UNIVERSIDAD AUTÓNOMA DE MADRID
ESCUELA POLITÉCNICA SUPERIOR

ONLINE CONTEXTUAL UPDATING IN MULTI-CAMERA SCENARIOS

Alejandro López Cifuentes
Director: Marcos Escudero Viñolo
Supervisor: Jesús Bescós Cano

-MASTER THESIS-

Departamento de Tecnología Electrónica y de las Comunicaciones
Escuela Politécnica Superior
Universidad Autónoma de Madrid
July 2017

ONLINE CONTEXTUAL UPDATING IN MULTI-CAMERA SCENARIOS

Alejandro López Cifuentes

Director: Marcos Escudero Viñolo
Supervisor: Jesús Bescós Cano

Video Processing and Understanding Lab
Departamento de Ingeniería Informática
Escuela Politécnica Superior
Universidad Autónoma de Madrid
July 2017

Abstract

The

Keywords

Keywords.

Acknowledgements

Alejandro López Cifuentes.
2017.

Contents

Abstract **v**

Acknowledgements **vii**

1 Introduction **1**
1.1 Motivation . . . 1
1.2 Objectives . . . 2
1.3 Thesis Structure . . . 3

2 State of the Art **5**
2.1 Pedestrian detection and crowd patrons . . . 5
2.1.1 People detection approaches . . . 6
2.1.2 Available Datasets . . . 9
2.1.3 Object Proposals . . . 9
2.1.4 Research Directions . . . 11
2.1.5 Crowd Dynamics . . . 12
2.2 Contextual Information . . . 12
2.2.1 Global . . . 12
2.2.2 Local . . . 13
2.2.3 Offline . . . 13
2.2.4 Online . . . 13
2.2.5 Semantic segmentation . . . 13
2.3 Multi-camera scenarios . . . 16

3 Developed Application **19**
3.1 Single-thread Application . . . 20
3.2 Multi-thread Application . . . 21
3.2.1 Classes Distribution . . . 25
3.2.2 Engineer Application . . . 26
3.2.3 User Application . . . 26

4 Proposed System **27**
4.1 Contextual Model Generation . . . 27
4.2 Pedestrian Detection . . . 29
4.2.1 Histogram of Oriented Gradients . . . 29

4.2.2 Deformable Part Model 30
4.2.3 Fast Region-Based Convolutional Network 31
4.2.4 PSP-Net 32
4.3 Fusion and Filtering 32
4.3.1 Multi-camera Scenario 32
4.3.2 Pedestrian Filtering 37
4.4 Statistical Usage Data 40
4.5 Induced Plane Homography 40

5 Results **41**
5.1 Application Performance Results 41
5.2 Hardware 41
5.2.1 Camera Specifications 41
5.2.2 Camera User Web Interface (GUI) 42

6 Conclusions and Future Work **45**
6.1 Conclusions 45
6.2 Future Work 45

Bibliography **46**

List of Figures

2.1 Edge Detector Example Diagram 7
2.2 Image summary from all the analyzed datasets. 10
2.3 Scene parsing on ADE20K [1]. Image from [2]. 14
2.4 Semantic Segmentation result examples on Cityscapes Dataset 16

3.1 QT Window Designer 19
3.2 Flow-chart legend 20
3.3 Flow-chart diagram for the single-thread application 21
3.4 Flow-chart diagram for the multi-thread application 22
3.5 Main application window 23
3.6 Application menu bar 23
3.7 Options Menu 24
3.8 Information Display 24
3.9 Results Display Area 25

4.1 Overview of the proposed PSPNet 28
4.2 HOG Pedestrian Descriptor 30
4.3 Example detection obtained with the person model. From left to right: Original image + detections, global coarse model, part templates and spatial model for each part. [3] 31
4.4 Fast R-RCNN architecture. [4] 32
4.5 Multi-camera configuration 33
4.6 Initial Camera Views 33
4.7 First cenital plane approach 34
4.8 Second cenital plane approach with measures 35
4.9 Second cenital plane approach with camera positions 35
4.10 Multi-camera configuration with panning set up 36
4.11 View selection process and homography between views computation. IMAGEN NO TERMINADA Y PENDIENTE DE CAMBIOS $N_{Views} = 5$ 37
4.12 Cylinder estimation for camera instance projections 38
4.13 Cylinder estimation for cenital view projections 38
4.14 Gaussian representation examples 39

5.1 Sony SNC-RZ50P Pan/Tilt Range diagram 41

5.2 Visualization and control menu 43
5.3 Preset position setting menu 43
5.4 Tour setting menu 44

List of Tables

2.1 Pedestrian Detection Performance 8
2.2 Cityscapes Dataset Class Definitions 15
2.3 Cityscapes Dataset Challenge Results 15

4.1 Final classes list from ADE20K to use in PSP-Net 29

5.1 Sony SNC-RZ50P Specifications 42

Chapter 1

Introduction

1.1 Motivation

Nowadays, we live surrounded by electronic devices which objective is to ensure the safety and security of the global population or to ease our lives on everyday tasks. These range from biometric systems[5] to all kind of different electrical sensors, including video surveillance cameras. These cameras are the ones that are of real interest when developing Image Processing and Computer Vision algorithms in the video surveillance scope [6].

The combination of these veins of research could lead to the automation of high-level human semantic tasks such as people detection [7], object detection and recognition [3, 8, 9] and extraction of contextual information [2]. The automation of these processes permits end-users build on these information sources to define the latest stages of video surveillance systems. These are usually the critical ones, e.g. alarm raising when some predefined event occurs.

Usually, mentioned video surveillance systems could be focus either from a single-camera point of view, whcih will lead to a simple scenariao in which the potential actions/events to detect will be observed from one point in the scene, or, on the contrary from a multi-camera setup. This sort of configuration will arise multiple benefits when dealing with big spaces as it will provide the user different views of the scene which will help the application purposes.

Among Computer Vision applications running on a multi-camera scenario, a pivotal field of research is the analysis of public spaces. These are often crowd-populated scenarios which analysis requires the combination of the data obtained by all recording cameras. It is of real interest to analyze people behavior patterns [10, 11, 12] and temporal usage of a given area in large-scale scenarios such as shopping malls, univer-

sities and, generally, in any public-use building, either to extract statistical measures of behavior or to detect anomalous unexpected events [13]. This analysis will come from a combination of complementary algorithms such as contextual or semantic area classification, people detection and crowd behavior analysis.

1.2 Objectives

The main objective of this master thesis is to extract contextual description from a large-scale populated scenario while extracting temporal statistical usage data from relevant scene spaces using various cameras. This should be supported with a Graphical User Interface application. To fulfill this objective, this work will embrace two different blocks of objectives that will complement each other. The first one will have to do with the design of a graphical user interface (GUI) whereas the second block will deal with algorithm and research-related objectives.

Graphical User Interface

The main user interface should be able to visualize and dynamically arrange usage statistics from different areas of interest in a public space either pre-generated or generated under a real-time constraint, under a user-friendly environment.

Algorithm

The algorithm related objectives are:

1. To integrate a semantic segmentation algorithm to perform contextual element analysis in video sequences. The objective is to detect and classify and determine the spatial extend on each frame of the video of relevant elements such as doors, desks, corridors and floor areas. We aim to:
   (a) Identify the current state of all these elements in each of the processed cameras. The state should distinguish between visible and occluded.
   (b) Identify the usage rate of some important elements of the scene measured by number of people per time interval.
2. To integrate state-of-the-art people detection algorithms results per view. To this aim, we need to:
   (a) Create a fusion mechanism to take advantage of the multi-camera scenario. The results from all the cameras are projected on a common space and com-

bined such that by result's refinement the individual detector performances are increased.

(b) Analyze people and crowd motion patterns to by combining them with contextual information define usage statistical measures on predefined spaces where specific activities take place.

1.3 Thesis Structure

The master thesis is divided into the following chapters:

- Chapter 1. Introduction.
- Chapter **??**. State of the Art.
- Chapter 3. Developed Application.
- Chapter 4. Proposed System.
- Chapter 5. Results.
- Chapter 6. Conclusions and Future Work.
- Appendix.
- Bibliography.

Chapter 2

State of the Art

As explained in Chapter 1 the process of analysis of a common crowed space will embrace many algorithms from different Computer Vision disciplines. In the scope of our work it will be worthy to analyze deeply state of the art pedestrian detection approaches and how to improve them as it will be a pivotal field for our goals. In addition contextual information an its main algorithms will be also important an the actual literature should be analyzed. Both of these disciplines will be used in a multi-camera setup and so, scenarios working with this configuration will be defined.

Throughout this Chapter we will summarize the actual and most used algorithms in the different mentioned categories.

2.1 Pedestrian detection and crowd patrons

Pedestrian detection has been a major issue in computer vision during the past few years due to its potential uses for all sort of modern applications. Its main objective is to detect and identify a potential object as a person and by the end, obtain its position in the scene.

Nowadays, it is consider a half-solved problem. Although there are many excellent pedestrian detectors in the literature the is not a detector that performs with perfect accuracy. This is one of the reasons why it keeps being one of the most researched area in the computer vision field.

The difficulty of pedestrian detector lies in the large amount of available datasets with different video and people characteristics such as people occlusions, poses, scales, illumination ranges.

2.1.1 People detection approaches

Main pedestrian detections that have been used so far can be divided in many categories and from many points of views. One possible easy classification depending on the used descriptor may start with the ones that use Histograms of Oriented Gradients like [14] in combination with linear Support Vectors Machine in order to describe pedestrian shape with gradients to create a model and classify possible candidates. Secondly, Discriminative Part Models as [15] that will divide the human body into different parts (head, trunk, legs...) and search for them into the image for a later combination into a single person detection, or finally Aggregate Channel Features [16] which will use a combination of different channels such as normalized gradient magnitude, histogram of oriented gradients (6 channels), and LUV color channels in order to achieve the final detection.

In [17] an extensive evaluation of the state of the art of people detection is given. Here algorithms are characterize and differentiate ones from the others depending on the object detection approach and the person model used. Object detection will be the extraction of the possible candidates to be a person from the scene. Main algorithms usually will use:

- Sliding window: The so called exhaustive search will lead to an efficient classifier to test, in search of pedestrian, every possible image window. Parameters such as window size, or overlapping are common tuning values that will increase or decrease the performance of the detector. Those methods usually need from 10^4 to 10^5 windows per image and this number will grow exponentially for multi-scale detection. If the complexity of the core classifier is increased in every window testing, the computational time will end up being not affordable.

- Segmentation: This approach will introduce some sort of segmentation as a preliminary step for the pedestrian detection. Algorithms such as background subtraction will lead to some moving parts of the image that will generate interest objects from moving patterns. Others such as color segmentation are based on color skin to restrict the future people search only to those objects that fulfill the color condition. By all means, segmentation will directly produce those scene candidates to be a person and so the computational time is highly decreased.

- Segmentation + Exhaustive search: The final approach will be the combination of both previously analyzed techniques. In this case the previous step of segmentation will not lead to final candidates objects but to an area that could

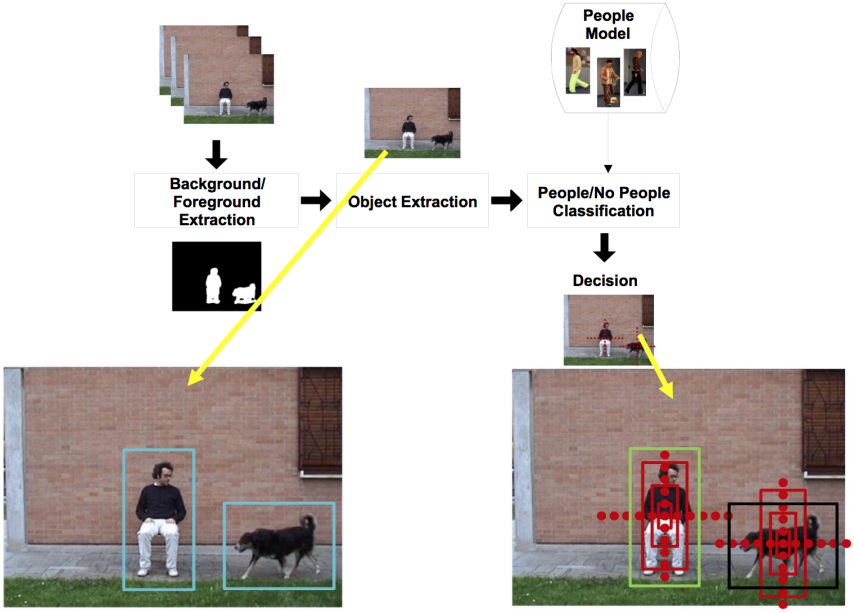


Figure 2.1: Edge Detector Example Diagram

contain some candidates objects. After segmentation process and sliding window technique is perform over the reduce scene area and so the final candidates are extracted. In this case improvements from both approaches are used as the computational cost of the exhaustive search (that is the main drawback) is reduce by the use of some sort of segmentation. In Figure 2.1 we can observe a diagram example of the Edge detector that uses the segmentation + exhaustive search approach.

Moving to person model, or in other words the set of characteristics that will be able to discriminate between people and any other object in the scene, [17] differentiates three different classification groups:

- Based on appearance: Most of the available detectors nowadays use appearance information to define the person model and so, to discriminate between person and the rest of the scene. In this group of person model one can differentiate two approaches to describe the shape of a person.

  Holistic: With this models people will be defined from the easiest point of view when a person is a region or a shape

  Part-based: These models however, use more complex model where people are defined as a combination of multiple shapes or regions from its body.

  Here one can find those that use silhouettes to classify people, either from an

Video	HOG	ISM	Fusion	Edge	DTDP	ACF	Faster-RCNN
1	89.3	71.4	34.9	84.9	96.7	99.3	99.7
2	69.2	82.9	92.5	90.2	77.1	77.1	98.2
3	55.6	75.7	64.3	71.7	68.9	68.9	82.9
4	10.1	1.0	0.5	5.4	33.9	33.9	37.5
Average AUC							

Table 2.1: Pedestrian Detection Performance

holistic or part-based point of view, or color distribution in people, however the most used approaches as said at the beginning of the section, use Histograms of Oriented Gradients like [14], Haar-like features [18], or combination of multiple features, aggregate channel features: HOG, gradient and color (ACF) [16].

- Based on motion: As said before many of the detectors usually are based on appearance, however, human appearance is likely to change due to environmental factors such as light conditions, cloths, camera settings. In addition, people variability in terms of height, weigh and poses will make appearance much likely to vary. Due to this, some approaches try to get rid of these factors and detect pedestrians using only its motion information. In [19] detections are based on periodic motion analysis. The algorithm performs motion segmentation and tracking to later on compute the self-similarity between objects.

- Based on appearance + motion: Although the main used algorithms are based on appearance there are some such as [20, 21] that will merge both appearance and motion in order to improve results. Most of these algorithms combine people detection and tracking, and so, they have been design to improve people tracking over video sequences.

In the Table 2.1, [17] propose some results from the State of the Art detectors depending on different datasets. The selected algorithm ranking for this evaluation is the average AUC. This metric will measure the area under a Recall Over Precision curve which means that an area of 1 represents a perfect test; an area of .5 represents a worthless test

However, last years a new sort of detectors have start to be use detectors based on deep Convolutional Networks have improve notably the accuracy of all the previous algorithms. Examples such as ImageNet [22] for image classification and Fast R-

CNN [4] for object detection expose that deep convolutional networks outperform the previous mentioned algorithms. This fact is clearly presented in the mentioned Table 2.1.

2.1.2 Available Datasets

As said at the beginning of the section pedestrian approaches will perform better or worse depending on the dataset characteristics that have been use for learning. Due to the high number of available dataset no detector can outperform on all of them. During this section main used datasets will be analyzed:

- Caltech [23], which consists on video taken from a vehicle diving through regular traffic in an urban environment.
- ETHZ [24]. It consists on recordings using a pair of AVT Marlins F033C mounted on a chariot which moves through pedestrian paths.
- TUD [25]. In this case, the recording has been done by a static camera in a crossing campus scene.
- INRIA [14] which collects precise people images both static and moving. In this case, images fully represent people without taking into account the environment.

As a result of this large amount of training and testing data, some detectors perform better in terms of precision and time consumption than others depending on the specific situations.

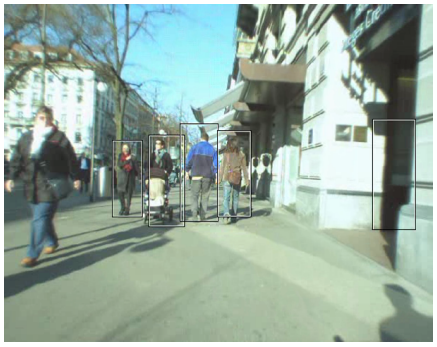
In Figure 2.2 one can observe some examples from the images that can be obtained from the different mentioned datasets.

2.1.3 Object Proposals

Described detectors such as HOG, DPM or ACF utilized the well known "sliding window" algorithm, which means that an efficient classifier will test, in search of pedestrian, every possible image window. Parameters such as window size, or overlapping are common tuning values that will increase or decrease the performance of the detector. Those methods usually need from 10^4 to 10^5 windows per image and this number will grow exponentially for multi-scale detection. If the complexity of the core classifier is increased in every window testing, the computational time will end up being not affordable.

One of the most successful approach to overcame this time consumption problem without losing detection quality is by means of object proposals [9].

(a) Caltech [23]



(b) ETHZ [24]

(c) TUD [25]

(d) INRIA [14]

Figure 2.2: Image summary from all the analyzed datasets.

Object proposals will be such objects that could be fitted into the desired label classification. This object will be share common visual properties that will distinguish them from the background. In order words, this approaches will perform a complete search over an image to previously select potential pedestrian candidates.

If lower number of object proposals than sliding windows lead to a higher object recall, a speed-up will be achieve which means that more efforts can be focus on more sophisticated classifiers.

Both ImageNet and Fast R-CNN use detections proposals, which is one of the main reasons why they outperform previous algorithms. In [9] two set of proposal methods are analyzed:

- Grouping proposal methods. This methods attempt to generate multiple, and so, overlapping segment that are likely to correspond to objects. Usually this method generate region proposals by grouping super pixels, solving multiple graph cut problems, or directly using edge contours. One of the best examples for this methods is MCG [26].

- Window scoring proposal methods. An alternate approach is to score each candidate window according to the probability to contain an object. Best results for this methods are provided by EdgeBoxes [27] which uses a coarse sliding window pattern which uses boundary estimation and a refinement step to improve performance.

2.1.4 Research Directions

As said before this computer vision field is constantly in change and so some future work lines can be set. In [7] some research directions are proposed that could be of interest in the scope of this work.

1. Context information should be added. Starting from the hypothesis that a person should be placed on the floor, the ground plane assumption can reduce errors if the detection for both the person and the floor are accurate. This could be achieve by extracting good contextual information from the scene which is one of the main objectives of the work.

2. Occlusion treatment. Usually pedestrians, due to other scene elements such as columns or even other pedestrians get occluded. When this happens performance degrades rapidly under even mild occlusion so, some improvements in this area will increase the performance in real life situations such as crowded spaces.

2.1.5 Crowd Dynamics

Finally, there is many literature concerning pedestrian dynamics in the real world such as the Social Force Model (SFM) [10] or people behavior in a built environment [11]. The information coming from these algorithms will lead to predictions in people behavior and the paths they could follow which could help us to extract more accurate statistical data, refine people detections based on this previous behavior knowledge or even improve detection computational time searching pedestrian only in those paths they usually follow.

2.2 Contextual Information

One can describe contextual information as the set of circumstances or facts that one can extract from a scene. Usually, this set of circumstances is a wealth of information that is not captured by the image but extracted by humans based on previously acquired knowledge. By taking a look to an outdoor image one can derive where the sky will be, what the weather conditions are, which time of the day... Also, by knowing the place the photo was taken, one can lead which objects will be more probable to appear in the scene.

When talking specifically about computer vision, contextual information will embrace for instance camera information (such as position, configuration, distance to an object, static or not), the set of objects that one could detect in the scene or how many views will be available.

All this set of information that is not extracted from the frame will provide a more general understanding about the scene and will help further algorithms. We can divide contextual information into two different levels, global and local contextual information, and also into two different categories offline, and online.

2.2.1 Global

Global context will consider image detections from the image as a whole, this means that for instance an image will be consider as a kitchen to predict the presence of a stove. This kind of approaches are focused on psychology studies [28, 29] that suggests that human perceptual processes work following a hierarchically organized process. Our perception system will go from a global structure towards a more detailed analysis.

Global context approaches aims to define a scene as an extra source of global information. The structure of a determinate scene image can be estimated by the

mean of global image features.

2.2.2 Local

On the contrary, local context will consider context information from the neighbor area of an object, e.g. a kitchen table will predict the presence of a spoon. The neighboring areas will be defined as a set of other objects, patches or even pixels. The aim is to try to correctly define the area that surrounds the object to precisely detect the desired object.

2.2.3 Offline

Offline contextual information will be the set of circumstances that are computed and available before starting any kind of procedure. This information will be previously computed and so, it will lead to external image information that will added to any algorithm knowledge.

2.2.4 Online

Online information, on the other hand, will be computed during procedures, and so it will not previously computed knowledge. Online extraction will have the computational time drawback as it will be considered as part of an algorithm.

If we now focus with the set of objects that one can observe from a scene, semantic segmentation techniques, which will be analyzed in the following section, aim to separate a scene into different observable objects.

2.2.5 Semantic segmentation

Semantic information can often lead to low-level characteristics [30] such as global color or, on the other hand, it can describe an image by the position and status of relevant objects [31] such as cars according to its moving paths or, what is more important to our related work, concrete areas in a scene such as walls, corridors, walking paths, etc. [2].

In other words, semantic segmentation will have the goal to assign each pixel of an image a category label. If the prediction is accurate, it will provide complete semantic understanding from the scene (Figure 2.3) which means that one could have the position and label of some object.

Cityscapes Dataset [32] is a large-scale dataset that contains stereo video sequences recorded in street scenes from among 50 different cities around the world. This dataset

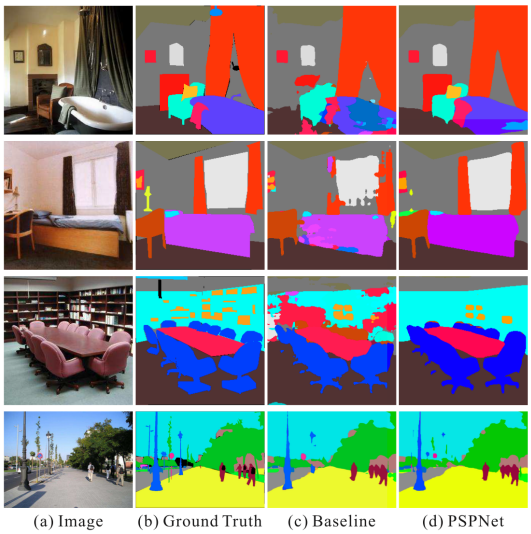


Figure 2.3: Scene parsing on ADE20K [1]. Image from [2].

presents class annotations over pixels in more than 5000 frames. In the following table 2.2 available dataset classes are presented.

[32] offers a benchmark suite with an evaluation server. This makes easy that authors can upload semantic segmentation results in order to be able to compare their performance with the rest of tested algorithms. In Table 2.3 we can observe the four most accurate algorithms in the scope of this dataset.

The main used metric for this Pixel-Level semantic classification results is the commonly known as the PASCAL VOC intersection-over-union metric which formula is presented in Eq 2.1.

$$IoU = \frac{\#TP}{\#TP + \#FP + \#FN} \tag{2.1}$$

with $TP = TruePositives$, $FP = FalsePositives$ and $FN = FalseNegatives$

Both ResNet-38 and PSPNet are based on convolutional networks. Since the presentation of AlexNet [22] in 2012 those networks have been growing deeper and deeper each year ending up with 1,202 trainable layers.

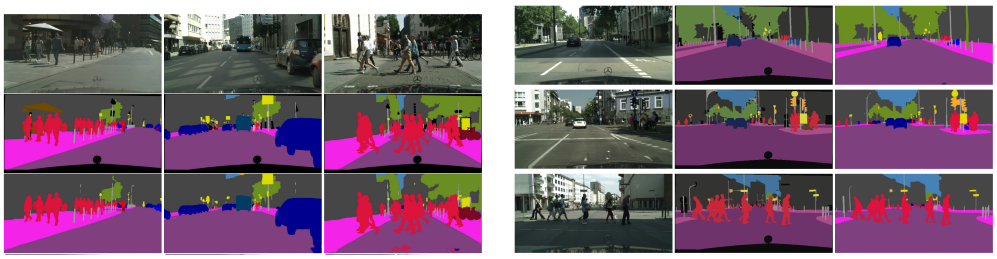
The main difficulty of scene parsing is related to the type of scene and by all means to the label variety that one can predict. [2] has dealt with this problems assigning relationships between different dataset labels, i.e. a an airplane is likely to be in runway or flying in the sky while not over a road. This relationships will reduce slightly the complexity of having large amounts of labels to predict and will improve the general performance of the algorithm. In Figure 2.4a and 2.4b we can observe

Category	Classes
Flat	road · sidewalk · parking · rail track
Human	person · rider
Vehicle	car · truck · bus · on rails · motorcycle · bicycle · caravan · trailer
Construction	building · wall · fence · guard rail · bridge · tunnel
Object	pole · pole group · traffic sign · traffic light
Nature	vegetation · terrain
Sky	sky
Void	ground · dynamic · static

Table 2.2: Cityscapes Dataset Class Definitions

Algorithm Name	iOU Category	IoU Class
motovis	91.4	81.0
ResNet-38 [33]	91.0	80.6
NetWarp	91.0	80.5
PSPNet [2]	90.6	80.2

Table 2.3: Cityscapes Dataset Challenge Results



(a) ResNet-38 Algorithm (b) PSPNet Algorithm

Figure 2.4: Semantic Segmentation result examples on Cityscapes Dataset

some visual examples of how this two algorithms perform on some Cityscapes frames.

2.3 Multi-camera scenarios

The use of multi-camera is a common setup when dealing with video surveillance problems. One can define as multi-camera scenario one space that has more than one video camera working and recording at the same exact time. Having N multiple camera instances allows to observe the same event or object of interest from up to two different points of view. This will lead to a set of advantages in the scope of our work when dealing with pedestrian detection and semantic classification and also, some unavoidable disadvantages.

- Advantages

  As said in 2.1 one of the main research paths in pedestrian detection is the occlusion treatment. In this case, the use of a multi-camera scenario with computed homographies between camera frames will help to reproject detections from one camera to another whose miss rates are high so one could solve partially the problem of occlusions or at least improve the performance.

  When talking about semantic segmentation the inclusion of a multi-camera system will arise some benefits. While a single-camera system could lead to misclassification of labels in the image or even to represent an object with many classes due to for example, camera capture problems, in a multi-camera system one camera instance could help to refine the classes in another one provided that, evidentially, they are analyzing the same common area.

- Disadvantages

  The main disadvantage when dealing with multi-camera systems will be the exponential grow of computational time as algorithms should be performed N

times being N the number of camera instances. This issue could be solved by the use of some parallel coding that will perform cameras process separately.

Other disadvantages will be related with synchronization problems, as if the camera instances will be used to reproject objects/pedestrians from one camera to another they must evidentially represent the same exact moment in time in the same sequence frame, which depending on the setup will be more or less difficult.

Chapter 3

Developed Application

Within this Chapter the developed application will be analyze. This application will be the base for the visualization and arrangement of the usage statistics from the different areas of interest. As said in Section 1.2 this usage data can be pre-generated or generated ideally under a real-time constraint. The main application environment should be user-friendly to ensure its correct usage.

The application has been developed under QT Creator coding environment in Mac OS Sierra. This decision has been fundamentally based on:

1. Its cross-platform characteristic which makes it easily portable from one operating system to another.

2. Its application window designer that enables the programmer to design the software windows by using an interface instead of having to code to create the mentioned window (Figure 3.1).

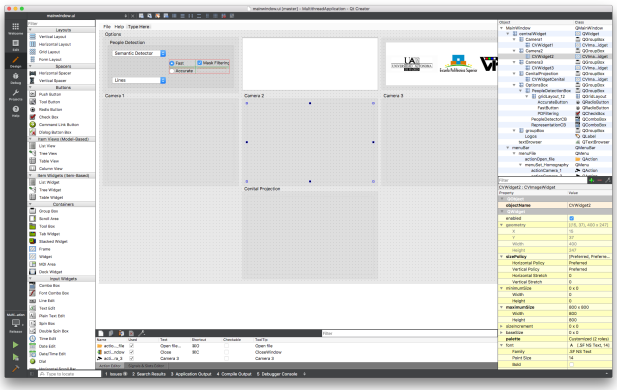


Figure 3.1: QT Window Designer

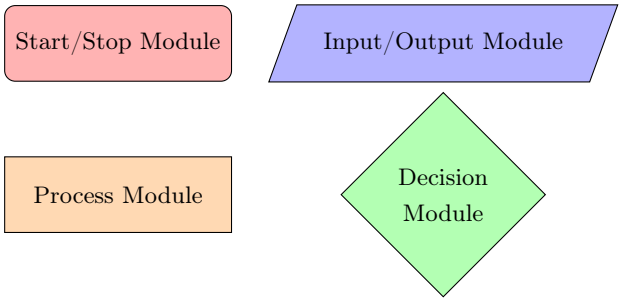


Figure 3.2: Flow-chart legend

3. The possibility to add OpenCV libraries to the project.

Two main applications, single-thread and multi-thread, have been developed and will be analyze on the following lines, however at the end of the project only the multi-thread application will be the one in use. All the code for both approaches is available in the following GitHub Repository.

During the analysis of the application trough out the Section some flow-charts will be displayed. The used legend can be observed in Figure 3.2.

3.1 Single-thread Application

During the first stages of the development and for simplicity sake the application has been design and developed to run under a single thread. This means that all the processing has been done sequentially camera by camera. A simple flow-chart diagram can be seen in Figure 3.3 that illustrates the execution path.

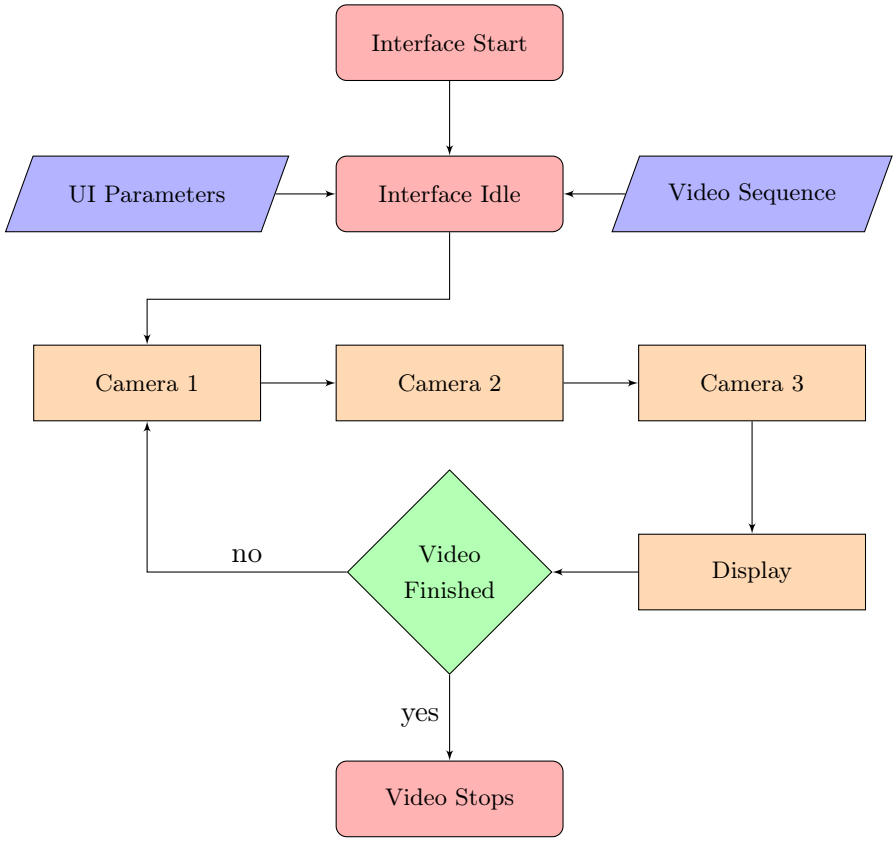


Figure 3.3: Flow-chart diagram for the single-thread application

This design approach will be valid if the computational effort is minimum because before displaying any results, all the process for the three cameras should be computed, however, when working with such a multi-camera system with heavy algorithms running as the proposed in our work, the computational time increases exponentially and this design is no longer worthwhile.

3.2 Multi-thread Application

The new and valid approach can be observed in the flow-chart displayed in Figure 3.4. As one can observe, now the three threads are running in parallel and so, all the process is no longer done sequentially and computing power from the CPU can be much more usable.

Now, however, as threads are running separately one have to create some sort of synchronization between them. This is done because one thread can process a frame faster than another one due to multiple external reason. Nevertheless, the application should display the same exact frame for all the cameras more over if the information

is going to be shared between threads. In our case the synchronization is perform by the barrier that can be observed in the diagram. This barrier will create a meeting point for all the threads that all of them must reach before displaying any result. This is done as said to ensure that all the process have been completed before displaying or saving any results.

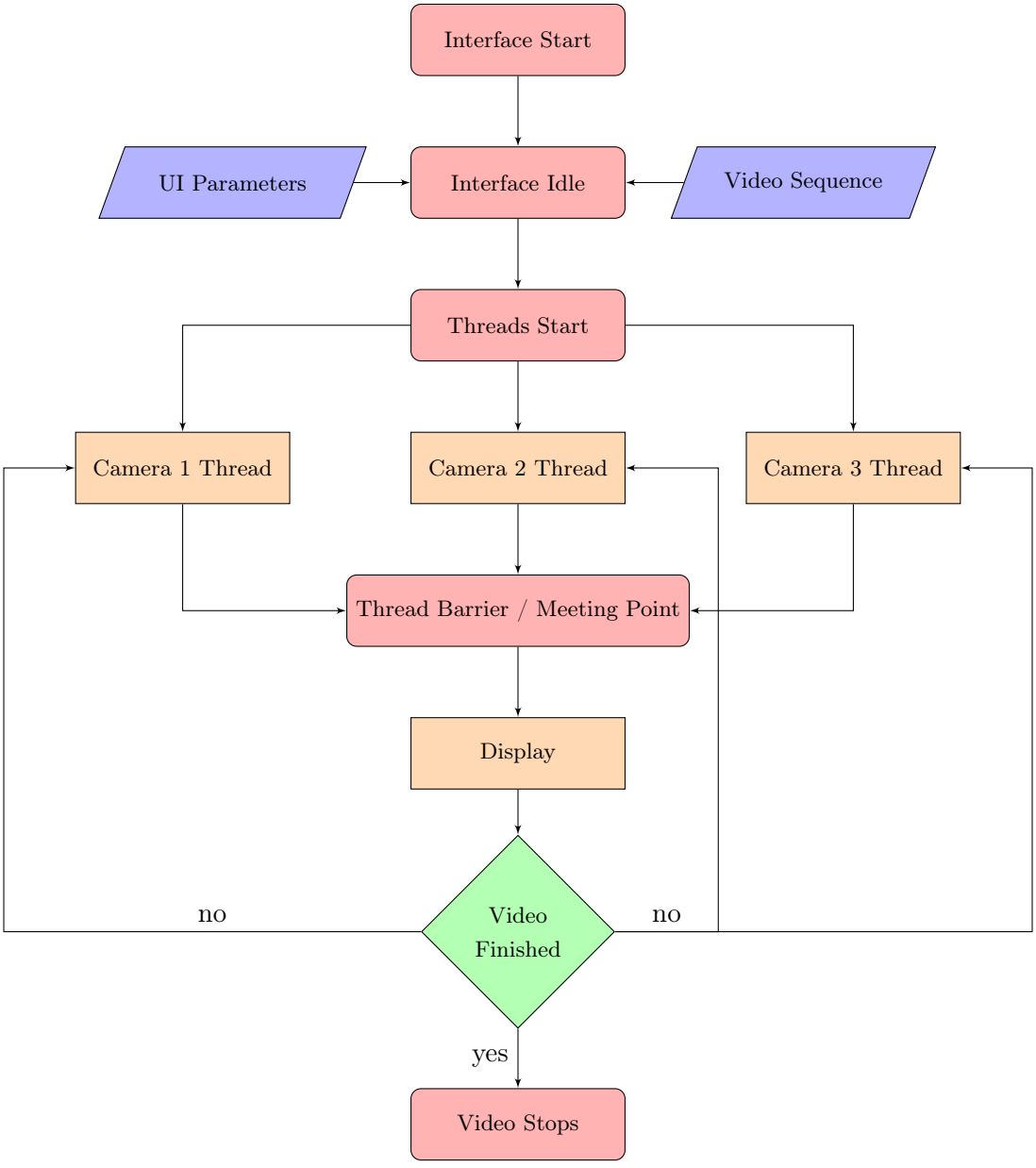


Figure 3.4: Flow-chart diagram for the multi-thread application

Main application window is shown in Figure 3.5. As one can observe it is compose

of four separate areas:

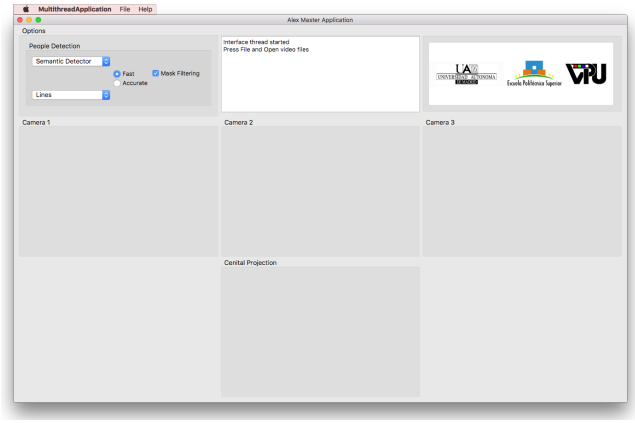
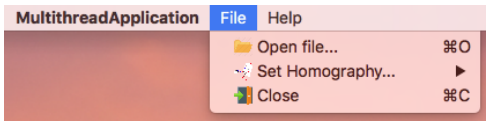


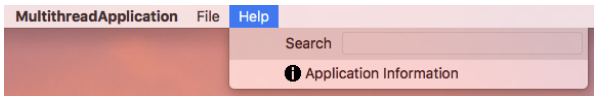
Figure 3.5: Main application window

1. Application menu bar.

   In this menu we will have the main application actions. From here the user can open a new video sequence, compute the set of desired homographies or close the program. The application also provides a help searcher and an external information window. This set of functions can be seen in Figure 3.6.



(a) File Submenu



(b) Help Submenu

Figure 3.6: Application menu bar

2. Options menu.

   This options box (Figure 3.7) in the application will contain all the possible parameters that can be tuned during the execution of the program. From here algorithms can be changed in real time so there is no need to restart the running before changing some parameter. The user can select among various pedestrian detectors such as semantic detector, DPM or even Fast-RCNN as well as the type of representation that one wants for these detections, lines or gaussians which will be explained later. In addition the application provides an option to perform mask filtering.

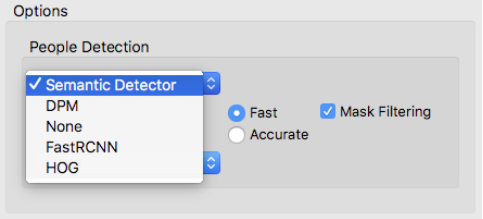


Figure 3.7: Options Menu

3. Information Display

Along this text box some information will be provided to the user. Messages such as "Open video files", "Processing starts now" or "DPM Pedestrian Detector is now in use" will appear during the execution of the application so the user can obtain some information about what to do, or what algorithm are in use as we can observe in Figure 3.8.

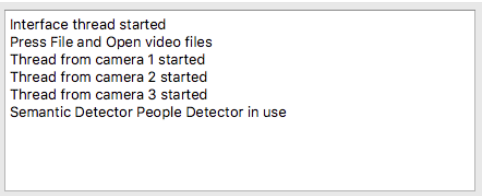


Figure 3.8: Information Display

4. Results Display

This is the main display area in the application in which all the visual results will be presented. We have three separate windows for each of the used camera as well as one more display window for the cenital plane. Here the camera frames with its pedestrian detection will be shown and all the projected semantic can be observed on the cenital frame. An example of its performance is presented in Figure 3.9.

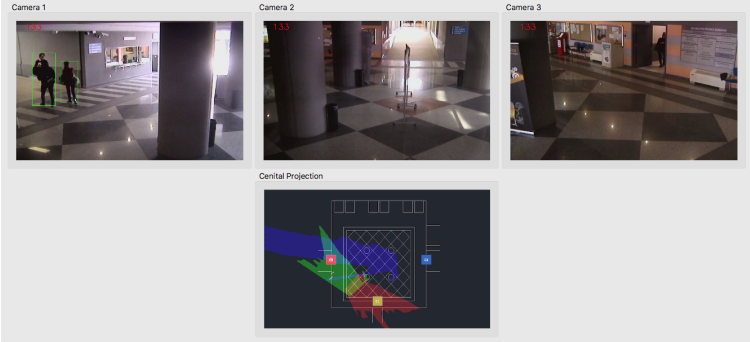


Figure 3.9: Results Display Area

3.2.1 Classes Distribution

In terms of C++ blocks the application has been divided into several classes whose functions will be explained in the following lines:

- MainWindow. This class corresponds to the main interface window and main application thread. It is the base for all the further processing. The functions that should be done by this class is opening the video files, creating and starting the threads, update all the algorithms configurations from the UI and displaying results through the CVImageWidget class.

- AboutWindow. Class that execute the second window that simply displays information.

- CameraWorker. Main class for the cameras. CameraWorker class is linked with a thread to process all the algorithms inside. It has CameraStream, PeopleDetector and Barrier classes declared within it.

- CameraStream. This class will perform all the processes that has to deal with the video processing except pedestrian detection. The main reading loop, homographies and computations will be computed in this class.

- PeopleDetector. Main class to perform everything with respect to pedestrian detection. All the algorithms to detect, project and draw will be from this class.

- Barrier. This class will implement everything that has to deal with thread synchronization.

- CVImageWidget. Display representation class that will paint OpenCV Mat images into the QT interface Widget.

3.2.2 Engineer Application

3.2.3 User Application

Chapter 4

Proposed System

During this Chapter our proposed system will be explained. We will start from the main contextual model generation and pedestrian detectors to finally explain the fusion of all the elements in a multi camera system in addition to the statistical usage data.

4.1 Contextual Model Generation

One of the main objectives in the scope of this work is to segment one frame into different semantic areas. The relative position in the scene for elements such as doors, walls, paths, columns and windows will be necessary to achieve further objectives such as multi camera fusion and statistical data. For this complex task we will use an algorithm presented in Chapter 2, PSP-Net [2]. The goal for this algorithm as said before, is is to assign each pixel in the image a category label and predict the label, location, as well as shape for each element.

This process is based on a deep convolutional neural network (CNN) called Pyramid Scene Parsing Network (PSPNet) which is designed to improve performance for open-vocabulary object and stuff identification in complex scene parsing. The structure of the network can be observed in Figure 4.1

Given the image in (a) the first step is to process it through a pre trained CNN called ResNet [34] to get the feature map (b) of the last convolutional layer. The final feature map in this step is 1/8 of the size of the input image.

The second step is to apply the main contribution of [2], which is called the Pyramid Pooling Module (c). The main objective of the module is to collect a few levels of information, much more representative than global pooling. It separates the feature map into different sub-regions and forms pooled representations for different

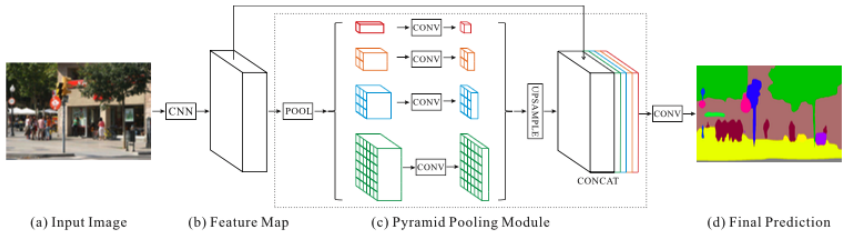


Figure 4.1: Overview of the proposed PSPNet

locations. Here a set of pooling, convolutional and upsampling layers are applied that will harvest different sub-region representation in N different scales.

After the bilinear interpolation upsampling, concatenation layers are used to form the final feature representation by fusing the feature map extracted in (b) and the Pyramid Pooling Module output. This final feature carries both local and global context information. In the end, the representation is fed into a convolutional layer which gets the final per-pixel prediction (d).

PSP-Net comes with a set of three different pre trained Caffe models for three different datasets. The main difference between the models for our scope is the environment in which the network has been trained.

- ADE20K: This dataset is the most challenging as it has up to 150 different labels in a wide range of scenes. The scenes go from interior room places to outdoor scenarios.
- VOC2012: It contains 20 object categories and one background class from diverse indoor and outdoor scenes.
- CityScapes: The last dataset defines 19 categories containing both stuff and objects. All the available sequences have been recorded from a driving car while driving in the street.

So, as one can observe the three different models represent different object categories in different real spaces. In our case we will select the model based on two main reasons:

1. The model should be trained with indoor scenes. This will lead to discard those models that perform only in outdoors scenes as we would like our approach to be used in interior scenarios. This will exclude CityScapes dataset from our options as all the classes and seqeunces used for training are from outdoor scenes.
2. From the trained indoor models we have to choose between those whose categories fit best in our work. In this case VOC2012 dataset uses classes such as

wall	building	floor	ceiling	road
window pane	person	door	table	chair
seat	desk	lamp	column	counter
path	stairs	screen door	stairway	toilet
poster	bag			

Table 4.1: Final classes list from ADE20K to use in PSP-Net

boat, airplane or table which are not interesting for our segmentation and it does not have classes such as door or wall which are really important for us.

With this two reasons being taked into account we have selected the model ADE20K because it has elements such as walls, floor, person and column in its model. However, we consider that most of the 150 label categories will be unused in our procedure, so, the number of classes in the model has been reduced to the 21 classes of our interest so we only obtain scores for those objects that we want. This class limitations will lead above all in a considerably hard drive space saving. In Table 4.1 can be observed the final 21 classes that have been used.

4.2 Pedestrian Detection

Along this section we will present the pedestrian State of the Art detectors that have been selected for our system. Some of them have been chosen due to the algorithm simplicity and computational cost and others due to its contrasted performance while working in real sequences as we have seen in Chapter 2. However, one of the main reasons of choosing the following three algorithms among others in the SoA is the possibility of either having the original source code or the practical implementation within an external library.

4.2.1 Histogram of Oriented Gradients

Histogram of Oriented Gradients, i.e. HOG, is one of the main used detectors along pedestrian detection field. This fact is due to it's extremely simplicity in terms of the descriptor complexity. Pedestrians are described as set of HOG, this means that its shape and appearance can be described by a set of gradients and intensities organized as orientation histograms. These histograms will describe intensity distributions from local gradients or border directions. This descriptor can be observe graphically in Figure 4.2.

Once HOG have been used to describe the person shape, Support Vector Machines are used to train a person model and to classify potential candidates as people. SVM

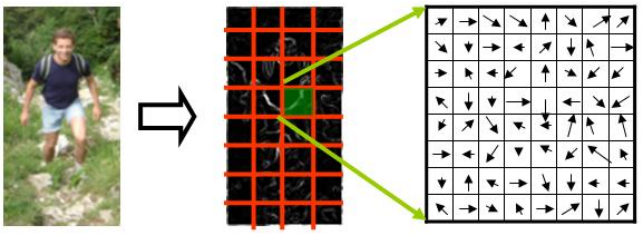


Figure 4.2: HOG Pedestrian Descriptor

are a data classification method formed by a set of supervised training. The aim of this kind of approaches is to produce a model which will be able to predict classification labels on a test set based only on the descriptors of the set.

The idea behind SVM is that training vectors will be mapped on to a bigger dimensional space in which the data separation by means of one or many hyperplane will be much easier than in the original dimensional space.

The combination between HOG and SVM will lead to a fast detector that depending on the situation will perform decently, although it has some main drawbacks as its lack of occlusion treatment which will not make this algorithm usable when working with crowded spaces.

The main implementation of Histogram of Oriented Gradient Pedestrian Detector is in OpenCV library for C++.

4.2.2 Deformable Part Model

As was mentioned in the previous paragraph one of the main drawbacks when working with the simple HOG pedestrian detector is that it describes the person model as a hole which will lead, inevitably, to the described drawbacks. Deformable Part Model tries to solve this problem, among others, by defining the model as first, a global coarse template, secondly, several higher resolution part templates and finally a spatial model for the location of each part. This description is the one that can be observed in Figure 4.3.

Both global and part templates are modeled with histogram of gradient features and the model is built by using an improving over SVM called latent SVM. In addition, scores for every detection are obtained by applying a root filter on the window plus the sum over parts of the maximum over placements of that par, of the part filter score on the resulting sub window minus the deformation cost.

The use of this detector will lead to a set of advantages than when working with others simpler approaches. In terms of pedestrian occlusion treatment we will be able to detect those people that have been occluded by something in the scene just

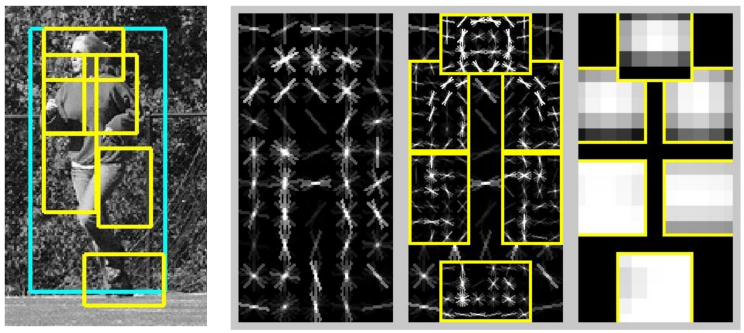


Figure 4.3: Example detection obtained with the person model. From left to right: Original image + detections, global coarse model, part templates and spatial model for each part. [3]

by detecting some visible part. This will outperform other detectors while working with crowded scenes in which holistic methods will have problems that will lead for instance to groups of people being detected as a unique detection while DPM will be able to separate them into different person instances.

However, its scanning window approach as well as the part based model will lead to some computational cost that will increase the time needed to obtain detections.

The main implementation of Deformable Part Model Pedestrian Detector can be found in OpenCV library for C++.

4.2.3 Fast Region-Based Convolutional Network

Fast Region-based Convolutional Network (Fast R-CNN) will be used to detect pedestrian in our proposed system, however, as said in Chapter 2 it is not only a pedestrian detector but an algorithm for object detection. In 2.1.3 we mentioned that Fast-RCNN must use objects proposals for its usage. In our developed system MCG [26] grouping method will be used for this purpose due to its great results.

Fast-RCNN method sets its contributions in a several number of innovations to improve training and testing speed over its fundamental base R-CNN:

1. Higher detection quality (mAP).
2. Training is single-stage, using a multi-task loss.
3. Training can update all network layers.
4. No disk storage is required for feature caching.

In Figure 4.4 one can observe the general Fast R-RCNN architecture. The input for the Fast-RCNN network are the entire desired image that one wants to process and

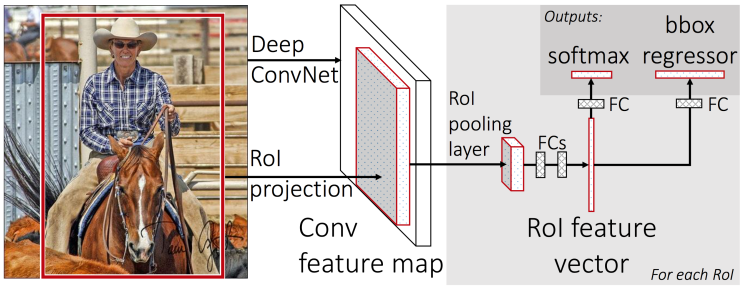


Figure 4.4: Fast R-RCNN architecture. [4]

the set of object proposals. The first step in the network is to process the whole image with several convolutional and max pooling layers to produce a convolutional feature map. Then, for every object proposal present in the image, a region of interest (ROI) pooling layer extracts a fixed-length feature vector from the feature map.

Each feature vector is after, introduced into a sequence of fully connected (FC) layers that finally diverge into to output layers. The first one produces probabilities estimates over K object classes. The second one outputs four real numbers for each of the K object classes. This set of 4 values encodes the final bounding-box positions for one of the objects from the K classes.

In this case, both Fast-RCNN detector and MCG object proposal extract are implemented within external Matlab libraries.

4.2.4 PSP-Net

As we explained in Section 4.1 PSP-Net working along with ADE20K dataset has been trained to detect people as a class. In our proposed system we will also use this class along with the previously explained pedestrian algorithms.

4.3 Fusion and Filtering

Once the semantic model and pedestrian detections are implemented the next step is to combine the information coming from different camera sources in a process that we have called fusion and filtering.

4.3.1 Multi-camera Scenario

Our system is developed in a multi-camera scenario. This means that we will be able to observe the scene from different point of views. In Figure 4.5 we can observe how the scene is configured with 3 different video-surveillance cameras. Two of them are

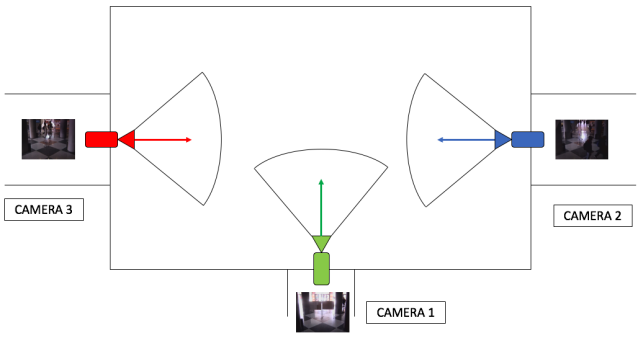
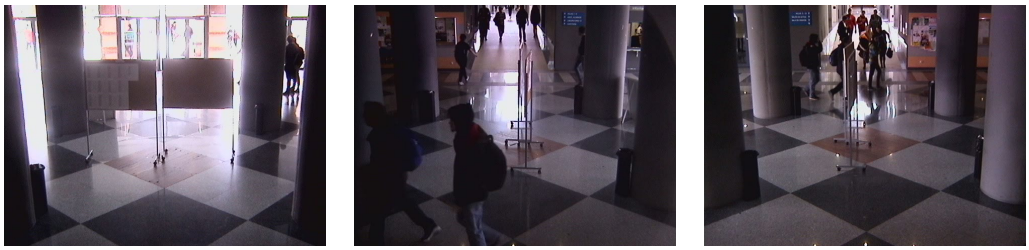


Figure 4.5: Multi-camera configuration



(a) Camera 1 (b) Camera 2 (c) Camera 3

Figure 4.6: Initial Camera Views

placed at the sides of the scene, while one is at the bottom part. The starting views from the three cameras is displayed in Figure 4.6.

As said in Chapter 2 working with a multi-camera scenario has some advantages. As one can see in Figure 4.6 we can observe the same scene area from three different point of views, which for example means, that detections from one camera can be used to detect in other camera or even refine the available detections.

4.3.1.1 Cenital Plane Homography

Homography calculation is a pivotal task in our work and the main base for the fusion of all the different information coming from the three cameras. The objective is to compute an homography matrix that will relate a camera frame to a so called cenital plane, i.e. a bird-eye representation of the scene. With this homography matrix we will be able to transform every frame pixel to its correspondent position in the cenital plane, i.e. it will transform the frame as if it were being viewed from the top.

In order to compute an homography, a relation between at least 4 points in each of the two images should be computed. In this case we are trying to create a relationship between a real camera frame, and a cenital view plane created by computer. This

Figure 4.7: First cenital plane approach

means that there will not be a real correspondence for pixels and so it is not possible to use a point descriptor to extract common points in both images. User has to manually select the points that represent the same spatial place in both images using the Graphical User Interface and then the algorithm will compute the transformation matrix.

4.3.1.2 Cenital Plan Design

As said in the previous paragraph, one of the main objectives of the work is to project the extracted detections from the three cameras, i.e. pedestrian and semantic, into a common plane for all of them. To achieve this goal a cenital plane from the scene is needed. The first approach used for the cenital plane can be observe in Figure 4.7.

As we can observe the first approach of the cenital plane lacks of details from the scene. The information about the details of the scenario is minimum and also, the scene proportions are not correct. To compute a correct homography between the camera frame and the cenital plane we should be able to identify the same scene points in both images in the ground plane. This means that the cenital plane should have enough detail so the point selection is done correctly by the user and the homography is correctly computed.

For this reason another cenital plane has been compute starting from zero. In this new approach the scene has been correctly measure by hand and the plane has been done with real measures and high floor detail.

For correctly drawing the plane AutoCAD 2017 software has been used. The second plane approach can be seen in Figure 4.8 with all the manual measures extracted from the real scene and in Figure 4.9 with the correct camera positions.

It is easily observable that differences between Figure 4.7 and 4.9 are outstanding both in floor details and in general construction proportions. This detail increase

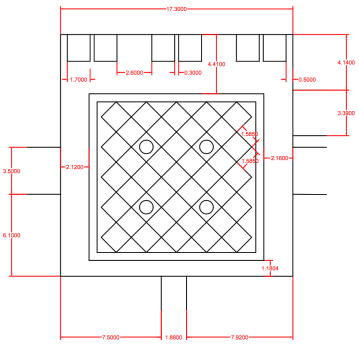


Figure 4.8: Second cenital plane approach with measures

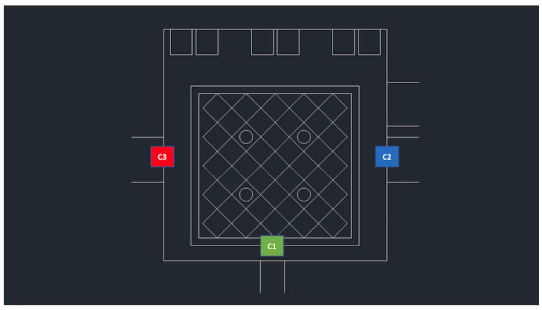


Figure 4.9: Second cenital plane approach with camera positions

will lead to a much more easy homography selection points by the user as it now has much more point options to choose and evidentially this will mean that the final homography matrix, and so, all the projections will be increased accurately talking.

4.3.1.3 Camera Panning

When cameras are static and pointing to the center of the scene as in Figure 4.6 one can easily observe that the vision range compared to the hole scenario representation is quite limited due to the low vision range of the cameras. The cameras are covering the middle part leaving completely unattended the lateral parts of the scenario. The solution to this problem is to include panning movement in all the cameras from left to right so cameras view range is increased and no longer focus in the middle of the scene as before. This will lead also that the common areas that the cameras will share are increased and not static. This solution is presented in Figure 4.10.

4.3.1.4 Camera Panning Discretization

Before including camera panning in the set up, homographies where calculated, at the beginning, for only one video frame per camera. This means that every camera only

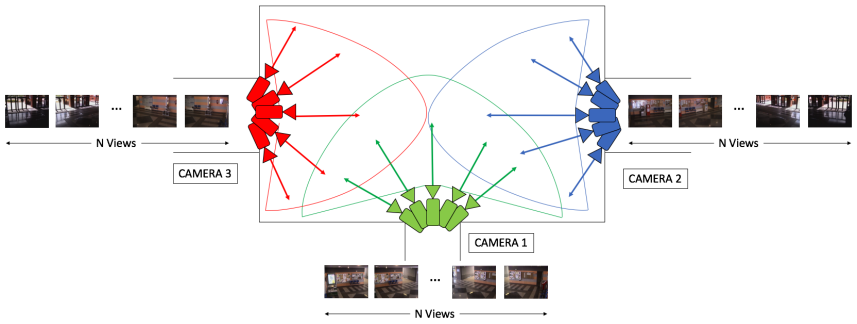


Figure 4.10: Multi-camera configuration with panning set up

had one homography matrix as all its sequence was static and all its video frames where projected using the same matrix. However, when including video panning this solution is no longer valid as the scene view is constantly changing. The ideal solution for this problem would be having one homography matrix H per video frame. As the homographies are calculated by the user selecting the points this solution is evidentially impossible in terms of usability. We propose discretization of the panning tour in which N views are selected from the video sequence. This will lead to an homography codebook in where the user will compute a homography matrix H for each of the N camera views. This codebook of views and homographies will be responsible for projecting into the cenital plane the hole video.

4.3.1.5 View Selection

Due to the creation of the homography codebook we now have to choose between a set of N homography matrices to project detections into the cenital plane. The actual analyzed frame should be compared to each of the N views to obtain an spatial correspondence and so, use the correct homography matrix.

In order to compare two images we have done comparison between points of interest. AKAZE detector and descriptor [35] has been used for this task.

AKAZE will detect and describe points of interest in any image and then by a brute force comparator we will extract how many coincidence we have between a pair of images. It is essential to say, that, as we have to compare the current frame with all of the N vies we will have a trade off. More views will mean a better space discretization, however, the computational time will increase exponentially, this means that we have to choose the number of views N so it represents correctly the space and keeps the computational time relatively low.

As we said before, the homographies are calculated for each of the views, this means, that if a frame is not positioned exactly as a view the homography matrix

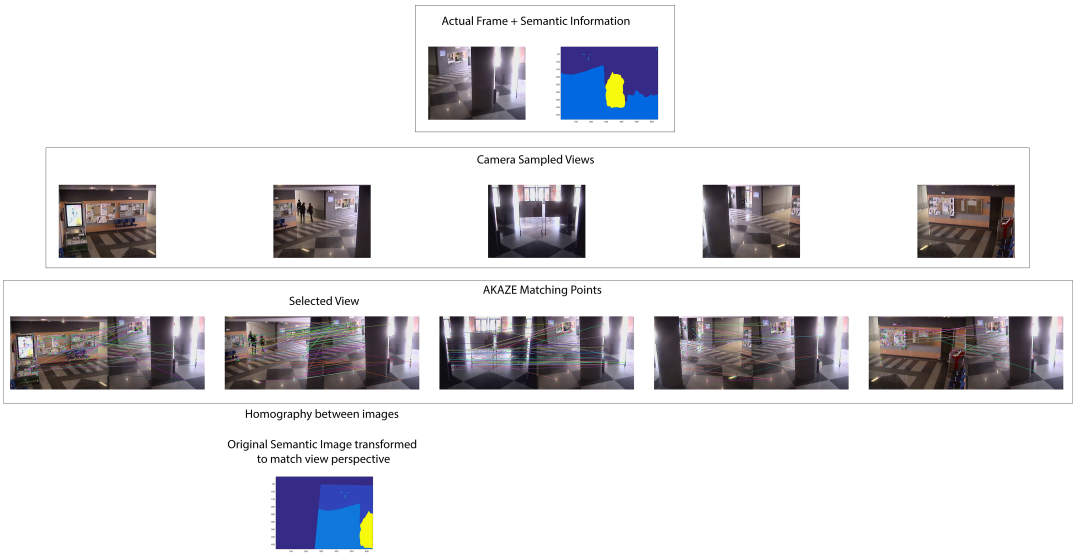


Figure 4.11: View selection process and homography between views computation. IMAGEN NO TERMINADA Y PENDIENTE DE CAMBIOS $N_{Views} = 5$

will not project the points correctly, there will be a small error. Taking advantage of the calculation of common points of interest between the frame and its correspondent view we can solve this problem by calculating automatically the homography between both images. This means that now, to project detections from the current frame to the cenital plane, first we will change the perspective of it to the perspective of the view. This way we will ensure that the used perspective is the same one as the one that was used to compute the homography. This process is graphically explained in Figure 4.11.

4.3.1.6 Video Sequence Synchronization

Synchronization between different camera videos by using concrete common events in the sequences.

4.3.2 Pedestrian Filtering

During this Section we will explain how the pedestrian detections are filtered and also how the information between people from the three cameras is fusion in the system to increase the general algorithm performance.

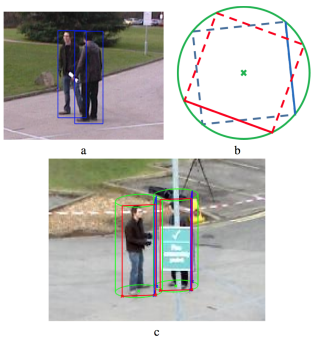


Figure 4.12: Cylinder estimation for camera instance projections

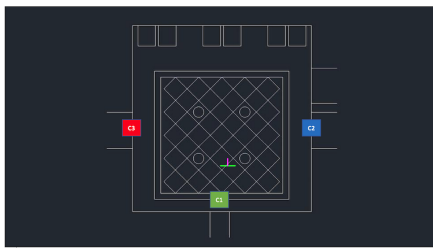


Figure 4.13: Cylinder estimation for cenital view projections

4.3.2.1 Cylinder Estimation

In [36] a cylinder estimation technique is proposed. The detected bounding boxes on the camera frame do not correspond spatially with the position of the detected object due to the camera perspective. If this detection error is not corrected when the bounding boxes are projected either to another camera instance or to the common cenital plane there will be a distance between the bounding box and the real object. Figure 4.12 shows the case for bounding box transference between cameras and Figure 4.13 for the cenital frame.

As we can observe in Figure 4.12 when the blue bounding boxes are projected from image (a) to image (c) they are not correctly on the pedestrian. The solution is to compute the cilinder that embrace the square whose side is the bounding box (b). Once the cylinder is estimated, the person will be in the middle of the cylinder. In (c) we can observe that the estimation, if correctly computed has great accuracy.

In Figure 4.13 the same methohd is applyed but in this case, the bounding box is not projected to another camera perspective but to the cenital plane. In this case we can see that the projection of the bounding box, represented by the green line is not in the same position as the center of the cylinder, whcih will be at the end of the purple line. The center of the cylinder so, will correspond to a more approximate position of the detected person.

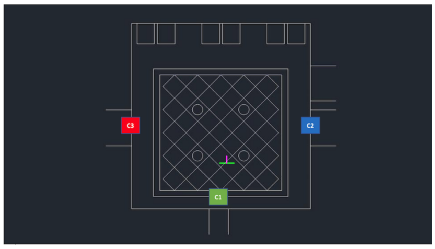


Figure 4.14: Gaussian representation examples

Poner aqui o en resultados?

4.3.2.2 Gaussian Bounding Box Representation

Using also the cylinder representation one can change the way bounding boxes are represented in the cenital plane. As said before, in the simple cylinder representation pedestrian will be represented with two perpendicular lines, however the middle point from the cylinder can be used to present a gaussian.

In this case we will project every pedestrian detection as a Gaussian function of the form described in Eq. 4.1.

$$f(x,y) = Aexp\left(-\left(\frac{(x-x_0)^2}{2\sigma_x^2} + \frac{(y-y_0)^2}{2\sigma_y^2}\right)\right) \tag{4.1}$$

In this case A will be the amplitude, x_0, y_0 the mean of the gaussian which, in our case will be the center of the blob and σ_x, σ_y which is the standard deviation and in our case will represent the accuracy of the detection. This means that detections with higher scores will be represented as a narrow gaussian function while lower scores will lead to wide gaussians that will represent a bigger area in where one can find that person. Some Gaussians examples depending on the scores can be observed in Figure 4.14.

Aqui o en resultados?

4.3.2.3 Pedestrian Reprojection

When all the pedestrian detections have been projected into the cenital plane we can start reprojecting those projections from the cenital plane back to the camera frames. This process will have a fundamental important as it will be the process from where the pedestrian detector accuracy can be increase by the use of the multi camera system. For instance, detections from Cameras 2 and 3 will be reprojected into Camera 1. Sometimes, those reprojections will not be in the frame because the cameras will not be aiming to the same spatial area, however, when the are seeing the

same spatial space cameras will share those detections. Ideally, if the three cameras detect the person reprojection will not be necessary, but, if one of them misses the detection, the other two detections, by the use of reprojection, will lead to suppress that miss detection.

4.4 Statistical Usage Data

Along this Section the extraction of the statistical usage data will be deeply explained

---

4.5 Induced Plane Homography

Promediado de semanticas projectadas en la imagen cenital para las 3 camaras.

Puntos comunes entre promedidados de semanticas para pares de camaras.

Puntos comunes entre los promedidados de las tres camaras -> Primera hipotesis de puntos pertenencientes al suelo. Extraccion de scores para esos puntos.

Generacion de planos paralelos al cenital para proyectar nuevos puntos comunes a otra altura.

Chapter 5

Results

5.1 Application Performance Results

5.2 Hardware

The project has been developed in the Escuela Politécnica Superior (Universidad Autónoma de Madrid). Due to this fact, the testing environment has been the hall of the mentioned engineering school which has a set up of three different Internet Protocol Cameras (IP Cameras). This type of cameras can send and receive data via a computer network and the Internet which allows the user to set the configuration and receive frames from the cameras.

5.2.1 Camera Specifications

Specifically, the camera model used along the project has been the Sony SNC-RZ50P PTZ Camera . This is a PTZ camera which means that it will be able to Pan, Tilt and Zoom all over the scene are. Precisely this camera will have a pan range of 340 degrees and a tilt range of 115 degrees, enabling users to monitor a wide area over the scene if the camera is moved (Figure 5.1) .

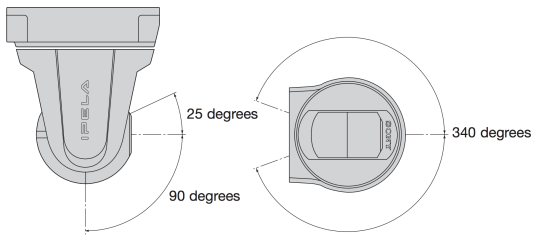


Figure 5.1: Sony SNC-RZ50P Pan/Tilt Range diagram

The complete and relevant specifications are detailed in Table 5.1:

Camera	
Horizontal viewing angle	1.7 to 42.0 degrees
Focal length	f = 3.5 to 91.0 mm
F-number	F1.6 (wide), F3.8 (tele)
Minimum object distance	320 mm (wide),1,500 mm (tele)
Pan angle	-170 to +170 degrees
Pan speed	300 degrees/s (max.)
Tilt angle	-90 to +25 degrees
Tilt speed	300 degrees/s (max.)

Image		
Image size (H x V)	640 x 480, 320 x 240, 160 x 120	
Compression format	JPEG, MPEG-4, H.264	
Maximum frame rate	JPEG/MPEG-4	25 fps (640 x 480)
	H.264	8 fps (640 x 480)

Table 5.1: Sony SNC-RZ50P Specifications

5.2.2 Camera User Web Interface (GUI)

The camera comes with a built-in web interface that will help us to visualize the visual range and set the different parameters that would change the camera behavior. The most important features that we will be able to tune are described as follow:

- Camera control: Through this interface one can control and set the position of the camera in terms of pan, tilt and zoom (Figure 5.2). Changes in this three variables will lead to different visualizations of the scene.

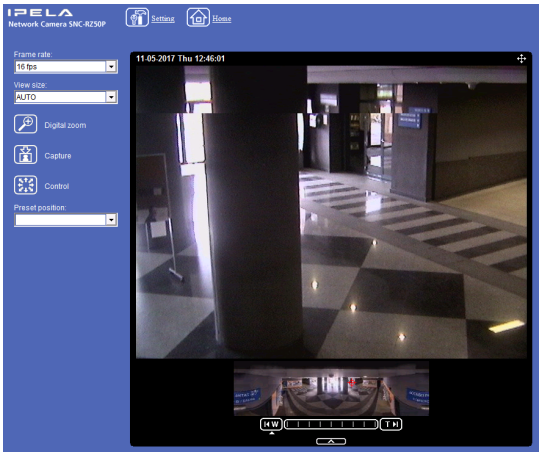


Figure 5.2: Visualization and control menu

- Preset position: In this menu (Figure 5.3) one could save the position that has been set in Figure 5.2 in order to recover the same position if the camera has been moved before in precise and easily manner.

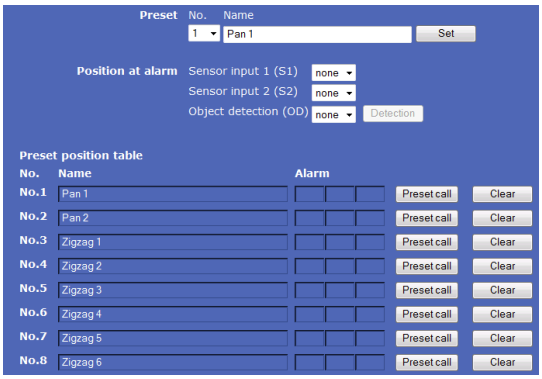


Figure 5.3: Preset position setting menu

- Tour setting: One could apart from pointing the camera to a static point, set it to describe a tour over the scene. This process is done by setting at least two different preset positions from where the camera will be moving from one to the other at a set speed. The menu to configure this behavior is displayed in Figure 5.4.

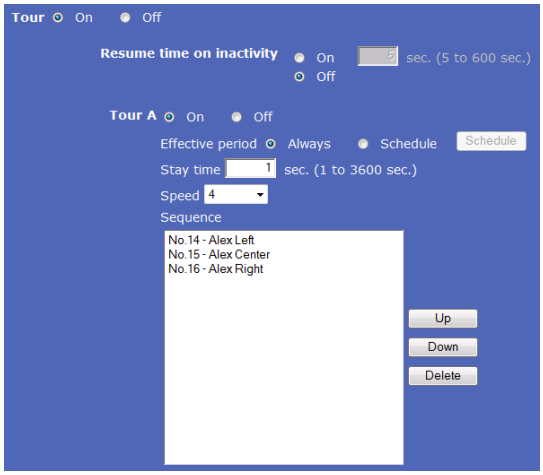


Figure 5.4: Tour setting menu

Chapter 6

Conclusions and Future Work

6.1 Conclusions

6.2 Future Work

Bibliography

[1] B. Zhou, H. Zhao, X. Puig, S. Fidler, A. Barriuso, and A. Torralba, "Semantic understanding of scenes through the ade20k dataset," *arXiv preprint arXiv:1608.05442*, 2016.

[2] H. Zhao, J. Shi, X. Qi, X. Wang, and J. Jia, "Pyramid scene parsing network," *arXiv preprint arXiv:1612.01105*, 2016.

[3] P. F. Felzenszwalb, R. B. Girshick, D. McAllester, and D. Ramanan, "Object detection with discriminatively trained part-based models," *IEEE transactions on pattern analysis and machine intelligence*, vol. 32, no. 9, pp. 1627–1645, 2010.

[4] R. Girshick, "Fast r-cnn," in *Proceedings of the IEEE International Conference on Computer Vision*, pp. 1440–1448, 2015.

[5] A. K. Jain, L. Hong, and Y. Kulkarni, "A multimodal biometric system using fingerprint, face and speech," in *2nd Int'l Conf. AVBPA*, vol. 10, 1999.

[6] X. Wang, "Intelligent multi-camera video surveillance: A review," *Pattern recognition letters*, vol. 34, no. 1, pp. 3–19, 2013.

[7] P. Dollar, C. Wojek, B. Schiele, and P. Perona, "Pedestrian detection: An evaluation of the state of the art," *IEEE transactions on pattern analysis and machine intelligence*, vol. 34, no. 4, pp. 743–761, 2012.

[8] S. Ren, K. He, R. Girshick, and J. Sun, "Faster r-cnn: Towards real-time object detection with region proposal networks," in *Advances in neural information processing systems*, pp. 91–99, 2015.

[9] J. Hosang, R. Benenson, P. Dollár, and B. Schiele, "What makes for effective detection proposals?," *IEEE transactions on pattern analysis and machine intelligence*, vol. 38, no. 4, pp. 814–830, 2016.

[10] R. Mazzon and A. Cavallaro, "Multi-camera tracking using a multi-goal social force model," *Neurocomputing*, vol. 100, pp. 41–50, 2013.

[11] A. Turner and A. Penn, "Encoding natural movement as an agent-based system: an investigation into human pedestrian behaviour in the built environment," *Environment and planning B: Planning and Design*, vol. 29, no. 4, pp. 473–490, 2002.

[12] P. Scovanner and M. F. Tappen, "Learning pedestrian dynamics from the real world," in *Computer Vision, 2009 IEEE 12th International Conference on*, pp. 381–388, IEEE, 2009.

[13] F. Jiang, J. Yuan, S. A. Tsaftaris, and A. K. Katsaggelos, "Anomalous video event detection using spatiotemporal context," *Computer Vision and Image Understanding*, vol. 115, no. 3, pp. 323–333, 2011.

[14] N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection," in *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on*, vol. 1, pp. 886–893, IEEE, 2005.

[15] P. Felzenszwalb, D. McAllester, and D. Ramanan, "A discriminatively trained, multiscale, deformable part model," in *Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on*, pp. 1–8, IEEE, 2008.

[16] P. Dollár, R. Appel, S. Belongie, and P. Perona, "Fast feature pyramids for object detection," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 36, no. 8, pp. 1532–1545, 2014.

[17] Á. García-Martín and J. M. Martínez, "People detection in surveillance: classification and evaluation," *IET Computer Vision*, vol. 9, no. 5, pp. 779–788, 2015.

[18] I. P. Alonso, D. F. Llorca, M. Á. Sotelo, L. M. Bergasa, P. R. de Toro, J. Nuevo, M. Ocaña, and M. Á. G. Garrido, "Combination of feature extraction methods for svm pedestrian detection," *IEEE Transactions on Intelligent Transportation Systems*, vol. 8, no. 2, pp. 292–307, 2007.

[19] R. Cutler and L. S. Davis, "Robust real-time periodic motion detection, analysis, and applications," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 22, no. 8, pp. 781–796, 2000.

[20] J. Giebel, D. Gavrila, and C. Schnörr, "A bayesian framework for multi-cue 3d object tracking," *Computer Vision-ECCV 2004*, pp. 241–252, 2004.

[21] K. Okuma, A. Taleghani, N. d. Freitas, J. J. Little, and D. G. Lowe, "A boosted particle filter: Multitarget detection and tracking," *Computer Vision-ECCV 2004*, pp. 28–39, 2004.

[22] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Advances in neural information processing systems*, pp. 1097–1105, 2012.

[23] P. Dollár, C. Wojek, B. Schiele, and P. Perona, "Pedestrian detection: A benchmark," in *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*, pp. 304–311, IEEE, 2009.

[24] A. Ess, B. Leibe, and L. Van Gool, "Depth and appearance for mobile scene analysis," in *Computer Vision, 2007. ICCV 2007. IEEE 11th International Conference on*, pp. 1–8, IEEE, 2007.

[25] C. Wojek, S. Walk, and B. Schiele, "Multi-cue onboard pedestrian detection," in *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*, pp. 794–801, IEEE, 2009.

[26] P. Arbeláez, J. Pont-Tuset, J. T. Barron, F. Marques, and J. Malik, "Multiscale combinatorial grouping," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 328–335, 2014.

[27] C. L. Zitnick and P. Dollár, "Edge boxes: Locating object proposals from edges," in *European Conference on Computer Vision*, pp. 391–405, Springer, 2014.

[28] D. Navon, "Forest before trees: The precedence of global features in visual perception," *Cognitive psychology*, vol. 9, no. 3, pp. 353–383, 1977.

[29] R. A. Rensink, J. K. O'Regan, and J. J. Clark, "To see or not to see: The need for attention to perceive changes in scenes," *Psychological science*, vol. 8, no. 5, pp. 368–373, 1997.

[30] B. Manjunath, P. Salembier, and T. Sikora, "Introduction to mpeg 7: Multimedia content description language. ed," 2002.

[31] S. Oh, A. Hoogs, M. Turek, and R. Collins, "Content-based retrieval of functional objects in video using scene context," in *European Conference on Computer Vision*, pp. 549–562, Springer, 2010.

[32] M. Cordts, M. Omran, S. Ramos, T. Rehfeld, M. Enzweiler, R. Benenson, U. Franke, S. Roth, and B. Schiele, "The cityscapes dataset for semantic urban scene understanding," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 3213–3223, 2016.

[33] Z. Wu, C. Shen, and A. v. d. Hengel, "Wider or deeper: Revisiting the resnet model for visual recognition," *arXiv preprint arXiv:1611.10080*, 2016.

[34] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 770–778, 2016.

[35] P. F. Alcantarilla and T. Solutions, "Fast explicit diffusion for accelerated features in nonlinear scale spaces," *IEEE Trans. Patt. Anal. Mach. Intell*, vol. 34, no. 7, pp. 1281–1298, 2011.

[36] A. Miguélez and R. Nieto, "Detección de personas en entornos multicámara utilizando informacion contextural," Master's thesis, Universidad Autónoma de Madrid. Escuela Politécnica Superior, 2016.