KU LEUVEN

Knowledge-driven image segmentation

Ignacio Garrido Botella

Thesis submitted for the degree of
Master of Science in Artificial
Intelligence, option Engineering and
Computer Science

Thesis supervisor:
Prof. dr. ir. Herman Bruyninckx

Assessor:
Ir. Filip Reniers

Academic year 2018 – 2019

Preface

The reason I decided to do this thesis is because I have always been curious about the way humans process information and the best way to translate this knowledge into a computer. Specifically and within the framework of this thesis, we, the humans, are incredibly good at recognizing objects and understanding the relationships of the environment that surrounds us. While this ability seems to come in an innate way, it is extremely difficult to understand the process we carry out to do so. Furthermore, it is even more difficult to adapt this knowledge to a machine. Thus, with this thesis I wanted to understand better the process we follow to understand so well the environment that surrounds us, and infer into the ways to adapt this knowledge to a computer-based analysis.

As well, I would like to thank everybody that helped me to achieve my goals. First, I would like to thank my supervisor Prof. dr. ir. Herman Bruyninckx and assessor Ir. Filip Reniers for the tremendous support you have given me to achieve this better understanding of what truly means a knowledge-based image segmentation. Indeed, when starting this thesis I kept falling in the "magic numbers" procedures, but you helped me to pivot to a knowledge analysis, that led to "real" and more deterministic results.

I would also want to thank my family. Thank you Mom and Dad for the support you always gave me so I could have the wonderful experience of studying one year in Leuven. Indeed, it would have never been possible without you. As well, I would specially like to mention my sister Isa. Thank you for taking care of Borja and of the whole family while I was not there.

Last but not least, I would like to thank all the people that made this year one of the best years of my life. First, to Cris and Javi for those ways back to Parkstraat 137 at night and for making it feel like a home. To Anna for those incredible Wednesdays of rock climbing. To Sofía for giving me one of the weirdest nights of my life. To the dream team, Victor, Rushil, Erick and Murilo, for showing me that there are no limits if you put passion and hard work in what you want to achieve. Finally, thank you Guille, Javi, Faustine, Abel, Isa, Manu, Jorge and Jesús for all the incredible experiences I have lived with you.

Thank you.

Ignacio Garrido Botella

Contents

Abstract

The aim of this thesis is to infer the knowledge needed by an autonomous system for performing image segmentation of a video footage that is taken inside a building. Precisely, the current document is based on the use case of an autonomous vehicle that has to move in a public building transporting goods and without posing any kind of hazard for the people moving around it. Thus, one of the most important capabilities is performing a semantic segmentation of the image in order to distinguish foreground (movable area, ie., floor) from background (non-movable area).

The approach taken in this work to discern between foreground and background consists in the segmentation of images in a fixed number of squared superpixels, followed by a posterior analysis of the features collected for each of those superpixels. The features analysed for each of the superpixels are divided into features extracted directly from the superpixel, local knowledge in the sense of relationships of the superpixel with its close neighbourhood of superpixels and global knowledge in the sense of features taken from the image as a whole. In this way, the image is summarized in a database of the features, from which a data analysis can be made. Accordingly, in the present work a feature selection process has been carried out and several tests have been performed for these feature collections and for the correct combination between them.

In addition, one of the cornerstones of this thesis is to facilitate the explainability of the decision-making process for the subsequently image segmentation. For this reason, the posterior analysis of the collected features has been done with a decision tree model. Thus, this work is at an intermediate point between entirely black box segmentation models such as CNN (global information completely learned) and segmentation models based purely on knowledge (global information user defined).

List of Figures and Tables

List of Figures

List of Tables

List of Abbreviations

Abbreviations

AV	Autonomous vehicle
EMD	Earth mover's distance
LBP	Local binary patterns
DT	Decision tree

Chapter 1

Introduction

Gradually, society is moving towards a robot-based culture. The natural step, as technology advances, is progressively introducing robots to develop the most dangerous and repetitive tasks. Concretely and in this context, autonomous driving can be adapted to numerous applications in diverse fields, such as transportation of goods and people, waste collection [36] or even it could be applied in an agricultural context for autonomous harvesting [24].

The first step for developing this kind of autonomous vehicles (AV) consist in providing them with a vision system that allows the AV to both, locate in a known environment and segment the area through which they can freely (or restricted to some rules) move. Moreover, doing this segmentation can be useful for mapping purposes or even for grounding the image to an already known map of the environment. While this positioning and segmentation can be done using several approaches, usually combining the input of a collection of sensors [12], most of the systems rely totally or partially on images [6] [37]. Following this line, the work carried out in this project will be based on the use case of the endowment of an image-based vision system to an AV. Concretely, this thesis supposes an AV that moves inside an office building (through corridors) for the transportation of goods. Furthermore, the effort has been focused in the creation of an image segmentation system in which the movable area in front of the robot (floor) is distinguished from the rest of the image. The final purpose of this work is that, by detecting the floor and by distinguishing it from the walls and obstacles, the robot can trace out a route planning and the subsequently series of actions that drives it to its goal.

Some approaches to model these kind of AV systems base their decisions in semi-stochastic analysis of data [7], modeled usually as a black box. As well, a common approach for this kind of vision systems for AV consist in not defining a clear boundary of the whole movable area of the image, but in defining the obstacles next to the AV by analysing the features of the images and, thereupon, avoiding them [27] [17]. Likewise, most of these systems make use of different features, weighting and comparing them according to the input [15]. However, this kind of algorithms usually incorporate in their analysis the output of other sensors such as lidar [23].

Another common approach consist in applying insight of the geometry of the

environment and reasoning over it [13]. Moreover and similarly to the present work, other approaches focus on directly detecting and segmenting the floor. Some of these approaches detect the ground plane by using information of the a sequence of previous images and by establishing a series of constraints of how the homography should behave [39]. Finally, some of the most interesting approaches, and the one in which this thesis is more inspired, is a floor segmenting approach based in previous user defined knowledge of how the floor is represented in images. An example of this approach is the one carried out by Yinxiao Li et al. [38], in which, by applying previous insights of some characteristics that differentiates the floor plane from the rest of the image and by weighting and combining these features, a decision of the location the boundary between the floor and the wall is made.

Altogether, the faced problem consist in a segmentation problem, in which the floor has to be segmented from the rest of the image. A common approach to face segmentation problems consist in dividing the image in superpixels (groups of pixels), and analyse their characteristics. This approach was first introduced by Jitendra Malik et al. [26], in which the image is first divided in superpixels, and according to the features extracted for each of these, a classification is made.

Following this line, the proposed algorithm should base its decisions on a deterministic approach that result in correct actions and from which it can easily be inferred the reasons by which the AV's image segmentation system takes certain decisions instead of others. Thus, the main purpose of this thesis is that the performance of the system can be easily explained and is based on a trustworthy decision making process.

On the basis of the previous ideas, the main goal of the present thesis consist in the segmentation of the floor from the background in images. Additionally, this segmentation has to be easily explainable and interpretable. Furthermore, the decisions made for discerning between floor and background should be based in features determined in an user defined knowledge basis (knowledge of the attributes that characterize the floor in images). As well, the main objective of this segmentation is being useful for driving the actions of an AV that moves inside a building. Thus, the segmentation is done at superpixel level.

For doing so, the image is divided in a fixed number of rectangular superpixels and several features are extracted for each of them. A database with the features for each of the superpixels is then constructed and they are classified either as floor, background or edge (superpixel that is between the floor and the background) superpixels. Furthermore, in order to make the model as explainable and closest to a white box approach as possible, a decision tree is trained on these features for doing the classification.

The features taken for each of the superpixels are divided into three categories. First, there is the knowledge extracted directly from the superpixel. This kind of knowledge consist in observations made directly from the own superpixel, without taking into account the surrounding ones. Some of these features are the detection of an edge in the superpixel, the entropy of the gray and local binary patterns distributions and the biggest gradient change in that superpixel. Second, it is taken features that correspond to "local" knowledge in the sense of relationships of the

superpixel with its neighbourhood of superpixels. For inferring this "local" knowledge, a neighbourhood of the twelve closest superpixels is taken and features such as class of previous classified superpixels, difference in mean gray intensity between the superpixel and its neighbouring superpixels and distance between gray and local binary pattern distributions of these neighbour superpixels and the one that is being analysed are calculated. Third, it is also taken "global" knowledge in the sense of features extracted from the analysis of the whole image. Some of the features derived from this "global" knowledge are the relative position of the tile with respect to the lower centered part of the image, maximum gradient change over, under and at both sides of the superpixel and size of the smaller area to which the tile belongs after performing a Felzenszwalb segmentation. Additionally, the analysis approach consist in a region growing-like algorithm. Starting from the centered bottom part of the image, i.e., the region in which there is a higher probability to detected the floor, the new superpixels to be analyzed are taken, growing to the top, right and left parts of the image.

Accordingly, in the present work a feature selection process has been carried out and several tests have been performed on the features collections. It has been proven that a correct feature selection is crucial for constructing a trustworthy classifier.

This thesis is divided in four parts. First, a brief theoretical introduction of the images, methods and algorithms used is made. Then, the collection of features is explained, followed by the explanation of the algorithm. Finally, the results obtained for each of the tests that have been carried out are showed.

Chapter 2

Data used and theoretical background

In this chapter the video sequences that have been used is introduced. This is followed by the proposal of image segmentation in which the thesis is based. Finally, the theoretical background for doing this segmentation, extracting the features and classifying the images is introduced.

2.1 Data used

The image dataset that has been used has been created by Grace Tsai and Benjamin Kuipers and it can be found in [34]. It was previously used by the authors for producing the testing of their IROS 2012 paper [35].

This dataset of images consist of 4 video sequences of indoor corridors acquired with a camera that has been mounted on a movable platform, thus, imitating the point of view of an AV that moves inside an office building. The camera is fixed at a height of 0.47 m and has a fixed angle with respect to the ground for each of the videos. Additionally, every 10 frames it is provided the ground truth images in which the floor, walls and ceiling are labeled. Moreover, a post-processing of the video sequences has been done, in which the images are divided into floor and background from the provided ground truth. Next it is shown these video sequences and the provided ground truth images.

Figure 2.1: Video sequence 1

Figure 2.2: Video sequence 1 - Ground truth

Figure 2.3: Video sequence 2

Figure 2.4: Video sequence 2 - Ground truth

Figure 2.5: Video sequence 3

Figure 2.6: Video sequence 3 - Ground truth

Figure 2.7: Video sequence 4

Figure 2.8: Video sequence 4 - Ground truth

Note that these images capture the already explained scenario in which an AV has to move through an indoor corridor avoiding obstacles. As well, and as explained, it is shown the post processing of the ground truth in which the segmentation of the image has been reduced to floor (1 - yellow) and background (0 - black). The final database consist of 203 images and their corresponding ground truth segmentations.

2.2 Proposal of segmentation for the image

The problem faced in this thesis can be reduced to a segmentation problem. As stated before, one common approach consist in first doing a segmentation in superpixels of the image and then classifying these superpixels [26]. Based on this approach, the proposed algorithm consist in dividing the images in rectangular superpixels and classifying each of them as either floor, background or edge.

Notice that most of the superpixel classification problems divide the image in superpixels with not prefixed shape [14] [16]. This is usually done because it is a

desired characteristic that there exists a clear boundary between superpixels that belong to different classes [3, p.187]. However, the proposed algorithm divides the images in rectangular superpixels, which incredibly eases the analysis of the local features and the relationships between the superpixel and its neighbour superpixels. Thus, to meet this requirement of a clear boundary between different classes, the edge has been modelled as an independent class, different from the floor and the background. Notice that this edge superpixel consist in a superpixel that contains pixels that belong to both, the floor and the background.

The relationships between the three possible classifications, floor, background or edge, can be seen in the next figure.

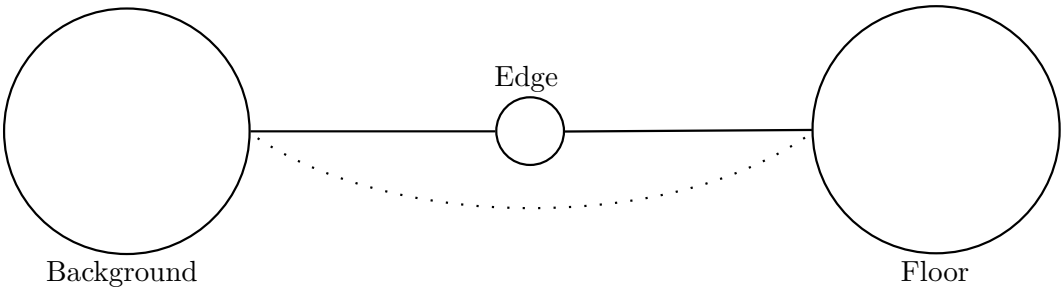


Figure 2.9: Categories of the segmentation and their relations.

All the superpixels will fall in one of these three categories, shown in Figure 2.9. As well, between a superpixel that correspond to the floor and a superpixel that correspond to the background there would be, in most of the cases, a superpixel that belongs to the edge. This will happen in most of the cases as, sometimes and because of the division of the images in rectangular superpixels, the edge between the floor and a wall falls exactly between two superpixels, resulting in a direct transition between the floor and the background (in Figure 2.9 this is represented with the dotted line).

Therefore, if an image like the one represented in Figure 2.10 is segmented, the expected result would be a classification like the one shown in Figure 2.11. Each of superpixels should be classified as either floor (green), edge (dark blue) or background (light blue).

In addition, an example of this direct transition between floor and background is shown in Figure 2.11. In the left upper part of the segmented image there are some superpixels that shift directly from floor to background. This is because the boundary between the floor and the wall falls exactly between two superpixels. Thereby, in this region of the image there is no superpixel containing pixels in both sides.

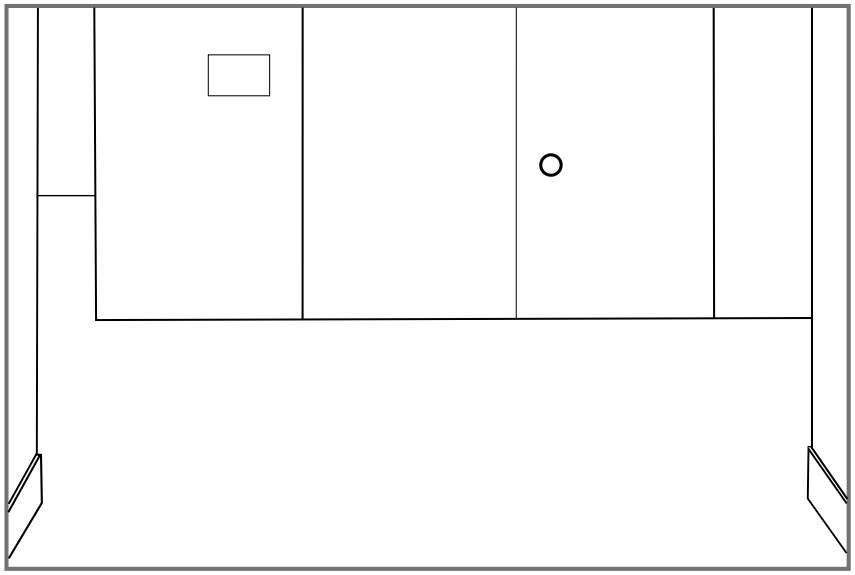


Figure 2.10: Example of a frame of a corridor

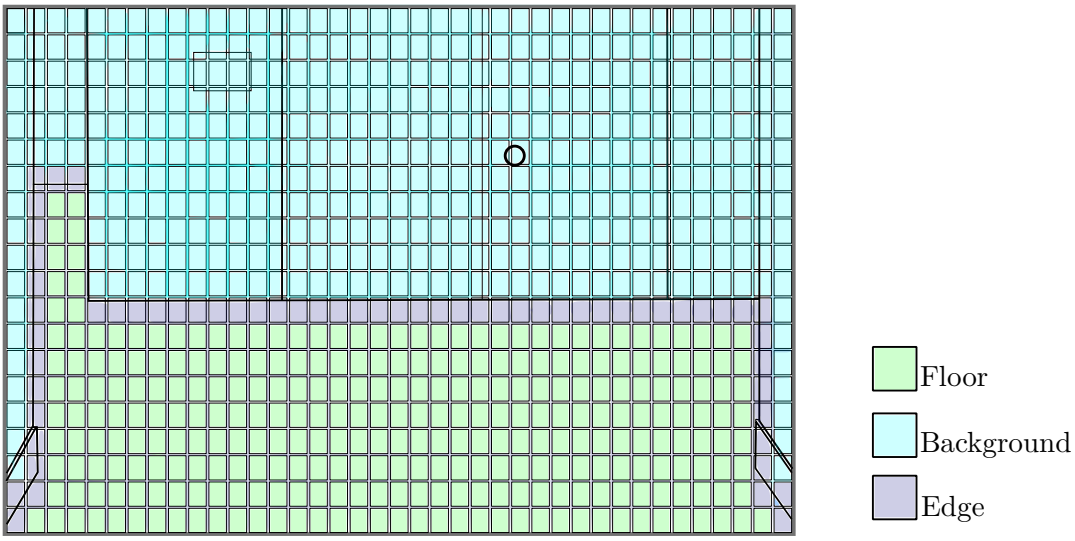


Figure 2.11: Desired segmentation. Image divided in superpixels that are classified as either floor, background or edge.

2.3 Features extracted and theoretical background

Once the image is divided in superpixels, an analysis of the features of each of these should be done in order to correctly classify them. In this section it is introduced the theoretical background of the tools used for extracting these features and for the posterior analysis of them.

2.3.1 Local binary patterns

Local binary patterns (LBP) are a well known local texture descriptor, first introduced by T. Ojala et al. in 1993 [19] [20], and it can be considered a variant of the spectrum analysis method proposed by Li Wang et al. in 1990 [8] [11].

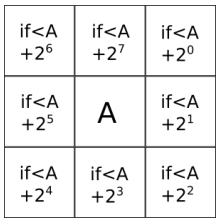


Figure 2.12: Example of the LBP kernel for a 3x3 neighbourhood.

The basic idea of LBP consist in analysing each of the pixels separately, and calculate the descriptor value for each one by comparing it with its neighbourhood. The most simple approach is represented in Figure 2.12 and it consist in comparing each pixel with the neighbourhood of the 8 closest pixels in a grayscale image. For each set of 9 pixels, if the center one is greater than any of its neighbours, it is noted by adding to the LBP descriptor of that pixel the value specified in Figure 2.12. This process is formalized in Equations 2.1 and 2.2. By doing this, each pixel has assigned a LBP descriptor that can get 256 different values. Finally, after a LBP descriptor is assigned to each pixel of the image, it can be computed the histogram of the all the values of the descriptors of the pixels of one region (histogram of the values that the LBP takes in one region).

$$LBPdesc(P_A) = P_1 * 2^0 + P_2 * 2^1 + P_3 * 2^2 + P_4 * 2^3 + P_5 * 2^4 + \\ + P_6 * 2^5 + P_7 * 2^6 + P_8 * 2^7 \tag{2.1}$$

$$where P_x = \begin{cases} 1, & \text{if Pixel(x)<Pixel(A).} \\ 0, & \text{otherwise.} \end{cases} \tag{2.2}$$

Notice that the values specified in Figure 2.12 can rotate among the 8 pixels, while each pixel has assigned it's own value, and the filter is fixed for the whole image. As well, bigger neighbourhoods than 3x3 can be taken, resulting in more values for each LBP descriptor. In addition, it can be done a finer quantization of the search space, reducing the number of values of the histogram and grouping similar values in one single bin (uniform patterns), by computing the so-called uniform LBP extension [2]. This finer quantization of the search space is as well rotation invariant.

The strength of this descriptor resides in that, by analysing in a region of the image the histogram of the values given by the descriptor, it can be inferred if that region is a flat surface, if there is a corner or if there is an edge. Likewise, a classification of the regions can be made by analysing the LBP histograms.

For implementing the LBP it has been used the code proportioned by Sklearn-image kit [29].

2.3.2 Gradient of an image

An edge is usually located at a local maxima of the derivative of the image. Thereby, calculating the gradient of an image can be a very useful tool for identifying the boundary between different surfaces.

A common approach to calculate the derivatives of an image consist in making use of the Sobel kernel [33] and filtering the image. These Sobel kernels are used for calculating the gradient at each pixel's position in the vertical G_y and in the horizontal G_x axis of the image. Afterwards, the absolute value of the gradient in each pixel can be calculated as shown in Equation 2.3.

$$Gradient = \sqrt{G_x^2 + G_y^2} \tag{2.3}$$

In the Figures 2.13 and 2.14 it is shown the 3x3 kernels used for calculating the vertical and horizontal first derivatives. Notice that the size of these kernels can be adapted to the image.

$$\begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix}$$

Figure 2.13: 3x3 vertical Sobel kernel.

$$\begin{bmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{bmatrix}$$

Figure 2.14: 3x3 horizontal Sobel kernel.

For calculating the gradient value of the image it has been used the code proportioned by OpenCV [22]. First, it has been calculated the vertical and horizontal derivatives with 3x3 kernels, and then it has been calculated the absolute value of the gradient, given by Equation 2.3.

Notice that the width of the detected gradient depends on the size of the Sobel kernels. With the chosen size, shown in Figure [22], it is detected the gradient at one pixel width. This is done at full image resolution (400x702).

2.3.3 Canny edge detector

Canny edge detector is a popular algorithm used for detecting edges in images. It was first introduced by John F. Canny in 1986 [5].

This algorithm consist in a multi-stage process in which the borders of the images are detected. It is composed by the next steps:

1. First, the noise is reduced by blurring the image applying a Gaussian filter.

2. Then, the intensity gradient of the image has to be found. For doing so, the image is filtered with a vertical (G_y) and horizontal (G_x) Sobel kernels. From

these it can be obtained the gradient value and the angle of the edge at each pixel:

$$Gradient = \sqrt{G_x^2 + G_y^2} \tag{2.4}$$

$$Angle = \tan^{-1}(\frac{G_y}{G_x}) \tag{2.5}$$

3. After getting the gradient direction, those points that are not associated to this edge are eliminated. This is the so-called non-maximum suppression step, in which all those points that do not belong to a local maxima in the direction of the gradient (normal to the direction of the edge) are suppressed.

4. The next step consist in the double thresholding stage and it is decided which points do belong to an edge or not. First, two pretuned values are given, $max_{threshold}$ and $min_{threshold}$. Basically, any points that have an intensity of the gradient over $max_{threshold}$ value, are considered edges. Additionally, all those points whose intensity of the gradient relies between $max_{threshold}$ and $min_{threshold}$ are considered edges whenever they are connected to other pixels whose gradient's intensity value is over $max_{threshold}$. Likewise, all those points whose gradient's intensity is under $min_{threshold}$ are discarded from being considered edges.

5. Finally, it can be done an edge tracking step, in which those weak edges that are connected to edges with a high gradient value are considered as well as actual edges.

For implementing the Canny edge detector it has been used the code proportioned by OpenCV [21].

2.3.4 Felzenswalb segmentation

Several image segmentation techniques can be used as pre-processing tools for analysing images. Concretely, Felzenswalb segmentation [9] is a graph-based method which tries to merge perceptually similar pixels, taking into account the size of the area segmented. A key point of this segmentation method is its high computational efficiency and that it merges big areas of the image that have a very high similarity into one single segmentation.

This algorithm takes a graph-based approach for performing the segmentation. Let $G = (V, E)$ be an undirected graph of vertices (pixels) $\in V$, and edges between two vertices (edges between two neighbour pixels) $\in E$. Additionally, each of the edges between two pixels has assigned a weight w_i, which is a measurement of the dissimilarity between those two pixels. Following this approach, a segmentation will be a partition of all the elements of V in such a way that the elements that belong to the same segmentation are connected in the graph G.

A segmentation will be considered as a good one if the elements in a component are similar between them, and dissimilar with the components of the other surrounding segmentations. For calculating the quality of a segmentation, two dissimilarity measurements are proposed.

- The internal difference of a component C is defined as the biggest weight of the edges that belong to the minimum spanning tree of that component.

$$IntDiff(C) = \max_{e \in MST(C,E)} w(e) \tag{2.6}$$

- The difference between two components C_1 and C_2 is defined as the minimum weight of all the existing edges between the pixels of the two components. Notice that if there is no edge connecting two pixels, its weight is ∞.

$$Diff(C_1, C_2) = \min_{v_1 \in C_1, v_2 \in C_2, (v_1,v_2) \in E} w(v_1, v_2) \tag{2.7}$$

The algorithm for segmenting an image, as explained in the original paper, with graph $G = (V, E)$, n vertices and m edges into a segmentation $S = (C_1, C_2, ..., C_r)$ is the following:

1. First, the edges of E have to be sorted by their weights in increasing order.
2. In the starting point, each vertex is its own segmentation S^0.
3. Repeat the next step for q = 1,...,m.
4. From S^{q-1} it is constructed the new segmentation S^q. First, it has to be taken the pixels v_i and v_j that are at position q in the ordered list of edges (step 1). If these two pixels belong to different segmentations, and the weight of the edge that connects both, $w(v_i, v_j)$, is small compared to the internal difference (Eq. 2.6) of both segmentations, then merge them.
5. Return the components of the final segmentation

For implementing the Felzenswalb segmentation it has been used the code proportioned by sklearn-image kit [28].

2.3.5 Wasserstein metric - Earth mover's distance

Wasserstein metric [4], also called Kantorovich-Rubinstein distance or "earth mover distance" (EMD), is a metric that measures the difference between two distributions. It was first introduced in 1969 by Leonid Vaserštešn.

This metric can be interpreted as the minimum amount of movement of the components of one distribution to become the other distribution. In other words, the EMD is the minimum amount of effort for matching two different distributions. Hence, the bigger is the value of the EMD, the more dissimilar are the compared distributions.

Because of this interpretation, EMD is usually used in optimal transport problems (cost of transporting a mass from one point to another) and for comparing color histograms of different regions of images [18].

Formally, the EMD between two distributions x and y is defined as stated in Equations 2.8 and 2.9.

$$EMD(x,y) = \frac{min_{F=(f_{i,j})\in F(x,y)} \sum_{i=1}^{m} \sum_{j=1}^{n} f_{ij} d_{ij}}{min(w_\Sigma, u_\Sigma)} \tag{2.8}$$

$$where P_x = \begin{cases} \sum_{i=1}^{m} \sum_{j=1}^{n} f_{ij} d_{ij}, & is\, the\, work\, done\, by\, a\, flow\, to\, match\, x\, and\, y. \\ f_{ij}, & is\, the\, amount\, of\, weight\, at\, x_i that\, matches\, y_j. \\ d_{ij}, & is\, the\, distance\, between\, the\, points\, x_i\, and\, y_j. \\ min(w_\Sigma, u_\Sigma), & is\, the\, total\, weight\, of\, the\, lighter\, distribution. \end{cases} \tag{2.9}$$

The strength of this metric, compared to other statistical distance metrics such as Kullback leibler divergence, resides in the property of being a bidirectional measurement between the compared distributions. In addition, it has a relatively easy and fast computation. For implementing the Wasserstein distance it has been used the code proportioned by Scipy-stats kit [32].

2.3.6 Decision trees

Decision trees are a supervised learning algorithm that can be used for either classification or regression (in the present work they are used for classification) [25]. Similarly, its input can be categorical or continuous variables, and its decision making structure is based in the rules-making set "if-then-else". Decision trees are a powerful tool for classification, with high stability and with a superior explainability than other classification techniques.

As a drawback, decision trees make rectangular partitioning of the data and they tend to overfitting. To avoid overfitting a common approach consists in bounding the size of the tree. Some of the ways to limit the size of the tree are setting a minimum number of samples in a node to split or to consider it a leaf node, setting a maximum vertical depth of the tree and setting a maximum number of terminal nodes. Another way to avoid overfitting, like in any other classification problem, is limiting the number of features used in the classification process.

The general process for splitting the data is the following.

1. Find the feature for which the most significant partition can be done (the one that enhances the homogeneity in each of the leafs more), according to some criteria (Gini index, entropy,...).

2. Perform this partition.

3. Repeat the first step for each of the new created leafs until the data is perfectly split or some stopping criteria is met.

The prediction is as easy as following the tree rules for each new data input, until obtaining its classification.

As said, the decision tree finds the most significant feature for doing the split according to some criteria. One of the most used splitting criteria is Gini index, which is a measurement of the impurity of a node. In other words, the smaller is the Gini index, the better is the split, obtaining purer leafs. The equation for calculating the Gini index is shown in 2.10.

$$Gini = 1 - \sum_{i=1}^{C}(p_i)^2 \tag{2.10}$$

In a leaf a dataset is divided into two smaller datasets. As a result, the Gini index of that node is defined as the weighted summation of the Gini index of each of the two partitions; $Gini_{node}(D) = \frac{|D_1|}{|D|}Gini(D_1) + \frac{|D_2|}{|D|}Gini(D_2)$. In this way, the feature chosen for doing the split will be the one that gives a biggest reduction in impurity, defined as $\bigtriangledown Impurity = Gini(D) - Gini_{node}(D)$.

In addition, it can be defined the feature importance as the proportional decrease in impurity over the whole dataset, due to all the splits done by a concrete feature. In other words, it is the addition of the proportional decrease in impurity of all the nodes in which a concrete feature makes the split. The feature importance is defined in Equation 2.11 and it can be interpreted as the amount of say of a concrete feature in the decision made by the decision tree.

$$\begin{aligned} Importance(feature_i) &= \frac{\sum_{j:node\ j\ splits\ on\ feature\ i} ni_j}{\sum_{k:all\ nodes} ni_k} \\ where\ \ ni_j &= Gini\ importance\ of\ node\ i. \end{aligned} \tag{2.11}$$

Note that Gini index is the splitting criteria used in this thesis. For implementing the decision tree algorithm it has been used the tool proportioned by Sklearn [30].

2.3.7 Feature selection techniques

A bad selection of features can lead to a poor performance of the model. This is because irrelevant features can lead the model to increase its overfitting, resulting in a lower accuracy than the one of a model without those features. Thus, two of the most well-known feature selection techniques are the usage of the feature importance, as shown in Chapter 2.3.6 and a technique called recursive feature elimination.

First, by simply calculating the feature importance, it can be inferred the amount of say that each feature has in the decision. With this, the most important features can be selected, discarding those with a lower importance.

Secondly, recursive feature elimination [10] is a technique introduced by I. Guyon et al. that consists in the elimination of the weakest features iteratively until the desired number is reached. The process follows the next steps. First, the model is trained with all the features. Then, the feature (or features) with the lowest importance is eliminated and the model is trained again with the new set of features

(without the weakest feature). This process is repeated recursively until it is reached the desired number of features. A variant of this feature selection technique consists in using cross validation in the process of elimination of features, using the validation process to get the best combination of features. For implementing the recursive feature elimination algorithm it has been used the tool proportioned by Sklearn [31].

However, despite the usefulness of these two techniques, it is crucial that some human insights on the posed problem are introduced when selecting features. Thus, it is a naive decision relying entirely on these feature selection processes without a posterior analysis of the selected features.

Chapter 3

Prior knowledge and collection of features

Once the image is divided in rectangular superpixels, the next step consist in gathering features for making the classification of the tiles. For making this segmentation possible, the collected features must match the knowledge of how the floor is characterized in the image and how it can be distinguished from the background. Furthermore, it is essential to represent the observations and features of the floor in a way that fits the approach that has been taken of dividing the image in tiles. In fact, the performance of the algorithm is highly dependent on the correct adaptation of the prior knowledge features to the tiling of the image.

For doing so, in this chapter it is first explained the insights of how the floor is represented in images and what characterizes it in comparison with the background. Then it is analysed how this knowledge can be adapted to the superpixels' approach. Notice that the aspects pointed out in the prior knowledge section are closely related to the fixed position of the camera in the AV and to the use case to which the system is being adapted (segmentation of the floor of the corridors in an office-like building).

3.1 Prior knowledge

For correctly doing the segmentation, first it has to be stated some of the knowledge and features by which it is possible to distinguish the floor from the background of the image, and then adapt it to the squared superpixel approach. The knowledge retrieved for this problem can be divided in two groups, local knowledge and global knowledge. The first one, local knowledge, consist in the insights that can be derived from the own superpixel that is being analysed and the close neighbourhood of that superpixel. The second, global knowledge, consist in the characteristics that can be derived from taking features directly from the analysis of the whole image.

3.1.1 Local knowledge

Local knowledge is the knowledge that can be extracted from analysing a superpixel and the nearby superpixels. Next, it is shown the statements of the local knowledge in which the features extraction process is based.

- A segmented area has a smooth texture all along it. This means that close sections in the image that have a similar texture and are located nearby probably belongs to the same class of the segmentation.
- The color/texture distribution of two superpixels that belong to different classes in the segmentation are, with high probability, quite different (the EMD is probably big).

In addition, more insights of the prior local knowledge can be inferred from the next figures.

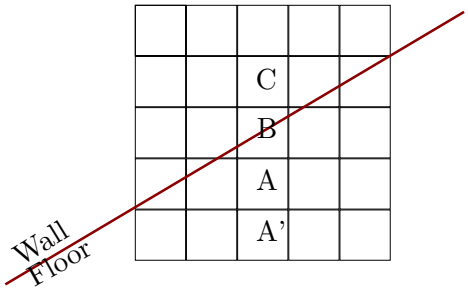


Figure 3.1: Example of wall-floor boundary divided in superpixels.

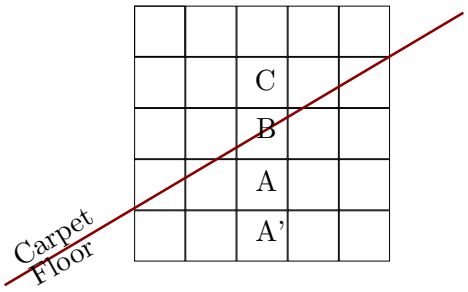


Figure 3.2: Example of carpet-floor boundary divided in superpixels.

Figures 3.1 and 3.2 represent a closer analysis of a small section of the image. Each of the squares represent a superpixel, and the red line represents an edge. In Figure 3.1 the red line represents the edge between the floor and the wall, and in Figure 3.2 it represents the edge between the floor and a carpet (also classified as floor). In this way, there are some superpixels that belong to the floor classification, some that belong to the background classification and some that belong to the edge classification.

Given that in Figure 3.1 the tiles A and A' represent floor superpixels, B represents an edge superpixel and C represents a background superpixel; and in Figure 3.2 the tiles A, A', B and C represent floor superpixels it can be inferred the next knowledge:

- If an edge passes through the superpixel B, the LBP distribution of that tile will have a high EMD (Earth Mover's Distance) with respect to the LBP distributions of the superpixels A and C. Similarly, the LBP distributions of A and C is, probably, quite similar.
- Given that superpixels A and A' belong to the same segmented class and they are neighbours in the image, the EMD between their grayscale distributions is, probably, very small.

- Suppose that superpixel A has already been inferred as floor, and superpixel B as an edge. If there is a big EMD between the colors' distributions of A and C, it is highly probable that C is a background tile.

Besides everything that has been stated before is fulfilled for the wall-floor boundary, similar properties fit in the carpet-floor boundary. This means that, by only analysing a small portion of the image it may be quite difficult to make a correct classification. The solution to solve this issue is the introduction of global knowledge in the analysis.

3.1.2 Global knowledge

In an attempt to solve some of the problems found in the previous section, it is introduced the global knowledge in the analysis. This global knowledge consist in the insights that can be retrieved from analysing the whole image. First, the faced scenarios in the current problem are the ones shown in Figures 3.3, 3.4 and 3.5.

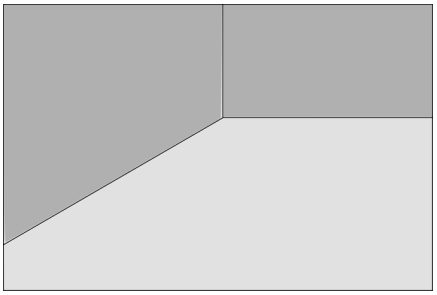


Figure 3.3: Scenario of floor-wall boundary with AV pointing left.

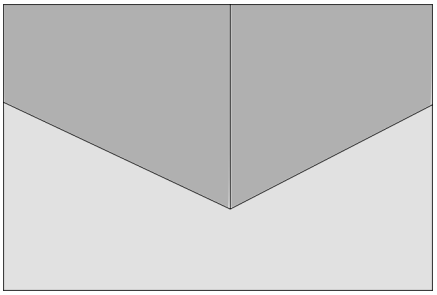


Figure 3.4: Scenario of floor-wall boundary in the intersection of 4 corridors.

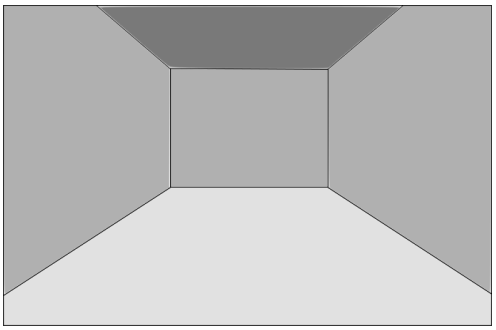


Figure 3.5: Scenario of straight corridor. Most common scenario.

From these figures it can be inferred some knowledge of the behaviour of the floor and the background in the whole image. Next, it is shown the statements of the global knowledge in which the analysis is based.

- Between two different areas (eg. Floor and wall) there is an edge (probably with a high gradient in grayscale intensity), i.e., the transition between two areas can be modeled and differs from both of these areas.

- The camera has a permanent and fixed position at the AV, with a height of 0.47 m and a fixed angle with respect to the ground. Because of this, the floor is always located at the lower part of the image. The closer is the analysed section to the bottom of the image, the more probable is that it belongs to the floor.

- The corridors usually have a symmetric structure. This means that, if the AV is moving through the central part of the corridor (expected route for this AV), and because of the fixed position of the camera, in the centered part of the X-axis of the image it is more probable to find the floor, and in the extremes of the X-axis of the image it is more probable to find walls. This scenario is shown in Figure 3.5, in which it is represented a straight corridor in which the AV moves. This scenario is the one present in most of the frames.

- Usually, and because of the position of the lights in the ceiling, the floor has lighter colors than the walls. This statement may not be true in corridors in which the walls and the floor have very different colors.

- After performing a Felzenszwalb segmentation, big areas may belong to the floor and the walls, and small areas may belong to obstacles, edges and glares.

- As seen in Figures 3.3, 3.4 and 3.5, walls and floor can usually be modeled as big homogeneous surfaces. Thereby, over a wall and the floor there may not be big gradient values.

- Furthermore, reflections can be characterized as small areas with high intensity and in which the gradient of the transition between the reflection and the current surface has a small value.

Hopefully, and by combining the local and global knowledge it can be inferred the boundary between the wall and the floor in general corridor images. However, the previous insights have to first be adapted to the superpixel approach that has been taken.

3.2 Adaptation of the prior knowledge to the superpixel approach

The knowledge that was introduced in Chapter 3.1 has to be adapted to the superpixel approach. For doing so, the features are divided in three categories. First, the tile's own features, i.e., the features that can be inferred directly from the superpixel that is being analysed. Second, the local features, which are the features that can be inferred from the close neighbourhood of the superpixel that is being analysed.

Finally, the global features, which are the attributes taken from analyzing the image as a whole.

Dividing the collected features into three categories is advantageous when doing the modeling. Each of the three categories takes a different approach for segmenting the floor. For example, it is improbable to get a good segmentation using only local knowledge, as it would not be easy to infer the floor section by only looking to a small portion of the image. However, local knowledge introduces very useful information, as, if a superpixel is very similar to its neighbour, it is highly probable that both, the superpixel and the neighbour, belong to the same class. In order to be able to use this useful information in an accurate way, it is indispensable to introduce other features such as the position in the image of the tiles (global knowledge). In a similar way, global knowledge introduces very useful information in the analysis, but it is not sufficient on its own.

Notice that each of the superpixels of the image has assigned all the features that are exposed in this section, but not all of them are used for performing the classification. This is better explained in Chapter 5.

3.2.1 Tile's features

First, the features that can be inferred by exploring the superpixel on its own are:

1. Entropy of the grayscale values. Usually, the floor and the walls are homogeneous big surfaces. Thereby, if a superpixel has a high entropy of the grayscale pixel values it may be because that superpixel corresponds to an edge superpixel. Furthermore, if there is a high entropy of the grayscale values it may mean, as well, that the superpixel is a reflection.

2. Entropy of the LBP (uniform LBP) values. As with the previous feature, if there is a high entropy of the LBP values in the superpixel, it may mean that the superpixel is an edge tile.

3. Biggest gradient change of the grayscale image in the superpixel. If there is a very big gradient change it may be because it is an edge superpixel.

4. Proportion of pixels that are detected as edge with the Canny edge detector. If there is a big enough number of pixels detected as edge, the superpixel may be in the boundary between the floor an background.

From analysing only the own superpixel features, it can only be made very simple conclusions. For example, if it is a very homogeneous superpixel, it is highly improbable that it belongs to an edge superpixel. Besides this is an useful outcome, more information is needed in order to perform a good segmentation. This is solved by introducing the local and global knowledge.

3.2.2 Local features

Local features take into account the characteristics of the own superpixel that is being analyzed, comparing them with the characteristics of its close neighbourhood of superpixels.

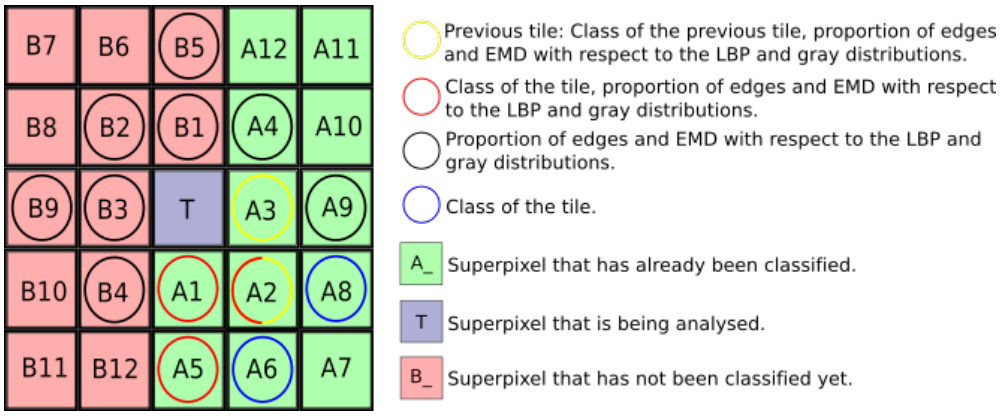


Figure 3.6: Neighbourhood of superpixel T. In green those superpixels that have already been classified and in red those to be classified.

In Figure 3.6 it is shown the neighbourhood from which the local knowledge of tile T is extracted. Each of the small squares represent a superpixel, and the local features of superpixel T are extracted from some of the neighbour tiles that are at a Manhattan distance $\leq 2\, superpixels$ (the circled superpixels). Note that not all the superpixels that meet this definition of Manhattan distance $\leq 2\, superpixels$ from tile T are used to extract the local knowledge.

Due to the order in which superpixels are analyzed by the proposed algorithm, which is explained in detail in Chapter 4, in the neighbourhood of each superpixel there are some tiles that have already been analysed, and some that have not. Hence, in Figure 3.6 the superpixel that is being analysed, i.e., T, is represented in blue. Those superpixels that have already been analysed and classified as either floor, edge or background, i.e., *A1, A2, A3, A4, A5, A6, A7, A8, A9, A10, A11* and *A12* are represented in green. Finally, those superpixels that haven't been analysed, i.e., *B1, B2, B3, B4, B5, B6, B7, B8, B9, B10, B11* and *B12*, are represented in red. It should be noted that, depending on the section of the image that is being analysed, the superpixels *A3, A9, A10, A11* and *A12* may have not been classified at the moment in which tile T is being analysed. This is better explained in the next chapter.

Thus, the features that have been extracted to constitute the local knowledge of a superpixel are:

1. Class of the nearby previously analyzed superpixels. These superpixels are represented in Figure 3.6 with the IDs *A1, A2, A5, A6 and A8*.

2. Class of the superpixel that has already been analysed (previous superpixel) that is more "in line" with the expansion followed by the algorithm (explained in Chapter 4). This superpixel can be interpreted as the one that has the closer relationship with tile T according to the expanding analysis process of the algorithm. In other words, it is the superpixel with highest probability to have the same classification (floor, edge or background) of tile T. In Figure 3.6 it corresponds, most of the times, with the class of the superpixel *A3* (sometimes, and due to the expanding algorithm it corresponds with the class of the superpixel *A2*).

3. EMD with the grayscale distributions of the superpixels with IDs *A1, A2, A3, A4, A5, A9, B1, B2, B3, B4, B5, B9*. If the EMD distance of the grayscale distributions between the superpixel T and some of the superpixels $A*$ is not very big, then it is highly probable that they all belong to the same classification (floor, edge or background). Likewise, if the grayscale distribution of superpixel T has a high EMD with respect to the superpixels *A1, A2, A3, A4, A5* and *A9*, it is probable that they belong to different classes. However, in this case it may be needed to analyze other features.

4. EMD with the LBP (uniform LBP) distributions of the superpixels with IDs *A1, A2, A3, A4, A5, A9, B1, B2, B3, B4, B5, B9*. As explained in section 3.1.1, this is an useful feature for detecting edges. The LBP distribution of a superpixel through which an edge passes may have a big EMD with those LBP distributions of the superpixels through which an edge does not pass.

   Note that in 3 and 4 it is collected all the EMD between the surrounding superpixels with respect to the one that is currently being analysed (tile T), but only the most important ones are used.

5. Proportion of pixels, of the superpixels with IDs *A1, A2, A3, A4, A5, A9, B1, B2, B3, B4, B5, B9*, that have been detected as edge when processing the image with Canny edge detector. This is a very useful feature when classifying superpixels that belong to edges. It has been chosen the proportion of pixels that are detected as edges instead of a simple boolean value that indicates whether an edge has been detected in that superpixel or not because this feature contains more information. For example, if a superpixel contains many pixels detected as edges (more than usual for an edge), it is probable that it is a corner, providing information to the model about the geometry of the surrounding superpixels.

6. Grayscale intensity with respect to the mean grayscale intensity of the nearby neighbourhood, i.e., with respect to the grayscale intensity of the superpixels *A1, A2, A3, A4, A5, A9, B1, B2, B3, B4, B5, B9*. If the value of this feature is big, it may be because there is a glare in the superpixel T.

Notice that, from the superpixels with ID *A6* and *A8*, the only feature taken is the class. This may be needed in an scenario in which an edge passes through the

superpixels *A1, A2, A3, A5* and *A9*. In this case, if the superpixels *A6* and *A8* have already been classified as floor, the superpixels *A1, A2, A3, A5* and *A9* have already been classified as edge, and there is a small EMD distance between the grayscale distribution of the superpixel T and the grayscale distributions of the superpixels *B1, B2, B3, B4, B5* and *B9*, it is highly probable that T is a background superpixel.

3.2.3 Global features

It is highly complicated to make a good prediction by only analysing a small portion of the image (tile's features and local features). To ease the analysis, some reasoning has to be done over the entire image. Thus, global features are retrieved by analysing the image as a whole. These are quite valuable as they prevent from doing impossible classifications such as having superpixels classified as floor in the top part of the image. The global features that have been retrieved are:

1. Position X and Y of the superpixel. This feature consists in the tuple (X,Y). It is is taken, in absolute value, with respect to the superpixel that occupies the position at the centered bottom part of the image (superpixel coloured in blue in Figure 3.7). It consists in the Manhattan distance with respect to this superpixel, being *superpixel X-axis length* and *superpixel Y-axis length* the unit values for each of the measurements.

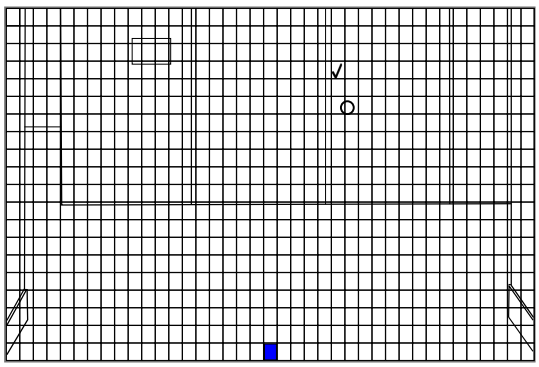


Figure 3.7: In blue, superpixel at position (X,Y) = (0,0).

2. After calculating Felzenszwalb segmentation in the image, it is taken the size of the smallest area to which that superpixel belongs. This can be an useful feature as big areas may belong to the floor and the walls, and small areas may belong to obstacles, edges and reflections. **In this work, the minimum size of the Felzenswalb segmentation is 1000 pixels for images with size 400x702.*

3. Mean of the grayscale and LBP values of a superpixel with respect to the mean values of the whole image. This may be useful as, because of the position of the lights in the ceiling, usually the floor has lighter colors than the walls. In this way, the floor, probably, has a value under the mean intensity of the image, while the walls have a value over the mean intensity value of the image.

4. Maximum grayscale gradient value (calculated as explained in Chapter 2.3.2) over, under, at right side and at left side of the superpixel. As said before, floors and walls are usually big homogeneous surfaces. Moreover, the edge between the floor and the background may have a big gradient value (of the grayscale image), while in the floor and in the walls there may not be big gradient values. In this line, it is taken the maximum gradient change of the grayscale image directly over the superpixel (those superpixels that have the same x-coordinates and whose y-coordinates are bigger), under, at right (those superpixels that have the same y-coordinates and whose x-coordinates are bigger) and at left. In this way, a superpixel that belongs to the floor may have a big gradient value over it, a small one under it and probably (depending if there are side walls) big gradient values at both sides. Similarly, a superpixel that belongs to a wall (background) may have a small gradient value over it and a big one under it. In addition, a superpixel that belongs to a door (background) probably has big gradient changes in the four directions.

3.2.4 Dataset and analysis

Following what was stated in the previous chapter, it is constructed a dataset for each image, in which all the features are collected for each of the superpixels. However, besides all these features are knowledge-based, they all may not be useful. The goal of this thesis is analysing which combination of the collected features results in a good segmentation. In this way, this collection of features could be of great help in further studies.

It should be noted that a correct segmentation can only be done when combining some correlated features. For example, the fact that a superpixel has a high value of intensity compared to the nearby superpixels does not give much information. However, if it is also taken into account other features such as the fact that the superpixel is at the lower part of the image, surrounded by other superpixels that have been classified as ground, and with a small value of the gradient of the grayscale pixel values, then with great certainty it can be deduced that the superpixel corresponds with a glare and it can be classified as floor. On the other hand, if this superpixel is in the upper part of the image, it may not be classified as floor.

3.3 Knowledge graph

The knowledge for segmenting the floor from the rest of the image following a superpixel approach can be summarized in the graph shown in Figure 3.8.

In the images there are several objects represented (door, walls,...) which conform to either background or floor. In addition, these classes are constituted by a group of superpixels. Moreover, these superpixels can be described with different features that conform to either local knowledge or global knowledge. Finally, these superpixels have also an unique class, which is floor, background or edge.

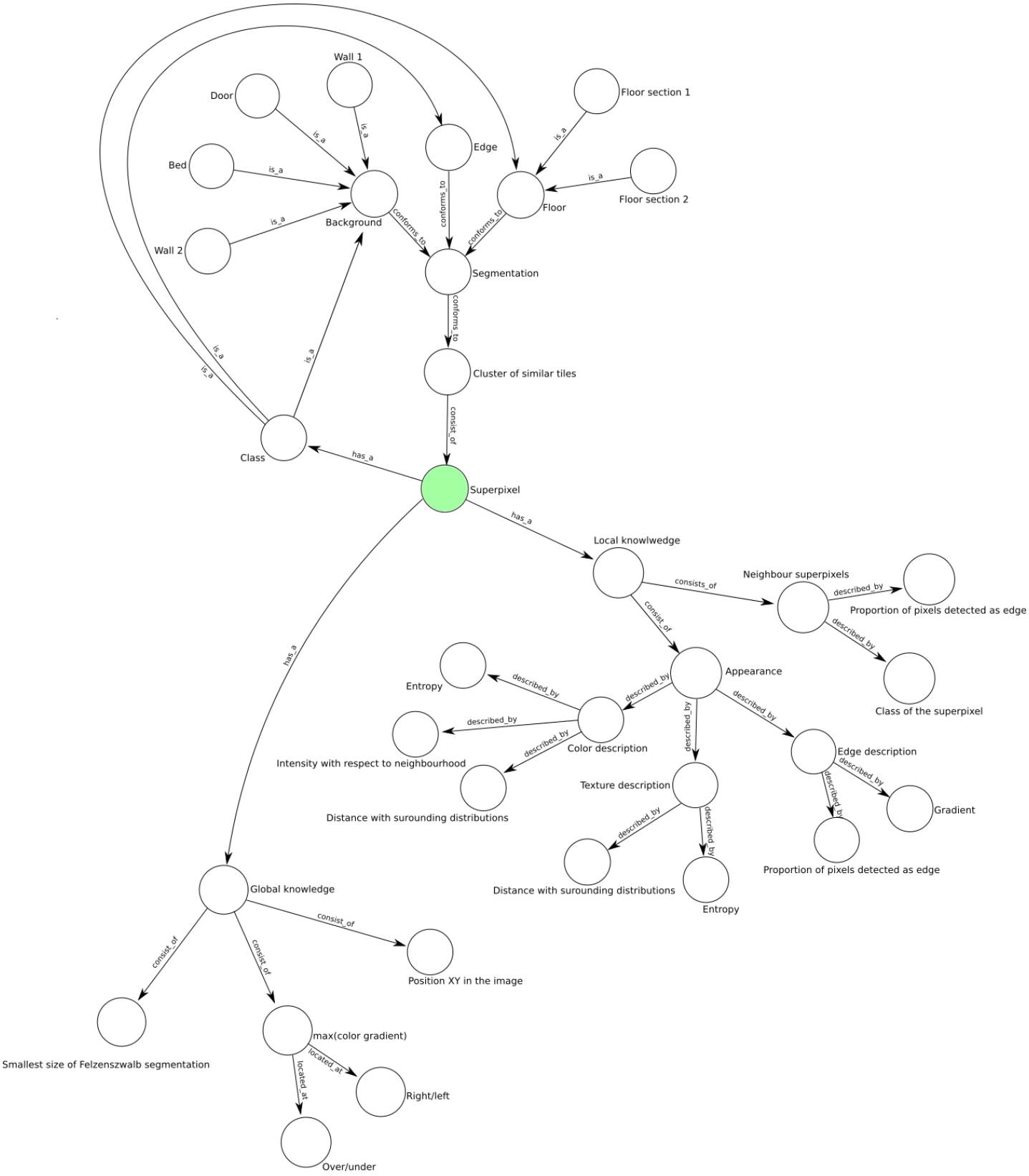


Figure 3.8: Knowledge graph of a tile.

Chapter 4

The algorithm

By localizing and inferring the movable area, the AV is be able to make a planning of its following movements. For doing so, and correctly segmenting the image, a series of steps have to be followed. Thus, it is not only important the feature retrieval, but also the order in which it is done.

Note that the algorithm is based in a region expanding process, in which the analysed superpixels have to be taken in a concrete order. Furthermore, due to the relationships between the superpixels and the classification of the neighbours, that are taken for classifying new superpixels, the order in which these are analysed is of vital importance.

In addition, the decision tree model has to be trained before the application of this algorithm to new images.

4.1 Order of analysis of the superpixels

The order of analysis is of great importance when classifying superpixels, as some of the features that may be taken for doing the analysis are the class of previously classified superpixels. The algorithm consist in a region growing process, starting at the bottom most-centered part of the image and expanding in the two directions of the X-axis (left and right) and up in the Y-axis. The directions in which the new superpixels are taken for their classification are shown in Figure 4.1.

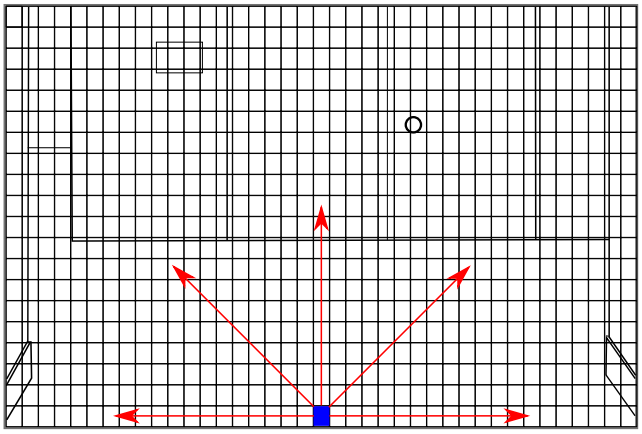


Figure 4.1: Directions of growth.

Following this line, in each region growth the group of superpixels that is closer to the already analysed area is taken. This process can be seen in Figures 4.2, 4.3 and 4.4. In these figures the already analyzed area is represented in blue color and the area that is being analyzed in the current step (new region growth) is represented in orange. Furthermore, in each step in which a region growth is done, the new superpixels are analyzed in the order signaled by the coloured arrows. First, those superpixels marked with a red arrow are taken from the bottom part of the image to the upper part. Then, those superpixels marked with a green arrow are analysed from the left part of the image to the right. Finally, those superpixels marked with a black arrow are taken from the bottom part of the image to the upper part.

This region growing process is carried out until the whole image is analysed. This is shown in Figure 4.4.

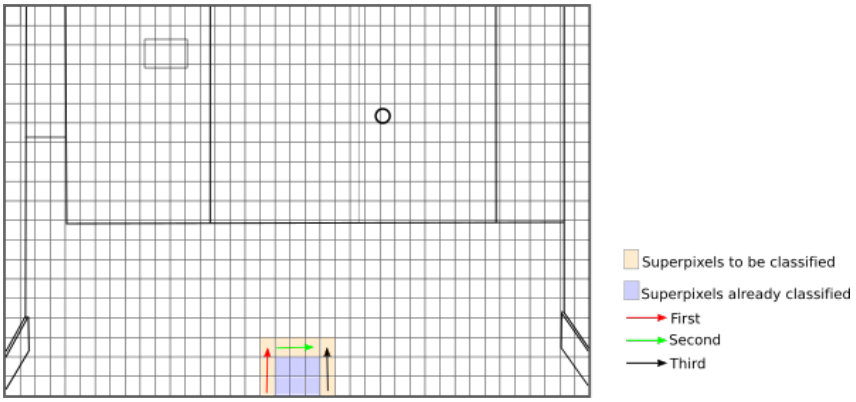


Figure 4.2: First growth.

Notice that the IDs of the neighbour superpixels shown in Figure 3.6 are adapted for each of the superpixels under the three arrows. Those superpixels under the red arrow in the above figures have the same neighbour IDs orientation than in Figure

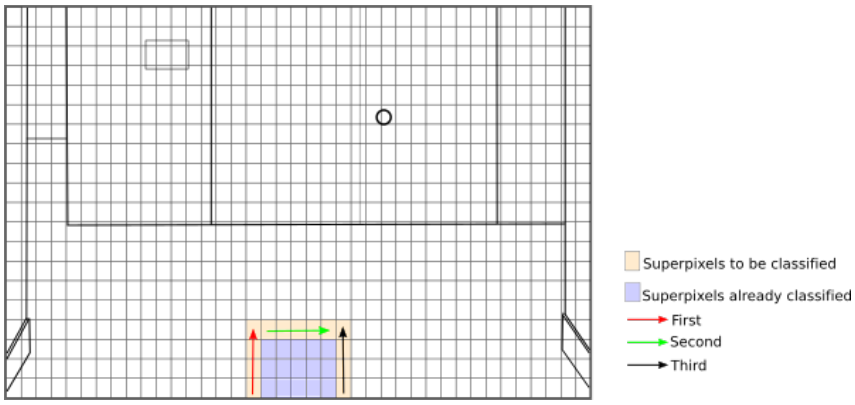


Figure 4.3: Second growth.

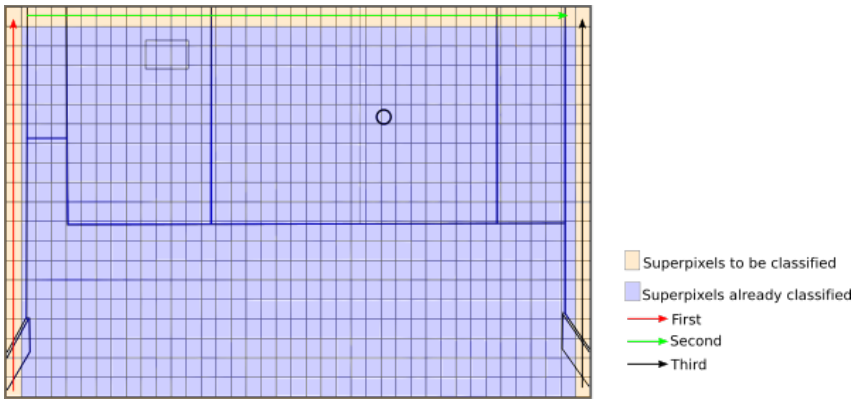


Figure 4.4: Last growth.

3.6. Those under the green arrow have the neighbour IDs shown in Figure 3.6 rotated 90° to the right. Finally, those superpixels under the black arrow have the neighbour IDs shown in Figure 3.6 flipping their orientation.

Furthermore, in Chapter 3.2.2, it is introduced the "previous superpixel" concept. As said, this superpixel is the one that is more "in line" with the region growing process, i.e, it is the one with a more similar relationship with respect to the superpixel that is being analysed. For those superpixels that are being analysed (in orange color in the above figures), this "previous superpixel" is the closer in distance superpixel of those that have already been analysed (in blue color in the above figures). In this way, it usually corresponds with the superpixel with ID A3 in Figure 3.6. However, for those orange superpixels that are in the corners it corresponds with the superpixel with ID A2.

4.2 Four decision trees - local knowledge

When using local knowledge for implementing the classification of the superpixels, four different decision trees have to be used. There are two reasons for this.

First, as it is shown in Figure 3.6, some of the local features are based in the neighbourhood of the tile that is being analyzed. However, for those superpixels that are at the borders of the image, some of these neighbour superpixels are out of the bounds. To solve this, it can be developed a custom decision tree for those superpixels that have less neighbours than the most centered ones, not using the information of those neighbour superpixels as a feature. This situation occurs with all those superpixels whose Manhattan distance to the edge of the image is lower or equal to one superpixel. However, because the stated problem consists in performing the segmentation of the image to drive the actions of an AV, it is not necessary to classify those superpixels that are in the top, right and left edges of the image. Thus, due to this problem, two additional and independent decision trees are needed. Notice that, in case it is wanted to segment all the borders of the image (right, top, left), it would be needed 12 additional decision trees.

Second, due to the order in which the superpixels are taken to be analyzed, shown in Figures 4.2, 4.3 and 4.4, the class of some of the neighbour superpixels is already inferred at the moment of classifying a tile. However, depending on the position in the image of the superpixel that is being analysed, a different set of neighbour superpixels (with IDs shown in Figure 3.6) would have been already classified at the moment of evaluating the new tile. In other words, those superpixels that are under the green arrow in Figures 4.2, 4.3 and 4.4 have a different set of already classified neighbours than those superpixels that are under the red and black arrows in these same figures. This means that two additional decision trees have to be inferred for all those superpixels whose Manhattan distance to the edge of the image is greater than one superpixel.

Altogether, when using local knowledge, four different decision trees have to be inferred, each of them for a concrete set of superpixels. The regions over which each of these four decision trees act are shown in Figure 4.5. First, those superpixels that are at the top, right and left borders of the image (gray color in the figure) are not classified. However, their features are used for the classification of the nearby superpixels. Then, two different decision trees are needed to classify those superpixels that are at the most bottom part of the image (red (DT1) and green (DT2) in the figure) because the do not have the same set of neighbour superpixels under them. In addition, for classifying the rest of the superpixels of the image (blue (DT3) and yellow (DT4) in the figure), two additional decision trees are needed. This is because they have a different set of already classified neighbours, as explained above.

Note that these additional decision trees are only needed when using local knowledge as it is represented in Figure 3.6. When not using the features of the neighbourhood, it is not necessary to have different decision trees, each for a different part of the image. Furthermore, all the decision trees can be merged into one big decision tree in which the first step separates the tiles according to their coordinates, directing them to their corresponding sub-decision tree.

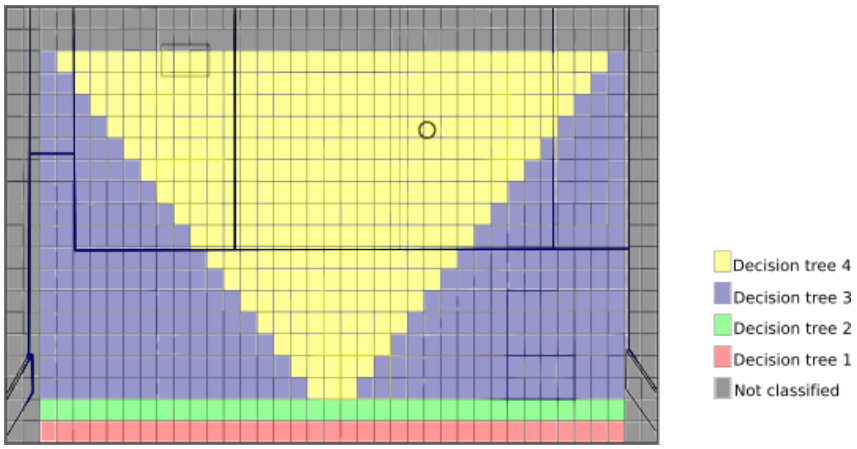


Figure 4.5: Regions of the four decision trees (local knowledge).

It is important to emphasize in the names given to the four decision trees:

- DT1: Decision tree that infers the class of the red superpixels in the above figure.
- DT2: Decision tree that infers the class of the green superpixels in the above figure.
- DT3: Decision tree that infers the class of the blue superpixels in the above figure.
- DT4: Decision tree that infers the class of the yellow superpixels in the above figure.

4.3 Formalization of the algorithm

The next steps are proposed for doing an image segmentation from a knowledge-basis.

1. First, the image is divided in superpixels. Concretely, all the superpixels should have a fixed size (x, y) that is kept for the whole sequence of frames. Additionally, and for the correct application of the region growing-like algorithm, there has to be n superpixels in the y-axis and $2 * n - 1$ superpixels in the x-axis. In order to adapt the algorithm to the image dimensions, the parameters (x, y) and n have to be pretuned. However, it may be needed to do a small resizing of the image.
2. The second step of the process consists in the creation of the data frame with the collection of features for each of the superpixels. In this way, a series of transformations for extracting the features are applied to the image and it is generated a database in which each row corresponds with a superpixel, and each column corresponds with a feature.

Note that some of the local features are the classification of previously analyzed superpixels. In case this feature is taken for performing the classification, this data would not be taken in this step, but instead the database will be actualized in an online basis as new classifications are made.

3. The next step consists in setting the ground truth. Because of the fixed position of the camera in the AV, and being the center part of the corridor its expected path, some superpixels can be set as floor (ground truth). Concretely, the six bottom most-centered superpixels are grounded to floor class. These superpixels are signaled in blue color in Figure 4.6

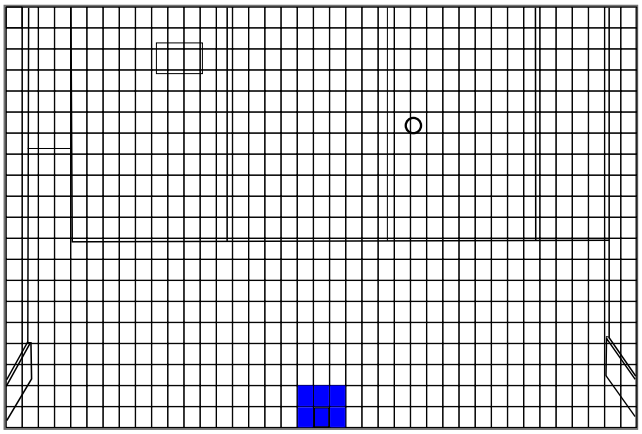


Figure 4.6: Ground truth of the image.

Note that this step is only needed for the cases in which the class of previously classified superpixels is used for the segmentation of new superpixels. In case these features are not used in the analysis, this step can be skipped and no prior ground truth is set (raw image). Furthermore, if desired, a fifth decision tree can be done for inferring the class of these superpixels. However, the preferred approach in this thesis was directly grounding them to floor.

4. Then, the superpixels are taken in a certain order and classified. For performing this classification the next steps are followed.

   4.1. Take the next superpixel to be classified following the order specified in Chapter 4.1.

   4.2. Classify the new superpixel with the decision tree model. Given some features, the decision tree is fed and thus, the superpixel is classified as either floor, edge or background.

   4.3. If the class of the neighbour superpixels is taken as an attribute, actualize the database of features with the class of the newly classified superpixel.

   4.4. If there are superpixels left for the classification, repeat step 4.1..

5. Return the segmented image.

Note that the final segmentation consist in a superpixel classification instead of a pixel-wise classification. However, for orientation and driving the actions of the AV there is no need of a finer pixel-wise segmentation. Furthermore, if needed a finer segmentation, the prior superpixel classification obtained with this algorithm will incredibly ease the pixel-wise classification.

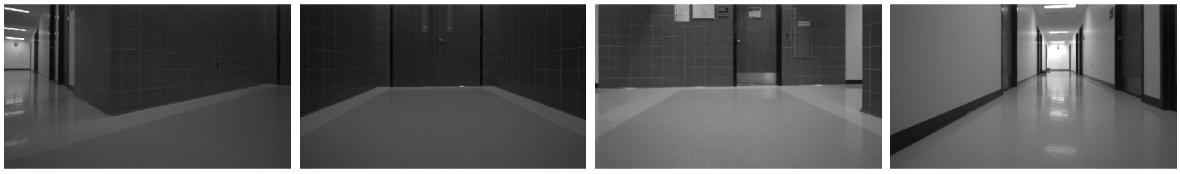
Chapter 5

Test and results

In this chapter it is shown the results obtained when applied the algorithm shown in Chapter 4 on images. This chapter consists in three different tests. The goal of the first two experiments is to infer the effects in the segmentation boundaries of the local and global features when tested independently. The third experiment consist in a wiser feature selection based in the previous two experiments and in an analysis of the importance of the features.

(a) (b) (c) (d)

Figure 5.1: Images used for testing the algorithm.



(a) (b) (c) (d)

Figure 5.2: Images used for testing the algorithm.

The procedure followed in all the experiments is the same. First, the images are divided in superpixels. As explained in 4, all the superpixels have a fixed size (X, Y). Additionally, there has to be n superpixels in the Y-axis and $2n - 1$ superpixels in the X-axis. In this way, the parameters chosen for the tests of this chapter are $(X, Y) = (18, 20)$ and $n = 20$.

Hence, in the present work each of the images is first resized from a size of 400x695 to a size of 400x702, and then it is divided into 780 superpixels, each with a size of 18x20. Note that these numbers were selected accordingly to the size of the image and in order to get a reasonable number of superpixels per image.

Then, the database of 203 images is divided in three groups. First, eight random images that represent all the scenarios shown in Figures 3.3, 3.4 and 3.5 are taken as test set. These images are shown in Figures 5.1 and 5.2. Note that, for comparing purposes, all the experiments are tested over this same set of eight images. Then, the remaining 195 images are divided in a train group (90% - 176 images) and a validation group (10% - 19 images). First, in order to avoid overfitting, the validation set is used to analyse the best depth of the decision tree trained with the train set. Then, the definitive decision tree is trained with the whole set of images, but the testing ones (195 images).

5.1 Experiment 1 - Classification made using only local features

In this first experiment, it is analysed the influence of the local features on the segmentation. For so, it is trained a set of four decision trees as explained in Chapter 4.2, using only local features. Notice that the purpose of this test is not to get a good segmentation, but to study the properties that have the local parameters in the segmentation of the image.

Thus, the features that are studied for creating the decision trees of this first test are the proportion of pixels detected as edge in the tile that is being analysed, its entropy of the grayscale and LBP distributions, the class of the previous tile and of the superpixels with ID A1, A2, A5, A6 and A8 in Figure 3.6, the EMD of the grayscale and LBP distributions with respect to the neighbours' distributions and the proportion of pixels detected as edge in all its neighbours.

5.1.1 Decision trees and feature importance

Using a little set of features for taking a decision may lead to a bad classification, while using too many features may lead to overfitting. To solve this, a study of the effect of the maximum depth of the decision trees on the accuracy is made.

In Figures 5.3, 5.4, 5.5 and 5.6 it is shown the accuracy of each of the four decision trees, dependant on their maximum depth. For the decision trees (DT in the figures) 1 and 2 (red and green in Figure 4.5) it is obtained the best results when setting a maximum depth of the decision tree to 4. Notice that, because of the simplicity of the features and because of the bigger number of superpixels that belong to floor or background than superpixels that belong to edge class, it is better setting a slightly big depth of the decision tree. This is because the classification of a superpixel as edge can be done when analysed a few number of features, and not basing the decision in only one or two characteristics of the superpixel. That is why 4 is a good maximum depth of these decision trees.

For the decision tree 3 (blue in Figure 4.5) the best results are obtained with a maximum depth of 8, and for the decision tree 4 (yellow in Figure 4.5) the best results are obtained, as well, when setting a maximum depth of 8. Notice that in the lower figures it is shown the accuracy of each of the decision trees on the validation set, supposing that previous classifications were correctly done. The depth is chosen in this way, and not basing it on previous real classifications, so the inference made by the decision trees relies always on correct previous predictions and they do not overfit trying to predict on incorrect previous decisions.

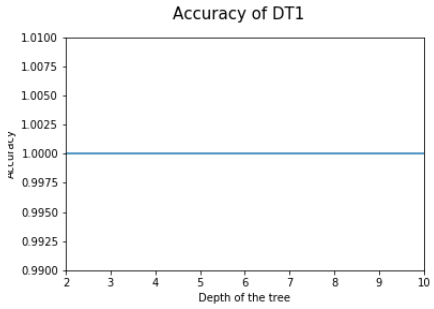


Figure 5.3: Accuracy - depth of DT1.

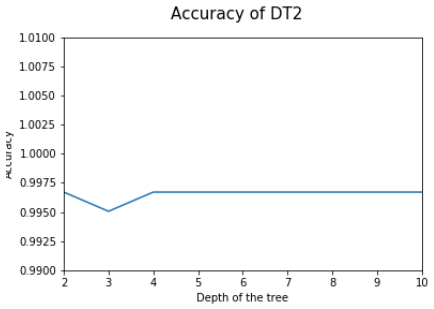


Figure 5.4: Accuracy - depth of DT2.

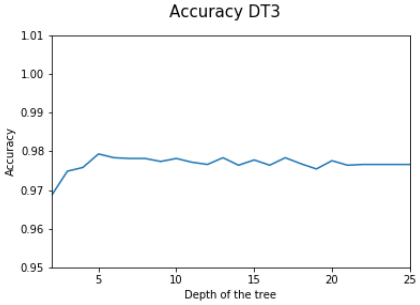


Figure 5.5: Accuracy - depth of DT3.

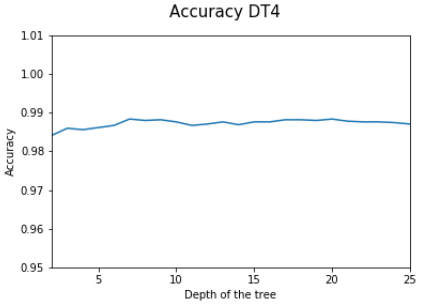


Figure 5.6: Accuracy - depth of DT4.

Note that DT1 and DT2 infer in the lower part of the image, where, in almost all the frames, the superpixels are classified as floor (in a few frames there is an edge at the corners of these regions). That is the reason for their very high accuracy. Thus, the main decision trees (the ones that suppose a bigger challenge) are DT3 and DT4.

However, it is noticeable the small dependency of the accuracy on the decision tree depth. This is better seen when analysing Table 5.1, in which it can be seen the importance of the used features for each of the four decision trees (DT1, DT2, DT3 and DT4 in the table). Note that some of the features of the neighbours (marked with "-") are not available for the decision trees DT1 and DT2, as these neighbours are out of the bounds of the image for these superpixels.

As it is shown in the Table 5.1, the most important features for classifying new superpixels is the class of the surrounding superpixels. This makes a lot of sense, as a superpixel would, in most of the cases, have the same class of its neighbours.

Feature	**DT1**	**DT2**	**DT3**	**DT4**
Class of the previous tile	0.503	0.72	0.746	0.768
Class of tile A2	-	0	0.162	0.011
Max gradient of the grayscale pixels of tile T	0	0	0.029	0.011
Class of tile A1	-	0	0.010	0.160
Gray intensity relative to neighbourhood	0.117	0.030	0.006	0.004
Proportion of pixels detected as edge of tile T	0	0.188	0.006	0.003
Proportion of pixels detected as edge of tile B4	-	0	0.005	0.001
Proportion of pixels detected as edge of tile A3	0	0	0.005	0.002
EMD with the LBP distribution of tile A2	-	0	0.002	0
Proportion of pixels detected as edge of tile B2	0	0	0.002	0.002
Proportion of pixels detected as edge of tile B3	0	0	0.002	0.003
EMD with the LBP distribution of tile B5	0	0	0.002	0
EMD with the grayscale distribution of tile A1	-	0	0.001	0
Proportion of pixels detected as edge of tile A1	-	0	0.001	0
Proportion of pixels detected as edge of tile A2	-	0	0.001	0.001
EMD with the grayscale distribution of tile A5	-	-	0.001	0
EMD with the grayscale distribution of tile B4	-	0	0.001	0
Entropy of the LBP distribution of tile T	0.102	0	0.001	0.001
EMD with the grayscale distribution of tile B3	0.042	0	0.001	0.001
EMD with the LBP distribution of tile B9	0.032	0	0.001	0.001
Entropy of the grayscale distribution of tile T	0	0	0.001	0.003
EMD with the grayscale distribution of tile A3	0	0	0.001	0
EMD with the grayscale distribution of tile A4	0	0	0.001	0
EMD with the LBP distribution of tile A4	0	0	0.001	0.001
Proportion of pixels detected as edge of tile A4	0	0.049	0.001	0.008
EMD with the grayscale distribution of tile A9	0	0	0.001	0
EMD with the LBP distribution of tile A9	0	0	0.001	0
Proportion of pixels detected as edge of tile A9	0	0	0.001	0
EMD with the grayscale distribution of tile B2	0	0	0.001	0
EMD with the LBP distribution of tile B2	0	0	0.001	0.001
EMD with the LBP distribution of tile B3	0	0	0.001	0
EMD with the grayscale distribution of tile B9	0	0	0.001	0
EMD with the LBP distribution of tile A1	-	0.01	0	0
Class of tile A6	-	0	0	0.001
Proportion of pixels detected as edge of tile B1	0.204	0	0	0.008
EMD with the LBP distribution of tile A3	0	0	0	0.001
EMD with the LBP distribution of tile B1	0	0.002	0	0
EMD with the grayscale distribution of tile B5	0	0	0	0.001
Proportion of pixels detected as edge of tile B5	0	0	0	0.001
Proportion of pixels detected as edge of tile B9	0	0	0	0.001

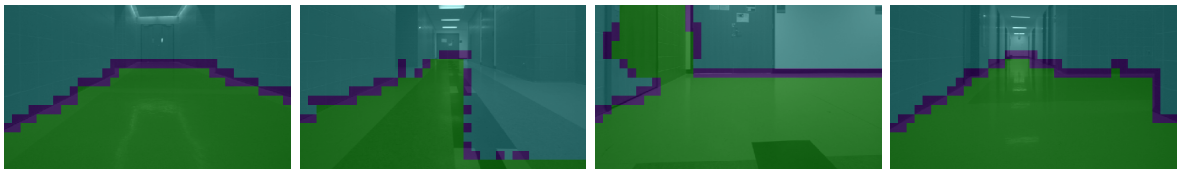
Table 5.1: Importance of the local features.

In addition, the next more important features after the class of nearby superpixels, are those that can be used for detecting the edge between the floor and the wall. Some of these are the gradient in grayscale of the superpixel, the proportion of pixels detected as edge and the entropy of the LBP. For example, in DT1 and DT2, in the last leaf of one of the branches of the decision trees, a superpixel is classified as edge if the entropy of the LBP or the proportion of pixels detected as edge of that tile are over a threshold.

Furthermore, by looking at the above numbers, it is noticeable the high dependence in the class of the previous classified superpixels in the decision-making process. That is the reason why the accuracy of the decision trees varies very little when changing the maximum depth of the decision tree, as shown in Figures 5.3, 5.4, 5.5 and 5.6. However, this could lead to problems as the inference done on new superpixels may be based in wrong previous classifications. This is better shown in the next section.

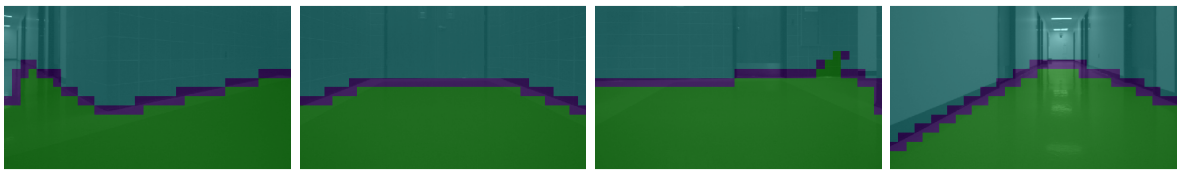
5.1.2 Result of the segmentation

In Figures 5.7 and 5.8 it is shown the segmentation results on the eight test images. In green it is represented those superpixels classified as floor, in dark blue it is represented those superpixels classified as edge and in light blue it is represented those superpixels classified as background.



(a) (b) (c) (d)

Figure 5.7: Classification made using only local features.



(a) (b) (c) (d)

Figure 5.8: Classification made using only local features.

In the above figures it can be seen that a relatively good segmentation is done for the images 5.7a, 5.7d, 5.8a and 5.8b, 5.8c and 5.8d. However, by looking at images 5.7b and 5.7c two remarks on the drawbacks of using local features can be done.

First, and because the inference of the decision tree on new superpixels is dependant on previous classifications, an early misclassification may drag the error all along the image. This is, somehow, linked to the region growing algorithm explained in Chapter 4, in which the classification of new superpixels is done from the most bottom-centered part of the image to the top, right and left parts of the image. Thereby, if a superpixel is misclassified, that error would be dragged to those superpixels that are on top and at the side of that superpixel. This can be seen in the image 5.7b, in which an early misclassification of a floor superpixel next to the bottom part of the image leads to the incorrect classification as background of the nearby floor superpixels. A similar effect, but in lower scale happens in the image 5.8a. This can be avoided by introducing global knowledge.

Second, as there is no spatial relationships in the image, the decision tree can do wrong inference such as classifying as floor superpixels those that are at the top part of the image. However, because of the fixed position of the camera in the AV, this is very improbable. An example of this is shown in the image 5.7c. This can be solved by adding global knowledge to the analysis, such as the position of the superpixel in the image.

Indeed, from an "expert" point of view the obtained results make a lot of sense. By analysing only a small portion of the image, a human can only infer that, if a region is quite homogeneous it, most probable, would belong to the same class of the nearby regions. As well, if there is a strong edge in that small portion of the image, a human can say that it is, with some uncertainty, an edge between the floor and the background. However, humans use much more information to perform this segmentation of the floor. Furthermore, despite the usefulness of local features, analysing the image as a whole is essential for this task.

As well, the accuracy of the prediction of the class of the superpixels of the eight test images is 0.935. The mean accuracy is greatly reduced due to the prediction of the images 5.7b and 5.7c.

5.2 Experiment 2 - Classification made using only global features

In the second experiment, it is tested the effects of the global features in the segmentation made on the whole image. Moreover, in contrast with the first experiment (Chapter 5.1), as the features of the neighbours are not being used, only one decision tree is needed for the segmentation of the whole image. Again and similarly to the previous test, the purpose of this experiment is not to get a good segmentation, but to study the properties that the global parameters have on the segmentation boundaries.

Thus, the features studied in this section are the position X and Y measured in superpixels with respect to the bottom most-centered superpixel (shown in blue color in Figure 3.7), the mean grayscale and LBP values of the superpixel with respect to the whole image, the minimum size of a Felzenswalb area to which the tile belongs and the maximum gradient change at top, right, bottom and left of the superpixel.

5.2.1 Decision tree and feature importance

First, similarly to the previous section, a study of the dependence of the accuracy of the decision tree on its depth is made. Notice that, due to the simplicity of the features and the need to use a few of them to do a good prediction, this step is crucial for every feature selection.

In Figure 5.9 it is shown the dependence of the accuracy of the decision tree on its depth. From this figure it can be said that a maximum depth of 10 is a correct one.

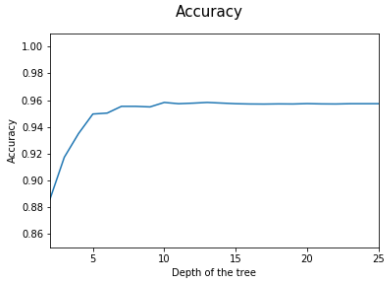


Figure 5.9: Accuracy dependent on the depth

Additionally, and similarly to the previous experiment, the accuracy slightly changes when setting the maximum depth of the decision tree over 5. This is because of the great importance of the first splits in comparison with the rest. It means that most of the superpixels can be classified using a very small set of features, and only some of them need a deeper analysis. Indeed, this is a quite intuitive. For example, knowing that a superpixel is in the lower part of the image and no edge is detected, one can classify it as floor with a very high probability. Only those superpixels that are located next to the boundary between the floor and the background have a more challenging analysis of features for their subsequently classification.

In Table 5.2 it is shown the importance of each of the features in the decision tree. As expected, the most important global feature is the position in the Y-axis of the superpixel. This actually makes a lot of sense, as those superpixels that have

Feature	**DT - Importance**
Position of the tile T in the Y-axis	0.71
Maximum gradient change over the tile T	0.132
Mean grayscale value with respect to the whole image	0.072
Mean LBP value with respect to the whole image	0.026
Maximum gradient change under the tile T	0.018
Maximum gradient change at left of the tile T	0.015
Minimum size of the Felzenswalb area	0.013
Maximum gradient change at right the tile T	0.01
Position of the tile T in the X-axis	0.004

Table 5.2: Importance of the global features.

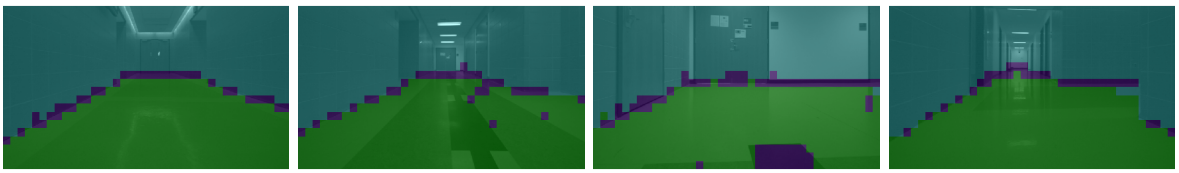
a high value on the Y-axis are, probably, classified as background and those that have a small value on the Y-axis are, probably, classified as floor superpixels. Those superpixels that are located in the middle of the Y-axis of the image are the ones that suppose a harder classification, and for which this feature has little importance. Furthermore, because of the straight decision boundaries of the decision trees, in order to classify a superpixel more than one cut is done in the Y-axis. As well, it is noticeable the contrast between the importance of the position of the tile in the Y-axis and the position of the tile in the X-axis. This is because the AV moves mainly in the centered part of the image and, thereby, the position of the tile in the X-axis has little importance.

In addition, the next feature with higher importance is the maximum gradient change over the superpixel. This is because it is a very useful feature for inferring the class of those superpixels that are in the middle part of the Y-axis of the image (next to the boundary between the floor and the background). If one of these tiles has a high gradient change over it, it is probably because it is a floor superpixel and the edge between the floor and the background is over it. As well, if this value is quite small it probably means that over the tile there is no edge and, thereby, it is a background superpixel.

Finally, as it is shown in the next section, global parameters are not good for inferring the class of the edge tiles. This is because no knowledge of the tile is used for the analysis. However, the maximum gradient change over the superpixel, the minimum size of the Felzenswalb area and the mean LBP value with respect to the whole image can be, with little success, used to solve this problem. However, as shown in the next section, local features would be needed for a correct classification of this type of superpixels.

5.2.2 Result of the segmentation

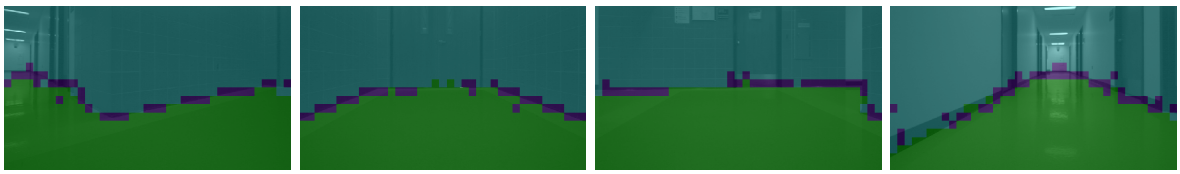
In Figures 5.10 and 5.11 it is shown the segmentation results of the inference made using only global knowledge. Similarly to the previous test, in green it is represented those superpixels classified as floor, in dark blue it is represented those superpixels classified as edge and in light blue it is represented those superpixels classified as background.



(a) (b) (c) (d)

Figure 5.10: Classification made using only global features.

Apparently, it is obtained an slightly better segmentation than when using only local knowledge. In all the images a quite good segmentation is done. This is mainly



(a) (b) (c) (d)

Figure 5.11: Classification made using only global features.

because, in comparison with the local features, an early misclassification does not drag the error over the surrounding superpixels, as each superpixel classification is independent from the rest.

However, as a drawback of the inference done by the global features, the classification of the edge superpixels is quite poor. This is because these features are not very powerful for this task. For example, the maximum gradient over the superpixel is useful for detecting an edge over it, but indeed, not for detecting it in the superpixel. Furthermore, as the neighbourhood of the tile is not being analysed, small gradient changes in the floor, as the ones caused by reflections, can be detected as edges. This misclassification can be seen in the images 5.10b and 5.10c. However, this problem can be solved by analysing the close neighbourhood of the superpixel.

Intuitively, this makes a lot of sense. If one does not have information of the superpixel that is being analysed or of its neighbourhood, it would be quite difficult to infer if it is an edge superpixel. However, it is impressive the good results obtained using only global knowledge.

As well, that the accuracy of the prediction of the class of the superpixels of the eight test images is 0.964. This is almost a 3% better than the accuracy obtained in the first experiment.

5.3 Experiment 3 - Feature selection

As it is shown in the two previous experiments, the selection of features deeply affects the performance of the segmentation. Not only it affects the accuracy of the classification, but also it affects the decisions made on the boundaries of the segmentation. Furthermore, the selection of irrelevant features may lead to overfitting and, thus, a decrease in performance of the model.

In order to avoid this undesired effect and to obtain a collection of features that gives a higher performance, it has been calculated the relevance of each feature by applying recursive feature elimination. Then, based in the results obtained with recursive feature elimination, in the segmentations performed in the previous two experiments shown in Chapters 5.1 and 5.2 and in Tables 5.1 and 5.2, it is suggested a better feature selection.

Additionally, and in order to avoid the problems found in the first experiment (Chapter 5.1), the features that consist in the class of the already analysed superpixels

are eliminated. Besides this is a very useful feature (the one with a highest importance), it may lead to unacceptable segmentations. This is because, as explained, an early misclassification of a single superpixel may drag the error all along the image.

5.3.1 Recursive feature elimination

To make a wiser feature selection it is first studied the importance (i.e., the amount of say) of the features in each of the four decision trees that were introduced in Chapter 4.2. Notice that in this experiment it is studied each of the four decision trees separately in contrast with the second experiment in which only one decision tree was used for the whole image. This is because the features of the neighbour superpixels are used again.

In Appendix A it is shown four tables with the features ordered from most important (1) to less important (50) of each of the four decision trees shown in Figure 4.5, DT1, DT2, DT3 and DT4. Each of the tables corresponds with a decision tree, and each of the columns corresponds with the maximum depth of that decision tree. For the decision trees DT1 and DT2 the maximum depth ranges from 2 to 6 and for the decision trees DT3 and DT4 the maximum depth ranges from 8 to 12. Note that it is important to study the feature importances for different depths because, depending on the depth of the decision tree, the ranking of those features that are at mid-table changes.

In tables A.1, A.2, A.3 and A.4 it is shown that the most important feature for the decision trees DT3 and DT4, is the position in the Y-axis of the image of the superpixel. In contrast, for the decision trees DT1 and DT2 this is not an important feature as all the tiles over which they make inference have the same position in the Y-axis.

Additionally, some of the most important features of the decision trees DT3 and DT4 are:

- Position of the tile in the Y axis.
- Maximum gradient change over the tile T. Feature used to infer the class of those superpixels that are next to the edge between the floor and the background.
- Proportion of pixels detected as edge of the tile T. Feature used to infer if it is an edge tile.
- Mean grayscale value with respect to the whole image. Due of the position of the lights in the ceiling, usually the floor has lighter colors.
- Maximum gradient of the grayscale pixels of tile T. Feature used to infer if it is an edge superpixel.
- Maximum gradient change over the superpixel. It is a more important feature for DT4 as it is used for detecting the edge between the floor and the front wall.

- Maximum gradient change at right and at left of the superpixel. It is a more important feature for DT3 as it is used for detecting the edge between the floor and the side walls.

- Entropy of the LBP distribution of tile T. Feature used to detect edges.

- For the decision tree DT3, the most important relationships with its neighbourhood are with the neighbours with ID A1, B1, A2, B4, A5 and A4.

- For the decision tree DT4, the most important relationships with its neighbourhood are with the neighbours with ID A3, B3, B9, A4, B2, A9 and B3.

  Notice that for the decision tree DT4 the most important neighbours are mainly those superpixels that are aligned in the direction to the Y-axis of the image, while for the decision tree DT3 the most important neighbours are those superpixels that are in the diagonal orientation. This is because DT4 has to infer, most of the times, the boundary of the edge that is in front of the AV (horizontal line) and the decision tree DT3 has to infer, most of the times, the boundary between the floor and the walls that are at both sides of the AV.

In contrast, some of the most important features of the decision trees DT1 and DT2 are:

- The proportion of pixels detected as edge in tile T. Feature used to detect edge superpixels.

- Max gradient of the grayscale pixels of tile T. Feature used to detect edge superpixels.

- Maximum gradient change at both sides of the superpixel. Feature used to detect edge superpixels.

- Mean grayscale value with respect to the whole image. Feature used to detect either an edge or background superpixel, in contrast with the floor superpixels.

- Relationship with the neighbours with ID B3, B2, A1, A9 and A3 Notice that the most important neighbour superpixels of the decision trees DT1 and DT2 are those that are in-line with the X-axis, as the edge superpixels are mostly detected in this direction.

  Furthermore, the most important features of the decision trees DT1 and DT2 are those that can be used to discern between floor and edge superpixels. This is because these two decision trees, in most of the cases, just have to infer floor superpixels, and sometimes edge superpixels.

5.3.2 Feature selection

By analysing the results exposed in Chapters 5.3.1, 5.1 and 5.2 and in Tables 5.1 and 5.2, a wiser feature selection can be make. Because of the different characteristics of the superpixels over which each of the decision trees make inference, a different set of features is chosen for each of them.

The decision tree DT1 is based in the next features.

- Gray intensity relative to neighbourhood.
- Position of the tile in the X-axis.
- Mean grayscale value with respect to the whole image.
- Max gradient of the grayscale pixels of tile T.
- Maximum gradient change at right of the tile T.
- Maximum gradient change at left of the tile T.
- Entropy of the LBP distribution of tile T.
- Entropy of the grayscale distribution of tile T.
- Proportion of pixels detected as edge of tile T.
- EMD with the LBP distribution of tile B1.
- Proportion of pixels detected as edge of tile B1.
- EMD with the grayscale distribution of tile B3.

The decision tree DT2 is based in the next features.

- Mean grayscale value with respect to the whole image.
- Position of the tile in the X-axis.
- Max gradient of the grayscale pixels of tile T.
- Maximum gradient change at left of the tile T.
- Entropy of the grayscale distribution of tile T.
- Maximum gradient change at right of the tile T.
- Entropy of the LBP distribution of tile T.
- Proportion of pixels detected as edge of tile T.
- Proportion of pixels detected as edge of tile A1.
- Proportion of pixels detected as edge of tile B1.

- EMD with the grayscale distribution of tile B3.

The decision tree DT3 is based in the next features.

- Position of the tile in the Y-axis.
- Position of the tile in the X-axis.
- Proportion of pixels detected as edge of tile T.
- Maximum gradient change over the tile T.
- Maximum gradient change at left of the tile T.
- Maximum gradient change at right of the tile T.
- Maximum gradient change under the tile T.
- Entropy of the grayscale distribution of tile T.
- Max gradient of the grayscale pixels of tile T.
- Entropy of the LBP distribution of tile T.
- Mean grayscale value with respect to the whole image.
- Mean LBP value with respect to the whole image.
- Minimum size of the Felzenswalb area.
- Gray intensity relative to neighbourhood.
- Proportion of pixels detected as edge of tile A1.
- Proportion of pixels detected as edge of tile A2.
- Proportion of pixels detected as edge of tile A3.
- Proportion of pixels detected as edge of tile A5.
- Proportion of pixels detected as edge of tile B1.
- EMD with the grayscale distribution of tile A5.
- EMD with the grayscale distribution of tile A4.
- EMD with the grayscale distribution of tile B1.
- EMD with the grayscale distribution of tile B3.
- EMD with the grayscale distribution of tile B4.
- EMD with the grayscale distribution of tile B5.

The decision tree DT4 is based in the next features.

- Position of the tile in the Y-axis.
- Position of the tile in the X-axis.
- Maximum gradient change over the tile T.
- Maximum gradient change at left of the tile T.
- Maximum gradient change at right of the tile T.
- Maximum gradient change under the tile T.
- Entropy of the grayscale distribution of tile T.
- Entropy of the LBP distribution of tile T.
- Proportion of pixels detected as edge of tile T.
- Max gradient of the grayscale pixels of tile T.
- Gray intensity relative to neighbourhood.
- Minimum size of the Felzenswalb area.
- Mean LBP value with respect to the whole image.
- Mean grayscale value with respect to the whole image.
- Proportion of pixels detected as edge of tile A3.
- Proportion of pixels detected as edge of tile A4.
- Proportion of pixels detected as edge of tile A9.
- Proportion of pixels detected as edge of tile B2.
- Proportion of pixels detected as edge of tile B3.
- Proportion of pixels detected as edge of tile B4.
- EMD with the grayscale distribution of tile B9.
- EMD with the grayscale distribution of tile B3.
- EMD with the LBP distribution of tile B3.

The choice of features for each of the decision trees was mainly based in the insights of the segmentation results obtained in the first experiment and in Tables A.1, A.2, A.3 and A.4. First, and in order to avoid undesired misclassifications due to early errors, the class of the surrounding already classified superpixels is eliminated from the features collection. Second, the most relevant features obtained when applying recursive feature elimination were taken (see Appendix A). As well, the number of features taken for each of the decision trees is dependant on their expected maximum depths (around 4 for the decision trees DT1 and DT2 and around 10 for the decision trees DT3 and DT4).

However, it is naive base the selection of features only in the recursive feature elimination outcome. Some features do not have a high importance in Tables A.1, A.2, A.3 and A.4, but are crucial for a good analysis of the image. This is because they are not useful features for classifying most of the superpixels, but they do make the difference for very concrete scenarios, and thus, they are actually useful. An example of this is the maximum gradient change under, at right and at left of the tile. They do not appear as an important features for the decision trees DT3 and DT4, but they are decisive for classifying as floor or background those superpixels that are next to the edge between the floor and the wall.

Note, as well, that if desired, a different selection of local features can be done in order to reduce the number of different decision trees (use less than four). However, in the current experiment it has been preferred to continue using the four different decision trees approach (Chapter 4.2).

5.3.3 Decision tree and feature importance

Similarly to the previous experiments and with an already defined collection of features for each of the four decision trees, an study of the dependence of the accuracy of the decision tree on their depth has to be made.

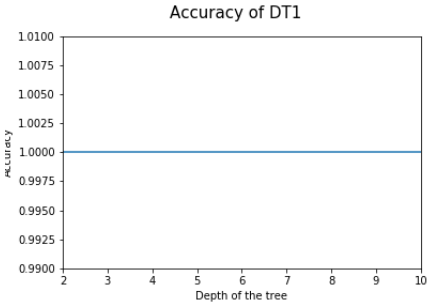


Figure 5.12: Accuracy - depth of DT1.

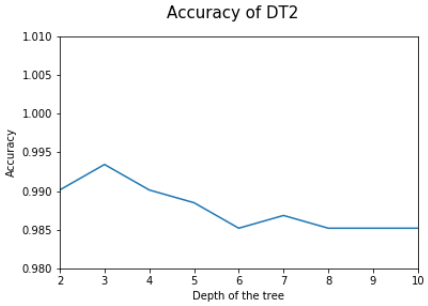


Figure 5.13: Accuracy - depth of DT2.

In Figures 5.12, 5.13, 5.14 and 5.15 it is shown the accuracy of each of the four decision trees, dependant on their maximum depth. Similarly to the first experiment, DT1 and DT2 have very high accuracy as, in almost all the cases, they only have to classify floor superpixels and in some cases edge superpixels. Thus, it can be set a

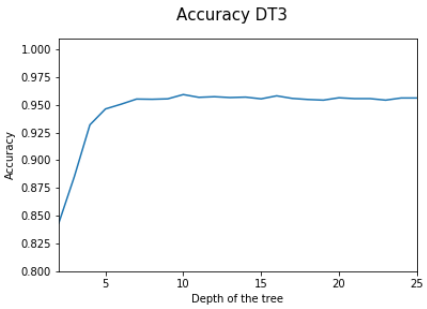


Figure 5.14: Accuracy - depth of DT3.

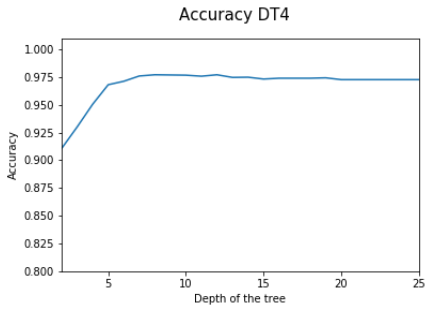


Figure 5.15: Accuracy - depth of DT4.

maximum depth of 3 for both, DT1 and DT2. As well, for DT3 and DT4 the best results are obtained when setting a maximum depth of 10.

Additionally, it is noticeable, again, the small dependence of the accuracy on the maximum depth of the decision trees. As said in the previous experiments this is because most of the superpixels can be classified using a very small set of features, and only those that are next to the boundary between the floor and the background suppose a more challenging classification.

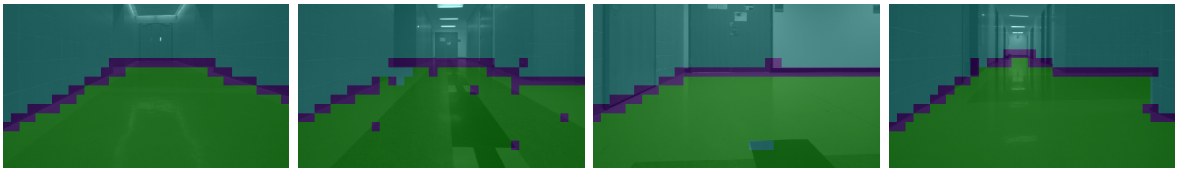
Likewise, in Table 5.3 it is shown the importances of each of the features for this segmentation. As expected, for DT3 and DT4 the most important feature is the position of the superpixel in the Y-axis, followed by the maximum gradient change over the tile.

Feature	**DT1**	**DT2**	**DT3**	**DT4**
Position of the tile in the Y-axis	-	-	0.555	0.685
Maximum gradient change over the tile T	-	-	0.160	0.070
Proportion of pixels detected as edge of tile T	0.041	0	0.078	0.029
Mean grayscale value with respect to the whole image	0.178	0.171	0.066	0.029
Max gradient of the grayscale pixels of tile T	0	0.812	0.049	0.028
Proportion of pixels detected as edge of tile B1	0.39	0	0.020	-
Proportion of pixels detected as edge of tile A1	-	0	0.013	-
Maximum gradient change at left of the tile T	0	0	0.012	0.022
Gray intensity relative to neighbourhood	0.324	-	0.004	0.007
Position of the tile in the X-axis	0	0	0.004	0
Entropy of the grayscale distribution of tile T	0	0	0.004	0.053
EMD with the grayscale distribution of tile B5	-	-	0.003	-
Proportion of pixels detected as edge of tile A2	-	-	0.003	-
EMD with the grayscale distribution of tile B1	-	-	0.003	-
EMD with the grayscale distribution of tile A4	-	-	0.003	-
Proportion of pixels detected as edge of tile A5	-	-	0.003	-
EMD with the grayscale distribution of tile A5	-	-	0.003	-
Mean LBP value with respect to the whole image	-	-	0.003	0.001
Entropy of the LBP distribution of tile T	0	0	0.003	0.001
Maximum gradient change under the tile T	-	-	0.002	0.004
EMD with the grayscale distribution of tile B4	-	-	0.002	-
Minimum size of the Felzenswalb area	-	-	0.002	0.003
Proportion of pixels detected as edge of tile A3	-	-	0.002	0.012
EMD with the grayscale distribution of tile B3	0.053	0.017	0.002	0.003
Maximum gradient change at right of the tile T	0	0	0.002	0.006
Proportion of pixels detected as edge of tile B3	-	-	-	0.014
EMD with the grayscale distribution of tile B9	-	-	-	0.006
Proportion of pixels detected as edge of tile A4	-	-	-	0.008
Proportion of pixels detected as edge of tile B2	-	-	-	0.007
Proportion of pixels detected as edge of tile A9	-	-	-	0.004
EMD with the LBP distribution of tile B3	-	-	-	0.004
Proportion of pixels detected as edge of tile B4	-	-	-	0.002
EMD with the LBP distribution of tile B1	0.014	-	-	-

Table 5.3: Importance of the selection of features.

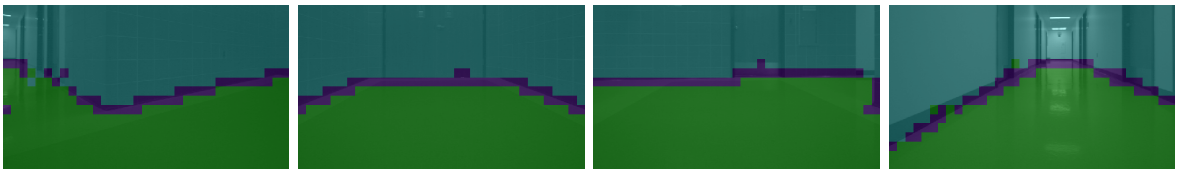
5.3.4 Result of the segmentation

In Figures 5.16 and 5.17 it is shown the results of the segmentation obtained when making a finer selection of knowledge.



(a) (b) (c) (d)

Figure 5.16: Classification made using only global features.



(a) (b) (c) (d)

Figure 5.17: Classification made using only global features.

Indeed, in the above images it can be seen the great increase in performance of the algorithm when doing a wiser feature selection. In images 5.16a, 5.16c, 5.16d, 5.17b, 5.17c and 5.17d a close to perfect segmentation is done. Additionally, the segmentation of two most challenging images, 5.16b and 5.17a, is improved with respect to the two previous tests. Indeed, the reason of the errors found in the segmentations of these two images are due to a very tiled floor or to strong reflections. This supposes a big gradient change in some of the floor tiles, and thus, they are classified as edge.

Furthermore, a finer segmentation could be done by performing some post-processing on the already segmented images. For example, some of the superpixels incorrectly classified as edge in image 5.16b could be easily discarded by inferring that they are not connected to any other edge superpixels and they are surrounded by floor superpixels. Note that this post-processing was neither developed nor applied to the above images.

In addition, besides it was obtained a slightly better classification for some images when using the class of the neighbour superpixels as features, a much more robust model is obtained when not using these parameters.

As well, that the accuracy of the prediction of the class of the superpixels of the eight test images is 0.981. This is, indeed, another proof of the better performance of the current selection of features.

Chapter 6

Conclusion

The main goal of this thesis is presenting a knowledge-based floor segmentation on images. The followed approach consists in the division of the image in squared superpixels and the subsequently analysis of a set of human-defined features based in the behavior of the floor in images, collected for each of these superpixels. In this way, this work is halfway between a pure black box segmentation model in which the information is completely learned and models in which the information for performing the segmentation is completely human defined. Three tests were performed on a set of 203 corridor images taken with a camera fixed in a movable platform. The first two consist in a study of the segmentation boundaries obtained when analysing different sets of features and the third one consists in a wiser feature selection based in the previous tests. Additionally, and because the explainability of the decision making process is one of the cornerstones of this thesis, the classification process has been based in a decision tree model.

Indeed, the obtained results with the superpixel approach are quite interesting and visible from a knowledge point of view. Dividing the set of features in superpixel's own knowledge, local knowledge and global knowledge have brought some useful insights of how each of these attributes contributes in the segmentation. From the local parameters, it has been probed that it is obtained a segmentation based in the fact that similar nearby regions in the image belong to the same class. In addition, because of the fixed squared shape of the superpixels, a set of relationships has to be fulfilled between superpixels that are in the surroundings of an edge. However, these relationships are not only present in the surroundings of the edge between the floor and the background. Consequently, it has been tested how this undesired effect can be avoided to some degree by introducing relationships that take into account the whole image (global knowledge).

As well, from the accuracy dependent on the depth of the decision tree figures, it has been probed that the classification of most of the superpixels do not pose a great challenge. Indeed, only a few features are needed to obtain a relatively good result with respect to the accuracy of the classification. This is because the floor is characterized as a big homogeneous surface, which makes it quite simple to be identified on the basis of some prior knowledge of its location. Only those superpixels

that are close to the edge between the floor and the walls pose a more challenging classification and need a deeper analysis, involving more features. Somehow, this is what humans do. We do not analyse every single detail of what we are seeing. For example, when we analyze the floor it is not important to infer in details such as if it is tiled, its color or if it has glares or irregularities. Instead, we are good at recognizing that a homogeneous surface belongs to the same object, without analysing it meticulously. Thus, it is only needed to make a deeper analysis of the boundary of that object, i.e., the boundary between the floor and the background. Besides this may be a straightforward outcome, the properties of this boundary must be defined in such a way that they are not mismatched with inner properties of this homogeneous surface that is being characterized. For example, in the posed problem a big gradient change in the grayscale image not only characterizes the edge between the floor and the background, but also the tiled texture of the floor. In an attempt to avoid this type of misclassification it has been added a set of global properties into the analysis. Furthermore, it has been of tremendous help modeling the transition between the floor and the background as a different class, the edge class. By segmenting the image in three different classes, floor, edge and background, it has been added a new level of abstraction for analysing those more challenging parts of the image, the edges, and thus, easing the classification of the most homogeneous parts of the image as floor or background.

Additionally, the goal of explainability has been successfully fulfilled. Most of the segmented images in the third test resulted in a reasonably good segmentation. As well, the reason of the misclassification of some of the superpixels in this last test can be easily explained as, because of the high gradient of the tiled texture of the floor, some parts of the floor with high gradient are classified as edge. However, as said before, this misclassification has been mitigated to some degree with the combination of global and local features.

Future work should include a post-processing of the obtained superpixel segmentation. First, some of the misclassified superpixels can be detected by following a simple analysis of the classification in which the outliers are reclassified. Additionally, the superpixel segmentation can be of great help for fitting a line in the edge between the floor and the background in the original image, which, subsequently, would result in a further elimination of misclassified superpixels.

Overall, in the current work it has been collected and analysed a set of features for segmenting the floor in images. Furthermore, it has been proposed an algorithm to divide the image in superpixels and reason over them. Additionally, some future lines to improve the segmentation have been established.

Appendices

Appendix A

Recursive Feature Elimination - Feature importances

In this appendix it is shown the order of importance of all the features that were introduced in Chapter 3, but the class of the neighbour superpixels. This order of importance is obtained when applying recursive feature elimination to the decision trees models and it is ordered from most important (1) to less important (50) for each of the four decision trees (DT1, DT2, DT3 and DT4) shown in Chapter 4.2.

Each of the four tables correspond with a decision tree, DT1, DT2, DT3 or DT4. Additionally, for the tables of the decision trees DT1 and DT2, each of the five columns corresponds with a maximum depth of the decision tree, that ranges between 2 to 6. Similarly, for the tables of the decision trees DT3 and DT4, each of the five columns corresponds with a maximum depth of the decision tree, that ranges between 8 to 12.

Feature	**d2**	**d3**	**d4**	**d5**	**d6**
Proportion of pixels detected as edge of tile B1	1	2	2	1	2
Gray intensity relative to neighbourhood	2	1	1	2	1
Mean grayscale value with respect to the whole image	3	3	3	3	3
EMD with the grayscale distribution of tile B5	4	17	17	17	9
Proportion of pixels detected as edge of tile B3	5	6	7	8	11
EMD with the LBP distribution of tile B3	6	8	8	9	18
EMD with the grayscale distribution of tile B3	7	5	4	4	17
Proportion of pixels detected as edge of tile B2	8	10	9	10	10
EMD with the LBP distribution of tile B2	9	11	12	12	13
EMD with the grayscale distribution of tile B2	10	7	10	11	7
Maximum gradient change at right of the tile T	11	9	11	14	12
Maximum gradient change under the tile T	12	14	13	15	14
Maximum gradient change over the tile T	13	13	14	13	25
Minimum size of the Felzenswalb area	14	15	15	16	15
Max gradient of the grayscale pixels of tile T	15	16	5	6	4
Mean LBP value with respect to the whole image	16	18	16	18	16
EMD with the LBP distribution of tile B5	17	19	19	19	20
Maximum gradient change at left of the tile T	18	12	18	5	5
Proportion of pixels detected as edge of tile B5	19	21	21	21	21
EMD with the grayscale distribution of tile A3	20	20	20	20	26
EMD with the grayscale distribution of tile B9	21	23	23	23	22
EMD with the LBP distribution of tile A3	22	22	22	22	28
EMD with the LBP distribution of tile B9	23	25	25	25	23
Proportion of pixels detected as edge of tile A3	24	24	24	24	30
Proportion of pixels detected as edge of tile B9	25	29	29	32	29
EMD with the grayscale distribution of tile A4	26	26	26	26	32
EMD with the LBP distribution of tile B1	27	4	6	7	8
EMD with the LBP distribution of tile A4	28	28	28	28	34
EMD with the grayscale distribution of tile B1	29	27	27	27	27
Proportion of pixels detected as edge of tile A4	30	30	30	30	36
Proportion of pixels detected as edge of tile A9	31	31	31	29	31
EMD with the grayscale distribution of tile A9	32	32	32	31	35
EMD with the LBP distribution of tile A9	33	33	33	33	33
Entropy of the LBP distribution of tile T	34	34	34	34	24
Entropy of the grayscale distribution of tile T	35	35	35	35	19
Proportion of pixels detected as edge of tile T	36	36	36	36	6
Position of the tile in the Y-axis	37	37	37	37	37
Position of the tile in the X-axis	38	38	38	38	38

Table A.1: RFE of the decision tree DT1 with a maximum depth between 2 and 6.

Feature	d2	d3	d4	d5	d6
Max gradient of the grayscale pixels of tile T	1	1	1	1	1
Mean grayscale value with respect to the whole image	2	2	2	3	3
EMD with the LBP distribution of tile B9	3	19	31	35	35
EMD with the grayscale distribution of tile B9	4	3	24	33	34
Proportion of pixels detected as edge of tile B5	5	4	23	31	33
EMD with the LBP distribution of tile B5	6	5	21	29	31
EMD with the grayscale distribution of tile B5	7	6	19	27	29
Proportion of pixels detected as edge of tile B4	8	7	18	25	27
EMD with the LBP distribution of tile B4	9	8	17	24	26
EMD with the grayscale distribution of tile B4	10	9	5	23	25
Maximum gradient change at right of the tile T	11	11	11	12	24
Maximum gradient change under the tile T	12	12	12	13	22
Maximum gradient change over the tile T	13	13	13	14	28
Gray intensity relative to neighbourhood	14	14	14	15	19
Minimum size of the Felzenswalb area	15	15	15	16	18
Mean LBP value with respect to the whole image	16	16	16	17	13
Proportion of pixels detected as edge of tile B3	17	10	6	21	23
Maximum gradient change at left of the tile T	18	18	3	2	2
EMD with the LBP distribution of tile B3	19	17	7	19	21
EMD with the grayscale distribution of tile A1	20	20	20	20	30
EMD with the grayscale distribution of tile B3	21	21	8	7	20
EMD with the LBP distribution of tile A1	22	22	22	22	32
Proportion of pixels detected as edge of tile B2	23	23	9	8	17
Proportion of pixels detected as edge of tile A1	24	24	4	6	6
EMD with the LBP distribution of tile B2	25	25	10	18	8
EMD with the grayscale distribution of tile A2	26	26	26	26	36
EMD with the grayscale distribution of tile B2	27	27	25	4	7
EMD with the LBP distribution of tile A2	28	28	28	28	38
Proportion of pixels detected as edge of tile B1	29	29	27	9	9
Proportion of pixels detected as edge of tile A2	30	30	30	30	40
EMD with the LBP distribution of tile B1	31	33	33	11	11
EMD with the grayscale distribution of tile A3	32	31	29	32	41
EMD with the LBP distribution of tile A3	33	34	34	34	44
Proportion of pixels detected as edge of tile B9	34	32	32	37	42
EMD with the grayscale distribution of tile B1	35	35	35	10	10
Proportion of pixels detected as edge of tile A3	36	36	36	36	46
Proportion of pixels detected as edge of tile A9	37	37	37	5	5
EMD with the grayscale distribution of tile A4	38	38	38	38	47
EMD with the LBP distribution of tile A9	39	39	39	39	37
EMD with the LBP distribution of tile A4	40	40	40	40	45
EMD with the grayscale distribution of tile A9	41	41	41	41	39
Proportion of pixels detected as edge of tile A4	42	42	42	42	43
Entropy of the LBP distribution of tile T	43	43	43	43	14
Entropy of the grayscale distribution of tile T	44	44	44	44	12
Proportion of pixels detected as edge of tile T	45	45	45	45	15
Position of the tile in the Y-axis	46	46	46	46	16
Position of the tile in the X-axis	47	47	47	47	4

Table A.2: RFE of the decision tree DT2 with a maximum depth between 2 and 6.

Feature	**d8**	**d9**	**d10**	**d11**	**d12**
Position of the tile in the Y-axis	1	1	1	1	1
Maximum gradient change over the tile T	2	2	2	2	2
Proportion of pixels detected as edge of tile T	3	3	3	3	3
Mean grayscale value with respect to the whole image	4	4	4	4	4
Max gradient of the grayscale pixels of tile T	5	5	5	5	5
Proportion of pixels detected as edge of tile B1	6	6	6	6	6
Proportion of pixels detected as edge of tile A1	7	7	7	7	7
Maximum gradient change at left of the tile T	8	8	8	8	8
Gray intensity relative to neighbourhood	9	9	9	9	9
EMD with the grayscale distribution of tile B1	10	10	14	16	11
Entropy of the LBP distribution of tile T	11	11	15	15	21
EMD with the grayscale distribution of tile A4	12	12	16	12	23
Proportion of pixels detected as edge of tile A5	13	19	17	18	19
EMD with the grayscale distribution of tile B4	14	16	18	19	17
Proportion of pixels detected as edge of tile B5	15	24	24	25	25
Mean LBP value with respect to the whole image	16	18	22	22	14
Proportion of pixels detected as edge of tile A2	17	13	12	13	12
Position of the tile in the X-axis	18	15	13	14	13
Entropy of the grayscale distribution of tile T	19	14	10	10	10
EMD with the grayscale distribution of tile A5	20	22	20	20	18
Minimum size of the Felzenswalb area	21	17	19	17	22
EMD with the grayscale distribution of tile B3	22	23	21	23	20
Proportion of pixels detected as edge of tile A3	23	21	23	21	16
Proportion of pixels detected as edge of tile B2	24	25	27	27	26
Maximum gradient change under the tile T	25	26	28	28	27
EMD with the grayscale distribution of tile A2	26	27	25	24	24
EMD with the grayscale distribution of tile B5	27	20	11	11	15
Proportion of pixels detected as edge of tile B9	28	28	26	26	29
EMD with the LBP distribution of tile B5	29	29	29	31	31
Proportion of pixels detected as edge of tile A4	30	31	37	33	36
EMD with the grayscale distribution of tile A3	31	30	31	30	28
EMD with the LBP distribution of tile A4	32	35	35	32	32
EMD with the grayscale distribution of tile A1	33	37	36	34	34
EMD with the LBP distribution of tile A2	34	34	32	37	37
EMD with the grayscale distribution of tile A9	35	32	40	41	41
Proportion of pixels detected as edge of tile B4	36	36	33	39	39
Proportion of pixels detected as edge of tile B3	37	39	34	36	35
Maximum gradient change at right of the tile T	38	33	30	29	30
EMD with the grayscale distribution of tile B9	39	49	50	47	47
EMD with the LBP distribution of tile B2	40	48	47	48	42
EMD with the LBP distribution of tile B3	41	44	42	46	49
EMD with the LBP distribution of tile B4	42	50	43	44	43
EMD with the LBP distribution of tile B1	43	41	48	49	48
EMD with the LBP distribution of tile A1	44	42	45	43	45
EMD with the LBP distribution of tile B9	45	43	46	42	44
EMD with the LBP distribution of tile A5	46	46	49	50	50
EMD with the LBP distribution of tile A9	47	38	39	40	40
Proportion of pixels detected as edge of tile A9	48	45	41	38	38
EMD with the grayscale distribution of tile B2	49	40	38	35	33
EMD with the LBP distribution of tile A3	50	47	44	45	46

Table A.3: RFE of the decision tree DT3 with a maximum depth between 8 and 12.

Feature	**d8**	**d9**	**d10**	**d11**	**d12**
Position of the tile in the Y-axis	1	1	1	1	1
Maximum gradient change over the tile T	2	2	2	2	2
Entropy of the grayscale distribution of tile T	3	3	3	3	3
Max gradient of the grayscale pixels of tile T	4	4	4	4	4
Mean grayscale value with respect to the whole image	5	5	5	5	5
Maximum gradient change at left of the tile T	6	6	6	6	6
Proportion of pixels detected as edge of tile T	7	7	7	7	7
Proportion of pixels detected as edge of tile B3	8	9	9	9	9
Proportion of pixels detected as edge of tile A3	9	8	8	8	8
EMD with the grayscale distribution of tile B9	10	11	11	11	11
Proportion of pixels detected as edge of tile A4	11	12	12	12	12
Gray intensity relative to neighbourhood	12	10	10	10	10
Proportion of pixels detected as edge of tile B2	13	13	14	15	15
Maximum gradient change at right of the tile T	14	14	13	13	13
Proportion of pixels detected as edge of tile A9	15	15	15	17	16
Maximum gradient change under the tile T	16	16	16	14	14
EMD with the LBP distribution of tile B3	17	17	18	16	18
Minimum size of the Felzenswalb area	18	18	17	18	17
Proportion of pixels detected as edge of tile B4	19	19	20	20	20
EMD with the grayscale distribution of tile A9	20	22	22	23	23
EMD with the grayscale distribution of tile B3	21	21	19	19	19
Entropy of the LBP distribution of tile T	22	23	24	24	24
Mean LBP value with respect to the whole image	23	27	27	25	27
EMD with the grayscale distribution of tile B5	24	25	29	28	30
Proportion of pixels detected as edge of tile B9	25	26	30	29	31
EMD with the LBP distribution of tile B9	26	32	38	32	38
EMD with the grayscale distribution of tile A4	27	29	33	30	32
EMD with the grayscale distribution of tile A3	28	30	23	22	22
EMD with the grayscale distribution of tile B1	29	31	25	27	25
EMD with the grayscale distribution of tile A5	30	34	35	36	40
Proportion of pixels detected as edge of tile A1	31	20	21	21	21
Proportion of pixels detected as edge of tile B1	32	37	39	40	37
EMD with the LBP distribution of tile A3	33	24	26	26	26
Proportion of pixels detected as edge of tile B5	34	35	40	38	35
EMD with the grayscale distribution of tile B2	35	36	37	39	36
Proportion of pixels detected as edge of tile A2	36	41	44	44	44
EMD with the grayscale distribution of tile A1	37	28	32	35	28
EMD with the LBP distribution of tile A9	38	48	46	45	49
EMD with the LBP distribution of tile A1	39	39	28	31	33
EMD with the LBP distribution of tile B1	40	33	31	33	29
EMD with the grayscale distribution of tile B4	41	46	47	48	50
EMD with the LBP distribution of tile B5	42	45	42	42	43
EMD with the LBP distribution of tile B2	43	49	36	34	39
EMD with the LBP distribution of tile A5	44	44	48	49	47
EMD with the grayscale distribution of tile A2	45	40	43	50	45
Proportion of pixels detected as edge of tile A5	46	47	49	46	46
EMD with the LBP distribution of tile A2	47	38	41	43	42
EMD with the LBP distribution of tile A4	48	42	34	37	34
EMD with the LBP distribution of tile B4	49	43	45	47	48
Position of the tile in the X-axis	50	50	50	41	41

Table A.4: RFE of the decision tree DT4 with a maximum depth between 8 and 12.

Appendix B

Suggestion for further studies

In this appendix it is suggested some possible lines for future studies.

B.1 Posterior analysis of the segmentation

As mentioned before, besides the good performance of the segmentation obtained in the third experiment (Chapter 5.3), it is, indeed, needed a posterior analysis of the results.

First, it is necessary an analysis to detect superpixels that are misclassified as edge superpixels. For example, in image 5.16b, it can be reasoned that some superpixels do not belong to the edge classification as they are not surrounded by any other edge superpixels. As well, they can be detected as misclassified because there are more reasonable edge superpixel classifications over them. Thus, it should be collected more image segmentations and make a subsequent reasoning on the misclassifications. In addition, it could be done an analysis of the probability of a superpixel being assigned a wrong class, given the classification of the surrounding superpixels.

Secondly, if needed a finer segmentation and in order to obtain better results, the superpixel classification could be used as a prior to fit the best line in the edge between the floor and the background. The combination of the original image with the superpixel segmentation can be of great help when facing the problem of fitting the best line between the floor and the background. Besides there are some misclassifications, most of the edge superpixels are correctly classified. By combining it with the image, it can easily be fitted the best line that complies with the scenarios shown in Figures 3.3, 3.4 and 3.5. Furthermore, by fitting this line, misclassifications done in some superpixels would be mitigated, as this best fit would follow the trend of the majority of the edge superpixels.

B.2 Time information

The next natural step would be adding time information into the analysis. Previously segmented images have most of the information needed for a new segmentation. Thus, it should be added two new features, the assigned class to the superpixel

in the previous frame, and its distance (measured in superpixels) to the edge. It is most probable that a new superpixel has assigned the same class than in the previous frame, and only those superpixels that are next to the edge may need a new classification. Furthermore, by introducing information such as the speed of the AV it can be calculated the position of the edge in the new frame.

However, misclassifications can led to errors as the ones experienced in the first experiment (Chapter 5.1). This is because an early misclassified superpixel can drag its error among frames.

B.3 Superpixels taken from other prior image segmentation (not squared superpixels)

In this work it has been decided to use squared superpixels because they bring some fixed relationships between neighbours that are present in the straight edges. This is incredibly useful for detecting edges when comparing the features of the actual tile with its neighbours.

Thereupon, this property is lost if introduced other prior segmentation that does not form a regular grid. However, it could be tested the same algorithm on a superpixel grid with a regular seed grid (for example, SLIC [1]).

B.4 Other changes in the algorithm and in the features collection

Next, it is written a list of variants of the algorithm and of the feature collection. Note that these variants consist in minor ideas I had while doing the tests and writing this report, or suggestions posed during the thesis meetings.

- Overlap superpixels: One of the problems found when characterizing the edge superpixels is that, sometimes and because of the squared shape of the superpixels, the edge between the floor and the background falls in between two superpixels and, thus, there is a direct transition between the floor and the background. If it is desired to fit a line in this edge using the classification of the superpixels as a prior, this may lead to errors. To avoid it, an overlapping of superpixels could be done.

- An obvious change that can be done to the algorithm is trying other "black box" techniques, such as reduction of features using PCA, or other classification methods, such as support vector machine or random forest. However, this mean a reduction in the explainability of the results. As well, a more Bayesian approach in which each of the superpixels is represented as a node and is connected to its neighbours could be done.

- As explained before, the classification of most of the superpixels of the floor and of the background suppose little challenge. Thereby, it may be interesting

to train a decision tree to detect only the edge superpixels and not train the other two classes, floor and background. In this way it could be obtained a model that prioritizes the detection of good edge superpixels, and not the detection of floor and background.

- A great improvement could be achieved if prior information of the layout of the building is introduced in the analysis. For example, if the area of the building in which the AV is located can be identified, information as the color of the floor or the tiling pattern could be introduced in the decision making process.
- The size of the superpixel could be adapted to the homography of the image. Thereby, the closer is the superpixel to the AV, the smaller it is.
- Another straightforward change consist in introducing more complex features, created from the combination of other features (second level of features) or experiment with different image representations (HSV, RGB,...).

Bibliography

[1] R. Achanta, A. Shaji, K. Smith, A. Lucchi, P. Fua, and S. Süsstrunk. Slic superpixels, 2010.

[2] T. Ahonen, A. Hadid, and M. Pietikainen. Face description with local binary patterns: Application to face recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 28(12):2037–2041, Dec 2006.

[3] L. A. Alexandre, J. S. Sánchez, and J. M. F. Rodrigues. *Pattern Recognition and Image Analysis*. Springer, 2017.

[4] Ane Berasategi. Earth mover's distance. URL: `https://towardsdatascience.com/earth-movers-distance-68fff0363ef2`, last checked on 2019-21-07, 2018.

[5] J. Canny. A computational approach to edge detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PAMI-8(6):679–698, Nov 1986.

[6] G. Casas Barcelo, G. Panahandeh, and M. Jansson. Image-based floor segmentation in visual inertial navigation. pages 1402–1407, 05 2013.

[7] T. Do, M. Duong, Q. Dang, and M. Le. Real-time self-driving car navigation using deep neural network. In *2018 4th International Conference on Green Technology and Sustainable Development (GTSD)*, pages 7–12, Nov 2018.

[8] Dong-chen He and Li Wang. Texture unit, texture spectrum, and texture analysis. *IEEE Transactions on Geoscience and Remote Sensing*, 28(4):509–512, July 1990.

[9] P. F. Felzenszwalb and D. P. Huttenlocher. Efficient graph-based image segmentation. Journal Volume 59 Issue 2, International Journal of Computer Vision, 2004.

[10] I. Guyon, J. Weston, S. Barnhill, and V. Vapnik. Gene selection for cancer classification using support vector machines. *Machine Learning*, 46(1):389–422, Jan 2002.

[11] C. . Hung, M. Pham, S. Arasteh, B. . Kuo, and T. Coleman. Image texture classification using texture spectrum and local binary pattern. In *2006 IEEE*

International Symposium on Geoscience and Remote Sensing, pages 2750–2753, July 2006.

[12] E. Javanmardi, Y. Gu, M. Javanmardi, and S. Kamijo. Autonomous vehicle self-localization based on abstract map and multi-channel lidar in urban area. *IATSS Research*, 43(1):1 – 13, 2019.

[13] D. C. Lee, M. Hebert, and T. Kanade. Geometric reasoning for single image structure recovery. In *2009 IEEE Conference on Computer Vision and Pattern Recognition*, pages 2136–2143, June 2009.

[14] J. Lijian, O. Tawose, P. , C. , J. Wang, and Z. . Weakly-supervised image semantic segmentation based on superpixel region merging. *Big Data and Cognitive Computing*, 3:31, 06 2019.

[15] L. M. Lorigo, R. A. Brooks, and W. E. L. Grimsou. Visually-guided obstacle avoidance in unstructured environments. In *Proceedings of the 1997 IEEE/RSJ International Conference on Intelligent Robot and Systems. Innovative Robotics for Real-World Applications. IROS '97*, volume 1, pages 373–379 vol.1, Sep. 1997.

[16] O. H. Maghsoudi. Superpixel based segmentation and classification of polyps in wireless capsule endoscopy. *2017 IEEE Signal Processing in Medicine and Biology Symposium (SPMB)*, Dec 2017.

[17] T. Mariya John. Humanoid robot moving in path and obstacle avoidance. *International Journal of Trend in Scientific Research and Development*, Volume-3:572–573, 04 2019.

[18] K. Ni, X. Bresson, T. Chan, and S. Esedoglu. Local histogram based segmentation using the wasserstein distance. *International Journal of Computer Vision*, 84(1):97–111, Aug 2009.

[19] T. Ojala, M. Pietikainen, and D. Harwood. Performance evaluation of texture measures with classification based on kullback discrimination of distributions, Oct 1994.

[20] T. Ojala, M. Pietikainen, and T. Maenpaa. Multiresolution gray-scale and rotation invariant texture classification with local binary patterns. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24(7):971–987, July 2002.

[21] OpenCV contributors. Canny edge detector — opencv. URL: `https://docs.opencv.org/trunk/dd/d1a/group__imgproc__feature.html#ga04723e007ed888ddf11d9ba04e2232de`, last checked on 2019-20-07, 2019.

[22] OpenCV contributors. Sobel — opencv. URL: https://docs.opencv.org/3.4/d4/d86/group__imgproc__filter.html#gacea54f142e81b6758cb6f375ce782c8d, last checked on 2019-20-07, 2019.

[23] Y. Peng, D. C. Qu, Y. Zhong, S. Xie, J. Luo, and J. J. Gu. The obstacle detection and obstacle avoidance algorithm based on 2-d lidar. *2015 IEEE International Conference on Information and Automation*, pages 1648–1653, 2015.

[24] T. Pilarski, M. Happold, H. Pangels, M. Ollis, K. Fitzpatrick, and A. Stentz. The demeter system for automated harvesting. *Auton. Robots*, 13:9–20, 07 2002.

[25] Rajesh S. Brid. Decision trees. URL: https://medium.com/greyatom/decision-trees-a-simple-way-to-visualize-a-decision-dc506a403aeb, last checked on 2019-24-07, 2018.

[26] X. Ren and J. Malik. Learning a classification model for segmentation. In *Proceedings of the Ninth IEEE International Conference on Computer Vision - Volume 2*, ICCV '03, pages 10–, Washington, DC, USA, 2003. IEEE Computer Society.

[27] K. Sabe, M. Fukuchi, S. Gutmann, T. Ohashi, K. Kawamoto, and T. Yoshigahara. Obstacle avoidance and path planning for humanoid robots using stereo vision. volume 1, pages 592 – 597 Vol.1, 01 2004.

[28] Scikit-image contributors. Felzenswalb segmentation — scikit-image. URL: https://scikit-image.org/docs/dev/api/skimage.segmentation.html#skimage.segmentation.felzenszwalb, last checked on 2019-20-07, 2019.

[29] Scikit-image contributors. Local binary patterns — scikit-image. URL: https://scikit-image.org/docs/dev/api/skimage.feature.html#skimage.feature.local_binary_patter, last checked on 2019-06-07, 2019.

[30] Scikit-learn contributors. Decision tree — scikit-learn. URL: https://scikit-learn.org/stable/modules/generated/sklearn.tree.DecisionTreeClassifier.html#sklearn.tree.DecisionTreeClassifier, last checked on 2019-08-07, 2019.

[31] Scikit-learn contributors. Recursive feature elimination — scikit-learn. URL: https://scikit-learn.org/stable/modules/generated/sklearn.feature_selection.RFE.html, last checked on 2019-08-07, 2019.

[32] Scipy stats contributors. Wasserstein distance — scipy stats. URL: https://docs.scipy.org/doc/scipy/reference/generated/scipy.stats.wasserstein_distance.html, last checked on 2019-20-07, 2019.

[33] I. Sobel and G. Feldman. A 3x3 isotropic gradient operator for image processing. *Pattern Classification and Scene Analysis*, pages 271–272, 01 1973.

[34] G. Tsai and B. Kuipers. Michigan indoor corridor dataset. URL: `https://deepblue.lib.umich.edu/data/concern/data_sets/3t945q88k?locale=en`, last checked on 2019-06-07.

[35] G. Tsai and B. Kuipers. Dynamic visual understanding of the local environment for an indoor navigating robot. In *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 4695–4701, Oct 2012.

[36] S. Viscelli. Driverless? autonomous trucks and the future of the american trucker. 2018.

[37] T. Wang and D. E. Chang. Improved reinforcement learning through imitation learning pretraining towards image-based autonomous driving. 2019.

[38] Yinxiao Li and S. T. Birchfield. Image-based segmentation of indoor corridor floors for a mobile robot. In *2010 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 837–843, Oct 2010.

[39] J. Zhou and B. Li. Robust ground plane detection with normalized homography in monocular sequences from a robot platform. In *2006 International Conference on Image Processing*, pages 3017–3020, Oct 2006.

Master's thesis filing card

Student: Ignacio Garrido Botella

Title: Knowledge-driven image segmentation

UDC: 621.3

Abstract:
The aim of this thesis is to infer the knowledge needed by an autonomous system for performing the image segmentation of a video footage that is taken inside a building. Precisely, it is intended to do a segmentation of the floor from the background. The followed approach consists in the segmentation of images in a fixed number of squared superpixels, followed by a posterior analysis of the features collected for each of those superpixels. The features analysed for each of the superpixels are divided into features extracted directly from the superpixel, local knowledge in the sense of relationships of the superpixel with its close neighbourhood of superpixels and global knowledge in the sense of features taken from the image as a whole. In this way, the image is summarized in a database of the features, from which a data analysis can be made. In addition, the posterior analysis of the collected features consists in a decision tree model.

Thesis submitted for the degree of Master of Science in Artificial Intelligence, option Engineering and Computer Science

Thesis supervisor: Prof. dr. ir. Herman Bruyninckx

Assessor: Ir. Filip Reniers

Mentor: