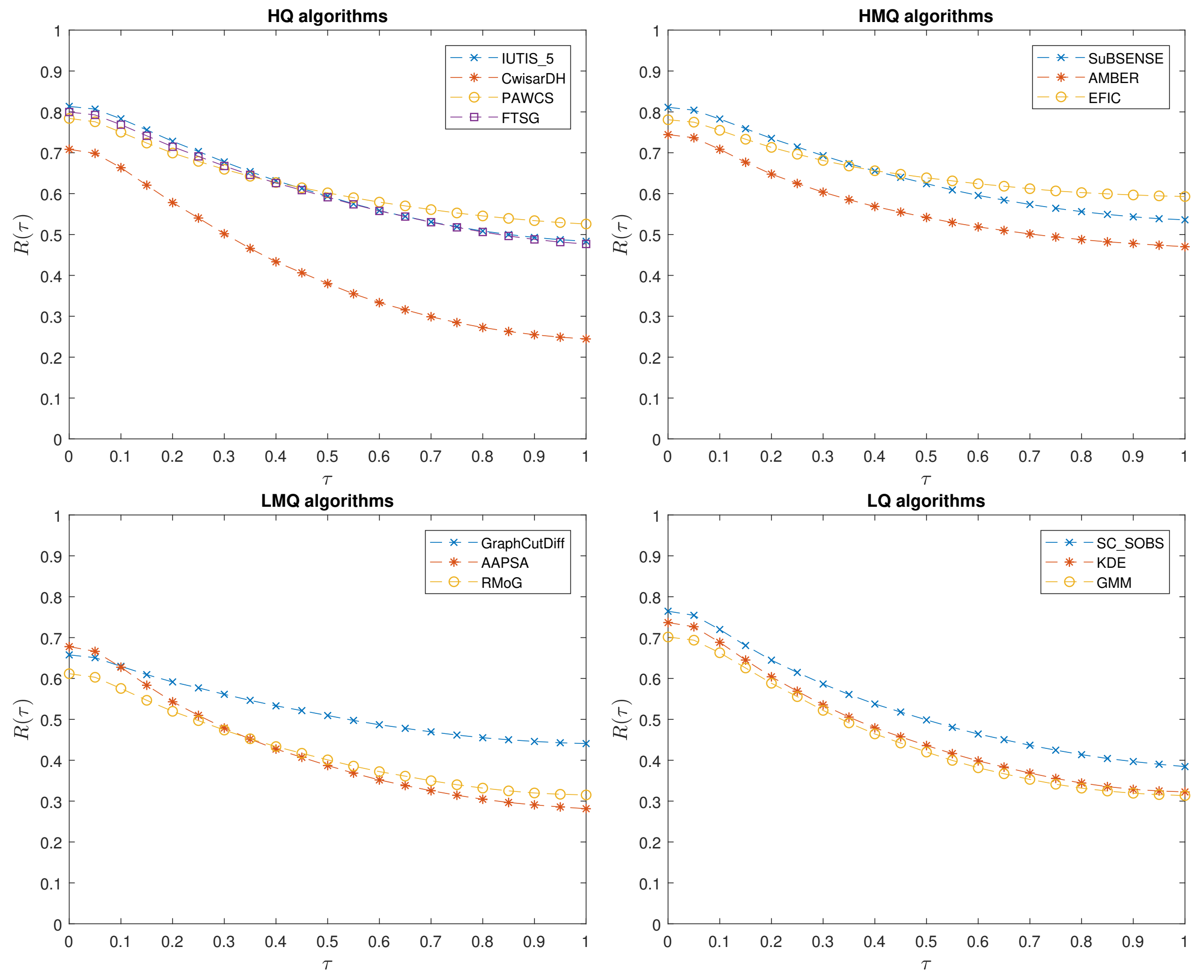

Figure 4.7: Average score against different thresholds.

pixels, this measure can be easily misinterpreted, e.g. a naive algorithm that detects every pixel as foreground would achieve a perfect score in our ranking. Under this assumption the ranking is only representative for algorithms in the same quality category.

Discussion: The obtained results seem consistent as the ranking obtained for each sequence-category does not present significant difference with that obtained for the others. Even though, there are some categories that present lower scores, these categories —e.g. "badWeather" and "thermal"— are commonly considered to contain camouflage (see subsection 4.1.1); hence, experimental results seem to confirm previous assumptions on these categories.

We can see in Figure 4.7 that algorithms that exploit other features different than color generally present a flatter slope, i.e. are robuster to camouflage. GraphCutDiff and EFIC seem to be highly robust against camouflage as their slope is pretty horizontal. If we take a closer look to he HQ algorithms chart, we can see that CwisarDH score decays more with the increase of threshold. Although it is true that it presents a lower score at the beginning, its graph present an steeper slope. As it may be checked in Table 4.1, this method is the only high quality method that relies exclusively on color. Another interesting observation is that low quality algorithms, which present an unexpected high Recall (score at threshold 0), do not maintain this performance with the increase of threshold suggesting that camouflage is a big problem for these algorithms. Otherwise, low medium quality algorithms present a worse Recall, but they are able to maintain similar performance despite the threshold used for evaluation.

These observations might be explained by the fact that these algorithms rely on other futures besides color (this is very notable for the "thermal" category in Table 4.3). Moreover, the HMQ graphs show that, as we presume, EFIC performs better against camouflage, but AMBER and SuBSENSE show a similar behavior despite the fact that SuBSENSE exploits LBSP features apart from color, in contrast to AMBER, which relies uniquely on color. It is also worth noting the dominance loss of methods exploiting motion in the "lowFramerate" category.

The most remarkable conclusion extracted from these results is that the use of some features such as motion can enhance significantly the robustness against camouflage in most of the different categories. This can not be extended to every alternative set of features, e.g. algorithms exploiting LBSP features do not show a remarkable improvement, in terms of camouflage robustness, with respect to algorithms relying exclusively on color.

4.3.2. Ranking challenging sequences

In the previous experiment we have evaluated algorithms using the whole dataset. An alternative solution, though less exhaustive, is to use only some representative sequences. We consider that the sequences that offer most valuable information, regarding robustness against camouflage, are those in which algorithms obtain lower scores,—i.e. the most challenging sequences in terms of camouflage. Therefore, we have averaged all the algorithms scores over all the different sequences and rank these sequences in increasing order of average score.

Table 4.4, shows the top-10 most challenging camouflage sequences. A low score means that the evaluated algorithms detect, in average, less camouflaged pixels; thus, the lower the score, the more challenging the sequence is in terms of camouflage. Besides, the table provides the proportion of fully camouflaged pixels ($\tau = 1$) with respect to the total number of foreground pixels.

Sequence	Category	Average score	% of camouflaged pixels
library	thermal	0.0731	0.0258
lakeSide	thermal	0.0879	0.0488
busyBoulvard	nightVideos	0.1446	0.3608
blizzard	badWeather	0.1462	0.0593
sofa	intermittentObjectMotion	0.1547	0.0335
turbulence3	turbulence	0.1788	0.1626
skating	badWeather	0.2331	0.0918
parking	intermittentObjectMotion	0.2501	0.0864
winterDriveway	intermittentObjectMotion	0.2654	0.0982
copyMachine	shadow	0.2688	0.0257

Table 4.4: Top-10 most challenging sequences in the CDNET2014 regarding camouflage.

Discussion: We can see that the leading categories, e.g. "thermal" and "badWeather", are categories that are expected to contain camouflage. Although, there are also some surprising results. There are three sequences of the "intermittentObjectMotion" category in the table. A discussion on these unexpected results is given in subsection 4.4.

It is worth noting that the average score seems to be uncorrelated with the ratio of camouflaged pixels. This means that a high presence of camouflaged pixels does not cause necessarily a challenging situation.

We believe that reported scores offer profitable information. The fact that the quantity of camouflaged pixels is uncorrelated with the level of challenge presented reinforces the approach of selecting challenging sequences. By doing this selection, we are picking truly challenging sequences, and discarding sequences in which camouflage may not present a real challenge for state-of-the-art algorithms.

4.3.3. Ranking BS algorithms in challenging sequences

We have ranked algorithms on a selection of the most challenging sequences regarding camouflage. We have used the top-10 most challenging sequences showed in the previous experiments. As these sequences are selected for the low average score obtained in these sequences, we expect lower scores than those reported in subsection 4.3.1.

Quality Category	Algorithm	library	lakeSide	busyBoulvard	blizzard	sofa	turbulence3	skating	parking	winterDriveway	copyMachine	Overall
HQ	PAWCS [21]	**0.1203**	**0.1831**	0.1206	0.0975	0.1671	**0.1329**	**0.2325**	**0.4821**	0.2022	**0.3234**	**0.2062**
	ITUIS_5 [16]	0.1063	0.1056	0.1656	0.1880	**0.2334**	0.1225	0.1615	0.2345	0.2846	0.2861	0.1888
	FTSG [13]	0.1143	0.0453	**0.2732**	0.1235	0.1648	0.0639	0.1805	0.2091	**0.3346**	0.2966	0.1806
	CwisarDH [18]	0.0643	0.0707	0.0271	**0.2199**	0.1359	0.0868	0.1288	0.2216	0.1280	0.1290	0.1212
HMQ	EFIC [1]	0.0633	**0.2939**	0.1318	0.1322	0.2510	0.2953	**0.4788**	**0.4969**	**0.5305**	**0.4907**	**0.3164**
	SuBSENSE [17]	**0.2021**	0.2809	**0.1829**	**0.2124**	**0.3096**	**0.3568**	0.1639	0.1358	0.3397	0.3648	0.2549
	AMBER [19]	0.0651	0.0414	0.1435	0.0288	0.0981	0.1091	0.2441	0.3950	0.3240	0.2738	0.1723
LMQ	GraphCutDiff [2]	0.0199	0.0256	**0.3007**	**0.5535**	**0.2735**	**0.3526**	**0.5696**	0.0454	0.1355	0.2658	**0.2542**
	AAPSA [3]	**0.0599**	**0.0158**	0.0636	0.1263	0.0522	0.1984	0.1736	**0.1569**	**0.2963**	**0.3275**	0.1471
	RMoG [27]	0.0029	0.0031	0.1496	0.0354	0.0267	0.0364	0.1930	0.1333	0.1619	0.1747	0.0917
LQ	GMM [14]	0.0160	**0.0472**	0.1011	**0.1412**	**0.1443**	0.1951	0.1285	**0.4398**	0.2411	0.1603	**0.1615**
	KDE [15]	**0.0588**	0.0250	0.1063	0.0331	0.0762	**0.2174**	0.1547	0.1473	**0.2745**	**0.2200**	0.1313
	SC-SOBS [30]	0.0574	0.0051	**0.1134**	0.0085	0.0785	0.1578	**0.2210**	0.1540	0.1977	0.1815	0.1175

Table 4.5: Overall and per sequence score of the different algorithms in challenging sequences.

Overall scores using a threshold of $\tau = 1$ are reported in Table 4.5, whereas performance against variable camouflage is shown in Figure 4.8. Even though scores are generally lower, these results should be studied under the same considerations than in subsection 4.3.1.

Discussion: We can see that the overall behavior is similar to that reported using the whole dataset. Some of the observations made over previous results apply to these as well. GraphCutDiff [2] presents a flatter slope than other algorithms in the same quality category. However the loss of performance of CwisarDH [18] is less noticeable in this case, although it can be verified that it still presents the steeper slope of the high quality category. The overall ranking exposed in Table 4.5 is similar than that reported in Table 4.3 for high quality and high medium quality algorithms. Although, it changes for low medium quality and low quality methods. Regarding low quality algorithms, all of them present similar performance, therefore slight changes in the obtained scores might have great impact on the final ranking. Taking a closer look to the low medium quality methods we can observe that RMoG [27] has been relegated to the last place. This could be because of a worse overall performance of the algorithm within these sequences, but, as seen in Figure 4.8 it still presents a flatter slope than AAPSA [3], suggesting that it is a robuster method against camouflage.

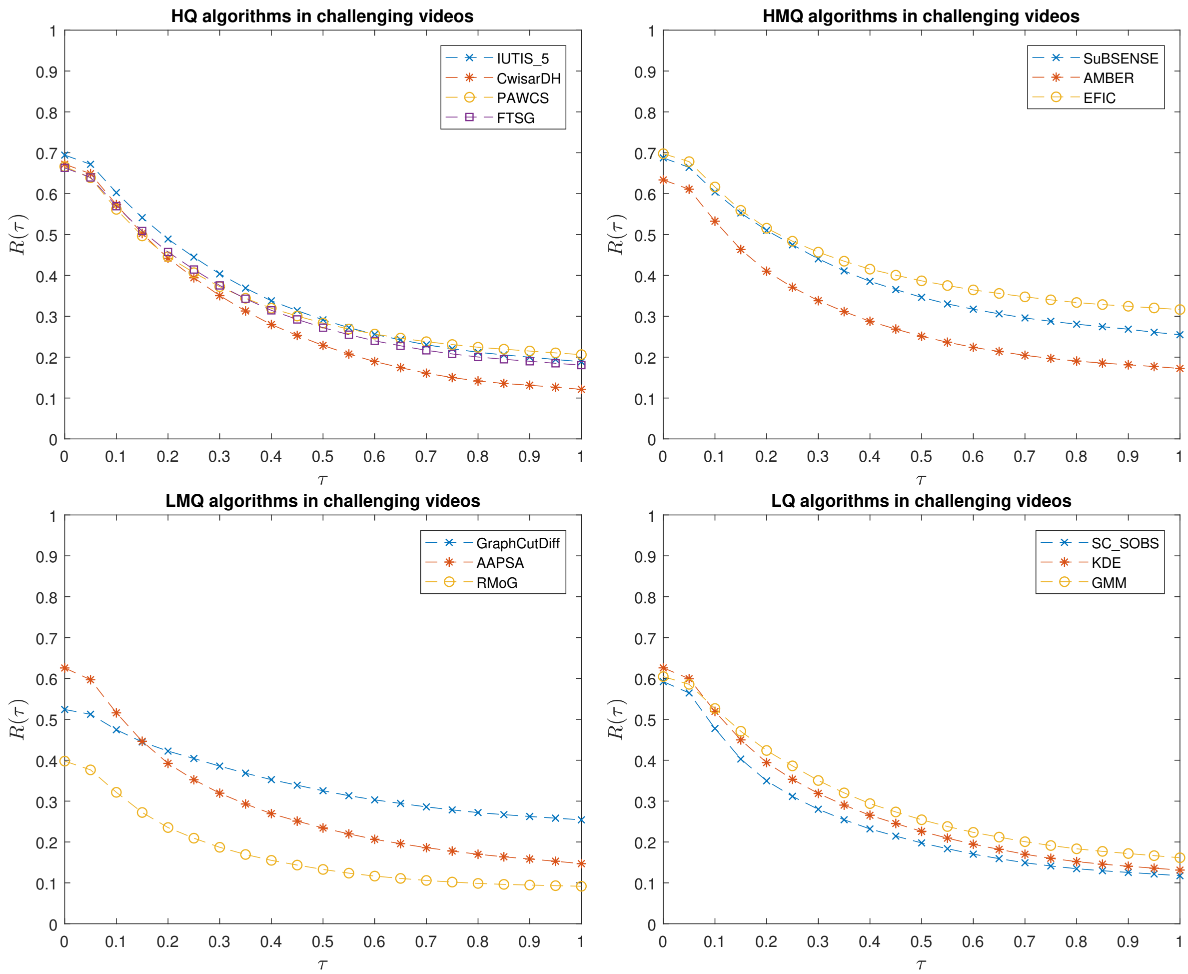

Figure 4.8: Average score in challenging sequences against different thresholds.

4.4. Drawbacks and limitations

The selection of challenging sequences reported in subsection 4.3.2 show some unexpected results. There are three sequences belonging to the category "inttermitentObjectMotion" which is not expected to present major issues regarding camouflage.

A closer inspection of these sequences revealed that, in the case of "parking" and "winterDriveway", a car that is initially labeled as background is suddenly classified as foreground when it starts moving. As both cars were considered background and relabeled without changes in their appearance, our model *considers* that those pixels are fully camouflaged until the car moves away from the region. This means that any possible false negative within this region penalizes the score, we provide visual results of these failure cases in Figure 4.9. Differently, the "sofa" sequence of this category turned out to contain true camouflage.

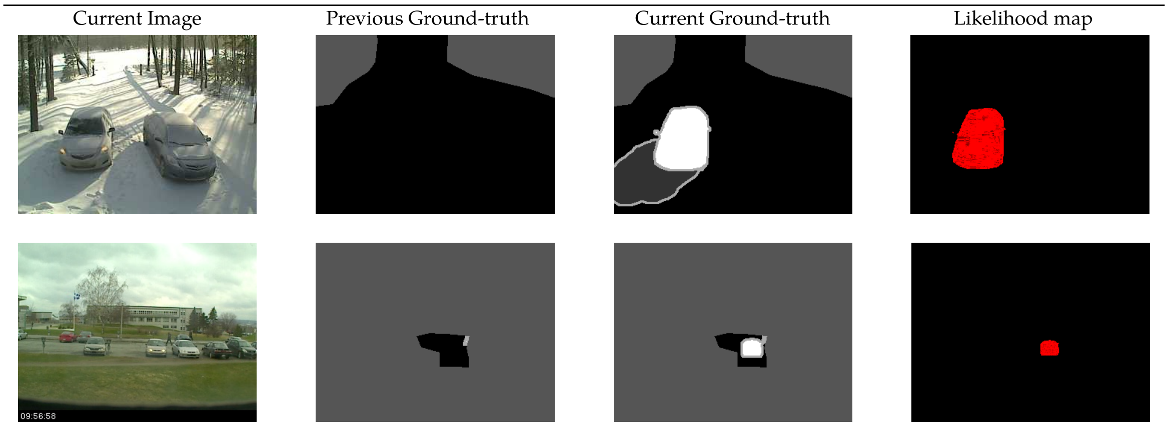

Figure 4.9: Examples of failure cases. The brighter the higher the likelihood.

5
Conclusions and future work

In this chapter we provide our conclusions on this work and we discuss possible future work.

5.1. Conclusions

We have described a new technique to quantitatively measure camouflage based on the Jung's theorem. To the best of our knowledge, this is the first attempt in modeling camouflage using this approach. We have been able to present a novel solution for this problem showing promising results as described in subsection 4.2.1.

Taking advantage of this measure we have computed camouflage likelihood maps for the CDNET2014 dataset. In order to do this, we have used of the hand-labeled ground-truth annotations provided in [12] to automatically generate camouflage likelihoods. We propose a simple non-parametric background model feed with ground-truth information. Given a set of background samples we compute its Jung's Radius and measure its necessary increase, Jung's Ratio, in order to embrace a new foreground sample. This measure provides a powerful tool to measure the saliency of foreground objects.

Throughout the setting experiments we have proposed a likelihood based on the Jung's Ratio. First, we have designed a function that maps this measure to a camouflage likelihood with satisfactory results. Then we have set the number of background samples to store by studying the impact of the background models size in the evaluation of background subtraction algorithms regarding camouflage.

We have ranked several state-of-the-art algorithms exploiting the generated likelihood maps and provided detailed scores for each method. We have provided quantitative confirmation of previous common assumptions on the level of camouflage expected in different categories. The evidence from these results suggests that taking advantage of alternative features to color may significantly enhance the robustness against camouflage. This behavior is particularly notable in methods using motion features, which seem to be surprisingly robust to color-camouflage.

We have selected the most challenging sequences regarding camouflage and sorted these sequences in increasing order of the average score obtained by the evaluated algorithms. We have perform another evaluation of the same methods but using only these challenging categories.

The obtained results confirm previous conclusions. We have experimentally demonstrated that the proportion of camouflaged pixels do not have direct impact on the level of challenge presented regarding camouflage.

One observed limitation is the inclusion in the background model of awakening foreground objects. In some sequences, an object may be labeled as background initially, but at a certain frame it will appear as foreground in the ground-truth because it is beginning to move. As it was labeled as background, our background model contains a representation of this object, hence, once it becomes foreground it does not differ from its background and is detected as camouflaged.

5.2. Future work

In our opinion these results represent a big step towards camouflage modeling. One possible improvement would be the inclusion of neighboring information, this could be done by working at region-level using superpixels.

Results show that some alternative features may enhance robustness against camouflage, however the set of methods ranked in this work do not encompassed all the possible alternative features. Particularly it would be interesting to analyze the performance of methods exploiting alternative domains, e.g. a wavelet domain.

The present findings might help to solve limitations of existing methods regarding camouflage challenging situations. We are currently in the process of developing a system to detect potential risk zones for a tracker in which it may get lost based on the Jung's Radius, presented in this work.

Bibliography

[1] Gianni Allebosch, Francis Deboeverie, Peter Veelaert, and Wilfried Philips. Efic: Edge based foreground background segmentation and interior classification for dynamic camera viewpoints. In Sebastiano Battiato, Jacques Blanc-Talon, Giovanni Gallo, Wilfried Philips, Dan Popescu, and Paul Scheunders, editors, *Advanced Concepts for Intelligent Vision Systems*, pages 130–141, Cham, 2015. Springer International Publishing.

[2] A. Miron and A. Badii. Change detection based on graph cuts. In *2015 International Conference on Systems, Signals and Image Processing (IWSSIP)*, pages 273–276, Sept 2015.

[3] Graciela Ramírez-Alonso and Mario I. Chacón-Murguía. Auto-adaptive parallel som architecture with a modular analysis for dynamic object segmentation in videos. *Neurocomputing*, 175:990 – 1000, 2016.

[4] Thierry Bouwmans. Traditional and recent approaches in background modeling for foreground detection: An overview. *Computer Science Review*, 11-12:31 – 66, 2014.

[5] S. Brutzer, B. Höferlin, and G. Heidemann. Evaluation of background subtraction techniques for video surveillance. In *CVPR 2011*, pages 1937–1944, June 2011.

[6] S. Li, D. Florencio, W. Li, Y. Zhao, and C. Cook. A Fusion Framework for Camouflaged Moving Foreground Detection in the Wavelet Domain. *ArXiv e-prints*, April 2018.

[7] X. Zhang, C. Zhu, S. Wang, Y. Liu, and M. Ye. A bayesian approach to camouflaged moving object detection. *IEEE Transactions on Circuits and Systems for Video Technology*, 27(9):2001–2013, Sept 2017.

[8] MIKHAIL KATZ. Jung's theorem in complex projective geometry1. *The Quarterly Journal of Mathematics*, 36(4):451–466, 1985.

[9] Marcos Escudero Viñolo. Contributions to region-based image and video analysis: feature aggregation, background subtraction and description constraining. 2016.

[10] Diego Ortego, Juan C. SanMiguel, and José M. Martínez. Stand-alone quality estimation of background subtraction algorithms. *Computer Vision and Image Understanding*, 162:87 – 102, 2017.

[11] R. Shi, K. N. Ngan, S. Li, R. Paramesran, and H. Li. Visual quality evaluation of image object segmentation: Subjective assessment and objective measure. *IEEE Transactions on Image Processing*, 24(12):5033–5045, Dec 2015.

[12] Y. Wang, P. M. Jodoin, F. Porikli, J. Konrad, Y. Benezeth, and P. Ishwar. Cdnet 2014: An expanded change detection benchmark dataset. In *2014 IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pages 393–400, June 2014.

[13] R. Wang, F. Bunyak, G. Seetharaman, and K. Palaniappan. Static and moving object detection using flux tensor with split gaussian models. In *2014 IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pages 420–424, June 2014.

[14] C. Stauffer and W. E. L. Grimson. Adaptive background mixture models for real-time tracking. In *Proceedings. 1999 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (Cat. No PR00149)*, volume 2, page 252 Vol. 2, 1999.

[15] Ahmed Elgammal, David Harwood, and Larry Davis. Non-parametric model for background subtraction. In David Vernon, editor, *Computer Vision — ECCV 2000*, pages 751–767, Berlin, Heidelberg, 2000. Springer Berlin Heidelberg.

[16] S. Bianco, G. Ciocca, and R. Schettini. Combination of video change detection algorithms by genetic programming. *IEEE Transactions on Evolutionary Computation*, 21(6):914–928, Dec 2017.

[17] P. L. St-Charles, G. A. Bilodeau, and R. Bergevin. Subsense: A universal change detection method with local adaptive sensitivity. *IEEE Transactions on Image Processing*, 24(1):359–373, Jan 2015.

[18] M. D. Gregorio and M. Giordano. Change detection with weightless neural networks. In *2014 IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pages 409–413, June 2014.

[19] B. Wang and P. Dudek. A fast self-tuning background subtraction algorithm. In *2014 IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pages 401–404, June 2014.

[20] M. Sedky, M. Moniri, and C. C. Chibelushi. Spectral-360: A physics-based technique for change detection. In *2014 IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pages 405–408, June 2014.

[21] P. L. St-Charles, G. A. Bilodeau, and R. Bergevin. A self-adjusting approach to change detection based on background word consensus. In *2015 IEEE Winter Conference on Applications of Computer Vision*, pages 990–997, Jan 2015.

[22] G. A. Bilodeau, J. P. Jodoin, and N. Saunier. Change detection in feature space using local binary similarity patterns. In *2013 International Conference on Computer and Robot Vision*, pages 106–112, May 2013.

[23] I. Aleksander, W.V. Thomas, and P.A. Bowden. Wisard·a radical step forward in image recognition. *Sensor Review*, 4(3):120–124, 1984.

[24] Filiz Bunyak, Kannappan Palaniappan, Sumit Kumar Nath, and Gunasekaran Seetharaman. Flux tensor constrained geodesic active contours with sensor fusion for persistent object tracking. *Journal of multimedia*, 2(4):20, 2007.

[25] X. Tan and B. Triggs. Enhanced local texture feature sets for face recognition under difficult lighting conditions. *IEEE Transactions on Image Processing*, 19(6):1635–1650, June 2010.

[26] D. M. Gavrila. A bayesian, exemplar-based approach to hierarchical shape matching. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 29(8):1408–1421, Aug 2007.

[27] S. Varadarajan, P. Miller, and H. Zhou. Spatial mixture of gaussians for dynamic background modelling. In *2013 10th IEEE International Conference on Advanced Video and Signal Based Surveillance*, pages 63–68, Aug 2013.

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

[29] Herbert Bay, Andreas Ess, Tinne Tuytelaars, and Luc Van Gool. Speeded-up robust features (surf). *Computer Vision and Image Understanding*, 110(3):346 – 359, 2008. Similarity Matching in Computer Vision and Multimedia.

[30] L. Maddalena and A. Petrosino. The sobs algorithm: What are the limits? In *2012 IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops*, pages 21–26, June 2012.