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
June 2017

université de BORDEAUX

Pázmány Péter Catholic University
Faculty of Information Technology and Bionics

UAM UNIVERSIDAD AUTONOMA DE MADRID

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
June 2017

Proyecto parcialmente financiado por el gobierno español bajo el proyecto TEC2014-53176-R (HAVideo)

Abstract

The following project has two main objectives. The first one aims to develop a system capable of perform pedestrian detection and semantic segmentation over a multi camera system. The fusion of both disciplines leads to combine pedestrian and semantic information in order to constrain detections. Also, the multi camera system is used to reproject detections from one camera to the others. In addition, statistical data usage of concrete semantic area is also extracted. Second objective is related to the development of a Graphical User Interface (GUI) from which algorithms can be controlled and results can be displayed.

In order to achieve these objectives a state of the art studied has been done. Pedestrian detection approaches have been deeply reviewed including those that need object proposals. Also, recent trends nowadays and next steps in the scope os Pedestrian Detection have been canalized. In addition, actual state of the art regarding contextual information and in concrete semantic segmentation has been studied. Finally, multi camera scenarios have been described.

A multithread application has been developed and analyzed to, as said in the objectives, present results and tune algorithms.

A new system has been proposed in order to achieve the objectives and perform pedestrian detections under different filtering and fusion conditions. Detectors such as HOG, DPM or PSP-Net have been integrated and a complete semantic segmentation has been performed. Both information has been combined in a common developed frame.

Finally, system performance has been tested in a generated dataset and manually annotated ground truth.

Keywords

Pedestrian detection, multi Camera, semantic segmentation, cenital plane.

Acknowledgements

Alejandro López Cifuentes.
2017.

Contents

List of Figures

List of Tables

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 and to ease our lives on everyday tasks. These range from biometric systems [2] 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 [3].

The combination of these veins of research could lead to the automation of high-level human semantic tasks such as people detection [4], object detection and recognition [5, 6, 7] and extraction of contextual information [8]. 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 on the analysis of a single-camera point of view, which will lead to a simple scenario in which the potential actions/events to detect will be observed from a single point in the scene or, on the analysis of a multi-camera setup. This last configuration will arise multiple benefits when analyzing big spaces as it will provide the user different views of the scene disambiguating occluded areas in the mono-camera views.

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 [9, 10, 11] and temporal usage of a given area in large-scale scenarios such as shopping malls,

universities and, generally, public-use buildings. Analysis range from the extraction of statistical measures of behavior to the detection of anomalous unexpected events [12]. This results may come from a combination of complementary algorithms such as contextual and semantic area classification, people detection and crowd behavior analysis.

1.2 Objectives

The main objective of this thesis is to extract contextual descriptions from a large-scale populated scenario while extracting temporal statistical usage data from relevant scene spaces using multiple 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 -under a user-friendly environment-statistics from different areas of interest in a public space either pre-generated or generated under a real-time constraint.

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, classify and determine the spatial extend on each frame of the video of relevant elements such as doors, chairs, 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 to a common space and combined such that by result's refinement the individual detector performances are increased.

(b) Increase people detection algorithms performance by the use of semantic constraining information to suppress false detections.

1.3 Thesis Structure

The master thesis is divided into the following chapters:

- Chapter 1. Introduction.
- Chapter 2. 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 analysis of a public crowed space embraces many different algorithms from Computer Vision disciplines.

This Chapter aims to study the State Of the Art pedestrian detection approaches and how to improve these as it constitutes a pivotal field for our goals. In addition, it also covers the topic of contextual information and in concrete, semantic segmentation and its algorithms. Finally as both of these disciplines are used in a multi-camera setup, scenarios working with this configuration are also defined, presenting their advantages an disadvantages.

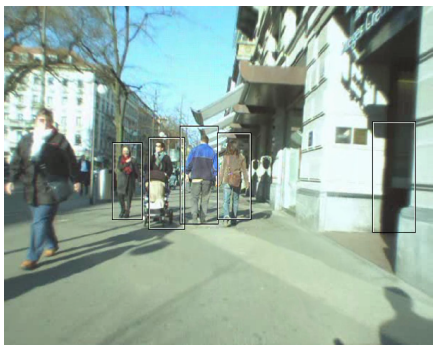
2.1 Pedestrian Detection

Pedestrian detection (PD) has been a major issue in computer vision during the past few years due to its relevance in all sort of computer vision applications. Its main objective is to detect and identify a potential object as a person by automatically detecting its position and relative size in the scene.

Nowadays, it can be consider a partially-solved problem. Although there are excellent pedestrian detectors in the literature, there is no algorithm able to effectively perform on a generic scenario. This is one of the reasons why PD is still one of the most researched areas in computer vision.

The complexity related to pedestrian detector lies on the large amount of available data-sets with different video and people characteristics including challenges such as: as people occlusions, different poses and scales and under different illumination conditions. Caltech [13] (recorded on a vehicle in an urban environment.), ETHZ [14] (recorded from a chariot which moves through pedestrian paths), TUD [15] (static camera in a crossing campus scene) or INRIA [16] (collects precise people images

(a) Caltech [13]



(b) ETHZ [14]

(c) TUD [15]

(d) INRIA [16]

Figure 2.1: Image summary from data-sets.

both static and moving) are some of the many literature data-sets that have been used through the years to train different pedestrain detectors. Figure 2.1 gathers some examples that can be obtained from the different mentioned data-sets.

2.1.1 Pedestrian Detection Approaches

Some people detection approaches are presented thorugh the following section. The PD State of the Art analysis is divided into three different classifications. Firstly, pedestrian descriptor and detection scheme is explained. Secondly, the person model and the algorithms to detect potential objects are analyzed. Finally, new trends in PD are briefly introduced.

2.1.1.1 Description and detection scheme

An organization of existing PD approaches based on the descriptor and detection scheme may start with the ones that use Histograms of Oriented Gradients (HOG) in combination with linear Support Vectors Machine (SVM) in order to describe

pedestrian shape with gradients to create a model and classify possible candidates [16]. Differently, Discriminative Part Models (DPM) such as [17] propose to 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.

Aggregate Channel Features [18] propose to 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.

2.1.1.2 Object detection and Model

In [1] an extensive evaluation of the state of the art of people detection is given. Here algorithms are characterized and differentiated from the others depending on the object detection approach and the used person model.

Object detection is defined as the extraction of the possible candidates to be a person from a scene. Mainly object detections algorithms are:

- Sliding window: -Also known as- exhaustive search leads to an efficient classifier to test, in search of pedestrian, every possible image window. Parameters such as window size, or overlapping between them are common tuning values that increase or decrease the performance of the detector. These methods usually need from 10^4 to 10^5 windows per image and this number grows 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 line of research introduces segmentation as a preliminary step for PD. Algorithms such as background subtraction lead to moving parts of the image that 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 directly produces those scene candidates to be a person and so the computational time is highly decreased.

- Segmentation + Exhaustive search: Alternatively, a combination of both previously analyzed techniques. In this case the previous step of segmentation does not lead to final candidates but to a delimited small area that could contain some candidates objects. After the segmentation process a sliding window technique is performed over the reduced scene area and so, the final candidates are extracted. In this case improvements from both approaches are exploited as the computational cost of the exhaustive search (which is its main drawback) is

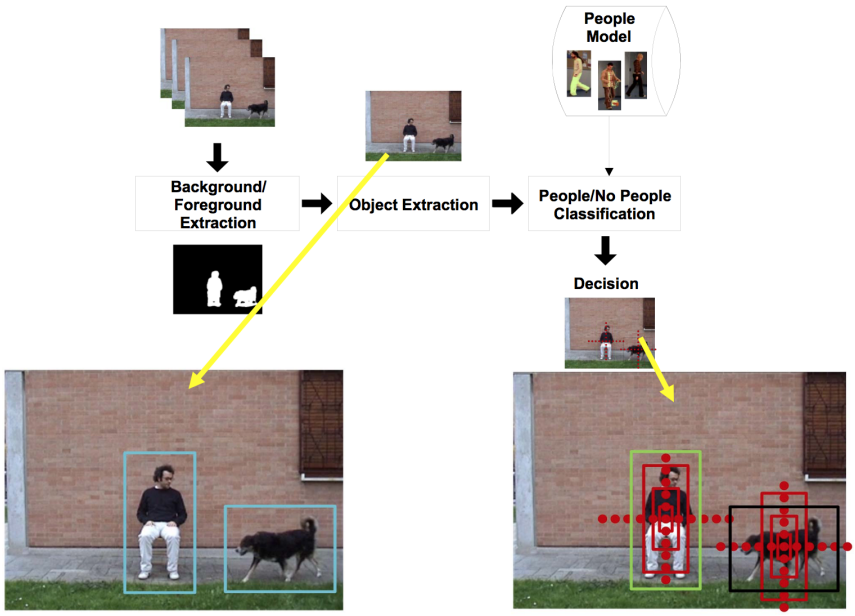


Figure 2.2: Generic Pedestrian Detector Example Diagram. Adapted from [1]

reduced by the use of segmentation. Figure 2.2 depicts a flowchart for a generic PD approach which relies on the combination of segmentation and exhaustive search.

The model of a person can be considered as the set of characteristics that are able to discriminate between people and any other object in the scene. In [1] three different kinds of person model are presented:

- Based on appearance: Most of the available detectors in the State of the Art use appearance information to define the person model. In this group of person models one can differentiate two approaches to describe the shape of a person.
  - ✧ Holistic: In these models people are defined as a unique and indivisible region or shape.
  - ✧ Part-based: These models however, rely on more complex characteristics where a person is defined as a combination of multiple shapes, regions or parts of its body.

  Examples of appearance-based detectors are those that use silhouettes to classify people, either from an holistic or part-based basis, or color distribution in people. However the group of appearance-based PD is mainly covered with approaches based on tthe use of Histograms of Oriented Gradients like [16], Haar-like fea-

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	56	57.5	48	63	69.1	69.8	**79.5**

Table 2.1: Pedestrian Detection Performance. Adapted from [23] (selected approaches). Metric for this evaluation is the average AUC. This metric measures the area under a Recall Over Precision curve. See VPU Website for the complete table.

tures [19], or combination of multiple features and aggregate channel features -e.g. the ACF [18] which combines HoG, gradient and color information-.

- Based on motion: Human appearance is likely to change due to environmental factors such as light conditions, cloths or camera settings. In addition, people variability in terms of height, weigh and poses make appearance likely to vary. For these reasons, some approaches try to get rid of these factors and detect pedestrians using only its motion information. In [20] 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: Algorithms such as [21, 22] merge both appearance and motion approaches in order to improve results. Most of these algorithms combine people detection and tracking, and so, they have been designed to improve people tracking over video sequences rather than to outperform State of the Art pedestrian detectors.

In [23] a comparison of the performance of PD on different data-sets is made. This comparison derives to a ranking which is partially included in Table 2.1.

2.1.1.3 Recent Trends in Pedestrian Detection

However, last years a new scheme of detectors have started to being used. Detectors based on deep Convolutional Networks have improved notably the accuracy of all the previous analyzed algorithms. Examples such as ImageNet [24] for image classification and Faster R-CNN [25] for object detection expose that deep convolutional networks usually improve the performance of aforementioned approaches. This fact is clearly presented Table 2.1 where Faster R-CNN outperforms every other approach.

2.1.2 Object Proposals

Described PD such as HOG, DPM or ACF utilized the analyzed "sliding window" algorithm, however, as mentioned before one of the main drawbacks of this approach is the final computational time needed to achieve good performance results.

One of the most successful solutions to overcome this time consumption problem without losing detection quality is the use of object proposals [7].

Object proposals approaches perform a complete search over an image to detect potential object candidates. These candidates are detected as image areas with visual properties that distinguish them from the scene background.

In general, object proposal approaches reduce pedestrian candidates respect to sliding-window like algorithms. Leading to a higher object recall and increasing the efficiency of the detection process. Successful examples of using proposal to improve and speed-up detection include the aforementioned ImageNet [24] and Faster R-CNN [25].

In [7] three set of proposal methods groups are proposed and analyzed:

- Grouping proposal methods attempt to generate multiple, and so, overlapping segments that are likely to correspond to objects. Here one can distinguish between three types of methods according to how they generate proposals. Methods can generate proposals by groping super-pixels (SP), solving multiple graph cut (GC) problems or finally, using edge contours (EC). Among those that use SP we can find Selective Search [26], Randomized Prim's [27], Rantalankia [28] or Chang [29]. Those that use GC are CPMC [30], Endres [31] or Rigor [32]. Finally Geodesic [33] and MCG [34] use EC to obtain proposals.

- Window scoring proposal methods are an alternative approach to score each candidate window according to the probability to contain an object. Usually this methods tend to be faster and, in addition, they typically extract only bounding boxes. One can find among other approaches Objectness [35], Rahtu [36], Bing [37], EdgeBoxes [38], Feng [39], Zhang [40], RandomizedSeeds [41].

- Alternative proposal methods. Apart from the main groups a set of alternate approaches such as ShapeSharing [42] or Multi-box [43] are also used to extract object proposals.

2.1.3 Next Steps Towards Generic PD

As said before PD is constantly in development and so, some future work lines can be set. In [4] 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 achieved by extracting useful contextual information from the scene. This 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 appear occluded. When this happens PD performance is substantially degraded under even mild occlusions. Improvements in this area could increase the performance in real life situations.

2.2 Contextual Information

One can describe contextual information as the set of additional circumstances or facts that one can extract from a scene besides the target of analysis. Generally, this set of circumstances is a wealth of information that is not extracted by the image analysis but extracted by humans based on previously acquired knowledge during their lifes. By just taking a look to an outdoor image a human can derive where the sky will be, what the weather conditions or which time of the day is. Also, by knowing the place the photo was taken, one would imagine which objects are more or less probable to appear in the scene.

Dealing with computer vision disciplines, contextual information sources also include camera information (such as position, configuration, distance to an object and camera motion), as well as set of objects that one could detect in the scene or how many views are available.

All this set of information that is not extracted from the image analysis provides a more general understanding about the scene and helps with further algorithms. We can divide contextual information into two different levels: global and local contextual information. Besides, we can also further divide contextual information two different categories offline, and online. Finally, semantic information is also analyzed.

2.2.1 Global

Global context considers image detections from the image as a whole. For instance, if the context of a scene is known -kitchen- we can use this information to search for typical objects in this context -e.g. a stove-.

This kind of approaches are focused on psychology studies that suggests that human perceptual processes work following a hierarchically organized process [44, 45]. Our perception system goes from a global structure towards a more detailed analysis in a top-down scene interpretation.

Global context approaches aim to define a scene as an extra source of global information. The structure of a determinate scene image can be estimated by means of global image features [46].

2.2.2 Local

On the contrary, local context refers to contextual information related to an object, e.g. a kitchen table may help predict the presence of a spoon.

The impact area of an object -in contextual terms- is defined as a set of other objects, patches or even pixels. Algorithms dealing with the extraction of local contextual information aim to correctly define the area that surrounds an object to precisely detect other object instances.

2.2.3 Offline

Offline contextual information is defined as the set of circumstances that are computed before starting any kind of analysis procedure. This information leads to external image information that is used to constraint analysis algorithms. Approaches such as [1] uses previously introduced contextual information to improve part-based pedestrian detections over a scene.

2.2.4 Online

Online information, on the other hand, is computed with the analysis. Online extraction entails a degradation of an algorithm efficiency, albeit allows to dynamically update the context.

Examples of online (and local) contextual information extractors are the algorithms in the semantic segmentation branch. Next section (2.2.5) discuss some of the approaches in this vein.

2.2.5 Semantic Segmentation

Semantic information is defined as the set of high-level elements from contextual information. This can often lead to characteristics such as global image color [47]. It can also describe an image by the position and status of relevant objects [48]. Or it

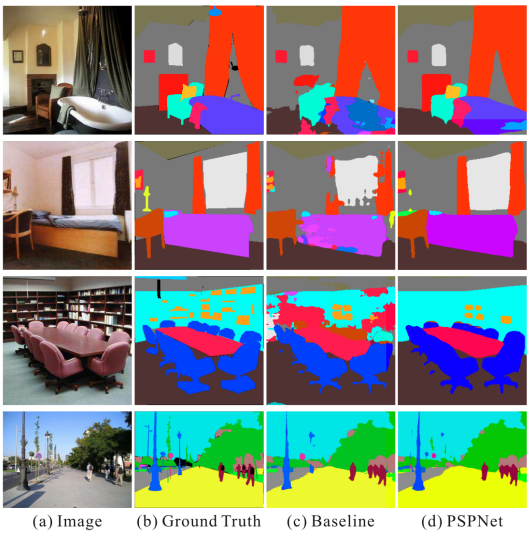


Figure 2.3: Scene parsing on ADE20K data-set [49]. Adapted from [8].

also defines concrete image areas in a scene such as walls, corridors or walking paths [8].

In other words, semantic segmentation has the goal to assign each image pixel a high level category label. If accurately done, it provides complete semantic understanding, which in terms of computer vision, it means that location of an object within an image will be known. The result of semantic segmentation is depicted in Figure 2.3, where scenes are analyzed and divided into separate semantic areas.

Semantic segmentation is a recent trend and nowadays, it remains a significant challenge for the computer vision community. Due to its short-life term there is not yet a complete survey available in which algorithms are deeply analyzed and compared. However, there are a set of benchmark suits where developers can upload their obtained results with a given dataset and so, algorithms performance can be compared.

An example of benchmark is Cityscapes Data-set [50]. It is a large-scale data-set that contains stereo video sequences recorded in street scenes from among 50 different cities around the world. This data-set presents class annotations over pixels in more than 5000 frames. This set of categories and classes are presented in Table 2.2.

In Table 2.3 we can observe an adapted version from Cityscapes Website in which some of the most accurate algorithms are presented. Only those algorithms that have more than 80% on IoU class metric have been mentioned in this thesis for practical reasons.

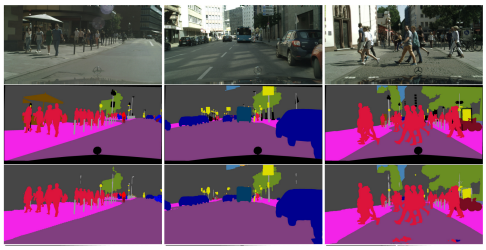
PSPNet, ResNet-38 and TuSimple_Coarse are all based on convolutional net-

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

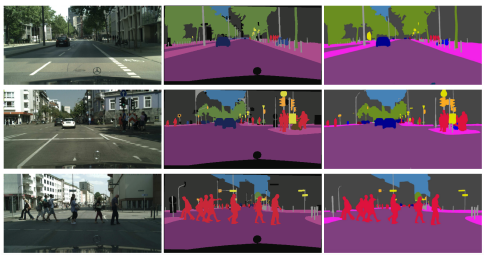
Table 2.2: Cityscapes Data-set Class Definitions

Algorithm Name	IoU Category	IoU Class	Available Code
motovis (Anonymous)	**91.5**	**81.3**	No
PSPNet [8]	91.2	81.2	Yes
ResNet-38 [51]	91.0	80.6	Yes
NetWarp (Anonymous)	91.0	80.5	Yes
TuSimple_Coarse [52]	90.7	80.5	No

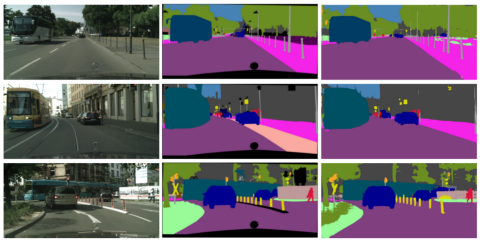
Table 2.3: Cityscapes Data-set Challenge Results (Intersection over Union metric)



(a) ResNet-38 Algorithm. From top to bottom: Input images, ground truth and results.



(b) PSPNet Algorithm. From left to right: Input images, ground truth and results.



(c) TuSimple_Coarse Algorithm. From left to right: Input images, ground truth and results.

Figure 2.4: Semantic Segmentation result examples on Cityscapes Data-set.

works. This fact reveals that deep convolutional neural networks (CNNs) have led to significant improvement over previous semantic segmentation systems since the presentation of AlexNet [24] in 2012.

However, even when using CNNs the main difficulty of scene parsing is related to the type of scene and to label variety. PSPNet deals with this problems assigning relationships between different labels, i.e. an airplane is likely to be in runway or flying in the sky while not over a road or in the water. This relationships reduce slightly the complexity of having large amounts of labels to predict and improve the general performance of the algorithm. ResNet-38 on the other hand, propose not only to not increase CNNs depth, but rather to ensemble many relatively shallow networks to increase performance. Their approach also improves usability reducing memory use and sometimes even training time. Finally, TuSimple_Coarse propose a combination between dense upsampling convolution (DUC) to generate pixel-level prediction and a hybrid dilated convolution (HDC) framework.

In Figure 2.4 some visual examples of how this algorithms perform on Cityscapes Data-set frames are displayed.

2.3 Multi-camera scenarios

The use of various cameras is a common setup when dealing with video surveillance problems. One can define as a multi-camera scenario one space that has more than one video camera recording. Ideally, the recordings for the different cameras are temporally aligned -synchronized-. Having N multiple camera instances allows to observe the same event or object of interest from at least two different points of view. This leads to a set of advantages in the scope of our work when dealing with PD and semantic classification and also, to some unavoidable disadvantages.

- Advantages

  As said in Section 2.1 one of the main research paths in PD is the occlusion treatment. In this case, the use of a multi-camera scenario with computed homographies between camera frames could help. This could be achieved by reprojecting detections from one camera to another whose miss rates are high [53].

  When talking about semantic segmentation the inclusion of a multi-camera system could 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 [54].

- Disadvantages

  The main disadvantage when dealing with multi-camera systems is the exponential grow of computational time. Algorithms should be performed N times. This issue could be solved by the use of parallel coding that perform cameras processing simultaneously.

  Besides, the use of multiple cameras entail two additional tasks: the temporal alignment of the views -synchronization- and the spatial arrangement of the different view -homographies-.

Chapter 3

Developed Application

Within this Chapter the developed application in terms of software developement will be analyze. This application will be the base for the calculation, svisualization and arrangement of the usage statistics from the different areas of interest, pedestrian detections and other results. 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 a correct and easy usage by the end user.

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

1. Its cross-platform characteristic which makes it easily portable from one operating system to another such as Windows or Linux distributions.

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

3. The possibility to add OpenCV libraries to the project as well as independent external libraries.

4. Its multi-thread capabilities that enables to perform different code segments in various threads to increase computational speed.

Due to the complexity of some of the algorithms used in terms of parameter tuning and configuration two separate applications have been developed, engineer and user version.

The first version of the software corresponds to the engineer application. As its name says, it is design so it can be used by programmers or engineers who generally understand concepts of the algorithms running at the backend application. This

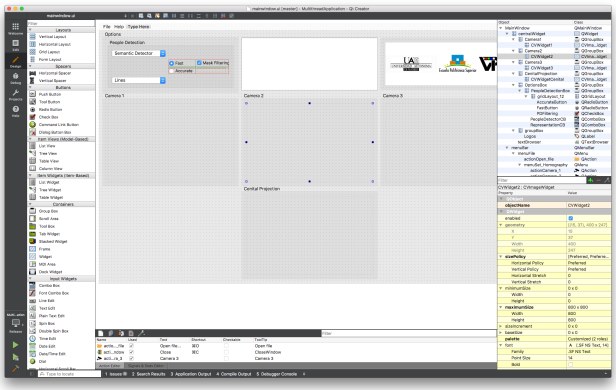


Figure 3.1: QT Main Window Designer

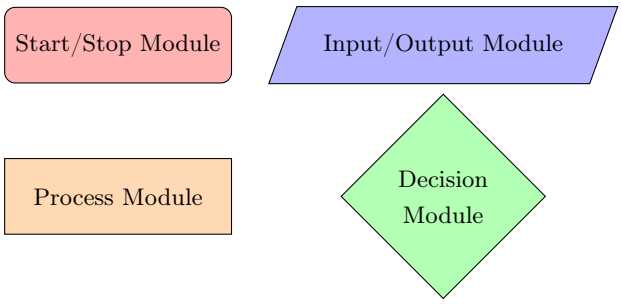


Figure 3.2: Flow-chart legend

means that all the variable parameters are available for tuning from the graphical interface, as well as different detector options and visualization. The final motivation for this version is to achieve high variability and tuning in terms of algorithm parameters, however, in order to have good performance the user should have a basic knowledge about how parameters are affecting the software performance.

For the engineer application both single-thread and multi-thread versions have been developed and are analyzed on the following lines. 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 are displayed. The used legend can be observed in Figure 3.2.

3.1 Single-thread Engineer 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 that illustrates the execution path can be seen in Figure 3.3 .

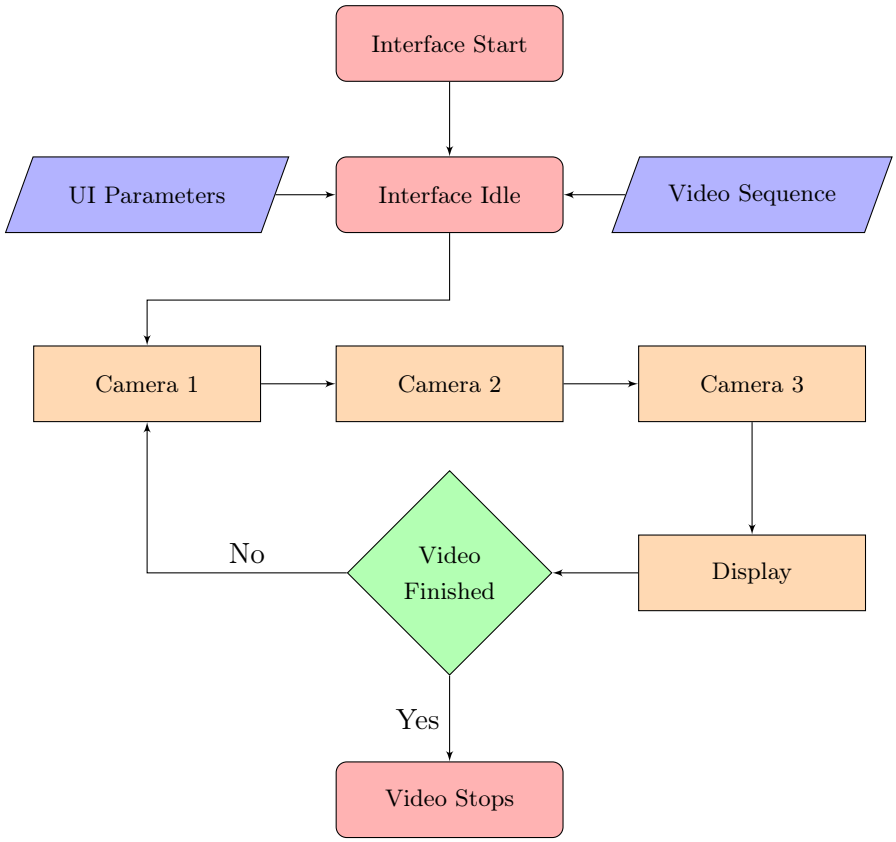


Figure 3.3: Flow-chart diagram for the single-thread application. NO DEFINITIVO

This approach has the advantage that all the code is executed in the same segment which makes really easy for instance to share information between cameras as the share the same memory segments, however, this design will only be valid if the computational effort is minimum. Before displaying any results, all the process for the three cameras should be computed one after the other. This means that when calculating detections for one camera the others will remain idle. 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 Engineer Application

The new and used approach can be observed in the flow-chart displayed in Figure 3.4. As one can observe, now different threads are running in parallel, one for each

camera, 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 to keep consistency in the application. One thread can process a frame faster than another one due to multiple external reasons, 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 diagram 3.4. This barrier will create a meeting point for all the threads that all of them must reach before continuing with their execution.

In the code two barriers will be available. The first one will ensure that all the threads have perform pedestrian detection before sharing those detections between the rest of the threads. The second one will be at the end of a frame execution, so threads wait to each other before sending results to the main window for representing processes.

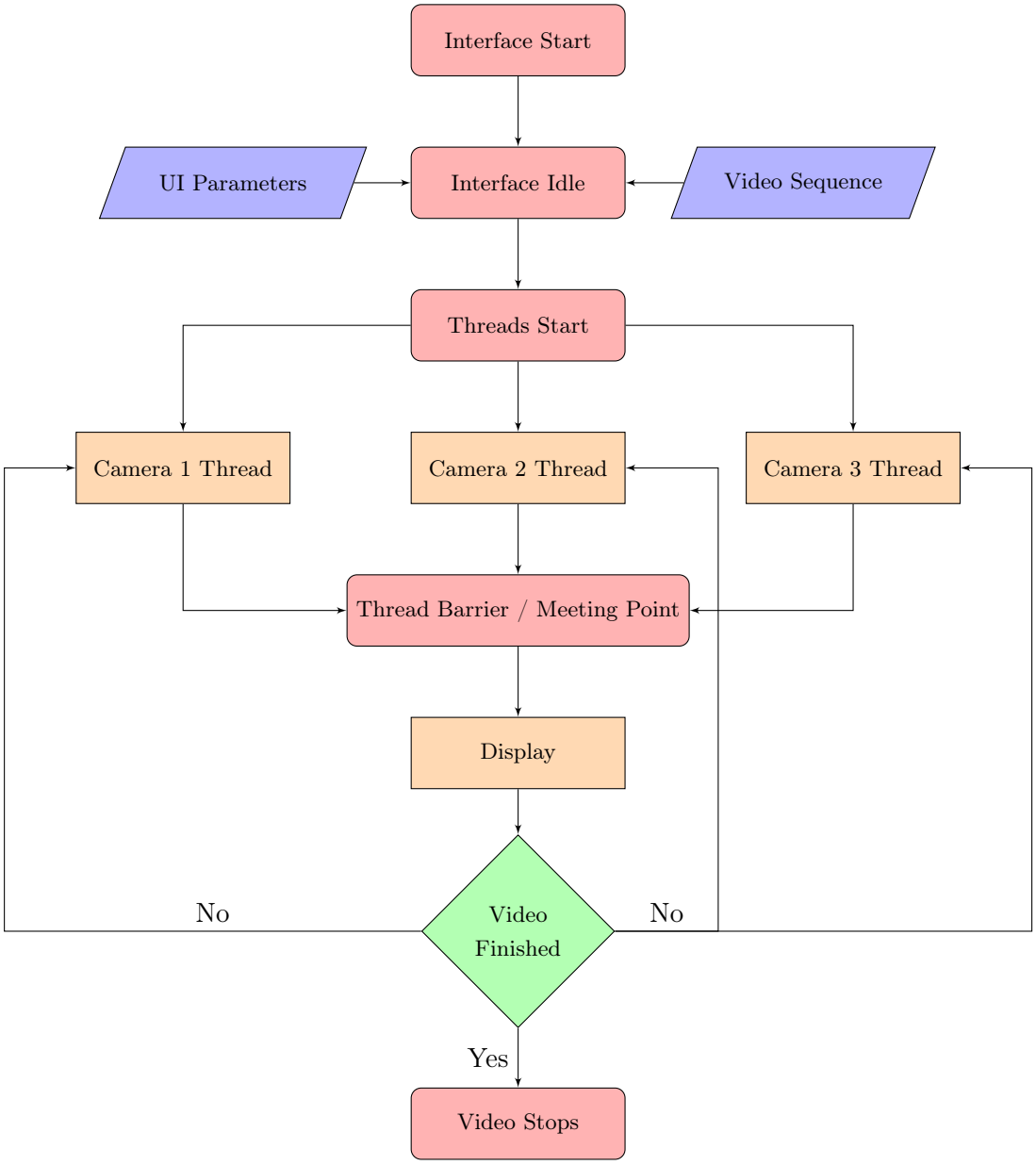


Figure 3.4: Flow-chart diagram for the multi-thread application. NO DEFINITIVO

Main application window is shown in Figure 3.5. As one can observe it is compose of four separate areas:

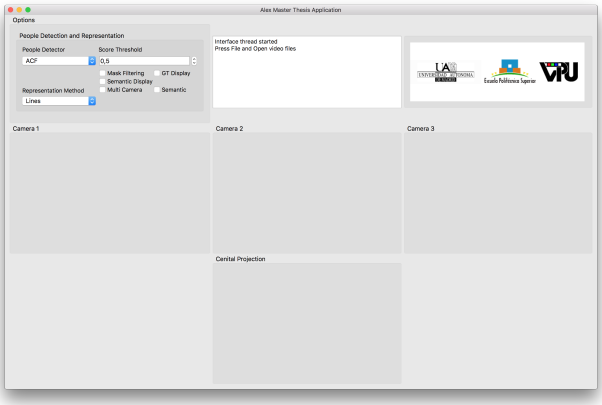
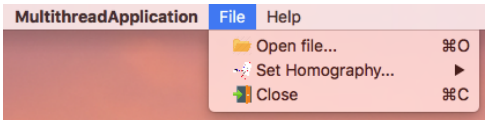


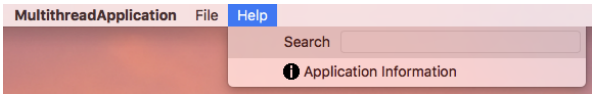
Figure 3.5: Main application window.

1. Application menu bar.

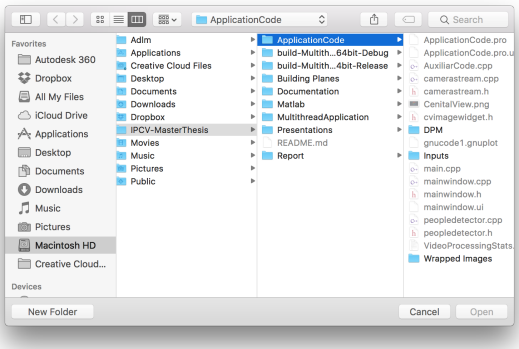
   In this menu we will have the main application actions. From here the user can open a new video sequence, compute the set of needed homographies for the algorithms or close the program. The application also provides a help searcher and an external information window. This set of functions can be seen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



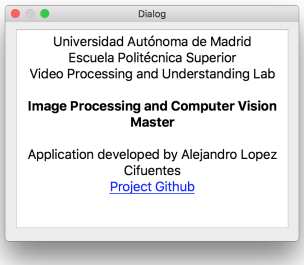
(a) File Submenu



(b) Help Submenu



(c) File open dialog.



(d) Application information window.

Figure 3.6: Application menu bar

2. Options menu.

The 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 PSP-Net detector, DPM, ACF or even Fast-RCNN as well as the type of representation that one wants for these detections, lines or gaussians which will be explained in Section 4.3.2. It also enables the user to select the desire pedestrian detection threshold as well as the filtering or constraint that is used. One can select between regular pedestrian detection, PD with semantic constraining or to perform multicamera projection. In addition mask filtering can be also perform so PD search area is reduced. Finally, a check box to display or not ground truth information is available too.

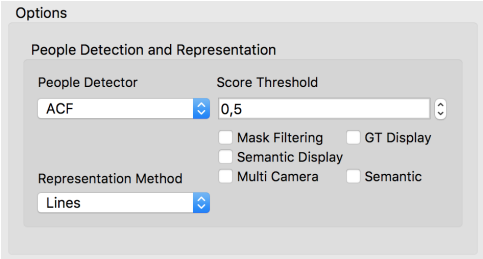


Figure 3.7: Options Menu.

3. Information Display

   Along this text box some guide 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 is in use. This can observe in Figure 3.8.

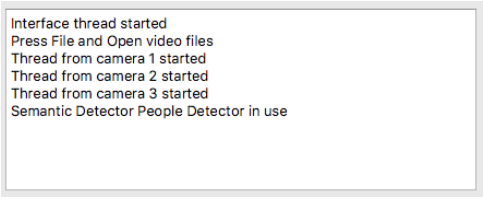


Figure 3.8: Information Display

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 or/and ground truth 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.

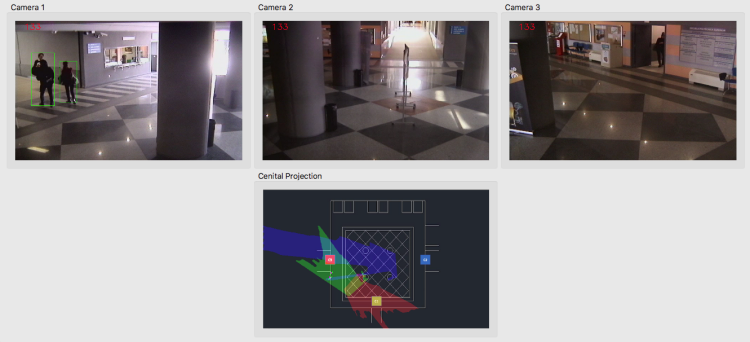


Figure 3.9: Results Display Area

3.2.1 Classes Distribution

In terms of C++ blocks the application has been divided into several classes for a better code design and easier understanding. Their 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 as everything will be inherit from this class. The functions that are going to be done by this class is creating file open dialogs, setting up and start all the camera threads, update all the algorithms configurations with value from the UI and displaying results through the CVImageWidget class. It is also responsible of the information sharing between threads.

- AboutWindow. Class that execute the second available window. This instance will display general information about the application and its developers.

- CameraWorker. Main class for all the execution in each of the cameras. CameraWorker class is linked with a unique thread that will process all the algorithms inside. It has CameraStream, PeopleDetector, Evaluation and Barrier classes declared within it to distribute the processing.

- CameraStream. This class will perform all the functions that has to deal with

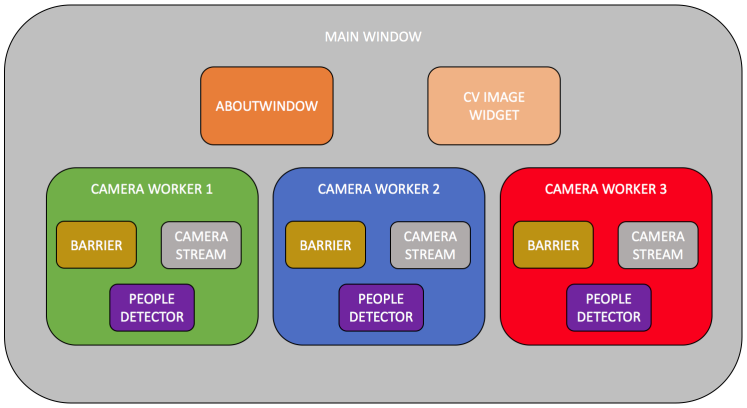


Figure 3.10: Hierarchical representation of the code. NO DEFINITIVO. FALTA CLASE EVALUACION. CAMBIAR LOS COLORES DE PD Y CS

the video processing except pedestrian detection. The main reading loop, homographies and semantic projections 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 pedestrian either on the camera frame or the cenital plane will be from this class.

- Evaluation: Here ground truth will be read and also the evaluation between system pedestrian detection and ground truth information will be performed.

- Barrier. This class will implement everything that has to deal with thread synchronization. It will be declared in Main Window and passed by arguments to the thread so each of them has the same exact barrier to perform the synchronization.

- CVImageWidget. Display representation class that will deal with all the process to finally paint OpenCV Mat images into a QT main window interface Widget.

In Figure 3.10 a hierarchical representation of how the different classes are related in the code is presented. As on can observe everything is under the heritage of the main window. Here we have the three camera worker threads which will have inside the same barrier, and each of them a camera stream, a people detector and a evaluator. In addition, main window will also be the parent of the extra about window and the CVImageWidget.

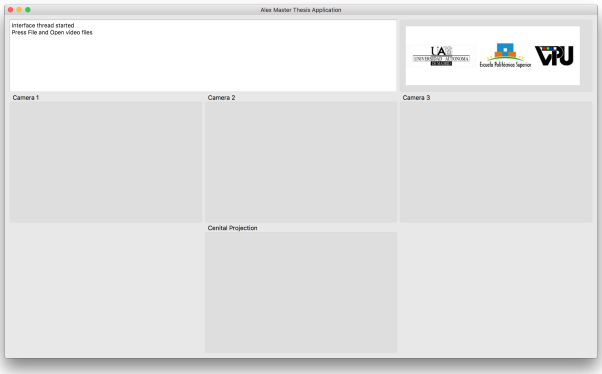


Figure 3.11: User version application main window.

3.3 Multi-thread User Application

On the contrary to the analyzed version, the second developed application is focus on general users. This version will only allow to load the video files and display results. All the parameters will be set by default so the user does not have to fine tuning any of them making the usage and the general perception of the application much more simple and easy. This method will ensure that achieved results will be the best ones that the proposed system can obtain and will be the best approach to display demos.

Figure 3.11 displays the general application window where we can observe the difference with the Engineer version.

As said, conversely to the engineer application the only functionality of the main window is to display results and guide information. All the options present in the engenieer application for tunning are no longer available.

Chapter 4

Proposed System

During this Chapter our proposed system will be deeply explained. We start from the main contextual model generation and pedestrian detectors to finally explain the fusion of all the obtained elements in a multi camera system with semantic constraints. In addition the generation of statistical usage data from semantic areas will be proposed. Figure 4.1 represents the flowchart that the application will follow for the computation of the main mentioned algorithms.

4.1 Contextual Model Generation

One of the main objectives in the scope of this work is to permorm semantic segmentation, wchih as explained in Chapter 2 means to divide one frame into different semantic areas. The relative position in the scene for elements such as doors, walls, paths, and columns are necessary to achieve further objectives such as multi camera pedestrian constraint and statistical data extraction. For this complex task the algorithm PSP-Net [8] presented in Chapter 2 is used. We choose PSP-Net because at the moment of this work was the one with available code achieving the best results. The goal for this algorithm as said before, is is to assign each pixel in the image a category label achieving by this the location and 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 4.2.

Given the image in 4.2.(a) the first step is to process it through a pre trained CNN called ResNet [55] to get the full feature map 4.2.(b) of the last convolutional layer. The final feature map in this step is 1/8 of the size of the input image.

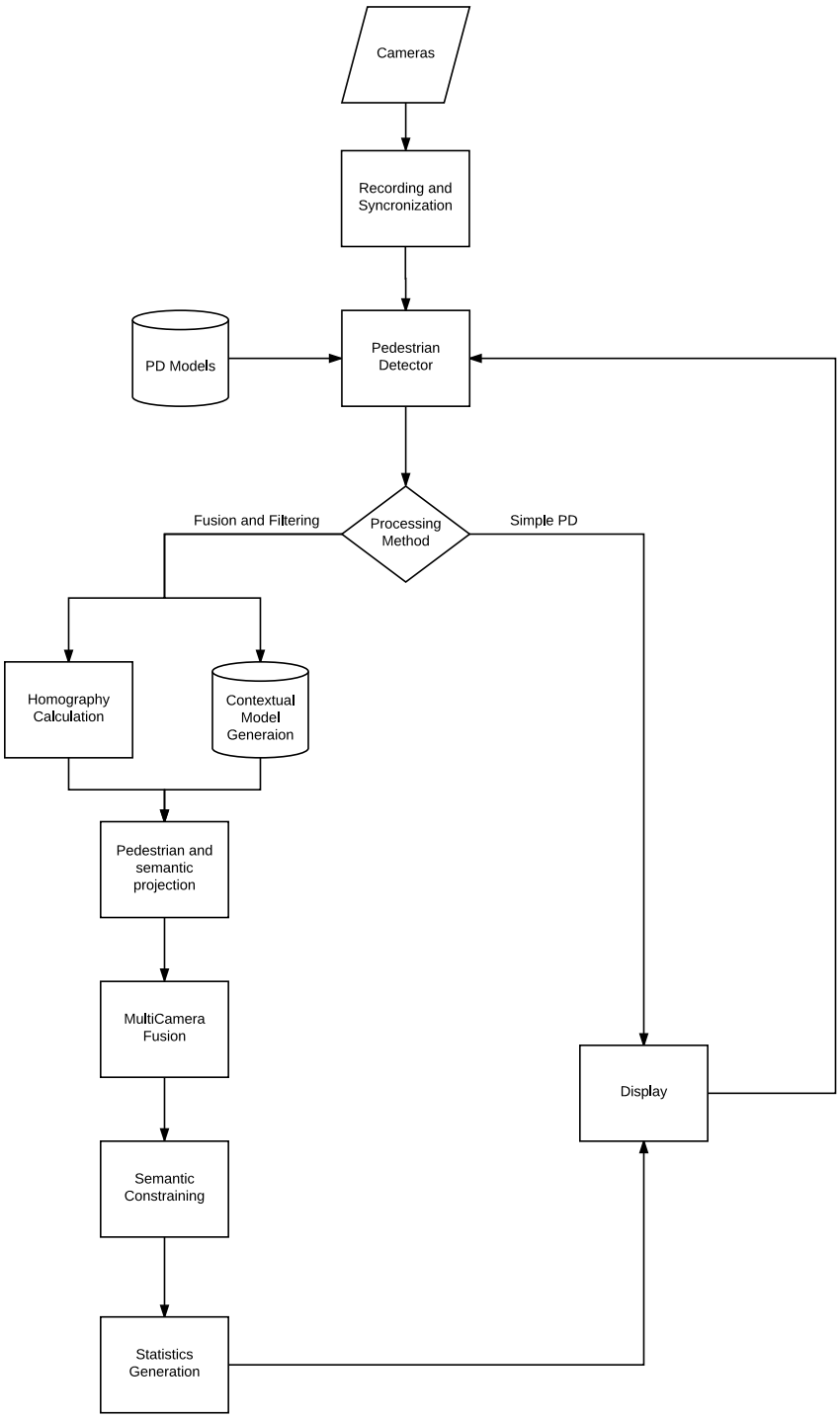


Figure 4.1: Proposed system flowchart

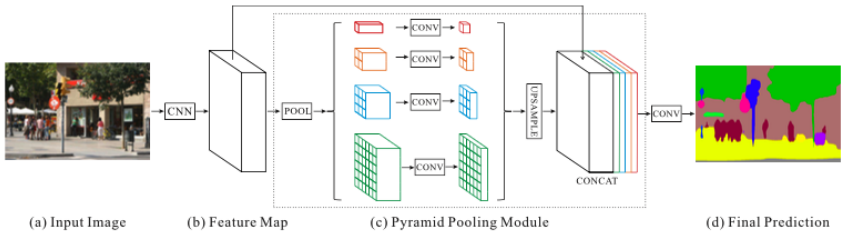


Figure 4.2: Overview of the proposed PSPNet

The second step is to apply the main contribution of [8], which is called the Pyramid Pooling Module 4.2.(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 locations. Here a set of pooling, convolutional and upsampling layers are applied to 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 4.2.(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 4.2.(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.

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

1. The model should have been trained with indoor scenes. This leads to discard those models that represent only outdoors scenes as we would like our approach

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

to be used in an interior scenario. This will exclude CityScapes dataset from our options as all the classes and sequences used for training are from outdoor scenes.

2. From the trained indoor models we have to choose between those whose categories best fit in our work. In this case VOC2012 dataset uses classes such as boat, airplane or table which are not interesting for our segmentation problem and it does not have classes such as door or wall which are really important for us.

With this two reasons being taken into account we have selected the model ADE20K because as said, 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 from the model has been reduced to the 21 classes of our interest. Position and scores for those objects are the only ones obtained. This class limitation leads above all in a considerably hard drive space saving. In Table 4.1 the final 21 selected classes are exposed.

4.2 Pedestrian Detection

Along this section we will present the pedestrian State of the Art detectors that have been selected to work in our system. Some of them have been chosen due to 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 OpenCV 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

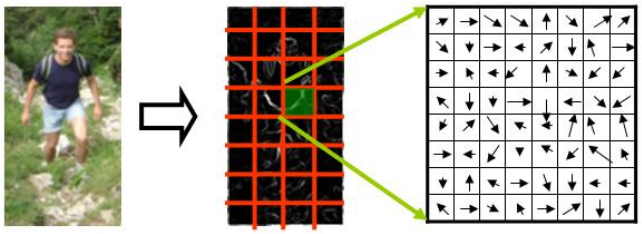


Figure 4.3: HOG Pedestrian Descriptor

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 4.3.

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 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 and the model.

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

The combination between HOG and SVM leads 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 mentioned occlusion 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 4.4.

Both global and part templates are modeled with histogram of gradient features

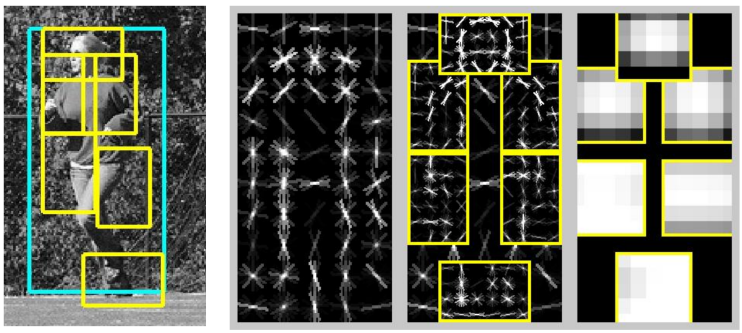


Figure 4.4: Example detection obtained with the DPM person model. From left to right: Original image + detections, global coarse model, part templates and spatial model for each part. [5]

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 the part filter score on the resulting sub window minus the deformation cost.

The use of this detector leads to a set of advantages than when working with others simpler approaches. In terms of pedestrian occlusion treatment we are able to detect those people that have been occluded by something in the scene just by detecting some visible part. This outperforms other detectors while working with crowded scenes in which holistic methods have problems that lead for instance to groups of people being detected as a unique detection while DPM is ideally able to separate them into different person instances.

However, its scanning window approach as well as the part based model 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 Aggregated Channel Features Detector

The basic idea behind ACF detector is to increase other approaches performance by the use of many different channels to describe an input image I. In Figure 4.5 one can observe the working path of the mentioned detector. Given an initial image I, several channels are computed. This channels are named $C = \Omega(I)$. After the computation, every block of pixels in C is summed and the resulting lower resolution channels are smoothed. After a vectorizing process features are single pixel lookups in the aggregated channels. Boosting trees are then used to learn this features (pixels) in order to distinguish people from the background. It is quite evidentially that the use

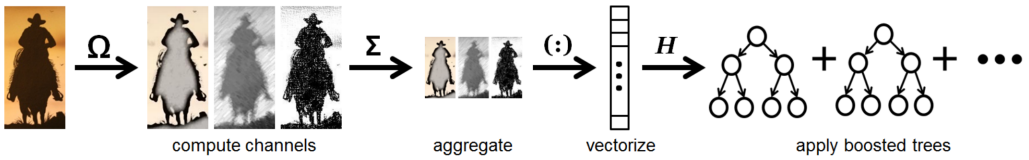


Figure 4.5: Aggregated Channel Features architecture. [18]

of the correct channels leads to better results. In [18] a set of 10 channels are used to achieve state-of-the-art performance in pedestrian detection:

1. Normalized gradient magnitude (1 channel).
2. Histogram of oriented gradients (6 channels).
3. LUV color channels (3 channels).

4.2.4 Fast Region-Based Convolutional Network

Fast Region-Based Convolutional Network (Fast R-CNN) is used to perform offline pedestrian detection in our proposed system, however, as said in Chapter 2 it is not only a pedestrian detector but an algorithm for object detection. In 2.1.2 we mentioned that Fast-RCNN must use objects proposals for its usage. In our developed system MCG [34] grouping method is used.

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 4.6 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 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

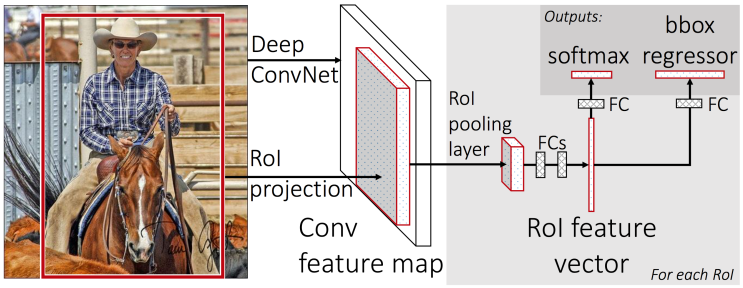


Figure 4.6: Fast R-RCNN architecture. [56]

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 and so, should be computed offline and introduced to the application.

4.2.5 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 approach by surrounding pedestrians labels with bounding boxes.

4.3 Fusion and Filtering

Once the semantic model and pedestrian detections are obtained the next step is to combine the information coming from different camera sources into a common plane. This is done in a process called fusion and filtering.

4.3.1 Multi-camera Scenario

Our system is developed in a multi-camera scenario. This is the essential setup to fusion information. This means that we are able to observe the scene from different point of views. In Figure 4.7 we can observe how the scene is configured with 3 static video-surveillance cameras. Two of them are placed at the sides of the scene, while one is at the bottom part. The starting views from the three static cameras are displayed in Figure 4.8.

As said in Chapter 2 working with a multi-camera scenario has some advantages. One can see in Figure 4.8 we can observe the same scene area from three different

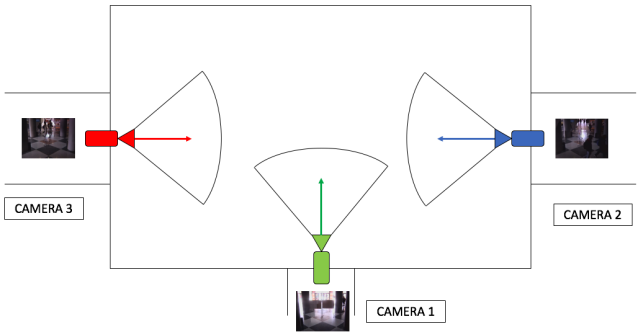
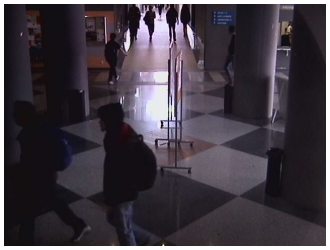


Figure 4.7: Multi-camera configuration

(a) Camera 1



(b) Camera 2

(c) Camera 3

Figure 4.8: Initial Camera Views

points of views, which means that for instance, 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 ${}^{\pi_{ref}}H_F$ that relates one camera frame F_t at a time t, to a so called cenital plane π_{ref}, i.e. a bird-eye representation of the scene. With this homography matrix we will be able to transform every frame pixel $F_t(x_1, y_2)$ to its correspondent position in the cenital plane $\pi_{ref}(x_2, y_2)$. Frame perspective is so, transformed as if it were being viewed from the top. The homography process is also important as it enables to project pedestrians detections to the cenital plane with the same proceeding.

The formulation of the generic homography problem between two cameras and a plane is as follows:

We have two cameras C_1 and C_2 looking at the same exact points P_i in a plane. One wants to pass from the projection $p_i(C_2) = (u_i(C_2); v_i(C_2); 1)$ of P_i in C_2 to the projection P_i in C_1 defined as $p_i(C_1) = (u_i(C_1); v_i(C_1); 1)$.

To do this process Eq 4.1 should be computed.

$$p_i(C_1) = \frac{z_i(C_2)}{z_i(C_1)} K_{C_1} \cdot H_{C_2C_1} \cdot K_{C_2}^{-1} \cdot p_i(C_2) \tag{4.1}$$

where $z_i(C_1)$ and $z_i(b)$ are the z coordinates of P in each camera frame and where homography matrix $H_{C_2C_1}$ is defined as in Eq 4.2.

$$H_{C_2C_1} = R - \frac{tn^T}{d} \tag{4.2}$$

where R is the rotation matrix by which C_2 is rotated in relation to C_1. t is the translation vector from C_1 to C_2. n and d are the normal vector and the distance to the plane respectively and finally K_a and K_b are the camera intrinsic parameters. All this variables can be observed in Figure 4.9.

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 a relationship between a real camera frame and a cenital view plane created by computer should be computed. This means that there is not 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.

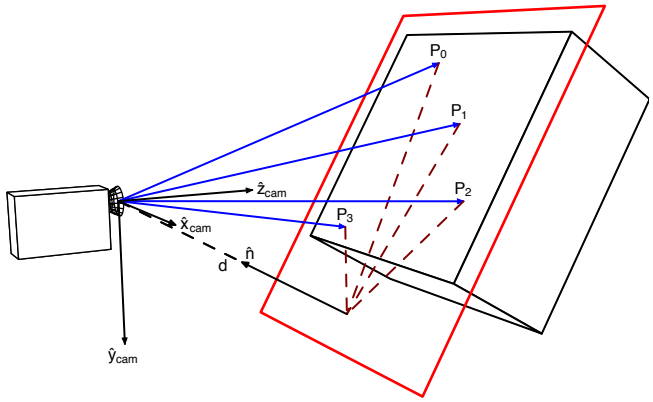


Figure 4.9: Generic homography problem. Camera and plane setup

Figure 4.10: First cenital plane approach

4.3.1.2 Cenital Plan Design

As said in the introduction of this Section, 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 that correctly represents the scene is needed. The first used approach for the cenital plane can be observe in Figure 4.10.

This 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 details so the point selection is done correctly by the user and the homography is correctly computed.

For this reason, and driven by bad results, 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.

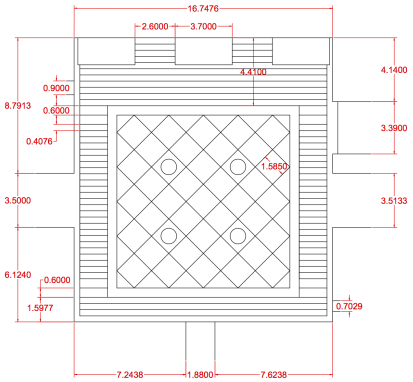


Figure 4.11: Second cenital plane approach with real measures

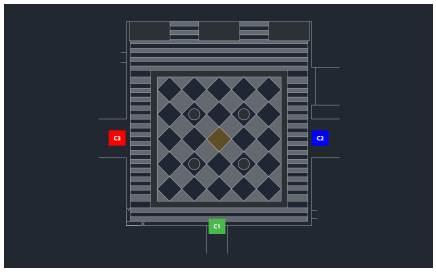


Figure 4.12: Second cenital plane approach with camera positions

For correctly drawing the plane AutoCAD 2017 software has been used. The second plane approach with all the manual measures extracted from the real scene superposed can be seen in Figure 4.11 and Figure 4.12 represents the final cenital map with the correct camera positions.

It is easily observable that differences between Figure 4.10 and 4.12 are outstanding both in floor details and in general construction proportions. This detail rise will lead to a much more easy homography selection points by the user as it now has more point options to choose and evidentially this means that the final homography matrix, and so, all the projections will be better accurately talking.

4.3.1.3 Camera Panning

When cameras are static and pointing to the center of the scene as in Figure 4.8 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. This approach is a limitation to the project objectives as we will not be able to extract a high percentage of the scene semantic. The solution for this problem is to include

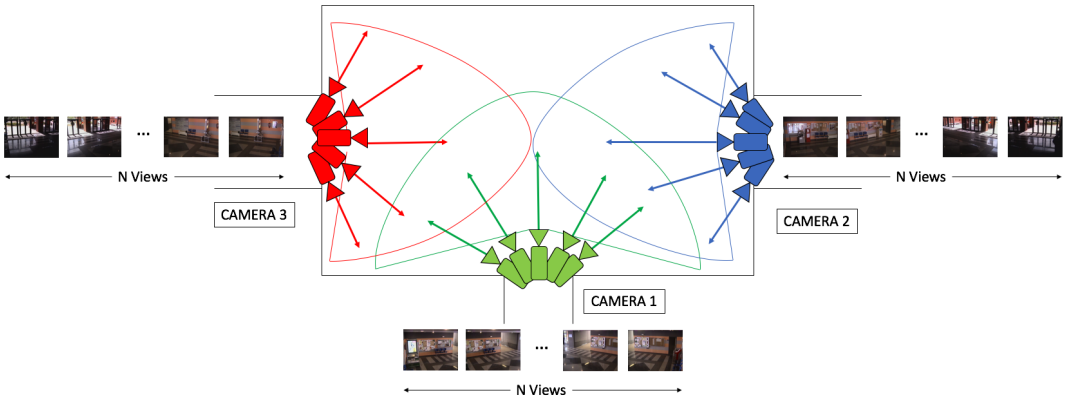


Figure 4.13: Multi-camera configuration with panning set up

panning movement thanks to the PTZ technology in all the cameras from left to right so cameras view range is increased and they are no longer focus only in the middle of the scene as before. This leads also to more common and not static areas which the cameras will share. This solution is presented in the scene diagram in Figure 4.13.

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 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 simple approach 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 leads to the obtention of an homography codebook in where the user will compute a homography matrix ${}^{\pi_{ref}}H_{View}$ 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 [57] has been used for this task.

AKAZE will detect and describe points of interest in any image and then by a brute force comparator in terms of distances we will extract how many coincidence we have between a pair of images.

DESCRIPCCION DE AKAZE

It is essential to say, that, as we have to compare the current frame with all of the N views we will have a trade off. More views will mean a better space discretization and more overlapping between them, 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. In our proposed system we have choose $N = 9$ to maintain a trade off between projection quality due to the overlap between the views and time consumption.

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 will not project the points correctly, there will be a small error. Taking advantage of the calculation of common descriptor 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 two homographies are used. First we change the perspective of the original frame to the perspective of the view with ${}^{View}H_{Frame}$. This process ensures that the used perspective is the same one as the one that was used previously to compute the homography. Finally, the frame and its new perspective are projected to the cenital plane by means of ${}^{\pi_{ref}}H_{View}$. This process is graphically explained in Figure 4.14.

4.3.1.6 Video Sequence Synchronization

One of the main issue when dealing with multi camera systems, and specially, those that combine information between the cameras, is that video sequences coming from them should be correctly synchronized. This is really important because if the same frame number f is not representing the same exact moment in time in the three cameras, combination and fusion of the information is not possible.

We have synchronized the processed videos using the so called clapperboard technique. This technique aims to synchronize using concrete events that can be observed from all the cameras at the same time. By this, we will be able to set a synchronize starting point from where the following video frames will be correctly representing the same moment in time.

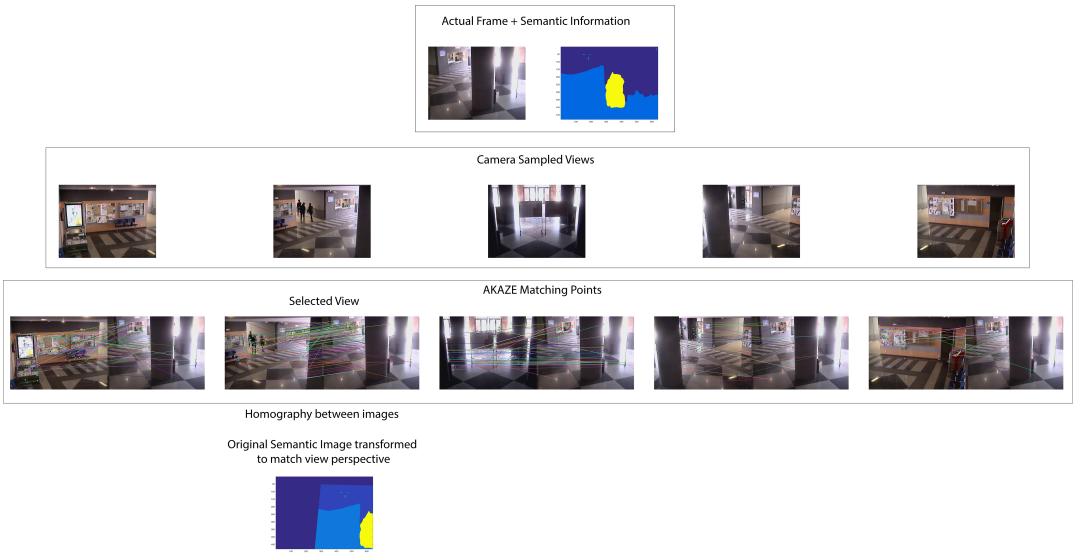


Figure 4.14: View selection process and homography between views computation. $N_{Views} = 5$. IMAGEN NO TERMINADA Y PENDIENTE DE CAMBIOS

4.3.2 Pedestrian Filtering and Constraining

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

4.3.2.1 Cylinder Estimation

In [53] a cylinder estimation technique is proposed. The detected bounding boxes on the camera frame do not correspond spatially with the exact 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 represented bounding box and the real object. Figure 4.15 shows the case for bounding box transference between cameras and Figure 4.16 for the cenital frame.

As we can observe in Figure 4.15 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 cylinder that embrace the square whose side is the bounding box (b). Once the cylinder is estimated, the person will be hypothetically in the middle of the cylinder. In (c) we can observe that the estimation, if correctly computed has great accuracy.

In Figure 4.16 the same method is applied but in this case, the bounding box is

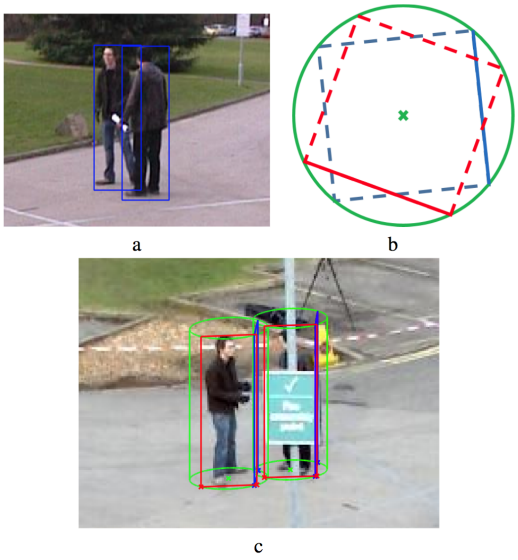


Figure 4.15: Cylinder estimation for camera instance projections

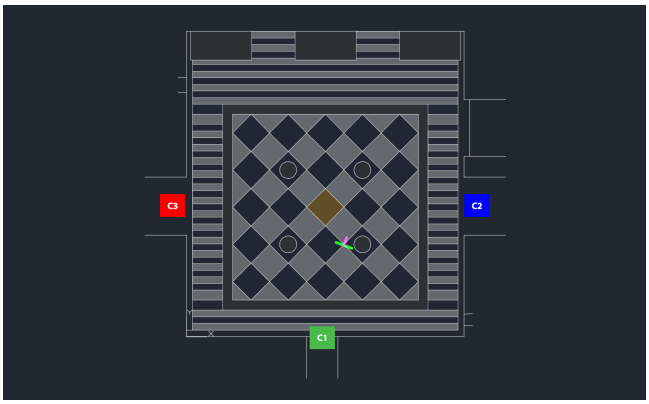


Figure 4.16: Cylinder estimation for cenital view projections.

not projected to another camera perspective but to the cenital plane. The projection of the bounding box, represented by the green line is not in the same position as the center of the cylinder, which 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.

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, in this representation one does have spatial information about the detection but none about the detection accuracy. This means that with the two perpendicular lines one can observed where the detection has been achieved but not the score associated with this detections. To solve this issue we propose a new representation method based on a Gaussian function placed at the middle of the cylinder.

Every pedestrian detection will be so represented as a Gaussian function of the form described in Eq. 4.3.

$$f(x,y) = Aexp\left(-\left(\frac{(x-x_0)^2}{2\sigma_x^2} + \frac{(y-y_0)^2}{2\sigma_y^2}\right)\right) \tag{4.3}$$

In this case A will be the amplitude which typically will be $A = 1$, x_0, y_0 will represent the mean of the gaussian which, in our case, will be the center of the cylinder associated to the bounding box and σ_x, σ_y which is the standard deviation and in our case will represent the accuracy of the detection.

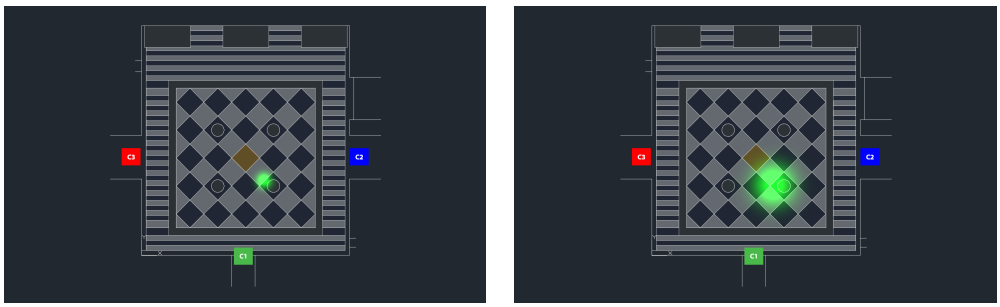
In order to relate a detection score S_n with a determinate standard deviation a simple rule explained in Eq 4.4 is proposed.

$$\sigma_x = \sigma_y = (S_n \cdot 10) + 5 \tag{4.4}$$

This Equation has been adapted to the size of the cenital plane. This means that detections with higher scores will be represented as a narrow gaussian function (Figure 4.17a) while lower scores will lead to wide gaussians that will represent a bigger area in where one can find that person (Figure 4.17b). Some Gaussians examples depending on the scores can be observed in Figure 4.17.

4.3.2.3 Pedestrian Reprojection Between Cameras

When all the pedestrian detections have been projected into the cenital plane one can start reprojecting those projections from the cenital plane back to the other camera



(a) High score detection. Frame 86 (b) Low score detections. Frame 87

Figure 4.17: Gaussian representation examples.

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.

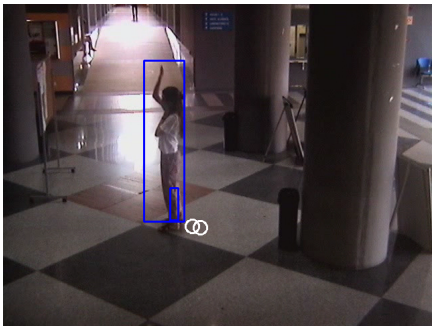
Projected frames from one camera to another will be treat as another detector blob. This means that if the detection is accurate enough in the original camera, when projected to another frame it will be either joined by a Non-Maximum Suppression (NMS) if that camera already has one blob, or maintained as a detection on the person. However, if the detection is not accurate enough projection will not be on the person and so that blob will lead to a false positive which is the main drawback of the method.

This process can be observed in Figure 4.18.

4.3.2.4 Pedestrian Semantic Constraining

Once all the detections have been reprojected into all the cameras one can use semantic information in order to filter false positives detections in weird scene areas . Ideally people are always walking on the floor so the common floor areas between pairs of cameras, whose calculation is deeply explained Section 4.4.2, in are used to constrain the pedestrian detections to this hypothesis. This process is done based on the cenital plane builded using the cylinder estimation explained before. The constrain idea is that a bounding box coming from a camera C_1 is assumed correctly computed if the

(a) Camera 1 + Reprojections from 2 and 3



(b) Camera 2 + Reprojections from 1 and 3

(c) Camera 3 + Reprojections from 1 and 2

Figure 4.18: Pedestrian detection reprojection.

(a) Camera 1

(b) Camera 2

(c) Camera 3

Figure 4.19: Pedestrian semantic constraining. Blobs that are not in the right semantic area are just displayed by a circle and text label.

center of its related cylinder is on the common floor between C_1 and C_2 and also between C_1 and C_3. An example of the constraining can be observed in Figure 4.19.

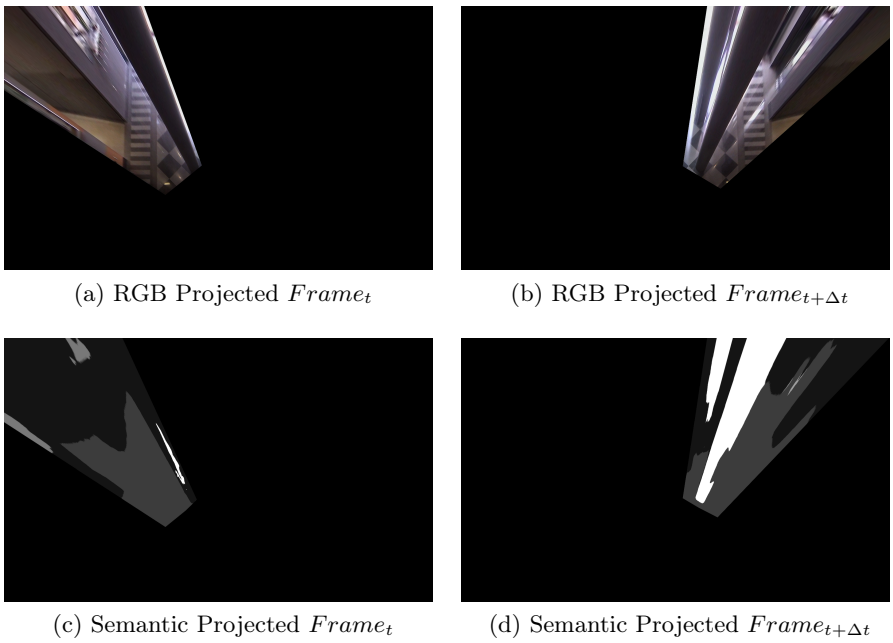
4.4 Reference and Inertial Homography Planes

During this Section we will explain our proposed method to create both RGB and semantic reference planes combining information from all the cameras positions. The final objective is to have complete maps in each camera that represent the scene.

4.4.1 Semantic and RGB Reference Plane Generation

Following with the process scheme of Figure 4.14 one can project every video frame into a cenital perspective. This way a set of projected RGB and semantic frames for each camera will be obtained. Some examples of these images can be observed in Figure 4.20.

Once all the video frames are projected we propose to apply a temporal average through a median filter for the three cameras separately. By this method a reference



(a) RGB Projected $Frame_t$ (b) RGB Projected $Frame_{t+\Delta t}$

(c) Semantic Projected $Frame_t$ (d) Semantic Projected $Frame_{t+\Delta t}$

Figure 4.20: RGB and semantic frames projected.

plane π_{ref-C} for each camera contained in the ground plane will be created by joining all the separated projected frames. This process is detailed in Eq 4.5(COMPLETAR ECUACION).

$$\pi_{ref-C} = \sum_{i=1}^{nFrames} Projected_i \tag{4.5}$$

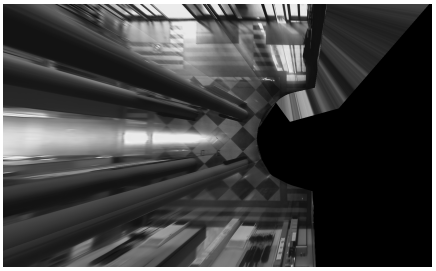
The result coming from the temporal averaging can be observed in Figure 4.21 for RBG case and in Figure 4.22 for semantic information.

As one can observe we get smooth semantic and RGB maps from all the hall area. It is important to recall that only if the homographies and the ground floor are correctly calculated and projected, the base of for instance, the columns from all the different cameras should lay in the same spatial point in these images.

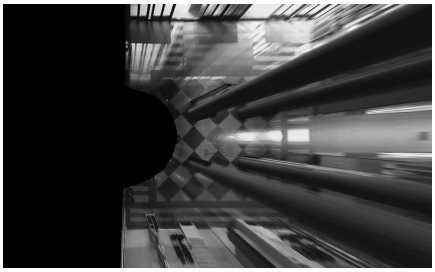
4.4.2 Semantic Fusion in the Reference Plane

Once the cenital maps have been extracted one can combine them to create semantic fusion in the reference plane. The main objective is to obtain common areas, i.e. pixels with the same label within two or more cameras. This method will allow us to increment the confidence for those pixels and so, we can use this information for example, to constrain pedestrians detections. As the reference plane is obtained by

(a) Camera 1. ${}^{RGB}\pi_{ref-1}$



(b) Camera 2. ${}^{RGB}\pi_{ref-2}$



(c) Camera 3.${}^{RGB}\pi_{ref-3}$

Figure 4.21: RGB Reference Planes ${}^{RGB}\pi_{ref-C}$.

(a) Camera 1 ${}^{Sem}\pi_{ref-1}$

(b) Camera 2 ${}^{Sem}\pi_{ref-2}$

(c) Camera 3 ${}^{Sem}\pi_{ref-3}$

Figure 4.22: Semantic Median Average ${}^{Sem}\pi_{ref-C}$.

(a) Cameras 1 and 2 ${}^{Sem}\pi_{ref-1} \wedge {}^{Sem}\pi_{ref-2}$ (b) Cameras 2 and 3 ${}^{Sem}\pi_{ref-2} \wedge {}^{Sem}\pi_{ref-3}$

(c) Cameras 3 and 1 ${}^{Sem}\pi_{ref-3} \wedge {}^{Sem}\pi_{ref-1}$

Figure 4.23: Common semantic areas between pair of cameras.

the homography to the ground plane we have only take into account floor pixels to create the common areas, as the others will have projection errors that will lead to fusion errors.

First, we have combine pairs of cameras to create three common semantic areas that are shown in Figure 4.23.

Once pairs of cameras have been combined one can go further and extract common areas for the three cameras at the same time. Those areas will be pixels with the exact same label for the three cameras so its probability to be that class is real high. Figure 4.24 represents the common areas.

As we can observe, due to the multi camera setup and some scene occlusions such as scene panels or columns, the common area shared by all the cameras represents a small portion of the room. This is the reason why we have choose common areas between pairs of cameras to work along with pedestrian detection rather than common area between all of them.

4.4.3 Parametric Homographies Between Inertial Planes

Homography matrices, as explained before, aim to relate floor plane for a frame with the cenital view. However, everything that is not exactly in the same floor plane is

Figure 4.24: Common semantic areas for all the cameras ${}^{Sem}\pi_{ref-1} \wedge {}^{Sem}\pi_{ref-2} \wedge {}^{Sem}\pi_{ref-3}$

not projected properly when the homography matrix is used. In our work, different semantic are needed to be projected, for instance a door. Using floor homography only its base is correctly projected.

In [58] a solution to this problem is proposed. The general idea is to create a multilayer reconstruction. Once the homography matrix ${}^{\pi_{ref-C}}H_{view}$ that relates the image view with the reference frame π_{ref-C} one can obtain another matrix that relates the same image frame with a parallel plane called inertial plane at a fixed height Δt.

H_v can be expressed as a function of H_v and Δh as described in Eq 4.6.

$$ {}^{\pi'}H_v^{-1}(\Delta h) = {}^{\pi}H_v^{-1} + \Delta h P \hat{\mathbf{k}}^T, \tag{4.6} $$

where $P = [u_0\, v_0\, 1]^T$ is the principal point of camera C and $\hat{\mathbf{k}}$ is the unit vector of the Z axis.

All this process is described in Figure 4.25.

By this process, ideally a number k of planes could be generate (Figure 4.26) in which different object sections will be correctly projected. It could lead to a complete semantic map in which all the pixels represent semantic areas that have been correctly projected.

4.5 Statistical Usage Data

Along this Section we explain the proposed system to extract statistical usage data frame by frame given a sequence. The aim is to obtain one graph per selected class that measures the amount of people at a moment t of time in the determined semantic area during the sequence.

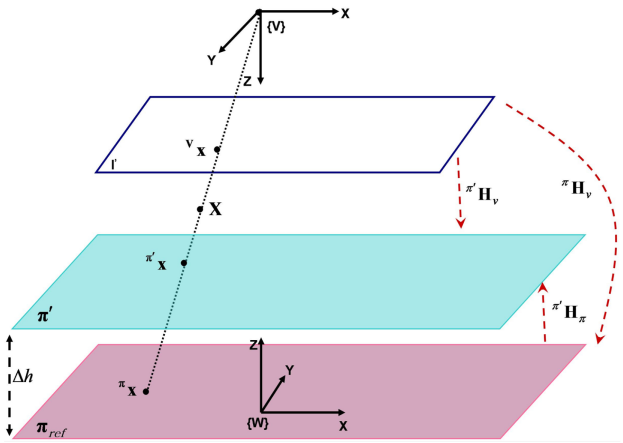


Figure 4.25: Extending homography for planes parallel to π_{ref}. ${}^{\pi_{ref}}H_V$ is the available homography between camera view and reference plane π_{ref} either semantic or RGB.

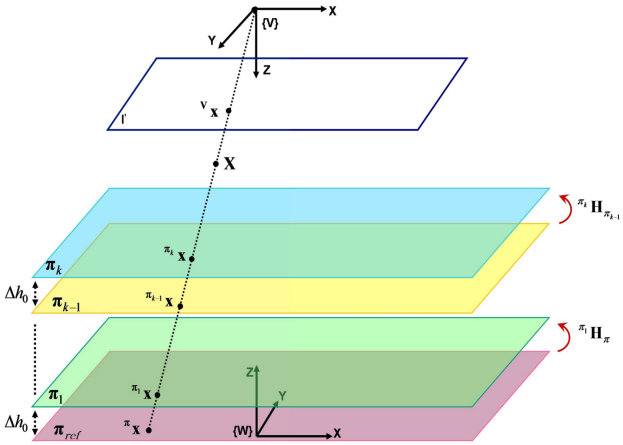


Figure 4.26: Set of k inertial planes π_k . Each inertial plane is separated from the other by the same Δh heigth

Figure 4.27: Statistical semantic map.

In order to extract usage data the first step is to create a projected common semantic map. For this task, rather than creating common areas between cameras as done for the PD fusion, information for all the cameras have been joined without strictly being shared. This have been done to preserve as much semantic as possible and so, have bigger floor or door areas than if we were more restrictive.

The result from this step is a unique map that represents the hole spatial area (Figure 4.27).

As one can observe we have decided to only take into account for this process floor, doors and chairs, which are the most accurate detections from the semantic segmentation. Following the same principle as in the previous Section, pedestrians will be projected on to the statistical semantic map and so we will be able to know how many people are in each of the designed areas. In addition,

Chapter 5

Results

During this Chapter all the achieve results are exposed and analyzed. Firstly, a brief analysis of the used hardware during the Thesis is done. Secondly, in detail, the experiment setup is explained, i.e. the generated dataset and ground truth, and the evaluation framework. Later, results concerning application performance are presented. Finally, the last two sections aim is to present pedestrian detection results in term of performance and usage data extraction.

5.1 Used 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 setup of three different Internet Protocol Cameras (IP Cameras). This type of cameras can send and receive data via a computer network and Internet which allows the user to set the configuration and receive frames from the cameras.

5.1.1 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. 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) .

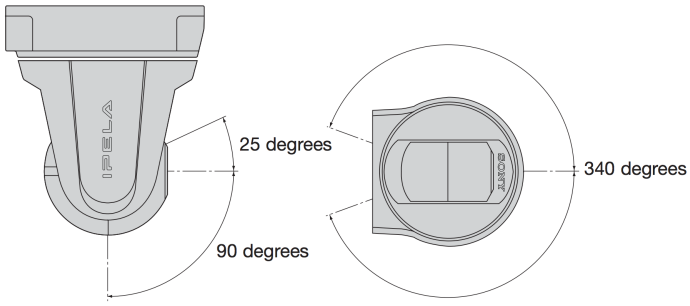


Figure 5.1: Sony SNC-RZ50P Pan/Tilt Range diagram

The complete and relevant specifications from the cameras are detailed in Table 5.1. In the scope of our work the most important features will be resolution and frames per second that will enable us to have more quality in both image and video.

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

5.1.2 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.

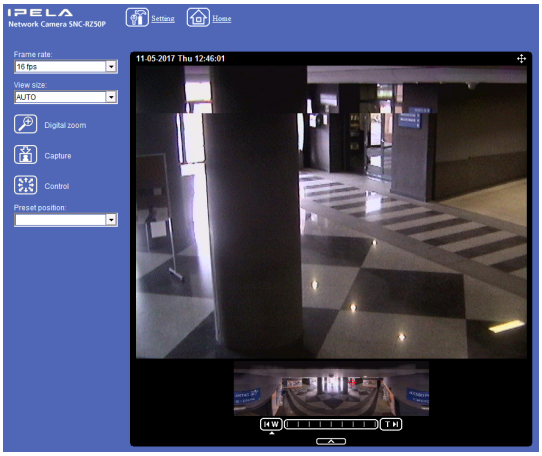


Figure 5.2: Visualization and control menu

- 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.

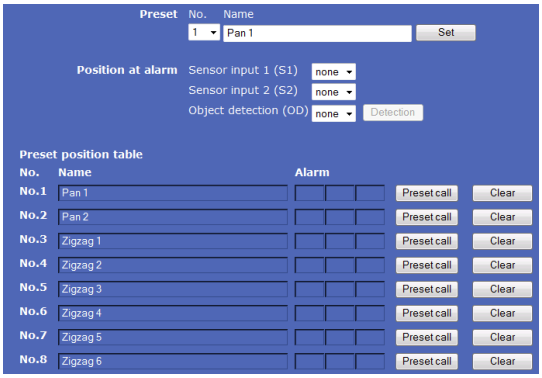


Figure 5.3: Preset position setting menu

- 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.

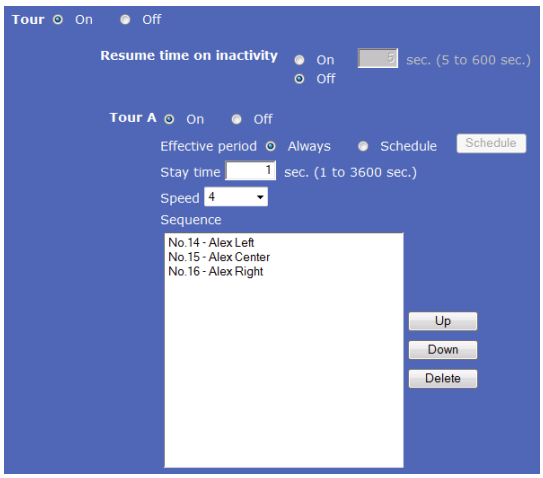


Figure 5.4: Tour setting menu

5.2 Experiment Setup

For all the experiments that are going to be analyzed along this Chapter the experiment setup should be explained. In this section, the generated dataset and ground truth and the proposed evaluation frame work used to obtain results will be detailed.

5.2.1 Dataset

As said at the beginning of the Chapter the dataset has been generated in the university hall. The number of available PTZ cameras in the university is three so we have been able to record different sequences from three points of view. In addition, as explained in the previous Chapter a tour for each camera has been set so they move from left to right only changing position in the X/horizontal axis and not in the Y/vertical direction because of the reasons presented in Chapter 4.

Both settings are configured as previously displayed in the diagram of Figure 4.13.

The complete dataset consists on a set of three different recordings about 5 minute long from three different days. In each of the recordings all the camera sequences are available. Figure 5.5 shows some example of them. The technical characteristics of the videos are the detailed in the following lines:

- Number of videos:

Figure 5.5: Dataset example frames

- Resolution: 640x480 pixels.
- Frame rate: 23.976 fps.
- Format: MPEG Video.

5.2.2 Ground Truth Generation

We have selected one of the three recordings from the dataset to completely evaluate the system. In order to do the evaluation ground truth information is needed. The video has been manually annotated with Via Annotation Tool to generate real pedestrian bounding boxes. This software allows the user to generate XML files with position and type of bounding box through the hole sequences. The recording consists on three different videos of 8300 frames, which means that the total number of manually annotated frames has been 25.200.

In Figure 5.7 annotated frames for the recording in different video situations can be observed.

As one can observe in Figure 5.7. The selected sequence for evaluation purposes is a mix between easy and difficult situations for a pedestrian detector. Situations go from people walking alone across the hall, to people pushing objects like a wheelchair in addition to big groups of people both inside and outside the building where people

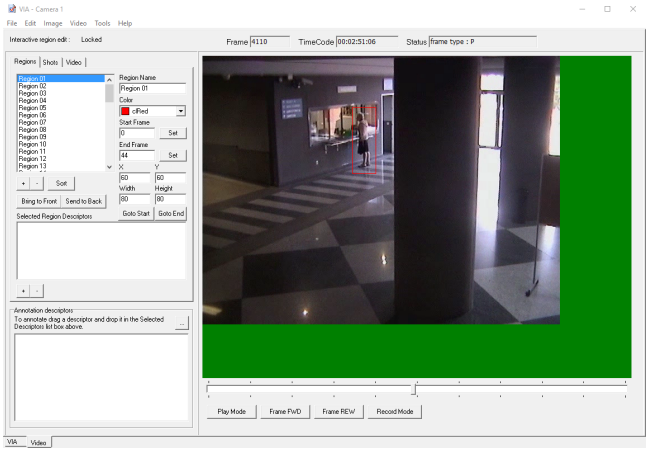


Figure 5.6: Via Annotation Tool software main window

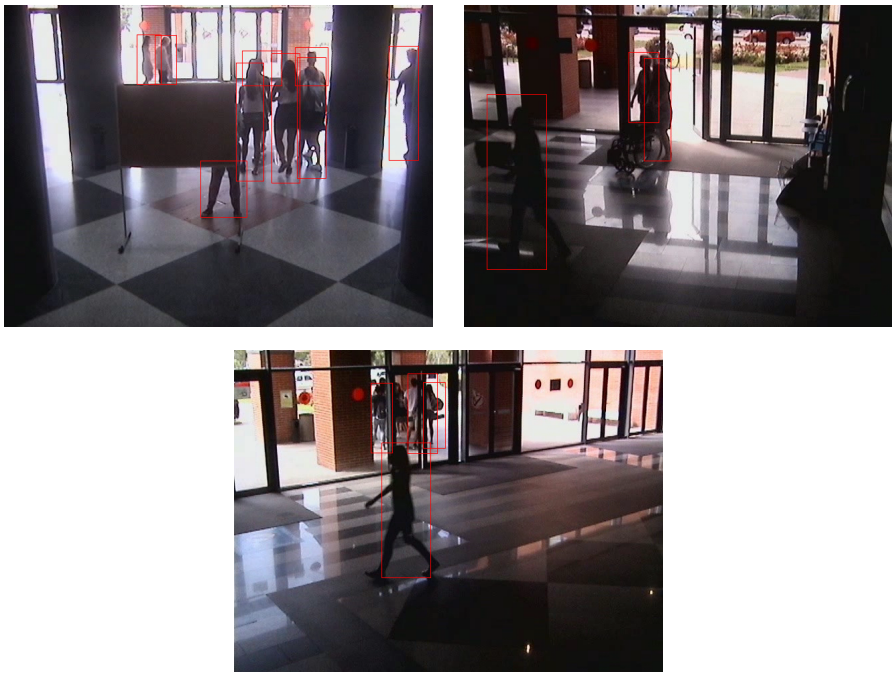


Figure 5.7: Annotated ground truth frames

are always occluded. In general due to image quality in terms of resolution and illumination and pedestrian situations the selected sequences are in the medium-high range of difficulty for pedestrian detection.

5.2.3 Evaluation Framework

Once the dataset recording has been correctly annotated one can proceed to evaluate the performance of pedestrian detectors. In order to evaluate these algorithms precision and recall metrics will be used. The calculation of these metrics is defined in Eq. 5.1 and Eq. 5.2, respectively.

$$Precision = \frac{\#True\,Positives}{\#True\,Positives + \#False\,Positives} \tag{5.1}$$

$$Recall = \frac{\#True\,Positives}{\#True\,Positives + \#False\,Negatives} \tag{5.2}$$

Precision gives us information about the classification stage, whereas the second one provides overall system performance information.

To extract the mentioned metrics we should have a method to compute which bounding boxes match with the ground truth. In order to achieve this objective Intersection over Union (IoU) or Jaccard metric has been selected. IoU is computed as in Eq 5.3 and it measures the relation between bounding boxes overlapping and its union areas.

This means that even if the bounding boxes overlap perfectly, but the union area of them is high, IoU metric will have a small value. This effect can be observed in Figure 5.8 where one can observe that a almost perfect overlapping between bounding boxes does not lead to a high IoU value. For our evaluation framework a detection is consider correct if $IoU(Detection, GroundTruth) \leq 0.5$ which is the usually metric value selected in the State of the Art.

$$IoU(Detection, GroundTruth) = \frac{OverlapArea(Detection, GroundTruth)}{UnionArea(Detection, GroundTruth)} \tag{5.3}$$

5.3 Application Performance Results

Along this section we will evaluate the application in terms of general computational time.

Firstly, results concerning single thread and multi thread application are exposed.

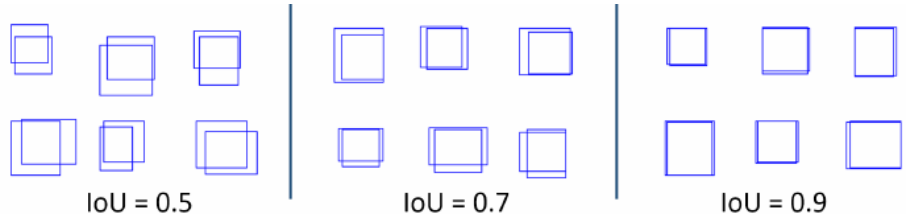


Figure 5.8: Intersection over Union or Jaccard Index

	Seconds/Frame	FPS
Single-Thread		
Multi-Thread		

Table 5.2: Comparison between the use of single thread or multi threads

All the computations to extract and visualize pedestrian detections are measured. In Table 5.2 one can observer the difference in term of seconds per frame and frame per second for the same video sequence and the same pedestrian detector approach.

In addition, all the PD are compared when evaluated in the multi thread environment with the same video sequence. This results can be seen in Table 5.3

5.4 Semantic Segmentation Results

In this Section some results concerning the performance of PSP-Net in out environment are presented. In order to test the algorithm and for the lack of semantic ground truth results should be visually evaluated. For this purpose Figure 5.9.

	Seconds/Frame	FPS
HOG		
ACF		
DPM		
PSP-Net (Offline)		

Table 5.3: Comparison between the use of single thread or multi threads

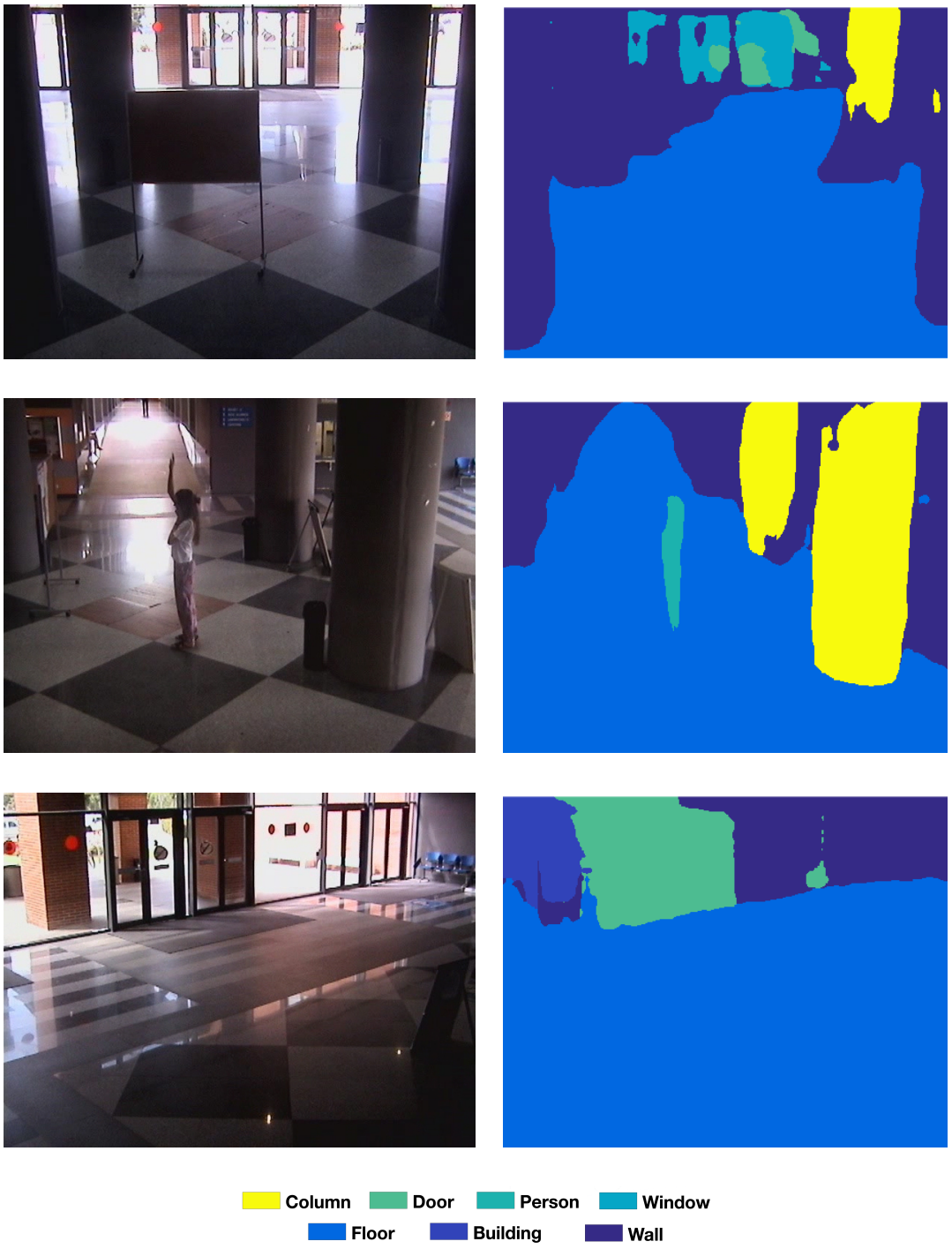


Figure 5.9: RGB frames and PSP-Net Semantic segmentation results. From top to bottom: Camera 1, Camera 2, Camera 3.

5.5 Pedestrian Detection Results

During this section results in the context of pedestrian detection will be presented. We propose a set of four different experiments to test our developed system.

The first one, will test all of the previously analyzed pedestrian detections working in a mono-camera environment. This means that neither information is shared by the cameras nor semantic constrains are applied.

We will follow testing the algorithms in a multi-camera setup. As explained in 4.3.2.3 detections from cameras will be projected into the others to try to achieve better results than in a mono-camera scenario.

Thirdly, we will analyze people detection again in a mono camera environment but this time, applying semantic constraining as proposed in Section 4.3.2.4 for all the detections.

Finally, the last test will embrace all the proposed algorithms working within the multi-camera setup and applying semantic constrains.

It is important to remind that results will be presented within Recall versus Precision curves. In order to create these graphs algorithms have been evaluated in all the four test using five different values for the score thresholding $[0, 1]$ interval. Each threshold iteration (0, 0.2, 0.4, 0.6, 0.8) for an algorithm will provide one point of the curve. To proceed with this process all the algorithms scores have been normalized from its original values using the maximum and minimum scores to the mentioned interval. In order to obtain scores distributions for all the detectors a train sequence has been used.

5.5.1 Mono-Camera

5.5.2 Multi-Camera

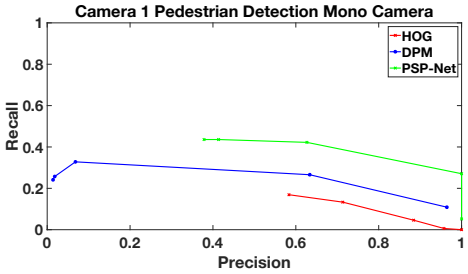
5.5.3 Mono-Camera with Semantic Constraining

5.5.4 Multi-Camera with Semantic Constraining

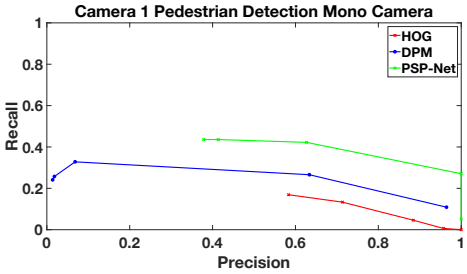
5.6 Statistical Usage Data

Finally, results concerning statistical usage data are presented. As said in the previous Chapter, we obtain one graphic per selected semantic area, which in our case are floor, doors and chars due to the mentioned map in Figure 4.27. The obtained curves are displayed both in Figure 5.14.

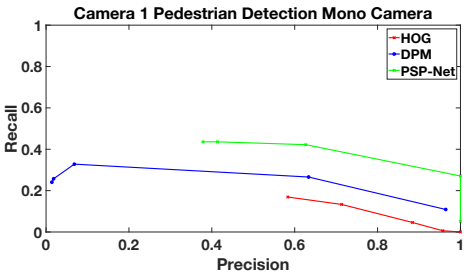
In addition, some results about the most used areas of the hall have been extracted. The hole area has been divided into regular subareas of fixed size in which pedestrian



(a) Camera 1

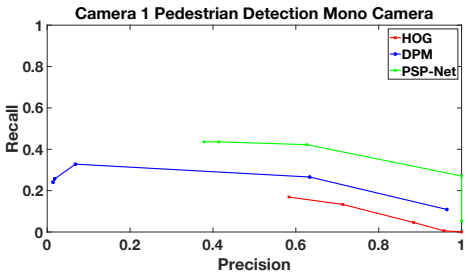


(b) Camera 2

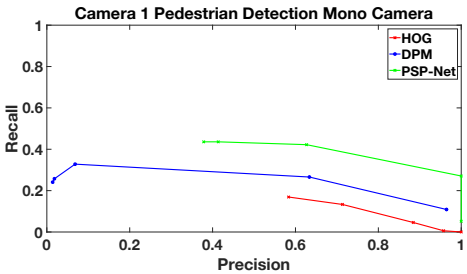


(c) Camera 3

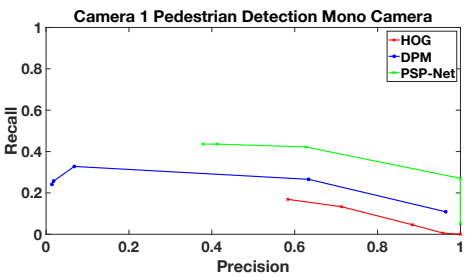
Figure 5.10: Recall / Precision graphs for mono camera pedestrian detection.



(a) Camera 1

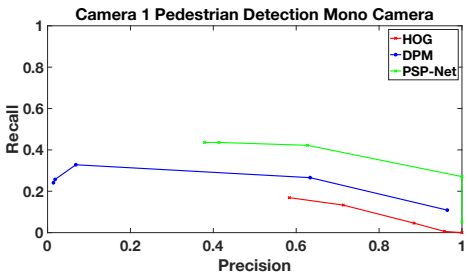


(b) Camera 2

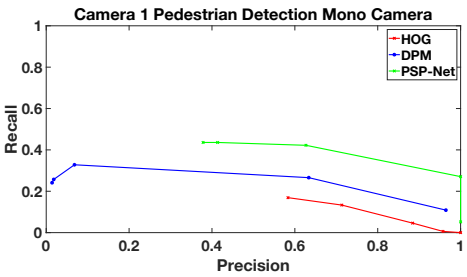


(c) Camera 3

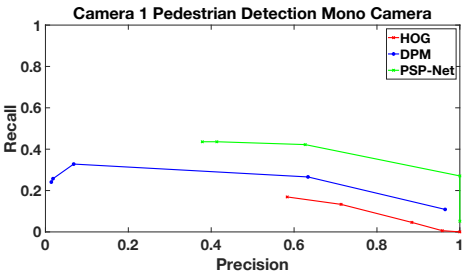
Figure 5.11: Recall / Precision graphs for multi camera pedestrian detection.



(a) Camera 1

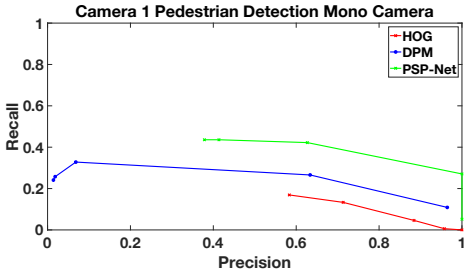


(b) Camera 2

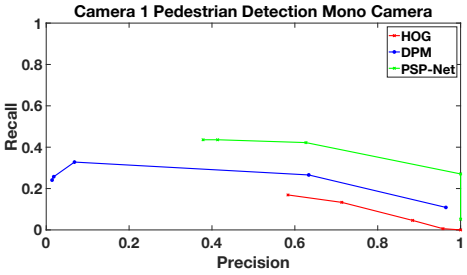


(c) Camera 3

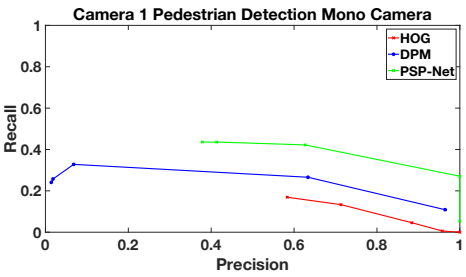
Figure 5.12: Recall / Precision graphs for mono camera pedestrian detection with semantic constraining.



(a) Camera 1

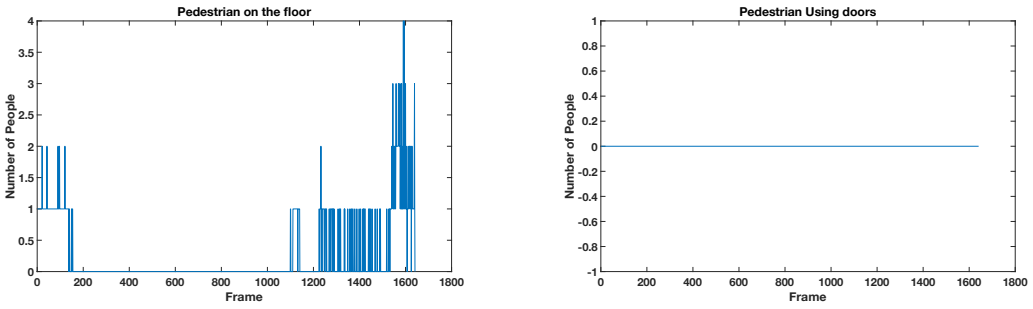


(b) Camera 2



(c) Camera 3

Figure 5.13: Recall / Precision graphs for multi camera pedestrian detection with semantic constraining.



(a) Floor Usage (b) Doors Usage

Figure 5.14: Statistical data usage curves in terms of pedestrian number/frames

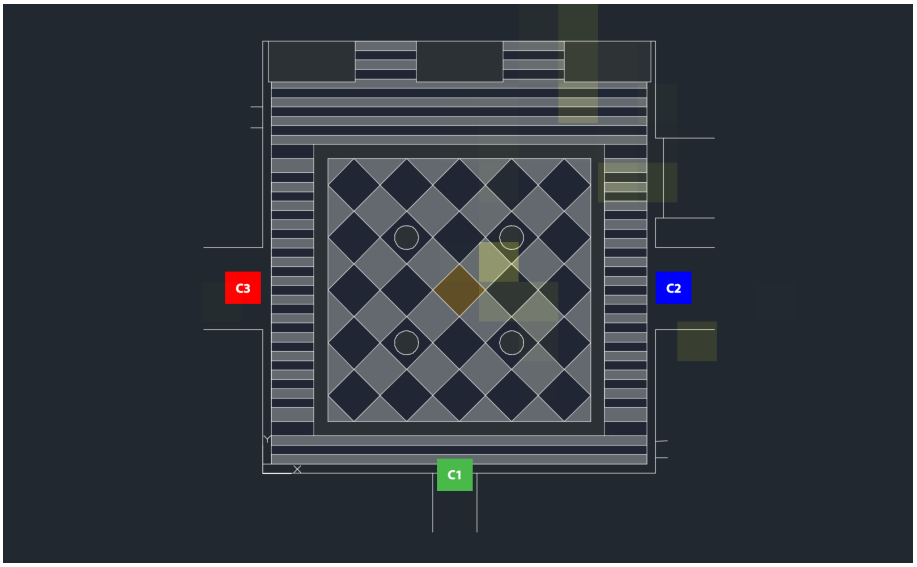


Figure 5.15: Pedestrians paths usage along the Hall

flow has been measured. By using this method one could have information about the most used paths by people over the scene. Results concerning this process are displayed in Figure 5.15.

Results are displayed as yellow squares. The more strong the color it means that during the sequence that subregion of the scene has been regularly populated with people.

5.7 Overall Discussion

Application performance.

Pedestrian Detector.

Statistical usage data.

Chapter 6

Conclusions and Future Work

6.1 Conclusions

Cuando hay cambios de iluminacion bruscos la seleccion de vistas falla.

Debido a fallos del detector las projeccciones no se corresponden exactamente con la persona.

Debido a las perspectivas, las reprojecciones de personas en otros frames tienen un ancho minimo y al applicar la fomrula para la altura, está, sale muy pequeña.

6.2 Future Work

Taking into account actual state of the art, obtained results and extracted conclusions one can set the stage for future work.

In term of application and software development some improvements are proposed to be done. Nowadays, heavy computational work is achieve almost in real time by the use of graphic cards. GPU computation should be implemented in the scoope of this work. Many of the used method are also developed with GPU functionalities and the inclusion of this kind of speed-up could lead to better and faster results.

In other term, the proposed system bases all its performance in the correct view selection as explained in Section 4.3.1.5. However, as said in Conclusions when high illumination changes occur this process fails. We propose to fix this problem by the use of camera spatial positions. With this information exact same position for each of the views can be obtained and so, they can be updated during the video sequence in order to adapt to illumination changes.

One of the main problems discussed in the conclusions section is that when pedestrians are reprojected the blob height is lost. Due to this, there are some frames that

have small detections compared to the person size. To correctly reproject the blob we propose as future work to use real distances between camera and bounding box to finally obtain the real height for the blob.

Bibliography

[1] 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.

[2] 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.

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

[4] 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.

[5] 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.

[6] 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.

[7] 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.

[8] H. Zhao, J. Shi, X. Qi, X. Wang, and J. Jia, "Pyramid scene parsing network," *CoRR*, 2016.

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

[10] 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.

[11] 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.

[12] 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.

[13] 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.

[14] 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.

[15] 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.

[16] 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.

[17] 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.

[18] 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.

[19] 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.

[20] 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.

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

[22] 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.

[23] A. García-Martín, B. Alcedo, and J. M. Martínez, "Pdbm: people detection benchmark repository," *Electronics Letters*, vol. 51, no. 7, pp. 559–560, 2015.

[24] 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.

[25] S. Ren, K. He, R. B. Girshick, and J. Sun, "Faster R-CNN: towards real-time object detection with region proposal networks," *CoRR*, vol. abs/1506.01497, 2015.

[26] K. E. Van de Sande, J. R. Uijlings, T. Gevers, and A. W. Smeulders, "Segmentation as selective search for object recognition," in *Computer Vision (ICCV), 2011 IEEE International Conference on*, pp. 1879–1886, IEEE, 2011.

[27] S. Manen, M. Guillaumin, and L. Van Gool, "Prime object proposals with randomized prim's algorithm," in *Proceedings of the IEEE International Conference on Computer Vision*, pp. 2536–2543, 2013.

[28] P. Rantalankila, J. Kannala, and E. Rahtu, "Generating object segmentation proposals using global and local search," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 2417–2424, 2014.

[29] K.-Y. Chang, T.-L. Liu, H.-T. Chen, and S.-H. Lai, "Fusing generic objectness and visual saliency for salient object detection," in *Computer Vision (ICCV), 2011 IEEE International Conference on*, pp. 914–921, IEEE, 2011.

[30] J. Carreira and C. Sminchisescu, "Constrained parametric min-cuts for automatic object segmentation," in *Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on*, pp. 3241–3248, IEEE, 2010.

[31] I. Endres and D. Hoiem, "Category independent object proposals," *Computer Vision–ECCV 2010*, pp. 575–588, 2010.

[32] A. Humayun, F. Li, and J. M. Rehg, "Rigor: Reusing inference in graph cuts for generating object regions," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 336–343, 2014.

[33] P. Krähenbühl and V. Koltun, "Geodesic object proposals," in *European Conference on Computer Vision*, pp. 725–739, Springer, 2014.

[34] 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.

[35] B. Alexe, T. Deselaers, and V. Ferrari, "What is an object?," in *Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on*, pp. 73–80, IEEE, 2010.

[36] E. Rahtu, J. Kannala, and M. Blaschko, "Learning a category independent object detection cascade," in *Computer Vision (ICCV), 2011 IEEE International Conference on*, pp. 1052–1059, IEEE, 2011.

[37] M.-M. Cheng, Z. Zhang, W.-Y. Lin, and P. Torr, "Bing: Binarized normed gradients for objectness estimation at 300fps," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 3286–3293, 2014.

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

[39] J. Feng, Y. Wei, L. Tao, C. Zhang, and J. Sun, "Salient object detection by composition," in *Computer Vision (ICCV), 2011 IEEE International Conference on*, pp. 1028–1035, IEEE, 2011.

[40] Z. Zhang, J. Warrell, and P. H. Torr, "Proposal generation for object detection using cascaded ranking svms," in *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on*, pp. 1497–1504, IEEE, 2011.

[41] M. Van den Bergh, G. Roig, X. Boix, S. Manen, and L. Van Gool, "Online video seeds for temporal window objectness," in *Proceedings of the IEEE International Conference on Computer Vision*, pp. 377–384, 2013.

[42] J. Kim and K. Grauman, "Shape sharing for object segmentation," *Computer Vision–ECCV 2012*, pp. 444–458, 2012.

[43] D. Erhan, C. Szegedy, A. Toshev, and D. Anguelov, "Scalable object detection using deep neural networks," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 2147–2154, 2014.

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

[45] 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.

[46] A. Torralba, A. Oliva, M. S. Castelhano, and J. M. Henderson, "Contextual guidance of eye movements and attention in real-world scenes: the role of global features in object search.," *Psychological review*, vol. 113, no. 4, p. 766, 2006.

[47] P. Salembier and T. Sikora, *Introduction to MPEG-7: Multimedia Content Description Interface*. New York, NY, USA: John Wiley & Sons, Inc., 2002.

[48] 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.

[49] B. Zhou, H. Zhao, X. Puig, S. Fidler, A. Barriuso, and A. Torralba, "Semantic understanding of scenes through the ade20k dataset," *CoRR*, vol. 1608, 2016.

[50] 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.

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

[52] P. Wang, P. Chen, Y. Yuan, D. Liu, Z. Huang, X. Hou, and G. W. Cottrell, "Understanding convolution for semantic segmentation," *CoRR*, vol. abs/1702.08502, 2017.

[53] 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.

[54] J. Xiao and L. Quan, "Multiple view semantic segmentation for street view images," in *Computer Vision, 2009 IEEE 12th International Conference on*, pp. 686–693, IEEE, 2009.

[55] 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.

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

[57] 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.

[58] H. Aliakbarpour, V. S. Prasath, K. Palaniappan, G. Seetharaman, and J. Dias, "Heterogeneous multi-view information fusion: Review of 3-d reconstruction methods and a new registration with uncertainty modeling," *IEEE Access*, vol. 4, pp. 8264–8285, 2016.