UNIVERSIDAD AUTÓNOMA DE MADRID

ESCUELA POLITÉCNICA SUPERIOR

Feedback-based integration of temporal and spatial coherence schemes for video-object segmentation based on background subtraction.

Marcos Escudero Viñolo

Supervisor: Jesús Bescós Cano

TRABAJO DE FIN DE MASTER

Departamento de Ingeniería Informática
Escuela Politécnica Superior
Universidad Autónoma de Madrid

Noviembre 2009

FEEDBACK-BASED INTEGRATION OF TEMPORAL AND SPATIAL COHERENCE SCHEMES FOR VIDEO-OBJECT SEGMENTATION BASED ON BACKGROUND SUBTRACTION

Marcos Escudero Viñolo

Supervisor: Jesús Bescós Cano

e-mail: {marcos.escudero, j.bescos}@uam.es

Video Processing and Understanding Lab.

Departamento de Ingeniería Informática

Escuela Politécnica Superior

Universidad Autónoma de Madrid

Noviembre 2009

Abstract

The work presented in this Master Thesis deals about video sequences region based object segmentation. Work starts from an efficient State of Art pixel level segmentation. In parallel, a new Mean-shift approach builds a region segmentation image of each frame by clustering pixels in a region focusing in the intrinsic characteristics of real objects. Obtaining a region based segmentation where illumination influence has been severely diminished. Region segmentation and pixel level segmentation are combined to discriminate between foreground and background regions in a scene.

Specifically, these regions are used to build and update a multilayer background and a foreground model. Regions in the models are characterized by a time varying covariance matrix which encloses a set of relevant features. Covariance matrix evolution along the video allows the system to discriminate between foreground and background regions. A new static region tracking approach is used to update background model while a dynamic region tracking is performed to update foreground model and identify regions frame to frame. After region discrimination a simple feedback scheme exports segmentation results to pixel level module. Finally, an approach to export foreground region tracking to connected-component tracking is presented.

Results show that segmentation approach practically avoids illumination artefacts from segmentation without any post-processing technique. Additionally, system fills objects holes and is exportable to multimodal backgrounds environments.

Work supported by:

the Spanish Administration Agency CDTI (CENIT-VISION 2007-1007)

and the Comunidad de Madrid (S-0505/TIC-0223 - ProMultiDis-CM).

Acknowledgements

First, I want to sincerely express my gratitude to my supervisor, Professor Jesús Bescós. I want to thank him for his deepness dedication, our inspiring conversations and his personal and technical support.

Then I want to thanks Professors Jesús Bescós and Jose Maria Martinez for giving me the opportunity of being part of the Visual Processing and Understanding Lab.

In this line, I can forget about my Lab partners, who are always by my side, helping me at any trouble, making me laugh even when the world has become blue.

I won't have been here without my family, without their help and support, without their education, without their love. Your confidence makes me better; your affection makes me sure. Thank you so much.

My friends have always been a light in the dark, an oasis in the desert, a continuously source of satisfaction, a necessary escape from routine. They are by my side when I need them. I won't have finished this work without them, especially without the mathematical support of Javier Ramos. Thank you all.

Finally, I want to thanks my muse, my partner, my love, the girl that makes me happy with a word, with a smile, with a hug, with a kiss. She has suffered and enjoyed this work with me. I can not ever thank you enough Elena.

Contents

Figure Index

Table Index

I Chapter 1 – Introduction

1 Motivation

Nowadays, one of the main objectives of video understanding techniques is to overcome the ‘semantic gap’. The concept of ‘semantic gap’ characterizes the difference between two descriptions formulated by different linguistic representations. In video analysis the ‘semantic gap’ is defined as the empty space between the human perception of content and the representation of the content that is included in the digital video signal [I 1]. In other words, it is the difference between the formulation of contextual knowledge in a powerful language (the human natural language) and its formulation in a formal language (in video signals, either the codification standard or the colour representation system in raw video).

The concept of ‘semantic gap’ can be extended to the ‘semantic pyramid’ concept, which can be understood as a division of the gap in several levels of understanding roughness. In video signal, the lowest level in the pyramid would be the pixel level. In the next level, pixels can be grouped to form the region level. Upper, a group of regions can be categorized as an object in the following level. Finally, the scene or group of interrelated objects is at the top of the pyramid.

When interacting with video content, people would like to access information (searching, indexing, viewing or tagging it) with high level scene descriptions. That is, with requests at the highest level of the pyramid instead of requests at the lowest. For instance, a user should query for a dog running in a park and not for a group of brown pixels over a bigger group of green ones.

Society demands research and advances in video analysis technology. Several areas of interest and applications have been derived from technology development and still need new research and results to increase its capabilities. Quoting some of the main ones; video-security, computer vision, multimedia content indexing or video coding. Weighting current necessities, the potential of an automatic semantic description server based in analysis of the digital video signal is enormous.

The digital video signal analysis techniques, which main objective is to generate high level semantic descriptions should ascend in the pyramid starting from the lowest level, that is, the pixel flow. Additionally, segmentation and tracking of objects is essential to describe what is in the scene and what is happening, that is, to describe the scene in a natural language.

Taking these premises under consideration, we believe that region level is not only the natural way to ascend from pixel analysis to objects segmentation, but also a key intermediate step to control pixels aggregation parameters, to consider illumination issues, as well as to characterize objects as bags of regions. Motivation of this project relays in the study of strong points, opportunities and benefits of region segmentation and tracking in opposition with systems that straight jump from pixel to object level.

2 Objectives

The main objective of this Master Thesis is to contribute to reduce the ‘semantic gap’ in as generic as possible environments (subordinating them to fixed camera situations). With this objective, we use region level analysis techniques both to feed forward object level analysis and to feed backward pixel level analysis.

Additionally, this work has specially focused in shadows and sparkles influence in final results. Illumination influence is not isolated but covers variable size areas (from small regions under objects to the whole frame). According to this, it seems adequate to consider illumination effects at region level instead of, as several works stand (see section II2), at pixel level. Consequently, the research, development and use of illumination-insensitive techniques are the other main objectives of this Master Thesis.

Objectives of current Master Thesis can be listed:

- Design and implementation of automatic and innovative region segmentation techniques based on region intrinsic characteristics.
- Categorization and tracking of segmented regions.
- Design and implementation of a feedback scheme from region to pixel level analysis. System works using two different paths: classical sequential flow and feedback flow from high to low level analysis stages.

Summarizing, main research has been made in:

- Selection of features to categorize unequivocally each region.
- Selection of robust approaches to avoid empirically setting of thresholds.

In conclusion, our aim is to allow the whole system to work as a black box, where video frames enter and semantic descriptions at region level are given at the output and feed next level. Every part of the work has taken under consideration potential use of regions and descriptors to feed next layer in the semantic pyramid. Furthermore, we have kept an eye at improvements obtained by feedback strategies at the low level analysis stage.

II Chapter 2 – State of Art

Region segmentation is a sub product of the proposed approach and a pixel level segmentation is the starting point of it. Consequently a description of existing approaches both in pixel level and region level segmentation is necessary to compare technology and show differences between current state of art and designed system.

Region tracking and characterization is the main objective as we have explained and motivated in I2. To show advantages of our system, existing approaches should also be included and commented.

Additionally, shadows influence has been diminished in the proposed work; so, the implemented technique needs to be compared with existing ones to be assessed in an overall view.

Finally, one of the significant points of the approach is the dual path: forward and backward. Therefore, developed feedback strategies are briefly described to motivate its advantage over layer-independent forward paths.

1 Segmentation approaches

II.1.a Low Level Segmentation

Low level Segmentation approaches are those that use low level information to perform segmentation. Low level information can be motion vectors, DC or AC coefficients and codification modes in compressed domains **[SEG 1], [SEG 2]**, or spatial configuration of each frame and temporal configuration of whole video extracted from pixel information (when we straight try to segment raw video).

II.1.a.1 Compressed-domain segmentation techniques

For completeness, we include in current state of art some of the existing segmentation approaches that work without decompressing the video stream, but developed work is far from these techniques. However, some of them share with the presented work a similar semantic-ascending scheme and that turns them relevant to our work.

The main advantage of compressed-domain segmentation is the fact that by working directly with compressed data, video does not need to be decompressed and consequently, the amount of data to process is between 4 and 64 times smaller in comparison with decompressed data. Therefore, analysis should be faster and results can be directly used by a video-codec (thus according with recently and poorly developed enhancements included in MPEG-4 **[SEG 3].**

Most of the existing techniques still work over MPEG-1/2 compressed domain available information to perform segmentation. According to parallelism with our work we can mention approach of **[SEG 4]** that uses motion vectors information in conjunction with colour information extracted form DC coefficients to segment moving objects in I frames (intra-predicted frames).

We can observe semantic pyramid ascension in the work proposed by **[SEG 5]** that preliminary over-segments each I frame by applying watershed transform (explained in section II.1.b) **[SEG 6]** and then uses motion information to combine previously segmented regions in moving objects. On the other hand, in **[SEG 7]** region segmentation is performed via motion information clustering along a group of frames, while objects boundary refinement is based in colour information extracted from DC coefficients.

Finally, the work developed by **[SEG 8]** proposed a modular technique in which motion and temporal tracking of motion vectors is the main source of information to achieve segmentation. **[SEG 9]** includes extensions and enhancements over that work. Specifically, their work uses colour information, adds a module that deals with intra-codified parts of a frame that includes motion information, a technique to avoid suddenly disappearance of previously segmented objects, and a starting approach to extend results to multi-modal backgrounds.

II.1.a.2 Raw pixel based segmentation techniques

According to bibliography, *background subtraction* is the core of pixel based segmentation techniques. Its relevance is even higher if we restrict segmentation to fixed cameras environments (as we have restricted in our work and is the common situation in scenarios as video-surveillance).

Background subtraction approaches are based on building and maintaining a model of the background and classify each pixel as either background or foreground depending on a measure of the dissimilarity with the stored background model. The nature of the model and the way to measure dissimilarity establish the differences between existing approaches.

As techniques in segmentation at region level are more similar among them (as we explain in section II.1.b) than at pixel level, most of the ideas that inspire our work are extracted from pixel level classical techniques and extrapolated to region level.

We can roughly divide existing methods following the well-known survey proposed by Piccardi **[SEG 10]** and make a simple to complex classification. Enumerating;

Running Gaussian average methods are those in where background is modelled independently at each pixel. Evolution of each pixel in time is fitted to a Gaussian distribution in which influence in the model of past and current samples of the pixels is weighted differently; in example for the mean of the Gaussian, see equation (1):

$$\mu_{x,y,f} = \alpha I_{x,y,f} + (1-\alpha)\mu_{x,y,f-1} \qquad \textbf{(1)}$$

Where, $I_{x,y,f}$ and $\mu_{x,y,f}$ are the value of the pixel *x*, *y* and of the mean of pixel *x*, *y* at frame *f* respectively, and α is the weighting factor (its value should be between 0 and 1). Running Gaussian average was first proposed by **[SEG 11]** and is used at the starting point of our approach.

Temporal median filter methods are those in which pixel value is computed just for a few frames that can be consecutives **[SEG 12]** or sampled along the video **[SEG 13]**. Obtaining more stable background model, by avoiding continuously influence of a pseudo-static object that appears somewhere in the middle of a video.

Mixture of Gaussians (MoG) based approaches increase **Running Gaussian average** methods capabilities by using more than a single Gaussian to model background at each pixel. With this approach, authors **[SEG 14]** enhances background modelling with the capability of work in multi modal background scenarios. In these scenarios Gaussians distributions of each pixel are supposed to model the different configurations of the pixel along the video. This approach has inspired region background modelling in our work as it is explained later in this document.

Kernel Density Estimation or KDE tries to add marginal samples influence to the model, that is, influence of an outlier sample would be located at the tails of a Gaussian in a **Mixture of Gaussians** approach while it is considered when using a **KDE** based method **[SEG 15]**. To avoid influence of foreground pixels in the model (which influence used to be marginal when observing a specific pixel) samples are added to a FIFO queue.

A quantitative and qualitative comparison among these background subtraction techniques can be found in **[SEG 16]**.

There are more complex techniques as **Co-occurrence of Image Variations [SEG 17**] **Eigen-backgrounds [SEG 18]** and **Bayesian Modelling [SEG 19]** but they are out of the scope of the proposed approach and in authors' opinion, its inclusion in current state-of-art can divert the attention of a patient reader.

All of the described approaches model the background to detect the foreground as exceptions over modelled background. Recent examples of evolutions and improvements over basic models can be also consulted in **[SEG 20]**. Other approaches also model the foreground, as this is the case of proposed work, we will briefly describe the philosophy of these out-of-standard techniques.

In background and foreground modelling techniques, foreground is detected by maximum a posteriori (MAP) measure of trained models for each class [S. Khan] [Mittal]. Probabilistic models can be any of the described *Background subtraction* techniques as explained in **[SEG 21]** where they finally decide for a feasible in computational time MoG model. These approaches have also inspired our work as can be checked in model description chapters (IV and I).

Results obtained by *Background subtraction* approaches are not usually good enough for authors' requirements either because there are inaccuracies in them or owing to an unsuitable processing time. Therefore, there are several works that pre or post process results obtained by *Background subtraction* techniques by using approaches fed with others sources of information, as in example; edges **[SEG 22]**, colour **[SEG 23]**, texture **[SEG 24]**, deepness **[SEG 25]** or change detection **[SEG 26]**.

Furthermore, there are several approaches that try to fuse information sources without giving priority to any of them. In this category we can include the works developed by

[SEG 27] and **[SEG 28]** where change detection is combined with *Background subtraction* via a Boolean logic and a Bayesan framework respectively. As a result of a deep study of these approaches, **[SEG 29]** proposed a low level fusion based segmentation approach as well as a feedback scheme. This technique has been considered a suitable one to be the starting point of our approach owing to the fact that proposed scheme perfectly fit with a whole semantic system scheme proposed by the authors. System is later explained and would be mentioned in other chapters of this Master Thesis documentation.

II.1.b Region based segmentation

In order to categorize and track regions we first need to segment each frame in such regions. Regions can be considered, as explain in (***I1Motivation***), as groups of connected pixels that share one or more features (colour, texture, spatial location inside a close boundary, etc.)

Region segmentation techniques can be very roughly divided in two groups as proposed in **[RSEG 1**: Region Growing, where a number of basic regions (seeds) are given and different strategies are used to join surrounding neighbourhoods; Split approaches, where the algorithm starts from non uniform regions and subdivides them until reaching uniform regions; Merge approaches, which start from non uniform regions and merge them until fulfilling asset of uniformity criteria among them. Usually, splitting and merge approaches are used together with merging as a post-processing stage after splitting.

Examples of classical *splitting* and *merge* methods can be found in papers from **[RSEG 2]** and **[RSEG 3]**. Both techniques follow a two-step process: first, homogeneity criteria are set and image is split in four quadrants if it does not fulfil those criteria. Then each quadrant is iteratively split in four smaller quadrants. Image is thus, segmented in progressive smaller quadrants (quadtree segmentation). This process locally stops in a quadrant if criteria are fulfilled for it. In the second stage, two adjacent similar quadrants are *merged* if they satisfied the same criteria used in the first stage

[RSEG 4] and **[RSEG 5]** presented evolved *splitting* and *merging* methods. In **[RSEG 4]** splitting is made by sequential histogramming for five colour features while in **[RSEG 5]** image is first splitting into chromatic and achromatic regions attending to human perceptual perception of colour. Then chromatic regions are *splitted* again. *Merging* is the post-processing stage in both papers. Additionally, we can consider as *Merging* processes the classical works proposed by **[RSEG 6]** and **[RSEG 7]** where authors use edge information to discriminate if a pixel is in a contour or not before measuring the similarity to candidate regions.

On the other side, differences between existing *seeds growing* approaches are essentially in features nature, similarity criteria used to divide/create/fuse regions and in the technique used with that purpose. As grouping sets of pixels to build regions is equivalent to clustering those pixels in classes, every classical clustering algorithm can be use for segmenting an image into regions. Examples of these methods are; the simple Nearest Neighbours (NN) which was used in **[RSEG 8]**, the distance between Karhunen-Loewe Transform (KL) of the original data, part of the work proposed by **[RSEG 9]**, an algorithm to peak selection in data distribution by Fisher projections, explained in

[RSEG 10] and the popular Mean-Shift **[RSEG 11]** algorithm (we go further in this algorithm in chapter IV). Fuzzy logic segmentation has also relevance in image region segmentation and we can consider these methods as an approximate way to perform *seeds growing.* Examples of fuzzy K-Means approaches can be found in **[RSEG 12]** and **[RSEG 13].**

In between seeds growing approaches and *split and merge* methods there are some others where image segmentation is formulated as a graph partitioning problem. The graph is subdivided in sub-graphs by pruning weighted edges of the graph. The total weight of the pruned edges between two sub-graphs is called a cut. Examples of works developing this technique are **[RSEG 14]** and **[RSEG 15].**

Other region segmentation techniques are the level set approaches, being the watershed transformation the more representative one. Watershed simulates a flooding process over the image obtaining a topological map representing the value of the gradient at each pixel. Inherently watershed results are very dependent on gradient estimation. A good description of watershed transformation can be found at **[RSEG 16]** and at **[RSEG 17].**

Most of the described region segmentation approaches are based on characteristics from the local neighbourhood of a pixel to decide whether adding it to an existing region or creating a new one. This process can be done either by finding discontinuities in the similarity criteria or by searching surrounding pixels that delimitate areas fulfilling established criteria. That is, finding every pixel in the boundaries of the region, we can also discriminate the region. Work developed by **[RSEG 18]** classifies methods to segment an image in uniform areas in boundary based and region based. Advantages of region based methods are that they do not rely in edge extraction and consequently, do not suffer from inaccuracies in this process (e.g., sudden cuts in extracted edges). However, region based methods usually need a high amount of pixels to compute reliable statistics in order to build the regions, and so, usually suffer for over-segmentation, losing in the best case, part of the image fine resolution.

Works mentioned in this section up to this point are of the region based type. While, the most relevant and interesting works in boundary based methods are the so called *Active Contours Models* (ACM) based approaches. ACM core is the minimization of an energy function that describes each contour, this energy function used to have two components, internal energy which is the part that tries to fit the *active contour* to region shape, and external energy which objective is to separate the region from the rest of the data.

ACM were first formulated by **[RSEG 19]** but most famous approaches based in ACM philosophy are those that use *snakes* (a simile to refer to deformable curves). Snakes can be classified either as parametric or as geometric. Parametric snakes are explicitly represented as parameterized curves in Lagrange formulation **[RSEG 20]**, while geometrical snakes are represented implicitly and evolve according to the Euler formulation based on the theory of surface evolution and geometric flows **[RSEG 21].** Drawbacks associated with snakes are mainly two: first its initialization dependence (partially solved by combining snakes and watershed **[RSEG 22]**) and inaccuracy to converge to the boundary concavities of a region. There are several techniques that have tried to reduce this drawback **[RSEG 23]** and **[RSEG 24]**, but they often result in very complex *snake* models.

In conclusion, attending to the complexity of the ACM models, the inaccuracies associated to *split* and *merge* methods, and the wide use of Mean-Shift region segmentations in current SoA, we finally choose this last data-set peak estimator to perform our region segmentation. Furthermore, we have introduced a couple of changes over the base algorithm to diminish influence of shadows in final segmentation. Next chapter is related to SoA in shadow detection and discrimination while Mean shift and the proposed improvements are described in chapter IV.

2 Shadows management

Shadows can be considered the illumination artifacts with higher influence in segmentation and tracking results, thus, there are a considerable amount of works that have tried to diminish this influence. In video analysis there are several interesting constraints related to the presence of a shadow:

- Pixel luminance decreases in comparison with that of the stored background model, but commonly, texture of the shadowed surface remains unaltered (in fact it always remains unaltered, but we can not distinguish it in a complete light absence situation).
- Light intensity reduction rate is smaller in the transition shadow-no shadow.
- *Cast shadows* are fused to the objects and are connected to them; those are the focus of most existing techniques. On the other hand, *self shadows* are part of the object and are not usually extracted.

Most of the developed segmentation and tracking techniques make results conditional to homogeneity in illumination and scenarios free of light artifacts. Failures in segmentation, and consequently in tracking, owing to these unpredicted, but common, situations are usually assumed, and its solution is delayed to specially targeted post processing techniques. These failures are related to the presence of more than one illumination source as well as to reflection phenomena produced on the illuminated objects surface. Moreover, post processing techniques do not usually stand for generic situations but, instead, focus on the suspicious wrong segmented areas, as cast shadows under objects, and try to discriminate them from applications targeted items (commonly moving objects).

Classical post processing techniques use to work in colour spaces in which one or more of the channels are less prone to shadows, for instance HSV (*Hue Saturation* and *Value*). Then, a set of sometimes empirical thresholds are configured based on ratios between channels, and pixels under or over those threshold are considered shadow or sparkle pixels respectively. Examples of these works can be found in **[SHD 1]**, **[SHD 2]** and **[SHD 3**].

In other works, a function to express light and colour capitation on the camera sensors has been tried to be modelled. Starting from functions similar to Equation (2), several approximations to simplify the model have been proposed.

$$I = \int s(\lambda) \cdot e(\lambda) \cdot r(\theta, v, n, \lambda) \mathrm{d}\lambda \qquad \textbf{(2)}$$

The model expresses the image brightness value, *I*, captured by a camera sensor with spectral response $s(\lambda)$, assuming an illumination source with a spectral distribution $e(\lambda)$ that emits over an object surface with an angle θ respect to its normal vector n. The distribution of the reflected light can be described by the reflectance function $r(\theta, v, n, \lambda)$ where v is the camera viewing angle.

Simplifications of the model are proposed in **[SHD 4]** by using colour, texture, darkness adjacency and temporal consistency properties, also in **[SHD 5]** which uses temporal continuity of reflectance and in **[SHD 6]** that combines physical properties of the objects with some empirical assumptions.

Finally, it can be interesting to mention works developed in the area of intrinsic image extraction. The term was proposed by Barrow and Tannembaum in 1978 when they were searching a way to decompose an image between illumination and reflectance sub-images. The illumination image should contain all the illumination present in the scene, while the reflectance (also known as intrinsic) shows the intrinsic inalterable properties of the objects.

Decomposition of an image in intrinsic and illumination image is obviously far from simplicity and several authors have proposed different ways to carry this division out. In **[SHD 7]** authors start from the illumination model described in Equation (2) to extract, by illuminating with different temperature lights a calibrated scene, the evolution direction of the colours, and thus, projecting the image on this direction they obtain the intrinsic image. In **[SHD 8]** the authors extend this work, and proposed a method to avoid the calibrating phase.

[SHD 9] and **[SHD 10]** use a different approximation to extract the intrinsic image. They make use of a set (normally a big one) of images showing the same scene under different illumination conditions. Extracting boundaries of each image, they can differentiate which of them are intrinsic to the objects and which are owing to light influence.

We can finally just mention some other works **[SHD 11]** **[SHD 12]** that also propose the use of low level techniques to detect illumination artifacts and correct its influence in analysis results (e.g., by using these pixels to update the background model). However, we believe that illumination influence does not appear over isolated pixels in the frame but, on the contrary, it affects variable-size closed regions (shadows and sparkles).

According to described objectives, the use of illumination invariant techniques or at least of techniques that diminish illumination influence is one of the key points of the proposed Master Thesis.

3 Feedback Schemes

The classical analysis path is sequential: results from pixel-level analysis feed region level modules whose results serve as input to object level analysis and so on. However, pixel-level analysis stages lack of semantic information available in higher level analysis stages. As the proposed work motivates the use of stratified layers to avoid the semantic gap, it seems adequate to provide the lower layers with feed-back of the higher level information extracted upper in the pyramid.

If we focus in the area related to the feeding of pixel level segmentation approaches from higher layers in order to improve both pixel level and global system segmentation results, we can mention three works in this line.

In **[FB 1]**, with same objects as this Master Thesis, a scheme to feed “Time-Adaptative, Per-Pixel Mixtures of Gaussians” segmentation approaches is proposed. In order to perform this feedback process they modified classical foreground segmentation by adding region based information and semantics related to objects in the Gaussians modeling scheme.

Differently, in **[FB 2]** a generic scheme to avoid failures in segmentation due to the scene noise is proposed. The authors suggest that by decomposing each frame in different description levels then, coherence and similarity between levels can be used to enable feedback among such levels.

Finally, the already mentioned work of **[SEG 29]** (see II.1.a) is of main relevance in work under presentation, as it first proposed the feedback scheme followed in this Master Thesis.

4 Tracking approaches

Simple tracking in video analysis can be defined as the problem of estimating the position of an object in the image plane related to the position of the same object in the previous frame, thus, estimating its trajectory along the video.

Object tracking is probably one of the main targeted applications of video analysis, especially in security applications. At present days, tracking is considered not only as a system final result, but as an essential intermediate step to extract semantically richer information. Consequently, relevance of a precise and truthful tracking is a key point in any system which aims either providing useful semantic descriptions or reliable detecting security violations events.

Object tracking approaches differ in: the nature of the features used to match objects frame to frame, the representation or container of those features and the metric used to measure the dissimilarity between features vectors. Every tracking method demands previous object detection or initialization; these detection can be done with any of the segmentation approaches described in section II.1.a.2 , or it can be manually done.

Colour is a main feature to track objects, but any other feature can be added to colour or used isolated to track objects; results are usually the judge to check if features selection has been appropriate. When tracking, an object is divided in fixed or variable-size image fractions that can be as small as a pixel or contain the object and part of the background. Features to characterize these image fractions are in some way extracted from pixels colour information, which is included in DC coefficients or in raw image in compress or decompress video respectively. Furthermore, the colour of a pixel is a function of the camera sensor, objects reflectivity and illumination sources, according to functions similar to Equation (2) but considering three channels and dependencies among them. We wont go deeper in colour systems representation, as research in this area can be fruit of whole Master Thesis (or even a Phd. Thesis), and so we recommend the biblical book of Wyszecki and Stiles **[TRC 1]** to deal with the selection of a suitable system for each problem.

Next chapters try to introduce, very briefly, some of the main tracking methods without loosing the general idea of this SoA that is, introducing works that have motivated ours. Classification of the systems is made according to well know object tracking survey proposed by **[TRC 2]**.

II.4.a Point Tracking

If the image fraction is as small as a pixel, tracking can be classified as point-tracking. Point correspondence is a difficult problem owing to occlusions, misdetections, new objects appearance and disappearance from the image plane.

Point tracking methods can associate a cost to each point and try to minimize that cost by combinatorial optimization (called deterministic methods). These methods try to search a some to some or one to one points association. Some approaches choose a set of characteristic points extracted for example by Harris **[TRC 3]**, SIFT **[TRC 4]** or SURF **[TRC 5]** and try to match these points frame to frame to follow the object. On the other hand, Hungarian algorithm **[TRC 6]** computes all points possible associations and choose minimal cost to perform the tracking, these methods are usually too heavy to fulfil time constrains.

Non deterministic approaches, called statistical approaches consider random noise influence in point characterization and so consider model uncertainty when assigning objects state (object predicted position). One the most known tracking statistical methods is the Kalman filter. A Kalman filter is used to estimate the state of a linear system where the noise has a Gaussian distribution. Recent approaches that deal with Kalman filtering can be found in **[TRC 7]** and **[TRC 8]**

If object state (and thus noise) is not assumed to be Gaussian, it is necessary to use particle filter methods **[TRC 9]** where a set of samples of the state of the objects are weighted according to its observation frequency (sampling probabilities).

II.4.b Kernel Tracking

Object is represented either as a whole area (primitive object), which encloses object and surrounding background or is divided in several parts and each part is tracked separately. Some of the Kernel tracking methods follows an object (or part of an object) template and compare it with possible templates in the image (usually this comparison is made in the vicinity of the object to minimize computational and temporal constrains).

This template can be in example its Mean-Shift segmentation **[TRC 10]** where templates or colour histograms are compared via the Bhattacharya coefficient **[TRC 11]** or a covariance matrix of a whole or part of an object and its surroundings **[TRC 12]**, the influence of this work in recent research as well as its flexibility has turn it a perfect state of the art for our tracking approach (we go deeper in this technique in system description chapters)

II.4.c Silhouette Tracking

Finally, and for completeness, we include the silhouette trackers, where either a shape model or a contour model (which is continuously updated) is searched frame-to-frame to provide object tracking. Shape matching can be considered similar to template matching mentioned in previous section: models are built via set of points features and comparison can be made by using the Hausdorff distance **[TRC 13]**.

Contour tracking methods iteratively evolve an initial contour frame-to-frame, demanding previous and current contour partial overlapping and adapting the contour to frame new configuration. Examples of contour tracking approaches are **[TRC 14]** where a posteriori contour probability is maximized and **[TRC 15]**, where an energy function modelling the temporal optical flow in the boundaries is minimiz

III Chapter 3 – System Overview

In this Chapter we present the architecture of the designed and implemented system, and explain how the system is able to discriminate foreground from background regions, and track foreground regions along time. In next chapters we go deeper in each module operation.

1 System Scheme

We start from a pixel level segmentation, a SoA approach based on an iterated combination of a background subtraction and a frame difference technique. This approach is explained in **[SEG 29]** and will be very briefly commented in this Master Thesis. Additionally, the region level segmentation module is the result of improving the design proposed by **[RSEG 11]** by introducing illumination invariant features in the clustering operation core. From there in advance, every other module depicted in the scheme (**Figure 1**) is fully original, and used approaches have been, even inspired in SoA ideas, fully designed and implemented.

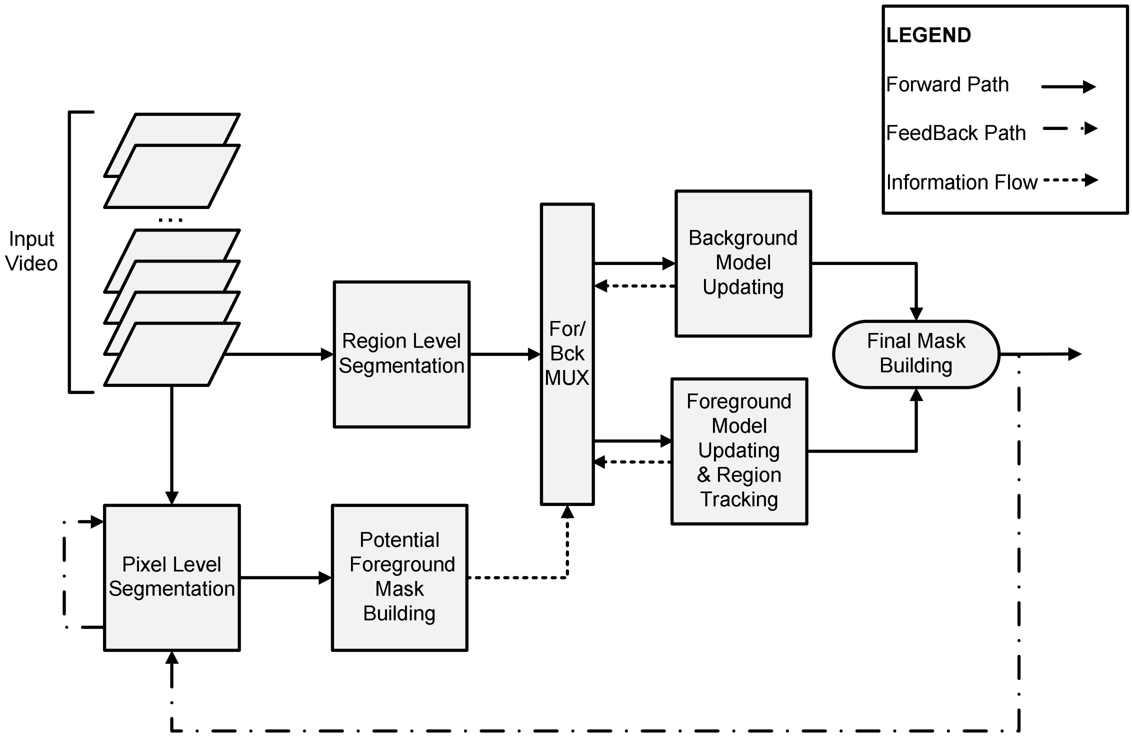

LEGEND
Forward Path
FeedBack Path
Information Flow
Input Video
...
Region Level Segmentation
For/ Bck MUX
Background Model Updating
Final Mask Building
Foreground Model Updating & Region Tracking
Pixel Level Segmentation
Potential Foreground Mask Building

Figure 1. ***System Scheme***

System operation flow at each frame can be described according to Figure 1.

III.1.a Pixel and region level segmentation

First, the every incoming video frame is separated in foreground and background by a pixel-level segmentation module **[SEG 29]**. In this module a background model based on a simple Gaussian is built and each frame's information updates the model. Additionally, frame difference is computed respect to the previous frame. Foreground is then detected as changes in the background model which are coherent with frame-difference information and previous frame segmentation. The module results in a binary mask, with foreground pixels activated ('1') and background pixels deactivated (´0´). Actually, this mask includes additional information (e.g., uncovered background). However, to provide our system with the ability of working at region level independently of the system used at pixel level, we finally decided to use just the described binary mask, which is commonly the output of almost every pixel segmentation module.

In parallel, region-level segmentation is performed over every incoming frame. The considered frame is divided in reflectance homogeneous regions by a Mean-Shift based approach. Shadows and light artifacts influence are diminished by using features and distance measures alternative to those used in classical Mean-Shift based approaches. This region-level segmentation is combined with the pixel-level segmentation to achieve an enhanced pixel-level segmentation mask, which additionally includes region-level information.

III.1.b Foreground/background region de-multiplexer

To provide the system with the capability of filling holes and recover miss-detected parts of the pixel-level foreground segmentation mask, this mask is dilated, with an NxN square structuring element, to build a *potential* foreground mask.

The potential foreground mask indicates to the foreground/background de-multiplexer if an input region should be considered either a background region or a potential foreground region. Confirmed background regions (those composed of deactivated pixels in the dilated mask) allow to build and maintain a consistent background model. With this information, we can robustly test the hypothesis of a confirmed new background region in the current frame to correspond or match to a region in the background model. Additionally, the potential foreground mask provides the capability of searching for foreground objects just in the regions activated at low level potential foreground mask.

III.1.c Foreground and background models updating strategy

Once the region segmentation has been performed for an incoming frame, the potential foreground segmentation mask is used to label them either as confirmed background or potential foreground.

Potential foreground regions will be finally assigned to the foreground or to the background, depending on how well they fit to the foreground or background model. A similar operation is performed for confirmed background regions. These models contain information to assess the cost of assigning a new region to an existing region in the model. Once a region has been decided to match a modelled region, the new region features are used to update the corresponding model.

To perform tracking of foreground regions, we just find correspondence between regions in the foreground model and regions classified as foreground in the current frame. Extension of this region tracking to object tracking is then a simple task, which is described in section.VI4.

When every region in the frame has been used to update either foreground or background models, the final mask resulting of activating foreground regions and deactivating background regions is used to update the background model of the pixel level segmentation module, this way, performing the feedback stage by using midlevel (region) results to improve low level (pixel) analysis.

In next Chapters we explain in detail each of mentioned approaches. First, the Mean Shift implementation used to segment the image in regions (IV), then, the designed background model and its updating strategy (I), and following, the foreground model used to detect foreground regions and perform the tracking between them(I).

IV Chapter 4 – Mean-Shift Based Region Segmentation

1 Mean-Shift

A deep explanation of the Mean-Shift clustering algorithm is out of the scope of this work. We propose the reader to consult the exhaustive description, reasonable motivation and original formal definition proposed first by **[MS 1]** and **[MS 2]** and then developed (including interesting applications of the system) in **[RSEG 11]**. However, a basic description of Mean-Shift is needed to assess improvements included in this Master Thesis over the basic system.

Most of the existing clustering techniques need a previous knowledge about the number or type of clusters to be built. For instance, classic K-Means clustering by expectation maximization (EM) requires the number of clusters as an input parameter **[MS 3]**. Mean-Shift is a non-parametric approximation. This turns to be a point in analysis of feature arbitrary spaces. Furthermore, it is computationally less expensive than other non-parametric approximations as hierarchical K-Means.

The objective of Mean-Shift is to find local extrema (peaks, modes) in the density distribution of a data set. For continuous distributions, Mean-shift just iteratively hill-climbs over the density distribution until it reaches a maximum. To provide robustness, Mean-Shift works in a delimitated part of the distribution. The window that encloses Mean-Shift working area is defined by a kernel and the size of the window by the bandwidth of that kernel. This way, the technique avoids the influence of outliers in peaks estimation and, by shifting the window, is capable to compute a set of peaks or modes that implicitly divide the data-set in a bandwidth-dependent number of clusters. These clusters are commonly fused in a post processing stage based on similarity criteria in order to avoid inaccuracies in the clustering owing to the window size restriction.

The bandwidth of the kernel (even over the kernel shape) is the most relevant parameter and consequently, several works have proposed approaches to choose its value. Value selection can be fixed (set based on the nature of the data-set **[MS 4]**) or it can dynamically be changed at each dimension of the data-set distribution **[MS 5]**.

In video analysis, most of the proposed approaches cluster colour-spaces data-sets **[RSEG 11]**, but there are also works that propose to include texture **[MS 5]**, or even oriented energies **[MS 6]** in the feature data-space. Moreover, it is common to include features vector's position in the data-set, in order to achieve a final set of connected component clusters. As the nature and range of features can be extremely different, use of different bandwidths for each dimension of the feature vector is strongly recommended.

Dealing with kernel selection, the Epanechnikov kernel appears to be the most commonly used kernel in application of Mean-shift to video analysis **[MS 7]**. Influence of points in the window falloff with the square of the distance between the point and the center of the window if the Epanechnikov kernel is used.

After this brief description of Mean-Shift, in next section we introduce the new approaches included in this Master Thesis project. We start from the popular SoA implementation proposed by **[RSEG 11]** and include some improvements both in features used for the clustering stage and in the distance used for the post processing fusion process. These improvements are motivated and described in the following sections.

2 Features description

As explained in the previous section, we have considered Mean-Shift the best tool to combine the proposed features for three main reasons:

- The method avoids the selection of fusion rules, which usually turns to heuristics.
- There is no need to estimate the number of clusters for each frame and for each video.
- Its main parameter, the bandwidth, can be easily defined for each of the inputs.

First of all, we present the used features; then, we describe their utility in our system and the way we combine them via Mean-Shift. Additionally, it is important to remark that our scheme is based on the fact that Mean-Shift has two differentiated phases: a clustering phase and a cluster fusion phase.

IV.2.a Albedo ratio

In **[SHD 5]** the authors prove that, under certain conditions, the albedo ratio is independent of the reflectance function and of the illumination spectrum. If we consider that the light source is white colored and that the sensor response remains constant across the visible light spectrum, Equation (1) becomes:

$$I = s \cdot e \cdot \rho \cdot R(\theta, v, n) \qquad \textbf{(3)}$$

, where dependence with λ has disappeared, ρ represents the integral of the reflectance function over the visible light spectrum, and *R(θ, v, n)* is the distribution of the reflected light for the particular wavelength of the incident light, hence discriminating between reflective power and reflectance distribution.

If we now consider a particular pixel in a small area surrounded by a smooth continuous surface, we can assume that *v, n,* and *θ,* are approximately the same for every neighbour pixel inside such area. According to this simplification we can define for two neighbouring pixels:

$$\begin{aligned} I_1 &= k_1 \cdot \rho_1 \cdot R(\theta, v, n) \\ I_2 &= k_2 \cdot \rho_2 \cdot R(\theta, v, n) \end{aligned} \qquad \textbf{(4)}$$

, where $k_1 = k_2 = k$ depends on the light source and sensor response. The authors consider it a constant, but we further discuss about its value in section IV.2.b

From (4) we can develop that neighbour pixels under the considered conditions will share the same reflective power if they belong to the same material but unequal if they belong to different ones. The albedo ratio, defined as it follows, will be an indicator of this situation:

$$P = \frac{\rho_1}{\rho_2} = \frac{I_1}{I_2} \qquad \textbf{(5)}$$

, or, to avoid indetermination when $I_2 \approx 0$:

$$P = \left| \frac{I_2 - I_1}{I_2 + I_1} \right| \qquad \textbf{(6)}$$

As we are computing the reflectance ratio between neighbour pixels, we can assume that all of them are illuminated with the same distribution emitted by the same sources. Hence, Equation (6) also holds for multiple illumination sources. All these expressions assume that pixel intensity has previously been gamma-compensated.

IV.2.b Color vectors angle

The term k defined in section IV.2.a depends on the camera sensor, and the illumination spectrum and intensity. We agree with the authors that sensor and spectrum can be the same for neighbouring pixels under detailed conditions, but light intensity can vary inside a reflectance shared region. The result presented in previous section fails in albedo homogeneous regions illuminated with different intensity light sources:

$$k_1 \neq k_2 \qquad \textbf{(7)}$$

, a situation closely related to shadow presence. When an object blocks a light source, the area behind the object in the trajectory defined by the light wave and the object becomes darker. This darkening can result in medium illuminated areas (penumbra) or poorly illuminated areas (umbra) depending on the relative position of the area with respect to the occluding object, the light source and the ambient illumination. In this situation, pixels belonging to the same material but in different shady areas won’t share similar ρ.

To tackle this problem we propose to use the color vectors angle measure as described in **[MS 8]**, which is claimed to be robust to changes in illumination intensity. The measure assumes that, in a gamma-compensated RGB space, if a pixel with a color vector *c* becomes under-illuminated, its modified color vector can be expressed as:

$$c' = \alpha\, c. \qquad \textbf{(8)}$$

Consequently, both vectors share the same orientation. The proportionality factor α is closely related with the light intensity component of k.

This scheme has some problems derived from the structure of the RGB color space (e.g., under low illumination conditions, colors are all almost proportional among them due to quantification), which have to be considered when using this descriptor.

3 Proposed approach

The novelty of the proposed approach lies in the integration of the two presented features in the base operation of the Mean-Shift algorithm: just the albedo ratio is used in the clustering phase and both features are used in the cluster fusion phase.

IV.3.a Bandwidth selection

The kernel bandwidth is a Mean-Shift parameter that controls the criteria or restrictions to cluster pixels in the clustering phase. Most segmentation approaches consider several pixel features (e.g., position, luminance, color) and, for each, a similarity range. These jointly define a multidimensional bandwidth. The range implicitly assumes pixel-feature comparison via the Euclidean distance.

We propose to combine pixel position, compared with the Euclidean distance, and pixel intensity, this compared via its ratio (as defined in Equation (6)), hence inherently including the albedo ratio in the bandwidth selection. In this line, the algorithm establishes that a neighbourhood for a pixel is defined as the set of pixels that are spatially closer than 10 pixels and present albedo ratios smaller than 0.01. These parameters are set motivated by the assumptions made in **[SHD 5]**.

IV.3.b Clusters fusion

The described Mean-Shift phase clusters regions attending to local estimations, which over-segments the scene in a large set of small regions, reflectance-homogeneous in our case. According to the Mean-Shift technique, a second phase performs a cluster fusion, typically based on inter-cluster similarity evaluation which is based on the same set of features over their centroids. In order to avoid shadow influence, as described in section IV.2.b, the proposed algorithm considers the color vector angle in the fusion procedure.

The designed technique first searches for connected regions whose centroid color vectors yield a normalized scalar product close to 1 (i.e., are colinear). As the albedo ratio restrictions should be conserved, every pair of connected regions satisfying the angle restriction are further examined: first, the α proportionality factor between their respective centroids color vectors is estimated by dividing them; then, this factor is used to correct the illumination intensity influence; finally, the albedo ratio is re-computed and the same similarity criteria applied in the first phase is applied to merge or not the pair of connected regions.

At the end of this fusion or merging process an image segmentation into reflectance homogeneous regions almost independent of the illumination intensity is achieved. It is necessary to remark that the result might be poor in very dark umbra and very bright spark areas where color information is occluded by the lack or excess of light.

Results for the implemented Mean Shift approach can be found in Figure 2 and Figure 3, where some illustrative results show system performance in shadows and reflect elimination.

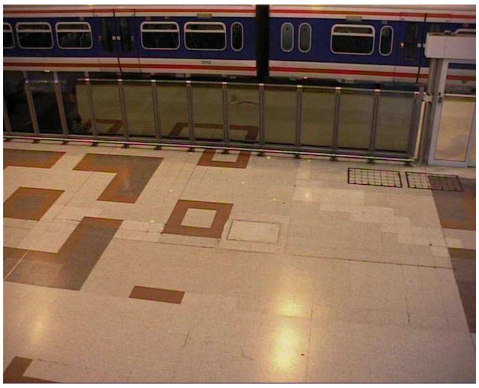
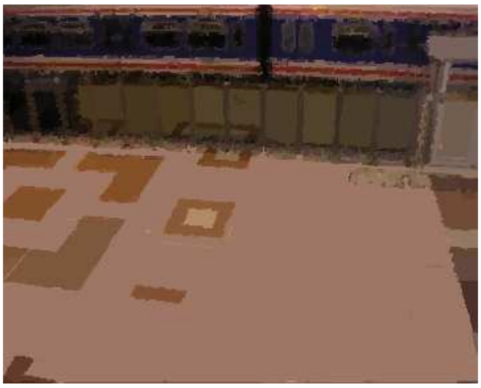
Original Frame	***Designed Mean-Shift Region segmentation***
	

Figure 2. ***Example of performance in avoiding reflects of designed Mean Shift approach***

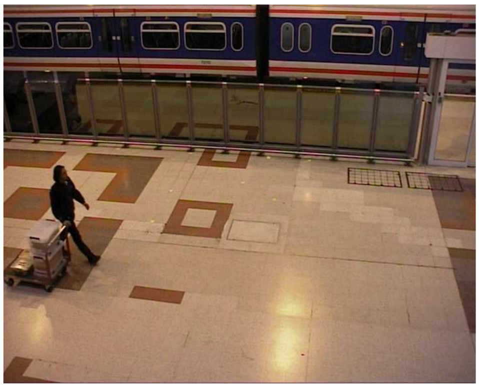
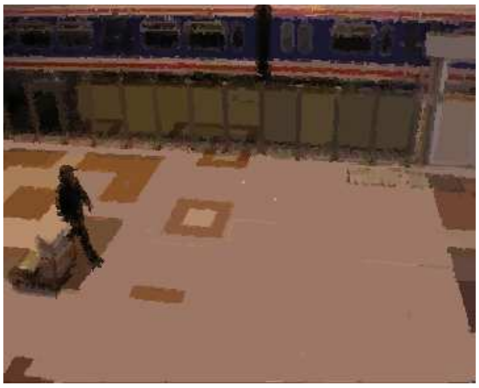
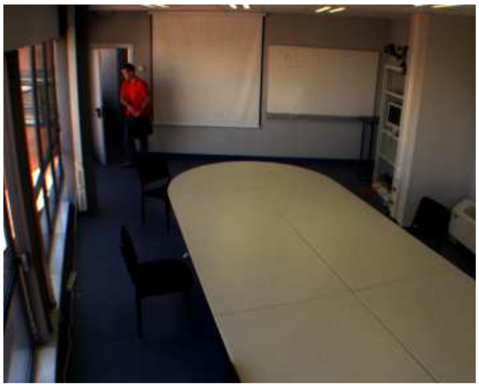
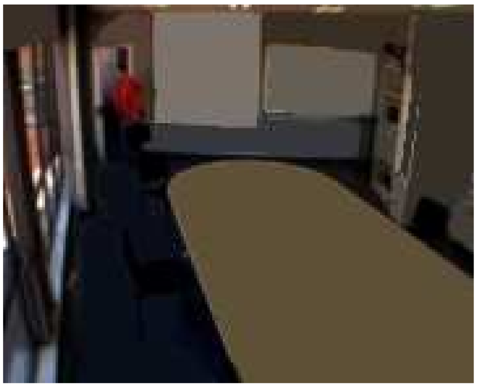
Original Frame	***Designed Mean-Shift Region segmentation***
	
	

Figure 3. ***Example of performance in avoiding shadows of designed Mean Shift approach***

V Chapter 5 – Background Modelling

The designed background modelling scheme tries to export classical pixel-level approaches to region-level analysis.

As explained in chapter III, the proposed scheme starts from a pixel-level segmentation mask; the aim is both to refine it and to account for a region-level description of such mask. Even though some of the inaccuracies in segmentation can be solved at region level, there are some problems that need to be considered in a higher semantic level, some of them in object level and others in scene level.

In brief, the implemented scheme can be described as a multilayer background model. The lowest level in the model contains a region based description of the most common background appearance. Upper background levels contain changes in the background over lowest layer, including moving background objects (as tree leaves moving by the wind), changes produced by reflectance properties of the objects in the background or inaccuracies of proposed region segmentation scheme, from this point in advance, we refer to these changes as background different region configurations. Changes modelled by this multilayer background model do not include those produced by foreground objects in the scene. Regions associated with those foreground changes are used to build and update the foreground model described in chapter V.

The following sections motivate the use of a multilayer scheme, explain and justify the use of the selected features to model each region at each layer, and present the structure to store those features. Finally, section V3 details the strategies used to discriminate background from foreground and those followed to update the multilayer background model.

1 Layer Motivation

The described Mean Shift region-segmentation scheme usually yields slightly different region distributions for every frame extracted from a sequence recorded by a fixed-camera. Variations are due to several factors. Specifically, there are three different sources of *problems*:

i. Unresolved illumination artifacts.
ii. Multimodal backgrounds.
iii. Variations of the proposed Mean-Shift scheme.

The influence of illumination sources has been diminished by the proposed Mean Shift scheme. However, incident light over reflective surfaces, as mirrors or crystal windows, sometimes results in unstable areas that change in color and position frame to frame. These areas can vary its location due to small camera vibrations, to object interactions and even to the nature of the light source. This results in different region configuration for a same scene background.

The presence of background moving objects as water, flames or tree leaves produces different regions configuration for every frame, which most pixel level approaches fail to model.

Finally, variations in region segmentation results owing to the threshold-based and bandwidth dependence nature of the implemented Mean Shift approach need to be also considered to design a robust background model.

Our proposal to overcome all these difficulties is to use a multilayer model, able to account for the different regions configuration. This scheme can cope with situations where a modelled region splits into several regions in the current background segmentation, and vice versa, when several modelled regions merge to one in the current frame segmentation.

Examples of multilayer background modelling are shown in Figure 4**.** Every row shows an instance of a background model, composed of three layers, resulting from different sources of variation. The model can be extended to more layers if the scene background nature demands it. Similarly, only one layer can be used if desired.

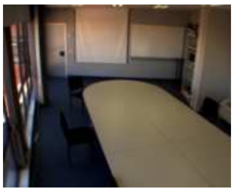
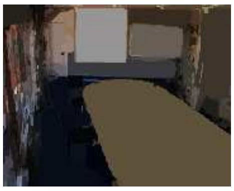
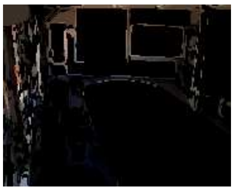
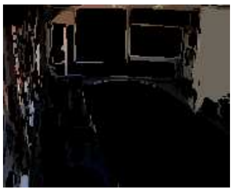
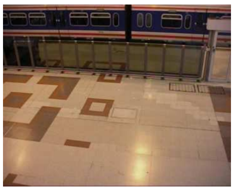
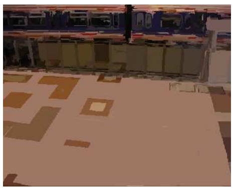
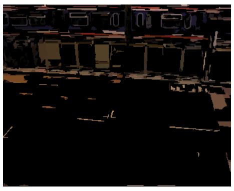
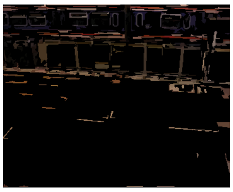
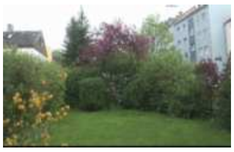
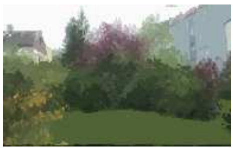
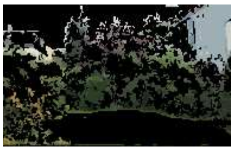
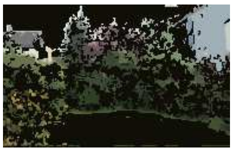
Original Frame	***Layer 1***	***Layer 2***	***Layer 3***	***Variation source***
				**i, iii**
				**i, iii**
				**ii, iii**

Figure 4. ***Multilayer background modelling***

Observing Layer 1, notice that only regions which do not match with those in this layer are assigned to Layer 2. Regions that match Layer 1 appear as black areas in Layer 2. This process can be successively extended to any number of layers.

In the first row, illumination artifacts over the window at the left side (reader point of view based) of the scene produces different region configurations that are stored in the different background layers. The same light artifacts are produced in the crystal panels placed in the upper part of the frame depicted in the second row.

Different regions configuration of the frame produced by typical elements from multimodal backgrounds are clearly shown in the third row of the Figure 4**.** Proposed background model handles these different region configurations by storing each in a different layer.

Variations in the region segmentation process are applicable to every row in the figure. This situation produces regions to split in several regions or fuse in one. Proposed background model store these regions merging and splitting processes in its different layers.

The aim of a background model is to assess for every incoming segmented frame which regions are considered background and which are considered foreground. If a new segmented region is declared a background region (in any of the layers), based on a similarity measure, such region parameters are updated in the background model. Consequently, only the first layer has to be fully partitioned in regions, being different regions configuration from stored in that layer those that make up the second layer and different regions to those in first and second layer the ones to constitute a third layer (and so on).

As can be expected, resources limit the number of layers. If a background region (so defined by the pixel-level segmentation) does not match any region in any layer, a new layer is created. This new layer replaces the *oldest* layer in the model, that is, the layer with lower recent update information. In order not to loose a whole frame in regions configuration, the first layer is not a candidate for replacement.

The implemented scheme provides the system with the capability of storing different configurations of a multimodal background even if pixel-level segmentation account for these situations, as, in example, when using a **MoG** (see section II.1.a.2) based approach. It is important to remark that system can handle these background changes only if its influence in pixel level segmentation approaches results in isolated pixels or in groups of pixels smaller than the region that encloses them at the initializing phase of the multilayer model.

2 Region Features and Region Similarity

Previous sections suggest the need for a region similarity measure. This section deepens into the region features involved in such measure. Selected features include:

- The RGB color vector (three values) of the region centroid, obtained from Mean shift region segmentation.
- The region size (one value).
- The set of color vectors angles between the considered region and each of its neighbouring regions (eight-connectivity, which results in eight values).

This results in a twelve values feature vector. Notice that some of these values are correlated, namely the RGB color vector values, owing to the nature of RGB color space (see **[TRC 1]**). Taking this under consideration and motivated by the popular work of **[TRC 12]** we present a temporal covariance scheme to measure similarity between feature vectors of two regions.

Each region in the background model is represented by a covariance matrix which includes its feature vector evolution along the time. Each position in this twelve-by-twelve matrix can be computed as:

$$\underset{R,L}{C}(i,j)=\sum_{t}^{t+\mathrm{T}}(f_{i,t}-\mu_i)\times(f_{j,t}-\mu_j)^T \qquad \textbf{(9)}$$

Where $f_{i,t}$ represent the value of feature *i* at frame *t* and μ_i is the mean value along time of feature *i* at region *R* and at layer *L*. That is, we compute the covariance of a set of temporal instances of a region, being each represented by a feature vector.

In order to obtain mean values, it is not efficient to store feature values from the start of the video till the current frame. Additionally, the updating solution proposed by **[TRC 12]** highly increases the computational cost of the algorithm because it is based on the extraction of the eigenvalues of each covariance matrix to the update of each position of the matrix in the Riemannian geometry.

We propose to define a sliding window scheme to compute covariance matrix just within the last $T-t+1$ frames. However, we also need to consider variations of the covariance matrix along the video to robustly model slow changes over the background (long time modelling).

The strategy used to model these slow background changes is based on accounting for the distance between covariance matrices of a region frame to frame and it is further explain in section V.3.c.

Construction of the covariance matrix is of main relevance in current work. In order to clarify the explained process

Figure 5 schematically depicts the computation of this matrix for a particular region ***R***.

According to Figure 5 first, region segmentation is performed over each frame in the sliding window (from frame *t* to frame *t+T*). Then, region ***R*** is isolated from the rest, and by a static region tracking strategy, explained in next section, matched with the stored representation of ***R*** in the model. Features of each frame ***R*** representation are then used to compute mean μ of each feature in the window. Means and features are then used to compute the covariance matrix of region ***R*** at frame *t+T* by using Equation (9).

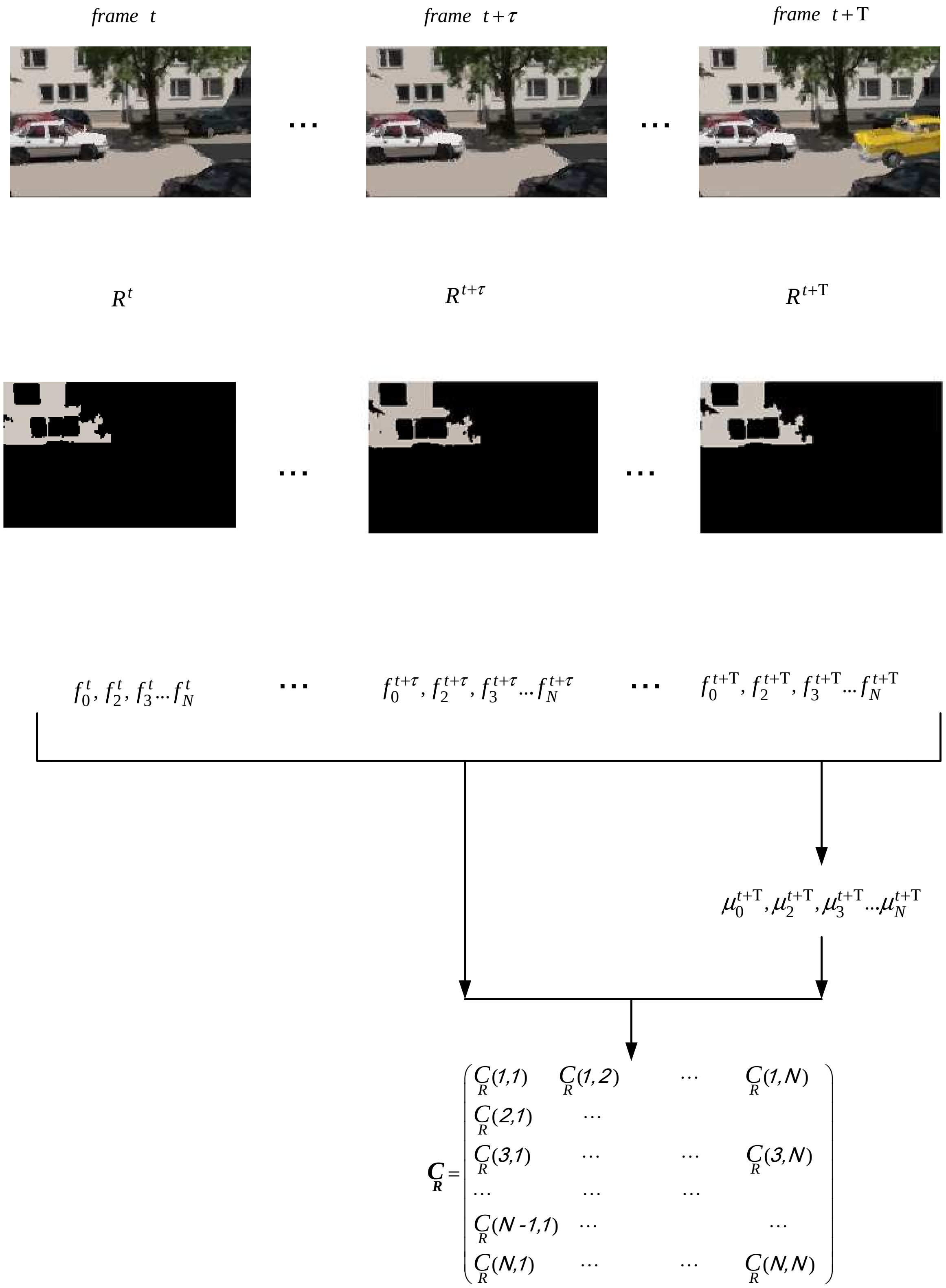

$frame\ \ t$
$frame\ \ t+\tau$
$frame\ \ t+\mathrm{T}$
…
…
R^{t}
$R^{t+\tau}$
$R^{t+\mathrm{T}}$
…
…
$f_0^t, f_2^t, f_3^t ... f_N^t$
…
$f_0^{t+\tau}, f_2^{t+\tau}, f_3^{t+\tau} ... f_N^{t+\tau}$
…
$f_0^{t+\mathrm{T}}, f_2^{t+\mathrm{T}}, f_3^{t+\mathrm{T}} ... f_N^{t+\mathrm{T}}$
$\mu_0^{t+\mathrm{T}}, \mu_2^{t+\mathrm{T}}, \mu_3^{t+\mathrm{T}} ... \mu_N^{t+\mathrm{T}}$
$\mathbf{C}_{\boldsymbol{R}} = \begin{pmatrix} C_R(1,1) & C_R(1,2) & \cdots & C_R(1,N) \\ C_R(2,1) & \cdots & & \\ C_R(3,1) & \cdots & \cdots & C_R(3,N) \\ \cdots & \cdots & \cdots & \\ C_R(N-1,1) & \cdots & & \cdots \\ C_R(N,1) & \cdots & \cdots & C_R(N,N) \end{pmatrix}$

Figure 5. ***Covariance Matrix Construction***

3 Modelling Scheme

This section tries to explain the two phases of the region matching process: selection of a search area and similarity searching. An explanation of the strategy designed to update the background concludes this section and this chapter.

V.3.a Selection of a search area

In environments where the camera is fixed, background can be assumed either to be almost static (if it is unimodal) or to present slight variation (if it has varying objects as explained when dealing with multimodal backgrounds). Multilayer background modelling can solve some of the problems derived from these different situations, and thus, if we assume that the number of background layers available is capable to model every possible background, we can consider that background regions are static at each layer.

As aforementioned, regions in each layer are characterized by a covariance matrix. As incoming frames are segmented, new regions should be matched to existing ones. Background regions are variable in size, and its shape can also vary frame to frame due to scene conditions and to the nature of the Mean shift implementation. Consequently, we need a way to robustly define the search area to find region matches for a new region.

The center of gravity of the region seems to be a good point to center the search area but its mathematical definition allows it to be out of the region. Mathematical morphology offers another possibility: to estimate it by computing the geodesic center, which is part of the region, via Symmetrical Ultimate Erosion. This consists in iteratively eroding a region until reaching an isolate point, which always belongs to the region under analysis. If the size of the set of points before the last erosion is lower than the active area of the structuring element, the whole region would be eroded. In order to avoid the region whole erosion there are two options: to decrease the size of the structure element or to choose one of the remaining points before the last erosion process.

An example of a Symmetrical Ultimate Erosion is depicted in Figure 6:

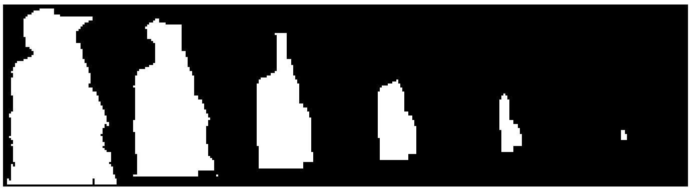


Figure 6. ***Ultimate Erosion***

Iterative erosion is a quite resource demanding operation. In order to perform this process in an efficient way, the distance transform of each region mask is performed. We keep the set of local minima of the transform; these are the geodesic centers candidates. Then, we should compute region Euclidean distances to every candidate and select the one that minimizes such distance. However, as for a search area we really do not need a perfect geodesic center extraction, we just choose the spatial medium point (from left to right and from up to down) of these potential candidates.

Two examples of geodesic center distribution after the explained estimation process are depicted in Figure 7. Observe that more textured areas (including foreground objects) in the frame are segmented in a high number of regions while homogenous areas are fused into a single region.

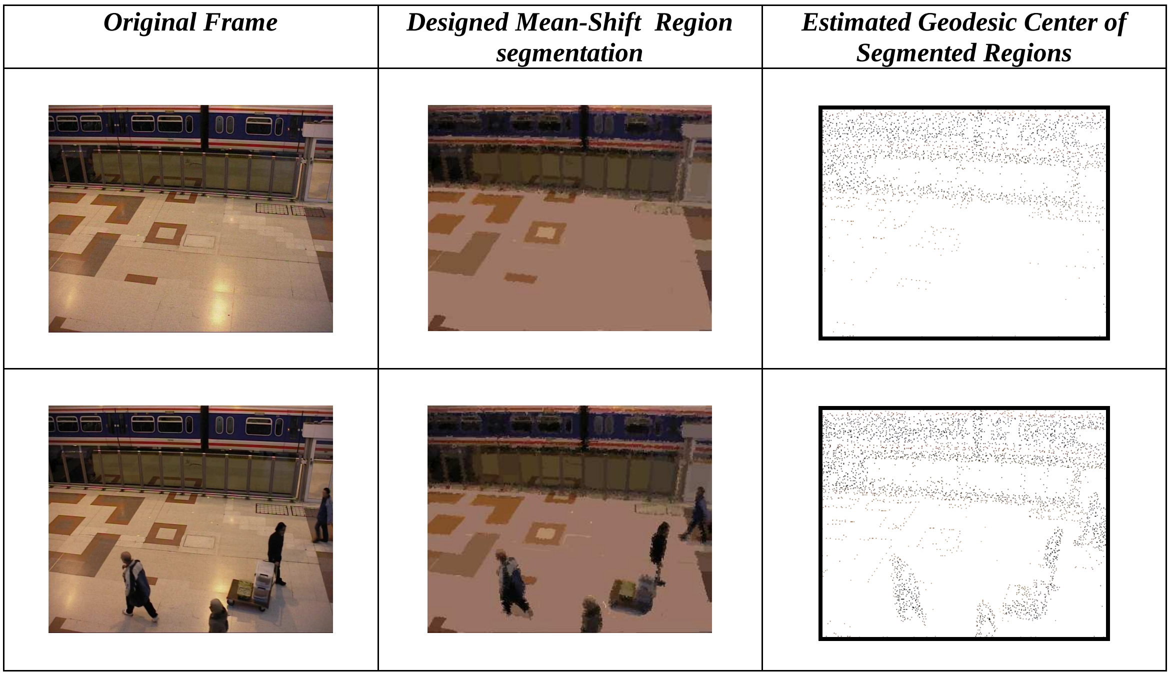

Original Frame
Designed Mean-Shift Region segmentation
Estimated Geodesic Center of Segmented Regions

Figure 7: ***Example of Geodesic Center Estimation***

Finally, after locating the new region via its estimated geodesic center, we define a circular search area with radio *r* around such point and search for matches among the set of regions in the model that overlap with the defined search area. The matching process is explained in the next section.

V.3.b Similarity searching

Once defined the set of candidate regions to match with a given one, we need to define the measure of similarity we are going to use.

Two alternative similarity measures are defined, depending of the representativeness of the covariance matrix associated to each region, which further depends on the number of region updates, which finally depends on the number of previous region matches.

In an initial phase, covariance matrices are not representative. Hence, we use the Euclidean distance between feature vectors to estimate similarity. As features are correlated, this measure is suboptimal.

After the initial phase (i.e., once a sufficient number on region matches results in a representative covariance matrix for regions under comparison), we use the cost of updating the covariance matrix to estimate similarity. Given the covariance matrix for a particular region *R* at layer *L* and frame *t*, and the covariance matrix of a candidate matching region at the next frame *t+1* we define the cost of updating the covariance matrix as the distance between these matrices, calculated following the work proposed by **[BM 1]** used as well by **[TRC 12]**:

$$d(\underset{R,L}{C^{t}},\underset{R,L}{C^{t+1}}) = \sqrt{\sum_{k=1}^{n} ln^2 \lambda_k (\underset{R,L}{C^{t}},\underset{R,L}{C^{t+1}})} \qquad \textbf{(10)}$$

, where $\lambda_k(\underset{R,L}{C^{t}},\underset{R,L}{C^{t+1}})$ are the generalized eigenvalues of matrices $\underset{R,L}{C^{t}}$ and $\underset{R,L}{C^{t+1}}$ computed from:

$$\lambda_k \underset{R,L}{C^{t}} x_k - \underset{R,L}{C^{t+1}} x_k = 0, \quad k = 1....n \qquad \textbf{(11)}$$

, where x_k are the generalized eigenvectors.

As expected, high distances are closely related with high differences in feature vectors and low distances with high similarity between regions. The use of this distance is only valid if matrices are positive defined, owing to the fact that its calculation requires to invert $\underset{R,L}{C^{t+1}}$. However, due to the temporal redundancy of a video, features of a region do not vary excessively from frame to frame, which results in a $\underset{R,L}{C^{t+1}}$ matrix which is likely to be just semi positive defined. Fortunately, we can compute the covariance matrix just with the features that make $\underset{R,L}{C^{t+1}}$ positive defined (i.e., those that are not identical in previous and current frame). This features removal, intrinsically results in diminishing the distance between covariance matrices (less values in the summation represented by Equation (10)), which is exactly what we were looking for: lower distances for similar feature configurations, but accounting for feature correlations.

The presented covariance-based distance is a key point of the proposed approach. Apart from, using it to match an incoming background region to a region in the background model, it will also be used to discriminate between background and foreground in incoming regions marked as potential foreground and to find matches between foreground regions for tracking as it is explained in chapter V.

Concluding, an incoming segmented region is matched to the most similar region in the model. Similarity can either be evaluated via the Euclidean distance (in absence of enough data) or via the cost of updating the covariance matrix of the region in model. Once matched, the covariance matrices distance between the incoming region and the matched region and the incoming region features are then used to update the region in the background model.

V.3.c Background model updating

Classical pixel-level segmentation approaches model pixel variation with a simple Gaussian. Using this distribution function, they try to be robust to impulsive noise added by the nature of the camera sensors (commonly categorized as measuring noise). Moreover, the use of multiple Gaussians allows these approaches to converge to every arbitrary distribution in the values of a pixel (if the number of Gaussian is big enough).

Exporting this idea to region-level segmentation, and considering that every error source but the impulsive noise influence has been compensated, we propose to model the values of the cost of updating the covariance matrix by a simple Gaussian. The use of multiple background layers can be understood as a similar (not exactly the same) approach that the MoG but at region level. Additionally, modelling the cost of updating the covariance matrices, allows to model somehow the region evolution, including both sudden and slow changes.

In this direction, we propose to use a Running Average scheme in which the mean (μ) and the standard deviation (σ) of the Gaussian modelling the region *R* at layer *L* are updated after a region-match according to the classical formulae:

$$\mu^{t}_{R,L} = \alpha\mu^{t-1}_{R,L} + (1-\alpha)d(C^{t}_{R,L}, C^{t+1}_{R,L}) \tag{12}$$

$$\sigma^{t}_{R,L} = \alpha\sigma^{t-1}_{R,L} + (1-\alpha)\left|\mu^{t}_{R,L} - d(C^{t}_{R,L}, C^{t+1}_{R,L})\right| \tag{13}$$

Summarizing, each background region is characterized by its geodesic center (which is able to change frame to frame), the background layer it belongs to, its feature vectors means, its temporal covariance matrix and by the Gaussian that model the cost of updating the covariance matrix.

Next chapter details the process followed to classify a region marked as potential foreground in the pixel-level segmentation as background or foreground, process that be call region discrimination. Furthermore, a tracking scheme of the foreground regions is also explained.

VI Chapter 6 – Foreground Modelling

This chapter describes the process follow to initialize and update a region based foreground model. This model is used to classify potential foreground regions as foreground regions (region discrimination) and to find similarities between foreground regions (region tracking).

1 Foreground model description

The foreground model designed in this Master thesis consists of a set of foreground regions characterized (similarly to background regions) by its geodesic center (which is able to change frame to frame), its feature vectors means, its temporal covariance matrix and by the Gaussian that model the cost of updating its covariance matrix.

System uses potential foreground mask to decide when create, update or reset check the foreground model. Foreground regions do not necessary remain static along the video, but they can appear, interact with the background and disappeared. Consequently foreground model needs to be created when an object (a set of regions) first enters in the scene, modified with each frame object characteristics, extended if more objects entered in the scene and reset when the objects walk out of the scene. Then the foreground model can enter in a sleep mode, waiting for new or previous detected objects entering in the scene.

Region discrimination proposed scheme allows us to create and initialize the foreground model. In turn, the foreground model allows the system to perform foreground region tracking. Next sections describe both process and connect then with changes in the foreground model.

2 Region Discrimination

Foreground regions are just searched in the area defined by the potential foreground mask. Regions that significantly overlap with this mask are considered potential foreground regions. Two hypotheses are formulated for each potential foreground region (RPF):

- "The region belongs to background" (H_0).
- "The region belongs to foreground" (H_1).

In order to check H_0, every potential foreground region undergoes the same process as an confirmed background region (described in chapter V); first geodesic center of the region is searched and static tracking candidates are set around this center (in a circular area defined by *r*), then we compute the cost of updating covariance matrix of each candidate region, getting temporal updated covariance matrix; $C^{t}_{RPF,L}$ and updating cost; $d(C^{t}_{RPF,L} ,C^{t+1}_{R,L})$.

If this cost falls inside the Gaussian that describes the evolution of this cost (see V.3.c) the hypothesis is temporary accepted, otherwise, it is neglected, That is:

$$\left| d(\underset{RPF,L}{C^t} , \underset{R,L}{C^{t+1}}) \right| \leq \underset{R,L}{\mu^t} + K \underset{R,L}{\sigma^t} \rightarrow H_0 \quad accepted \qquad \textbf{(14)}$$

, where *K* is a standard deviation factor.

Model region covariance matrix is not updated with RPF information in any case until final RPF discrimination has been performed.

At this point, a RPF can produce none, one or several temporal accepted H_0. If there are several temporal accepted hypotheses we just accept the one that produces a lower updating cost, this way reducing the possibilities of hypothesis state to temporal accepted and neglected.

If every H_0 is neglected and foreground model has not been initialized yet (thus H_1 can not be checked) or every possible H_1 is neglected or can not be computable, region is classified as a new foreground region and a region in foreground model is initialized with such region characteristics.

Definitive acceptance of a temporary accepted H_0 depends on the result obtained when formulating H_1. In order to check H_1 a similar process to H_0 formulation is followed. With the geodesic center extracted, we define a circular searching area of radio *R*, foreground model candidates are those of the stored foreground model (if any) which geodesic centers are contained in that area. To adapt to moving object displacements, this searching area is bigger than the static tracking searching area. Radios are in a proportion 3:1 (thus $R = 3 \times r$). If two or more H_1 are accepted for a RPF, we again temporary accept only the one that produces a lower updating cost.

If both H_0 and H_1 are temporary accepted, we neglect the hypothesis that produces a higher updating cost and finally used potential foreground region to update either foreground or background model, intrinsically classifying it as a foreground or background region.

With this hypothesis-based discrimination system, we are able to classify a RPF basing on its similarity to background and foreground models, while practically avoiding the use of thresholds (everywhere but in the concepts of region overlapping with the potential foreground mask and in the condition of being inside the Gaussian (standard deviation factor *K*) that describes the evolution of the cost of updating its covariance matrix).

3 Region Tracking and Foreground Model Updating

With the foreground model initialized, we are able to formulate H_1 for a particular RPF. In the case this hypothesis is accepted over H_0 we assign this RPF to foreground. Additionally, we can assign that RPF to the region in the model to which H_1 has been accepted, thus, matching these regions and performing a region based tracking.

According to previous section, we can summarize the operation structure of RPF discrimination and tracking process by means of a hypotheses-indexed table. Considering that, as explain in section VI2, there can be just four possible combinations of H_0 and H_1; H_0 accepted or H_0 neglected and H_1 accepted or H_1 neglected, this hypotheses-indexed table is depicted in Table 1:

	H_1 accepted	H_1 neglected
H_0 accepted	Background Region or **Tracked** Foreground Region (updating cost based)	Background Region or **New** Foreground Region (updating cost based)
H_0 neglected	**Tracked** Foreground Region	**New** Foreground Region

Table 1 Hypothesis based RPF discrimination and tracking processes

Foreground model updating scheme is equivalent to that performed in order to update the background. Consequently, both models share the same problems at the updating phase; covariance matrix needs to be representative enough to be used.

To overcome this initial lack of information, we first use the Euclidean distance to simulate the formulation of the hypothesis. This can be considered as a drawback of the algorithm and, therefore, we change to covariance based tracking as soon as region covariance matrix has been updated with the information of Euclidean based matching among three frames.

Finally, this region discrimination process is used to build a final pixel level segmentation mask by setting to ‘1’ every pixel belonging to either a tracked or a new region and to ‘0’ the pixels inside every new or static tracked background region. This mask is then use to feed pixel level segmentation by updating the pixel level background model only with the ‘0’ pixels in the final mask.

4 Objects tracking: extension to Connected-Component Tracking

Region level segmentation can be the base for object tracking, by just considering connected tracked regions. In this sensed, we have followed ideas proposed by **[FM 1]**. This work focuses in the context of segmentation in the H.264 compressed domain, but in our opinion, it is fully exportable to provide us a tracking approach robust to connected-component splitting, merging and occlusions.

Our approach and the one described in **[FM 1]** are very similar. However, while in the work developed in **[FM 1]** the segmentation unit is the macro-block, in our work it turns to be the region.

The process starts from a region based description of each connected-component. First, each connected-component is extracted from the final segmentation mask by performing a connected-component analysis. This results in a set of blobs, each defined by a mask that describes the shape and position of each connected-component present in a frame. Finally, we characterize each blob with the tracked and new foreground regions that overlap with its mask.

The process to export region tracking to connected-component tracking can be performed simply by checking the number of regions describing a connected-component in the current frame that have been tracked from regions describing a connected-component in the previous frame.

With this in mind, we (always following **[FM 1]**) build a so called Correspondence Matrix (CMM). In order to build this matrix we first need to compute three intermediate matrices; CM, CMR and CMC:

- CM has $M+1$ rows and $N+1$ columns, where M is the number of connected-components or blobs in the current frame and N the number of blobs in the previous frame. The extra row and the extra column represent the background area of each frame.

  Each position $CM(i,j)$ indicates the number of regions that simultaneously characterize blobs i and j in its respective frames (hence, these regions have been previously tracked for these two frames). Non-tracked regions add up to the background areas.

- CMR is defined as:

$$CMR(i,j) = \frac{CM(i,j)}{\sum_{k=0}^{N} CM(i,k)} \qquad \textbf{(15)}$$

  , so that each position of $CMR(i,j)$ indicates the proportion of regions from blob i in the current frame that are tracked from regions in blob j of the previous frame.

- CMC is defined as:

$$CMC(i,j) = \frac{CM(i,j)}{\sum_{k=0}^{M} CM(k,j)} \qquad \textbf{(16)}$$

, so that, similarly, each position of $CMC(i, j)$ indicates the proportion of regions from blob j in the previous frame that match with regions describing blob i in the current frame.

The highest value of CMR at each row i indicates the most correlated (from a region conformance perspective) blob in the previous frame. Similarly, the highest value of CMC at each column j in the previous frame indicates the most correlated blob extracted in the current frame.

Additionally, we have defined a set of tracking status for each blob. Letting *For* be a foreground blob and *Bck* a background one:

- New blob.(*0 →1 For*)
- One-to-one tracked blob. (*1 →1 For*)
- One-to-several tracked blob, splitting. (*1 →M For*)
- Several-to-one tracked blob, merging. (*M →1 For*)
- Several-to-several tracked blob. (*M →M For*)
- Disappeared blob. (*1 →0 For*)
- Frame background. (*Bck*)

With this simple scheme, we can just fill each position in CMM with a tracking status derived from the values of each position of CMR and CMC.

Specifically, we search the column j which maximizes i row in the matrix CMR and the row i that maximizes j column of the matrix CMC. We also search for non-zero positions (i, j) at each matrix.

From this point on, the proposed approach to fill matrix CMM differs to that explained in **[FM 1]**. We just take under consideration some of the possible combinations for each (i, j) at matrices CMR and CMC.

Table 2 presents the algorithm followed to fill CMM matrix as well as to define blobs status at each frame. Unconsidered combinations of positions (i, j) at matrices CMR and CMC are marked with an 'X' in the table.

Every combination (*Bck*, *Bck*) has not been considered due to the fact that static tracking has already been performed by the region background model updating procedure (see section V.3.c).

Dividing Table 2 in four quadrants, the up-left quadrant illustrates a classic one to one tracking, but allowing blob appearing and disappearing from the scene. Blobs that appear in the scene and are tracked one to one along it, preserve their identification until they disappear

The up-right quadrant handles blobs splitting; rules to identify split blobs should be fixed, based in final system application and system performance. However, it can be useful to provide each split blob a new identifier, but storing the identifier of the blob that splits.

The down-left quadrant represents another common situation of blobs tracking: when several connected-components merge into one. Even though identifiers assignation also depends on the context of application, in this case we assign to the merged blob i the identifier of the maximum j in the i row of the CMR.

Finally, the down-right quadrant deals with a multiple blob tracking. In our scheme each *M→M* tracking is treated as a multiple *1→1*.

Unconsidered (i, j) combinations in down-left and up/down-right quadrants are related to the progressive appearing and disappearing of blob i and blobs *M*, intrinsically considered in the region foreground discrimination strategy proposed in section VI2.

CMC CMR	i is the only non-zero value at column j			i is one of several non-zero values at column j		
j is the only non zero-value at row i		i is *Bck*	i is *For*		i is *Bck*	i is *For*
	j is *Bck*	X	(*0→1 For*)	j is *Bck*	X	X
	j is *For*	(*1→0 For*)	(*1→1 For*)	j is *For*	X	(*1→M For*)
j is one of several non zero-values at row i		i is *Bck*	i is *For*		i is *Bck*	i is *For*
	j is *Bck*	X	X	j is *Bck*	X	X
	j is *For*	X	(*M→1 For*)	j is *For*	X	(*M→M For*)

Table 2 Algorithm to fill CMM: Defining a blob status.

Although this blob tracking scheme has been implemented, its functionality has not been fully tested yet. Consequently, results of this blob tracking approach are not included in this Master Thesis documentation

Next chapter presents qualitative results to illustrate that the initial objectives have been fulfilled, as well as quantitative results over a set of Ground-truth sequences which compare our algorithm with one of the State of Art.

VII Chapter 7 – Results

This chapter presents final system segmentation results. First, qualitative results are shown by means of a set of frames extracted from real sequences that include light artifacts as shadows and light reflects. Ground-truth from these sequences is not available so quantitative results are computed with another set of sequences with publicly available ground-truth.

Results, both in the qualitative and in quantitative analysis are compared to those described in **[SEG 29]**, which is the initial pixel-level segmentation approach from which we start (see chapter III). The aim of this comparison is to show the improvements introduced both by our designed region based segmentation and by the feedback processing used to improve low level segmentation. From this point on, we refer to the work presented in **[SEG 29]** with the term: 'state of art approach'.

Presented and additional results are available at: http://www-vpu.ii.uam.es/~mev/

1 System configuration

To perform the evaluations, the system has been configured with the following parameters:

Minimum Region Size (MRS)	*5 / 3 pixels* *(Foreground absence/ Foreground presence)*
Background Searching Area (r)	*6 pixels*
Foreground Searching Area (R)	*18 pixels*
Initial Background Region Covariance Mean (μB):	*0*
Initial Background Region Covariance Deviation (σB):	*2*
Background Region Covariance Deviation Factor (K)	*1*
Initial Foreground Region Covariance Mean (μF):	*0*
Initial Foreground Region Covariance Deviation (σF):	*2*
Foreground Region Covariance Deviation Factor (K):	*3*

Table 3 System initialization parameters.

MRS is used during Mean-Shift segmentation to prune (angle distance based IV.2.b) very small regions. Foreground regions are usually smaller and more textured than background. Consequently, smaller regions can be of high significance in foreground areas, and so, we are more permissive in the minimum size restriction.

Background and Foreground searching areas are defined to select region candidates to which perform static background region tracking, and foreground region tracking; their utility is explained in sections V.3.a and VI2 respectively.

Finally, initial foreground and background covariance mean and covariance standard deviation are set at region initialization in the model. Foreground covariance models are used with a lower initialization time than background covariance. This situation requires a higher flexibility in the matching of foreground regions; thus, the deviation factor and the initial standard deviation are higher for the foreground model.

2 Qualitative Results

The aim of this section is to show the advantages of using our region-based segmentation system. These advantages are essentially a refinement of the final mask, in several aspects; shadows partial or complete elimination, boundary refinements and internal holes filling.

► First sequence.

Name: C0_UPC.avi

Description:

Static view of a video-intelligence room. Five people sequentially enter in the scene and shake their hands with each other. Four of them sit down in four chairs, already in the scene when the video started. The other one simulates to be a lecturer.

Complexity Factors:

I. Several objects interact in a real video; they produce shadows over the floor and the wall.

II. Scene is illuminated by fluorescents, their influence results in reflectance areas distributed along the frame.

III. Two of the people present in the video wear clothes with colors similar to some areas in the background.

IV. People interact among them, so that precise segmentation masks are needed to correctly detect those interactions in a higher semantic level.

Illustrative frames:

System performance in shadows elimination can be observed in every frame presented in Figure 8. Additionally, boundary refinement is clearly shown in frames 170 and 473. However, in frame 763, even most of the objects extracted boundaries are more adjusted to real ones, there are some background areas that are added to the foreground (as those surrounding the girl sitting in left side of the scene). The point is that these areas are whole regions that suffer iterative activation and deactivation along the video, consequently tracking scheme process described in VI4 seems to be a powerful tool to eliminate this system inaccuracies.

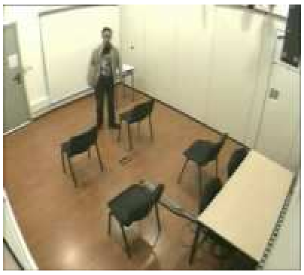
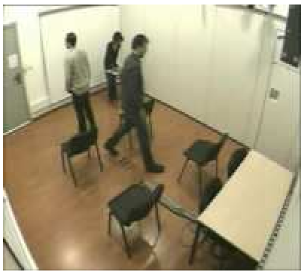
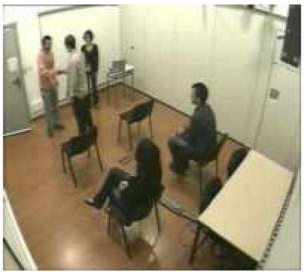
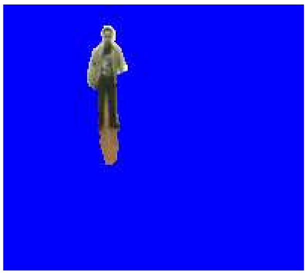
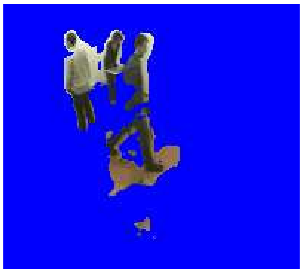
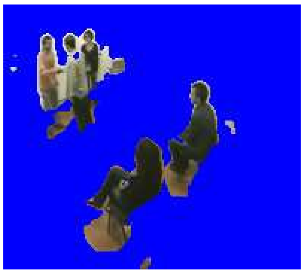
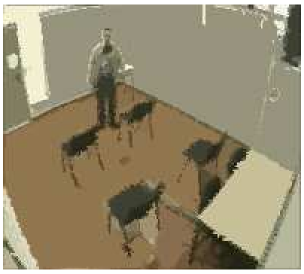
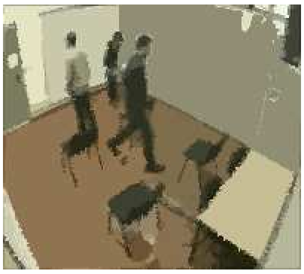
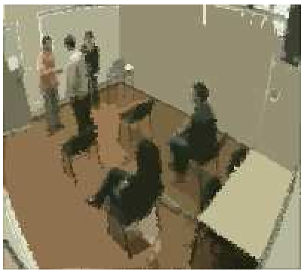
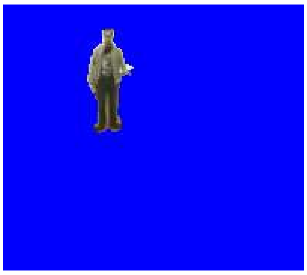
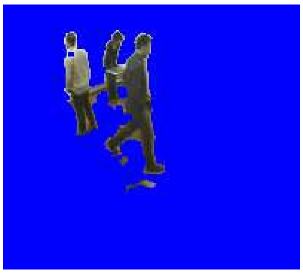
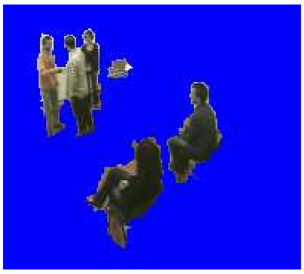
Frame Number	***170***	***473***	***763***
Original Frame			
State of Art approach			
Mean Shift Region Segmentation			
Implemented Approach			

Figure 8. ***Some illustrative frames from region based segmentation of C0_UPC .avi***

► Second sequence.

Name: PETS06_S7-T6-B_3_abandoned_object_4cif.avi

Description:

Static view of a train station hall where people walk trough, stop, occludes other people, leave objects unattended and run. Available at: pets2006.net/

Complexity Factors:

I. Several objects interact in a real video; they produce long shadows over the floor and the wall.

II. Scene is illuminated by fluorescents, their influence results in reflectance areas distributed along the frame.

III. A person remains static for a long time, enough to be considered as background if any tracking system is used.

IV. People interact among them, precise pixels segmentation masks are needed to correctly detect those interactions in a higher semantic level.

Illustrative frames:

System performance in shadows elimination is again depicted in Figure 9. Additionally, objects boundary refinement and hole-filling is carried out in frames 264 and 314. Reflects produced by crystal panels are eliminated in frame 469, but they are not in fame 314, probably background model has not been robustly updated with this new region configuration yet.

Observe the grey region appearing at the down-right corner of the frame 314. Ideally, this region should be merged with the rest of the floor, but as explained in section IV.3.b, designed Mean Shift segmentation can not handle very umbra-specially dark-areas where reflectance information has been strongly occluded by light absence.

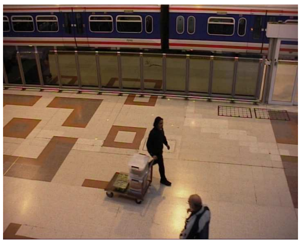
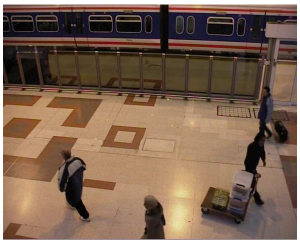
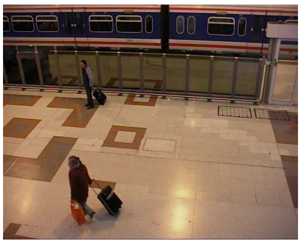
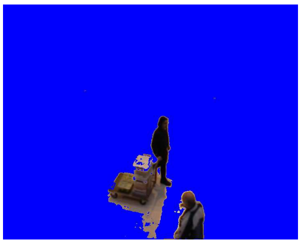
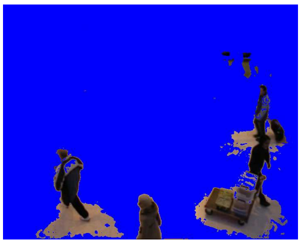
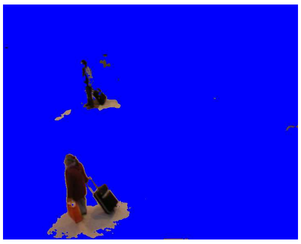
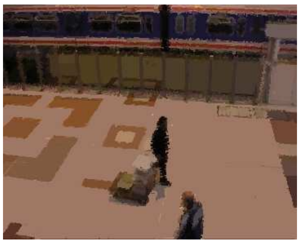
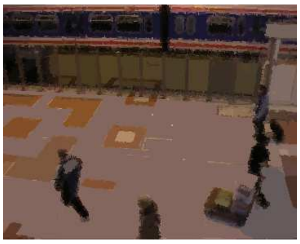
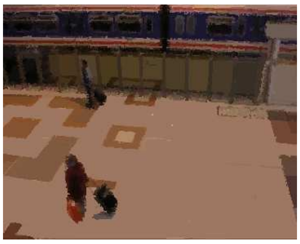
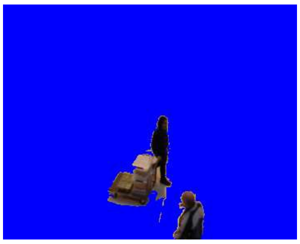
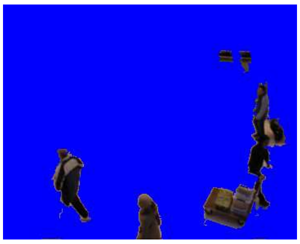
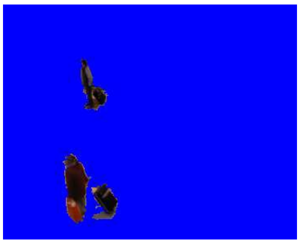
Frame Number	***264***	***314***	***469***
Original Frame			
State of Art approach			
Mean Shift Region Segmentation			
Implemented Approach			

Figure 9. *Frames from region based segmentation of PETS06_S7-T6-B_3.avi*

3 Quantitative Results

There are several quality parameters to measure the performance of an element binary classification system, where a class is consider as positive ('1') and the other class as negative ('0'). However, most of them are ratios of four parameters that can be extracted from an element wise comparison between obtained classification and a 'perfect classification'. These parameters are; the number of elements correctly classified for each class (also called true positives TP and true negatives TN) and the number of elements incorrectly classified as members of the other class (false positives FP and false negatives FN).

Dealing with video segmentation, the elements to classify are the pixels, the classes are foreground ('1') and background ('0'), and the perfect classification is usually called a Ground-truth.

The different measures we are going to evaluate for each video are the following:

- Sensitivity **[RLT 1]** or true positive rate (*S*). Measures the proportion of existing positive elements that are correctly identified as such. It can be computed by the formulae:

$$S = \frac{TP}{TP + FN} \tag{17}$$

, sometimes it is also called positives Recall rate. In pixel-level video segmentation, sensitivity represents the proportion of total foreground pixels correctly segmented.

- Specificity **[RLT 1]** or true negatives rate (*E*). Measures the proportion of existing negative elements that are correctly identify as negative by the classification system. It is common to compute *1-E*, that respond to the formulae:

$$1\text{-}E = 1 - \frac{TN}{TN + FP} \tag{18}$$

, sometimes *E* is also called negatives Recall rate. In pixel-level video segmentation, specificity represents the number of total background pixels correctly discriminated as such background.

- Positive precision **[RLT 2]** (*P_F*). Measures the proportion of elements classified as positives that are correctly identified as such. In the context of our problem, the number of pixels classified as foreground that are really part of the foreground:

$$P\_F = 1 - \frac{TP}{TP + FP} \tag{19}$$

- Negative precision **[RLT 2]** (*P_B*). Measures the proportion of elements correctly classified as negative from the total of elements identified as members of this class. That is, the proportion of pixels classified as background that are not foreground.

$$P\_B = 1 - \frac{TN}{TN + FN} \tag{20}$$

- Positive F1 Score **[RLT 3]** (*FS_F*). Is a measure that combines *P_F* and *S* to provide a global idea of the system performance in detecting positive elements, in our context, foreground pixel. Positive F1 Score best value is 1 and worst 0. We use the traditional definition of F1 score, that is:

$$FS\_F = 2 \times \left(\frac{P\_F \times S}{P\_F + S} \right) \tag{21}$$

- Negative F1 Score **[RLT 3]** (*FS_B*). Can be considered as measure that combines *P_N* and *E* to provide a global idea of the system performance when detecting negative elements. Negative F1 Score best value is 1 and worst 0. We have used again the traditional definition of F1 score, that is:

$$FS\_B = 2 \times \left(\frac{P\_N \times E}{P\_N + E} \right) \tag{22}$$

Tested videos with ground-truth have been extracted from the public databases available at **[RLT 4].** These videos have been artificially generated and thus they are shadow free; so comparison has just been made in a sequence where multimodality was present, in order to show the effectiveness of our multilayer background modelling.

► Third sequence.

Name: VSSN06-video4

Description:

The video presents a fixed scene of a house yard with a high amount of vegetation which leaves and flowers are constantly moving with the wind.

Complexity Factors:

I. Background multimodality

Illustrative frames:

Basically, results show that our algorithm is robust to the presence of moving elements typical form multimodality backgrounds, and State of the art algorithm is not.

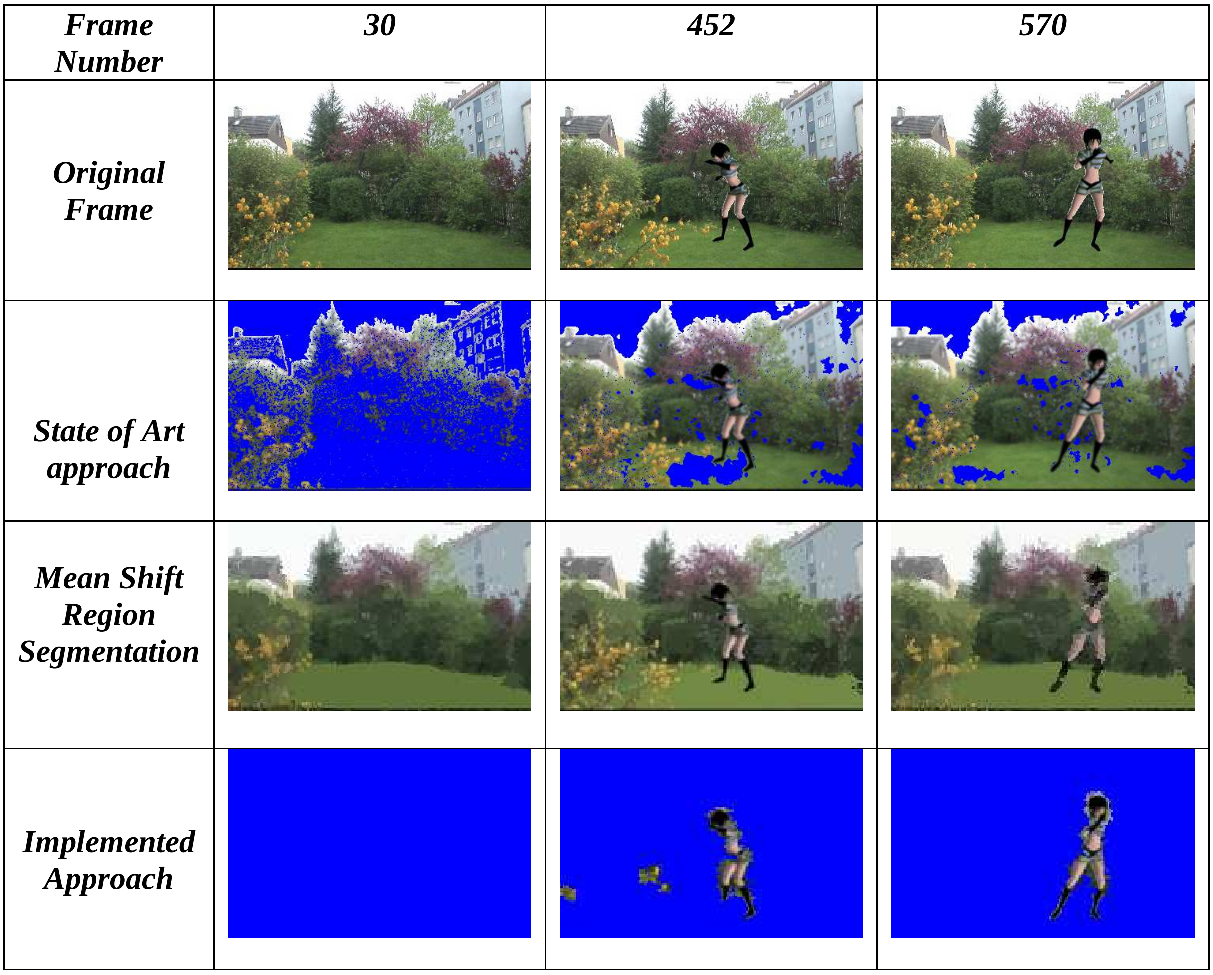


Figure 10. *Frames from region based segmentation of VSSN06-video4.avi*

Quantitative evaluation:

Quantitative Measure	*S*	*1-E*	*P_B*	*P_F*	*FS_B*	*FS_F*	*FS*
State of Art approach	6.03 %	40.67 %	98.69 %	0.377 %	0.007	0.734	0.741
Implemented Approach	77.0 %	0.89 %	99.41 %	68.73 %	0.993	0.7263	1.719

Table 4 Quantitative evaluation of VSSN06-video4.avi

Although our algorithm requires a higher initialization time to get robust foreground models, Figures Figure 11, Figure 12, Figure 13 and Figure 14 show that, its performance is better than the state-of-the-art approach for every quantitative measure computed.

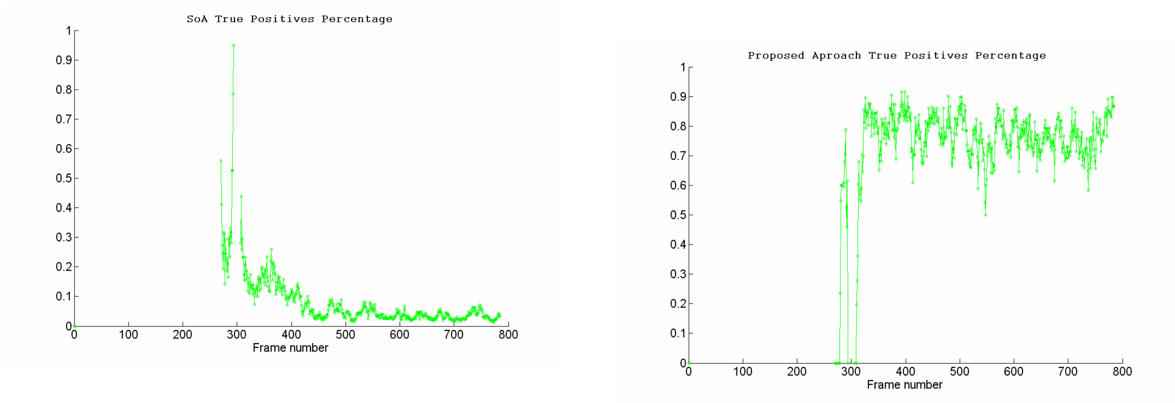


Figure 11. *True Positives percentage comparison (VSSN06-video4.avi)*

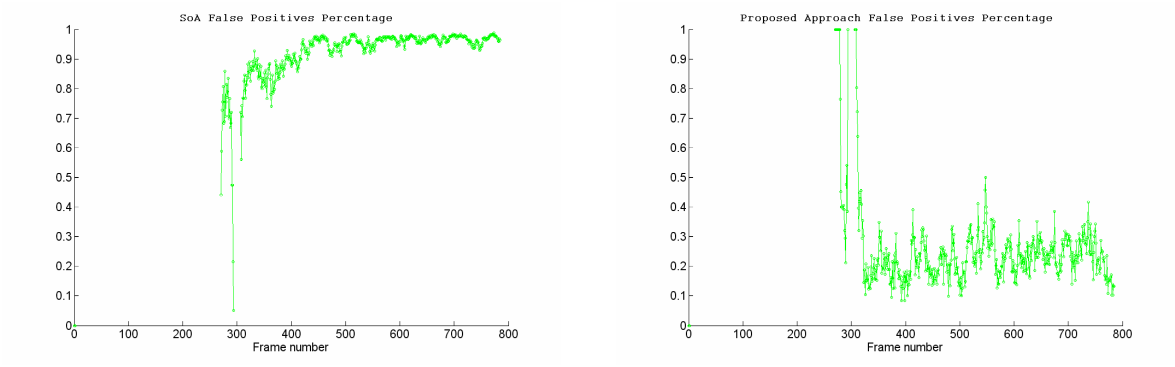


Figure 12. *False Positives percentage comparison (VSSN06-video4.avi)*

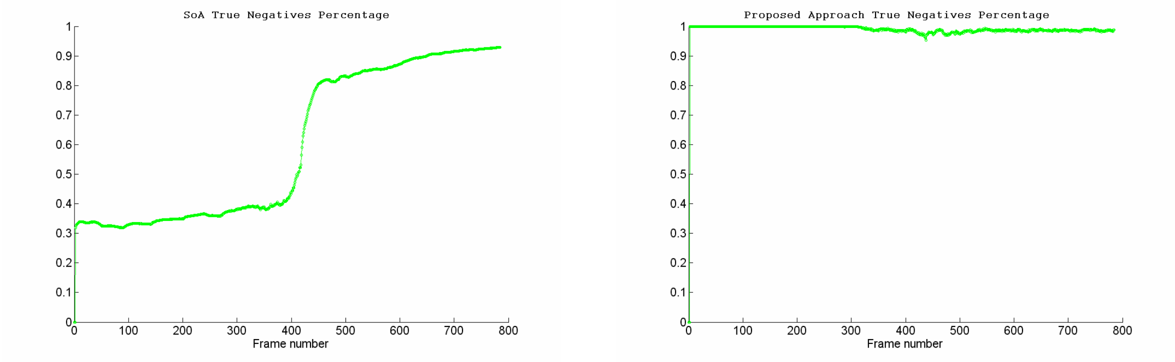


Figure 13. *True Negatives percentage comparison (VSSN06-video4.avi)*

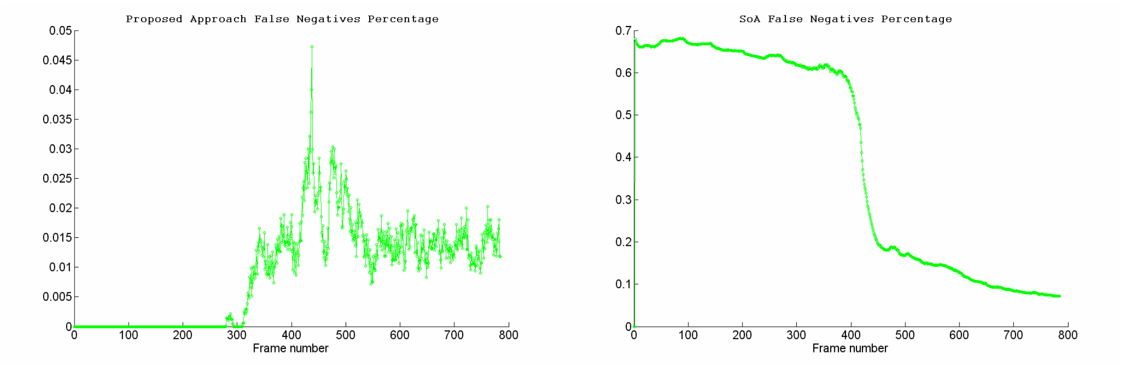


Figure 14. *False Negatives percentage comparison (VSSN06-video4.avi)*

VIII Chapter 8- Future Work and Conclusions

This chapter introduces future work required or suggested over the developed system. To organize it, we will divide future work in two areas, system opportunities and system limitations.

Systems opportunities refer to future work that may improve results, or make use of the presented system nature and results to perform video analysis at a higher semantic level. On the other hand, system limitations refers to developments that should be undertaken as soon as possible to allow video analysis systems to make use of this work and its advantages in real video analysis environments.

1 System opportunities

The most relevant future work required in this area is the use of the connected-component tracking described in section VI4 to improve the segmentation mask and, by extension, the system performance. Preliminary results of this module indicate that its integration in the system can refine final results and enhance system robustness.

Each region in the model is well characterized, and, at the same time, a set of those regions characterize each connected-component in the scene. Region characterization is quite flexible in the presented approach, so that alternative or complementary features can be used to model the scene without changing the system nature and philosophy. It would be of main interest to include textured based features in the region characterization.

Once we account for region and connected-component characterization, there is a wide range of extensions to the developed work. Modelled regions can be used to identify objects, which are, essentially, characterized connected-components. Obtained object based descriptions are potentially robust to perform typical high level analysis tasks which include, but are not limited to, object recognition, human detection, object-human interactivity and human activity recognition.

2 System limitations

Chapter VII does not provide indications on the system temporal efficiency or on computational costs, as the presented implementation is quite resource demanding (reaching several seconds per frame in the worst case), which is the main limitation of the system.

Even though a lot of effort has been made in code optimization, there are several bottle neck processes in the system which are consuming most of the processing time.

Heaviest processes relate to operations that require image inspection at pixel level, instead of at region level. These are operations are just those involved in region characterization:

i. Mean-Shift segmentation.

ii. Geodesic Center Estimation.

iii. Region neighbours searching.

Mean-Shift iteratively searches for convergence inside a window, centred in a particular pixel, and checks every pixel status in its clustering phase. Geodesic center is also a pixel, and iterative erosion process; even computed with the distance transform is a very heavy process if we compute it in highly textured areas where many regions are segmented. Finally, region neighbours searching requires also the extraction of the furthest pixel in the region at each searching angle.

These three processes are performed independently among them, consuming, in foreground absence, more than 90 % of the computational time. However, an approximated estimation of the geodesic center position and identifiers for each of the 8-connected neighbouring regions are directly available at Mean-Shift segmentation process core. Thus, region characterization can be done in a faster way.

Furthermore, as this process does not require any result from previous frame region segmentation, it can be performed in parallel to the region discrimination system explained in chapters V and VI.

3 *Conclusions*

We have designed, implemented and presented an automatic region-based segmentation system for video sequences recorded by fixed cameras. First a robust low level segmentation approach is used to identify potential foreground areas; in parallel, a region segmentation process robust to light artefacts is performed via a new Mean-shift implementation. The region segmentation and the pixel-level segmentation mask is combined to achieve region level analysis.

In foreground absence, confirmed background regions are used to build a background model characterized with the covariance matrix of each region accumulated features. Once the model is built, candidate background regions are assessed by matching them to regions in the modelled background, via a static region tracking driven by the evolution of the covariance matrix.

Foreground is extracted from potential foreground and discriminated from background by executing a hypothesis test. This test the hypothesis of a potential foreground region belonging to either the foreground or the background region model, and is also driven by modelled region covariance matrix evolution. Indirectly, this test performs a region tracking.

Region level segmentation results are used to update the background model of the pixel-level segmentation approach, thus performing an up to down information feedback scheme.

Finally, an approach to extend this region tracking to classical connected component tracking has been designed, inspired in state of art, implemented but not fully tested.

Concluding, we have designed and implemented an automatic and innovative region segmentation technique, we have robustly categorized and tracked such segmented regions and we have used final results to feed pixel-level segmentation. Results show that the combination of these processes results in a better segmentation mask, and that the feedback process efficiently improves pixel-level segmentation approach by avoiding the influence of this failure factors in background updating strategy. Consequently, we have fulfilled every initial objective.

System performance in segmenting objects seems to be, in the light of included results, better than state of art approach, as state of art approach is a combination of classical pixel segmentation techniques and its segmentation results are showed to be better **[SEG 29]** than them, we can transitively derive that presented system performance is better than classical pixel-level algorithm. However, an exhaustive studio with a higher amount of sequences is required to fully support this initial observations.

Finally, the main drawback of the system is its high processing time per frame, which turn its integration into a higher semantic analysis system inapplicable at present day; we would focus future work in this line, and with the aim to avoid this limitation as soon as possible.

References

[I 1] Smeulders AWM, Worring M, Santini S, Gupta A, Jain R. "Content-Based Image Retrieval at the End of the Early Years". IEEE Transitions on Pattern Analysis and Machine Intelligence. vol.22 (12): pp 1349-80. 2000

[SEG 1] **"**Coding of Moving Pictures and Associated Audio for Digital Storage Media and up to About 1.5 Mbit/s- Part 2: Video*"*, ISO/IEC 1172-2 (MPEG 1 Video) ISO/IEC JTC 1, Mar 1993

[SEG 2] "Generic Coding of Moving Pictures and Associated Audio information- Part 2: Video", ITU-T rec.H.262 and ISO/IEC 13818-2 (MPEG-2 Video), ITU-T and ISO/IEC JTC, Nov 1994

[SEG 3] "*Coding* of Audiovisual Objects Part 2-Visual", ISO/IEC JTC1 ISO/IEC 14492-2, (MPEG-4 Visual), Version 1: Apr. 1999, Version 2. Feb 2000. Version 3 May 2004.

[SEG 4] H. Eng, K. Ma, "Spatio-temporal Segmentation of Moving Video Objects over MPEG Compressed Domain", IEEE 2000

[SEG 5] M. L. Jamrozik, M. H. Hayes, "A Compressed Domain Video Object Segmentation System", IEEE ICIP, 2002

[SEG 6] S. Beucher and C.Lantuéjoul. "Use of watersheds in contour detection". International workshop on image processing, real-time edge and motion detection (1979).

[SEG 7] R. Venkatesh, K. Ramakrishnan, S. Srinivasan , "Video Object Segmentation: A Compressed Domain Approach", IEEE Transitions Circuits System Video Technology., April 2004

[SEG 8] V. Mezaris, I. Kompatsiaris, N. Boulgouris, M. Strintzis, "Real-Time Compressed-Domain Spatiotemporal Segmentation and Ontologies for Video Indexing andRetrieval", IEEE Trans. Circuits System Video Technology, May 2004

[SEG 9] M. Escudero, F. Tiburzi, J. Bescós: MPEG video object segmentation under camera motion and multimodal backgrounds. ICIP 2008: 2668-2671

[SEG 10] M. Piccardi, *Background subtraction techniques: a review*, in Proc. of IEEE SMC 2004 International Conference on Systems, Man and Cybernetics, The Hague, The Netherlands, October 2004.

[SEG 11] C. Wren, A. Azarhayejani, T. Darrell, and A.P. Pentland, "Pfinder: real-time tracking of the human body," IEEE Transitions on Pattern Analysis and Machine Intelligence., vol. 19, no. 7, pp. 780-785, 1997.

[SEG 12] B.P.L. Lo and S.A. Velastin, "Automatic congestion detection system for underground platforms," Proc. ISIMP2001, pp. 158-161, May2001.

[SEG 13] R. Cucchiara, C. Grana, M. Piccardi, and A. Prati, “Detecting moving objects, ghosts, and shadows in video streams,” IEEE Trans. on Pattern Analysis and Machine Intelligence, vol. 25, no. 10, pp. 1337-1442,2003.

[SEG 14] C. Stauffer and W.E.L. Grimson, “Adaptive background *mixture* models for real-time tracking,” Proc. IEEE CVPR 1999, pp. 24&252, June 1999.

[SEG 15] A. Elgammal, D. Hanvood, and L.S. Davis, “Nonparametric model for background subtraction,” Proc. ECCV 2000, pp. 751-767, June 2000.

[SEG 16] Herrero, S. Bescós, J. “Background Subtraction Techniques: Systematic Evaluation and Comparative Analysis”. ACIVS 2009: pp. 33-42

[SEG 17] M. Seki, T. Wada, H. Fujiwara, and K. Sumi, “Background subtraction based on cooccurrence of image variations,” Proc. CVPR 2003, Vol. 2, pp. 65-72, 2003.

[SEG 18] N.M. Oliver, B. Rosario, and A.P. Pentland, “A Bayesian computer vision system for modelling human interactions,” IEEE Trans. on Pattern Analysis and Machine Intelligence, vol. 22, no. 8, pp. 831-843,2000.

[SEG 19] Porikli, F.; Tuzel, O., "Bayesian Background Modelling for Foreground Detection", ACM International Workshop on Video Surveillance and Sensor Networks (VSSN), ISBN: 1-59593-242-9, pp. 55-28, November 2005

[SEG 20] T. Horpraset, D. Harwood, and L. Davis, “A statistical approach for real-time robust background subtraction and shadow detection,” in *Proc. Int. Conf. Comput. Vis.*, 1999, pp. 1–19.

[SEG 21] J.L. Landabaso, M. Pardàs. A Unified Framework for Consistent 2D/3D Foreground Object Detection. Circuits and Systems for Video Technology, Special Issue on Video Surveillance, August 2008.

[SEG 22] J. Sun, W. Zhang, X. Tang, H. Y. Shum, “Background cut,” in Proc. ECCV, 2006.

[SEG 23] F. Moscheni, S. Bhattacharjee, "Robust region merging for spatio-temporal segmentation", Proceedings of IEEE International Conference on Image Processing, vol. 1, 1996. p. 501–4.

[SEG 24] Ying-Li Tian, M. Lu, A. Hampapur, "Robust and efficient foreground analysis for real-time video surveillance," Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on, vol.1, no.pp. 1182- 1187 vol. 1, 20-25 June 2005

[SEG 25] M. Harville, "A framework for high-level feedback to adaptive, per pixel, mixture-of-Gaussian background models". In ECCV, page III: 543 ff., Copenhagen, Denmark, May 2002.

[SEG 26] S. Huwer, H. Niemann, "Adaptive Change Detection for Real-Time Surveillance Applications". IEEE International Workshop on Visual Surveillance, 2000.

[SEG 27] S.M. Desa, Q.A. Salih, "Image Subtraction for Real Time Moving Object Extraction". Proc. Intl. Conf. on Computer Graphics, Imaging and Visualization, CGIV'04. Jul. 2004.

[SEG 28] L. Li, W. Huang, I.Y.H. Gu, Q. Tia, "Foreground object detection from videos containing complex background". Proc. ACM Intl. Conf. on Multimedia, Nov. 2002

[SEG 29] Alvaro García, Jesús Bescós: "Video Object Segmentation Based on Feedback Schemes Guided by a Low-Level Scene Ontology". ACIVS 2008: 322-333

[RSEG 1] W. Skarbek, A. Koschan, "Colour image segmentation – a survey", Technical Report, Tech. Univ. of Berlin, October 1994.

[RSEG 2] Fukada, Y., "Spatial clustering procedures for region analysis". Pattern Recognition vol. 12, pp.395–403, 1980.

[RSEG 3] Chen, P., Pavlidis, T., "Image segmentation as an estimation problem". Computer Graphics and Image Processing vol. 12, pp. 153–172. 1980.

[RSEG 4] Schettini R. "A segmentation algorithm for colour images" Pattern Recognition Letters, vol. 14, pp. 499-506. 1993.

[RSEG 5] Tseng D.-C. and Chang C.-H. "Color segmentation using perceptual attributes", Proc. 11th International Conference on Pattern Recognition, Den Hague, 1992

[RSEG 6] Zucker, S. "Region growing: Childhood and adolescence". Computer Graphics and Image Processing, vol.5, pp. 382–399. 1976

[RSEG 7] Adams, R., Bischof, L., "Seeded region growing". IEEE Transactions on Pattern Analysis and Machine Intelligence 16 (6), 641–647.1994

[RSEG 8] Ferri F. and Vidal E. "Colour image segmentation and labelling through multiedit-condensing", Pattern Recognition Letters, vol. 13, no. 8, pp. 561-568. 1992

[RSEG 9] Umbaugh S.E., Moss R.H., Stoecker W.V., and Hance G.A. "Automatic colour segmentation algorithms with application to skin tumor feature identification", IEEE Engineering in Medicine and Biology, vol. 12, no. 3, pp. 75-82. 1993

[RSEG 10] Celenk M. (1988) "A recursive clustering technique for colour picture segmentation", Proc. International Conference on Computer Vision and Pattern Recognition, pp. 437-444,5-9 Ann Arbor, MI, USA, June 1988.

[RSEG 11] D. Comaniciu and P. Meer. "Mean shift: A robust approach toward feature space analysis". IEEE Transitions on Pattern Analysis and Machine Intelligence. 24(5):603–619, 2002.

[RSEG 12] Huntsberger T.L., Jacobs C.L., and Cannon R.L. "Iterative fuzzy image segmentation", Pattern Recognition, vol. 18, no. 2, pp. 131-138. 1985

[RSEG 13] Pham, D., Prince, J. “An adaptive fuzzy c-means algorithm for image segmentation in the presence of intensity in homogeneities”. Pattern Recognition Letters 20 (1), 57–68. 1999

[RSEG 14] Wu, Z. and Leahy, R. “An optimal graph theoretic approach to data clustering: Theory and its applications to image segmentation”. IEEE Transitions on Pattern Analysis and Machine Intelligence, vol. 11, pp. 1101–1113. 1993

[RSEG 15] Shi, J. and Malik, J. “Normalized cuts and image segmentation”. IEEE Transitions on Pattern Analysis and Machine Intelligence. Vol. 22, 8, pp. 888–905. 2000

[RSEG 16] Meyer, F., Beucher, S. “Morphological segmentation”. Journal of Visual Communication and Image Representation vol.1 (1), pp. 21–46. 1990

[RSEG 17] Vincent, L., Soille, P. “Watersheds in digital spaces: An efficient algorithm based on immersion simulations”. IEEE Transactions on Pattern Analysis and Machine Intelligence 13(6), 583–589. 1991

[RSEG 18] Muñoz, X., Freixenet, J., Cufí, X., Martí, J.: “Strategies for image segmentation combining region and boundary information”. Pattern Recognition Letters, vol. 24, pp. 375–392. 2003

[RSEG 19] Kass, M.,Witkin, A., and Terzopoulos, D. “Snakes: Active contour Models”. In Proc. 1st International Conference on Computer Vision, pp. 259–268. 1987.

[RSEG 20] C. Xu and J.L. Prince, “Snakes, Shapes, and gradient vector flow”, IEEE Transactions on Image Processing, vol **7** (3) , pp. 359–369. 1998

[RSEG 21] S.J. Osher and J.A. Sethian, “Fronts propagation with curvature dependent speed: algorithms based on Hamilton–Jacobi formulations”, *Journal* of Computational Physics, vol. 79 (1), pp. 12–49. 1988.

[RSEG 22] J. Park and J.M. Keller, “Snake on the watershed”, IEEE Transactions on Pattern Analysis and Machine Intelligence vol. 23 (10), pp. 1201–1205. 2001.

[RSEG 23] L.D. Cohen, “On active contour models and balloons”, CVGIP: Graphical Models and Image Processing: Image Understanding, vol. 53 (2), pp. 211–218. 1991

[RSEG 24] H. Park, T. Schoepflin and Y. Kim, “Active contour model with gradient directional information: Directional snake”, IEEE Transactions on Circuits and System for Video Technology vol. 11 (2), pp. 252–256. 2001

[SHD 1] Shan, Y., Yang, F., and Wang, R. 2007. “Color Space Selection for Moving Shadow Elimination”. In Proceedings of the Fourth international Conference on Image and Graphics. ICIG. IEEE Computer Society, Washington, DC, August 22 - 24, 2007.

[SHD 2] N. Herodotou, K.N. Plataniotis, and A.N. Venetsanopoulos, “A Color Segmentation Scheme for Object-Based Video Coding”. In Proc. IEEE Symposium of Advances in Digital Filtering and Signal Processing, pp. 25-29, 1998.

[SHD 3] Cucchiara, R.; Grana, C.; Piccardi, M.; Prati, A.; Sirotti, S., "Improving shadow suppression in moving object detection with HSV color information," Intelligent Transportation Systems. Proceedings. IEEE, vol., no., pp.334-339, 2001

[SHD 4] M.-T.Yang, K.-H.Lo, C.-C.Chiang, and W.-K.Tai. “Moving cast shadow detection by exploiting multiple cues”. In IEEE Int. Conf. On Image Processing (ICIP). Vol 1, pp. 413-416. 2005

[SHD 5] Nayar, S. K. and Bolle, R. M. Reflectance based object recognition. Int. Journal on Computer Vision 17, 3, 219-240. Mar. 1996.

[SHD 6] Nadimi, S.; Bhanu, B., "Physical models for moving shadow and object detection in video," IEEE Transactions on Pattern Analysis and Machine Intelligence, vol.26, no.8, pp.1079-1087, Aug. 2004.

[SHD 7] Graham .D. Finlayson, S.D. Hordley, and M.S. Drew. “Removing shadows from images”. In ECCV 2002: European Conference on Computer Vision, pp. 4: 823–836, Lecture Notes in Computer Science Vol. 2353. 2002.

[SHD 8] Graham D. Finlayson, Mark S. Drew, Cheng Lu. “Intrinsic Images by entropy minimization”. In ECCV 2004: European Conference on Computer Vision pp. 582-595. 2004

[SHD 9] Weiss, Y., "Deriving intrinsic images from image sequences," IEEE Eighth International Conference on Computer Vision. ICCV 2001. Proceedings. vol.2, pp.68 75. 2001

[SHD 10] Y. Matsushita, K. Nishino, K. Ikeuchi, and M. Sakauchi, “Illumination Normalization with Time-Dependent Intrinsic Images for Video Surveillance” In Proceedings of Conference on Computer Vision and Pattern Recognition, vol. 1, pp. 3-10, 2003.

[SHD 11] M.-T.Yang, K.-H.Lo, C.-C.Chiang, and W.-K.Tai. “Moving cast shadow detection by exploiting multiple cues”. In IEEE International Conference On Image Processing (ICIP).m vol. 1, pp. 413-416. 2005.

[SHD 12] Marshall F. Tappen, William T. Freeman, Edward H. Adelson, "Recovering Intrinsic Images from a Single Image," IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 27, no. 9, pp. 1459-1472, September, 2005.

[FB 1] Harville, M. , “A framework for high-level feedback to adaptive, per-pixel, mixture-of Gaussian background models”, ECCV, vol. 3, pp. 543_560, 2002.

[FB 2] Rincón, M., Carmona, E., Bachiller, M., and Folgado, E., “Segmentation of moving objects with information feedback between description levels”, In Proceedings of IWINAC 2007. 2007.

[TRC 1] Wyzecki, G.; Stiles, W.S. “Color Science: Concepts and Methods, Quantitative Data and Formulae “(2nd ed.). Wiley-Interscience. ISBN 0471021067. New York. 1982.

[TRC 2] Yilmaz, A., Javed, O., Shah, M., “Object tracking – A survey”.ACM Computing Surveys. 2006.

[TRC 3] Harris C., Stephens M.J., “A combined corner and edge detector”, In Alvey vision conference, pp147-152. 1988.

[TRC 4] Lowe, David G. "Object recognition from local scale-invariant features". Proceedings of the International Conference on Computer Vision. vol. 2, pp. 1150–1157. 1999.

[TRC 5] Bay H., Andreas E., Tuytelaars, T. Luc Van Gool "SURF: Speeded Up Robust Features", Computer Vision and Image Understanding (CVIU), Vol. 110, No. 3, pp. 346--359, 2008.

[TRC 6] Kuhn, H. “The Hungarian method for solving the assignment problem”.
Naval Research Logistics Quart. vol. 2, pp. 83–97. 1955.

[TRC 7] Beymer, D. and Konolige, K. “Real-time tracking of multiple people using continuous detection”. IEEE International Conference on Computer Vision (ICCV) Frame-Rate Workshop. 1999.

[TRC 8] Rosales, R. and Sclaroff, S. “3d trajectory recovery for tracking multiple objects and trajectory guided recognition of actions”. IEEE Conference on Computer Vision and Pattern Recognition (CVPR). pp. 117–123. 1999

[TRC 9] Tanizaki, H. “Non-gaussian state-space modeling of nonstationary time series”. J. Amer. Statistics Assoc. vol. 82, pp.1032–1063. 1987.

[TRC 10] D. Comaniciu, V. Ramesh, and P. Meer. “Real-time tracking of non-rigid objects using mean shift”. In IEEE Proceedings on Computer Vision and Pattern Recognition, pp. 673-678, 2000.

[TRC 11] Bhattacharyya, A. "On a measure of divergence between two statistical populations defined by their probability distributions". Bulletin of the Calcutta Mathematical Society vol. 35: pp. 99–109. 1943.

[TRC 12] F. Porikli, O.Tuzel, P. Meer: “Covariance Tracking using Model Update Based on Means on Riemannian Manifolds.”, Computer Vision and Pattern Recognition Conference, New York City, NY, vol. I, 728-735. June 2006.

[TRC 13] Haussdorf, F. “Set Theory”. Chelsea, New York, NY. 1962

[TRC 14] Terzopoulos, D. and Szeliski, R, “Tracking with Kalman snakes”.
In Active Vision vol. 3 (20), A. Blake and A. Yuille, Eds. MIT Press.1992

[TRC 15] Bertalmio, M. , Sapiro, G. and Randall, G., “Morphing active contours”. IEEE Transactions on Pattern Analysis and Machine Intelligence., vol. 22, (7), pp.733–737. 2000.

[MS 1] K.Fukunaga and L.D.Hostetler. “The estimation of the gradient of a density function, with applications in pattern recognition”. IEEE Transitions on Information Theory, vol. 21(1): pp.32–40, 1975

[MS 2] Y. Cheng, “Mean Shift, Mode Seeking, and Clustering”. IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 17 (8), pp. 790-799, Aug. 1995.

[MS 3] J. MacQueen. “Some methods for classification and analysis of multivariate observations”. In L.M. Le Cam and J. Neyman, editors, Proceedings of the Fifth Berkeley Symposium on Mathematical Statistics and Probability, vol. 1, pp. 281–297, 1967.

[MS 4] D. Comaniciu, V. Ramesh, and P. Meer. “The variable bandwidth mean shift and Data-Driven scale selection”. Proceedings of the International Conference on Computer Vision, vol.1, 2001.

[MS 5] Bugeau, A. and Pérez, P. 2009. “Detection and segmentation of moving objects in complex scenes”. Computer Vision and Image Understanding. vol.113 (4), pp. 459-476. Apr. 2009.

[MS 6] K. Cannons and R. P. Wildes. “Spatiotemporal oriented energy features for visual tracking”. In ACCV, pp. 532–543, 2007.

[MS 7] D. Comaniciu, V. Ramesh, and P. Meer, “Kernel based object tracking,” PAMI, IEEE Trans., vol. 25, (5), pp. 564–575, 2003.

[MS 8] R. D. Dony, S. Wesolkowski “Edge Detection on color images using RGB vector angles” In Proceedings of the 1999 IEEE Canadian Conference on Electrical and Computer Engineering. Shaw Conference Center, Edmonton, Alberta, Canada 1999.

[BM 1] W. Förstner and B. Moonen. “A metric for covariance matrices”. Technical report, Department of Geodesy and Geoinformatics, Stuttgart University, 1999.

[FM 1] Zhi, L., Yu, L. and Zhaoyan, Z.,” Real-time spatiotemporal segmentation of video objects in the H.264 compressed domain”. Journal of Visual Communication and Image Representation, pp. 275-290 Volume 18, Issue 3. June 2007.

[RLT 1] Altman,D.G. and Bland,J.M. “Statistics notes: diagnostic
Tests 1: sensitivity and specificity”. BMJ, 308, 1552. 1994.

[RLT 2] Makhoul, J., Kubala, F., Schwartz, R. and Weischedel, R. “Performance measures for information extraction”. Proceedings of DARPA Broadcast News Workshop, Herndon, VA, February 1999.

[RLT 3] Goutte, C. and Gaussier, E. "A probabilistic interpretation of precision, recall and F-score, with implication for evaluation," in ECIR, ser. Lecture Notes in Computer Science, D. E. Losada and Juan, Eds., vol. 3408. Springer, pp. 345-359. 2005

[RLT 4] VSSN 2006 “Call for Algorithm Competition in Foreground/Background Segmentation”. http://mmc36.informatik.uni-augsburg.de/VSSN06_OSAC/