UNIVERSIDAD NACIONAL DE COLOMBIA

Semantic Image Segmentation Using Learning Models

Paola Katherine Rozo Bernal

Universidad Nacional de Colombia
Facultad de Ingeniería, Departamento de Ingeniería de Sistemas e Industrial
Bogotá D.C., Colombia
2015

Semantic Image Segmentation Using Learning Models

Paola Katherine Rozo Bernal

In fulfillment of the requirements for the degree of:
Magister in Computer Science

Advisor:
Fabio Augusto González Osorio, Ph.D.

Research Field:
Machine Learning - Image Understanding
Research Group:
MindLab

Universidad Nacional de Colombia
Facultad de Ingeniería, Departamento de Ingeniería de Sistemas e Industrial
Bogotá D.C., Colombia
2015

Dedicatoria
En éste punto

Acknowledgements

Under Construction.

Abstract

This thesis presents a method for semantic image segmentation based on a structured output prediction strategy.

The method is applied to two different problems, road object detection and human gait analysis. In the first problem, the strategy can be summarized as follows: At first stage the image is oversegmeted to achieve a manageable number of entities to be classified, then these segments are characterized using the bag of feature representation. At a second stage, the image is modeled as a graph where nodes correspond to segments and arcs to neighborhood relationships, a Markov Random Field (MRF) model is trained using a set of labeled images, finally the segments are labeled by minimizing the MRF energy finding the most probable labels for each segment. The proposed strategy was tested on CamVid Dataset, and showed a good performance.

Regard to the human gait analysis, that is an important research area in several applications such as video surveillance, image retrieval systems, human interaction, and medical diagnostics, the problem emerged in this type of analysis is the segmentation of the human body in different parts of the body including head, limbs, torso and feet. This problem is challenging due to the occlusion of body parts (on sagittal gait), variability on the appearance on the images such as clothes with similar colors and finally the image perspective. In recent decades, the research works have been carried out in the segmentation of human body in different parts of videos and pictures directly but the perspective and appearance problems are still remain.

The strategy in this case can be summarized as follows: At first stage the division of the scene in small segments called superpixels using a oversegmentation technique is performed, and, at second, these superpixels are labelized defining a multi-class Support Vector Machine (SVM) and then finding a labeling that maximizes the probability given a set of classes. Superpixels are characterized using depth invariant features. The proposed strategy was tested on the Depth Gait Dataset, and showed a competitive performance.

Keywords: Structured output prediction, image segmentation, markov random fields, support vector machines, human gait analysis, kinect.

Resumen

Ésta tésis presenta un método de segmentación semantica de imágenes basado en una estrategia de predccion estructurada.

El método es aplicado a dos problemas diferntes, detección de bjetos en carretera y análisis de la marcha humana. En el primer problema, la estrategia puede ser resumidad de la siguiente forma: En la primera etapa, la imagen es sobresegmentada para obtener un número manejable de entidades a ser clasificadas, luego éstos segmentos son caracterizados usando la representación de bolsa de características (BoW). En la segunda estapa, la imagen es modelada como un grado, donde los nodos corresponde a superpixeles y los arcos a las relaciones entre éstos, un model de Campo Aleatorio de Markov (CAM) es entrenado usando un conjunto de imágenes etiquetadas. Finalmente los segmentos son clasificados a través de la minimización de la energía de CAM, encontrando las etiquetas más probables para cada segmento. La estrategia propuesta fue probada de el dataset CamVid, mostrando un buen rendimiento.

Con respecto al análisis de movimiento humano, que es un área de investigación importante en varias aplicaciones tales como video-vigilancia, sistemas de recuperación de imágenes, aplicaciones basadas en interacción humana y el diagnóstico médico, el problema que surge en este tipo de análisis es la segmentación del cuerpo humano en diferentes partes que lo componen, incluyendo la cabeza, extremidades, torso y pies. Este problema es un reto debido a la oclusión de partes del cuerpo (en marcha sagital), la variabilidad en la apariencia en las imágenes, tales como ropa con colores similares y, finalmente, la perspectiva de la imagen. En las últimas décadas, los trabajos de investigación se han llevado a cabo en la segmentación del cuerpo humano en diferentes partes de los vídeos y fotos directamente, pero los problemas perspectiva y apariencia siguen permaneciendo.

La estrategia en éste caso puede ser resumida como: En la primer etapa se hace una división de una escena visual en pequeños segmentos llamados superpixeles, usando técnicas de sobre-segmentación. En la segunda se lleva a cabo el etiquetado éstos superpixeles a través del entrenamiento de una máquina de vectores de soporte multi-clase, para entonces encontrar una configuración que maximice la probabilidad de éstas clases. Los superpixeles son descritos usando características invariantes a la profundidad. La estrategia propuesta fue probada en el Depth Gait Dateset, y probó tener un rendimiento competitivo.

Palabras clave: Predicción Structurada, segmentación de imágenes, campos aleatorio de markov, máquinas de vectores de soporte, análisis de marcha humana, kinect.

Publications

Portions of the work described in this thesis has also appeared in:

Conference Papers

- Rozo, Paola and González Fabio. Human Body Parts Identification in Sagittal Gait Videos using Support Vector Machines. IX Congreso Colombiano de Computación. 2014. Pereira, Colombia.
- Mathieu Dubois, Paola K. Rozo, Alexander Gepperth, Fabio A. González O. and David Filliat. A Comparison of Geometric and Energy-Based Point Cloud Semantic Segmentation Methods. ECMR 2013. Barcelona, España. [Poster]
- Rozo, Paola and González Fabio. Semantic Image Segmentation Using Markov Random Fields. VI Congreso Colombiano de Computación. 2011. Manizales, Colombia.[Poster]

Contents

List of Figures

List of Tables

Part I

Preliminaries

Chapter 1

Introduction

There are a variety of techniques for image understanding and multimedia information retrieval, that combine data properties, such as low-level features, metadata and labels. These techniques are used to find patterns that define, classify, or group relevant results according to user needs in an appropriate time period, or to deduce the different objects and scenes associated to semantic concepts. The definition of these patterns derives on the correct ontology and heuristic approach, that necessarily includes measures of similarity between different data (images), taking into account physical or semantic similarity, or both.

Due to the importance of semantic concepts, the image understanding techniques have developed annotation methods based on keywords or metadata associated with the image, ontologies that provide the form or hierarchy between the concepts of the image, segmentation and vocabulary building such as super-pixel, bag of features, salience maps and visual attention models, and finally classification using low-level features. In this context, image segmentation provides an interesting way to address the image understanding problem, because allows for the absence or presence of concepts associated with objects, at same time that turn up scenes that define the image type and emerging concepts according to the visual words spatial interaction.

The different approaches to image semantic segmentation can be group into several areas as Signal Analysis (SA), Machine Learning (ML) Probabilistic Analysis (PA) and Ontological and Human Perception Study (OHPS): on these areas, the common research strategy to address the problem is based on three steps: first, to calculate a visual vocabulary that involves bigger entities than pixels, second, to assign labels to these entities according to a trained classification model, and finally, group them into semantic objects using a SA/ML/PA/OHPS strategy.

Following this order, the different approaches have been divided in the three mentioned stages according to their contribution. At first,[53] ,[39] and [54] built the visual vocabulary through small blob based super-pixels (A group of pixels that are obtained using over segmentation methods), represented by descriptors such as color, texture [53], 2D frequency planes [12], SIFT, SURFT, among others. Those features could receive some kind of pre-processing for reducing the dimensionality of the variables to classify, such as PCA [12],. Once the super-pixels are characterized, they must be group in order to generate a more specific, reduced and non-redundant vocabulary.[39] and [44] adapted the bag-of-feature model based on hierarchical K-means, [1] built a randomized decision forest that uses simple pixel comparisons, performing an implicit hierarchical clustering into semantic texton.

At second stage is necessary to build a classifier for the labeling task, according to the previously visual vocabulary. Some approximations are based on a probabilistic framework, for example,[39] proposed a yielding second- order Markov Random Field (MRF) in conjunction with a MAP solution, [54] a MRF incorporating local interaction in unsupervised parameter learning, [?]turn raised a learning method that estimates the parameters by maximizing a lower bound of the data likelihood (CRF), same way than [43][and [53] used a MRF and SVM combination, [44] raised a multivariate iterative region growing using semantics (MIRGS) in a MRF context model with edge penalty, [56] assumed a different perspective of the problem through the template definition from similar image, that means with same objects or scenes, in different viewing angles and finally [59], at same pixel level, proposed a finite mixture model to approximate the class distribution of the pixels through an Expectation-Maximization (EM) algorithm.

Other solution is addressed to Multiclass SVM strategies such as the [53] model in which the variables are mixing with a MRF method, [45] where the features are unified through a weighted linear combination provided by relevance feedback, [12] incorporating Neuronal Networks in the optimization function and[19]in which salient objects are detected and defined using EM.

ML approaches are reported too, such as Bayesian perspective[34] in which model the effectiveness is due to the capacity to integrate knowledge to the network probability, and reduce a joint probability distribution to conditional independence relationships, [16]. Linear Discriminative Analysis (LDA), [32]Kernel version model based on Gaussian mixture strategy (KGMM). Graph based methods such as [32] that extend the bag of feature model, [17] in a connected coherence tree algorithm based on neighbor coherent segmentation criterion, and [4] with N-cut process. Generative Models [60] combines coarse shape information and robust parameter estimation, finally Relevance Feedback approximation [18] in which the system recomputed the weights of every relationship between the defined semantic concepts.

Some alternatives to this ML approaches including Ontological treatment,[5] that propose matching a region's low level descriptors against a knowledge base or fussy label set, defining merging process and fuzzy operations and [10] that calculates the regions similarity using the WordNet ontology enhanced with appearance information, that is orthogonal to visual similarity and finally Morphological Tools, [58] in which the regions are grouped according to morphological erosion and dilation.

Finally the labeled regions are group according to its similar labels, forming in this way, the analysis base semantic objects.

1.1 Problem Identification

The problem addressed in this proposal arises the solution of Image Understanding through Semantic Segmentation, where the goal is to assign one semantic category to every pixel's image and then group them into segments that can be related to scene objects. An alternative is to perform first a segmentation and them assign a label to each segment. An intermediate approach consist in performing an oversegmentation that generates a set of small homogeneous regions (called superpixels) that are later labeled and grouped together in segments. The latter approach is the one followed in this work.

The main problem here is how to assign labels to superpixels. A first approximation to solve it, is to model it as a classification problem, in which each superpixel is modeled as a variable that can be classified according to features such as color, texture, etc. This approach, although valid, does not take into account valuable information such as the way to relate the defined regions, which could lead to a new semantic concept definition or the establishment of tagging patterns, therefore, the question at this point would be, how to model the relationships among superpixels? We propose to model the problem as an structured output learning problem. The goal is then to build a prediction model that is able to assign all the labels at once and that models the image as a graph composed of superpixels (nodes) and their relationships (arcs).

According to this, the main objective is to answer questions that define the relevance, effectiveness and efficiency of structured prediction methods based on those reported in the literature, the same way to identify which structured prediction methods are more effective for the specific image segmentation problem.

1.2 Objectives

- To develop and evaluate a prototype of a semantic image segmentation system using a structured output prediction method based on a superpixel representation.
  - To develop a method for superpixels extraction and characterization.
  - To develop a method for image representation based on superpixels and a bag of features strategy.
  - To propose/adapt and develop a structured output prediction strategy to assign semantic labels to superpixels.
  - To evaluate the performance of the semantic segmentation system using a manually labeled image data set.

1.3 Structure of Thesis

The remainder of this thesis is organized as follows. Chapter 2 reviews and discusses the state of the art in image segmentation and a brief summary of the approaches to the problem of object road detection and gait segmentation analysis.

Chapter 3 describes the proposed the overall structured output prediction strategy for segmentation, describing the all phases, the oversegmentation process, the learning model construction and the inference problem.

Chapter 4 presents the experimental results and analysis on two datasets, CamVid and Gait Database.

Finally, Chapter 5 presents our conclusions and outlines some avenues for future research.

Chapter 2

Literature Review

The daily process of image understanding and interpretation that performs the human brain, defines the way we interact with the environment, allowing us to build knowledge according to the ability of the brain to detect objects, recognize their interactions and predict the state of them. The image processing research area, through the analysis of human visual strategy, establishes as a method of understanding the environment (measure through images) the recognition and tracking of objects in time.Research in this area of knowledge, can be used in practical applications such as medical tests (location of tumors and other diseases, measured tissue volume, computer-guided surgery, diagnosis, treatment planning, study of the anatomical structure) , locating objects in satellite images (remote sensing), fingerprint sensors, face recognition, iris recognition, traffic control systems, computer vision, among others. This work will focus on the application of computer vision, more specifically on the problem of image segmentation, strategy established as one of the most important stages, when the image understanding is performed Figure 2.0.1. In this chapter we summarize the state of the literature of recent techniques used on images segmentation. Later we will describe the particular problem of human gait analysis, and how the image segmentation solve the problem.

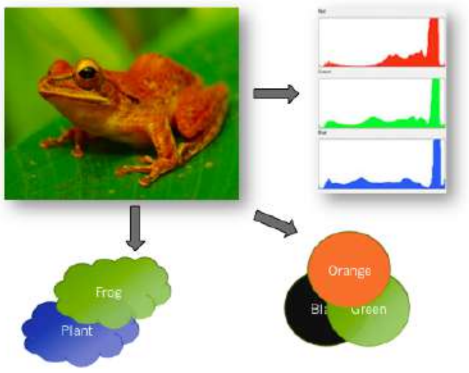


Figure 2.0.1: Image understanding examples

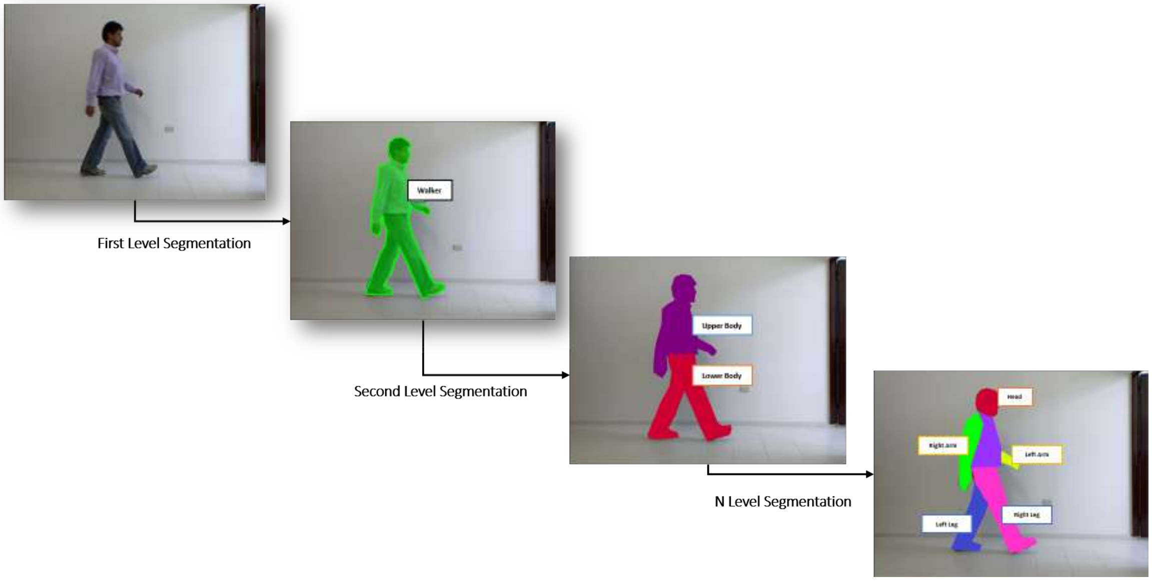


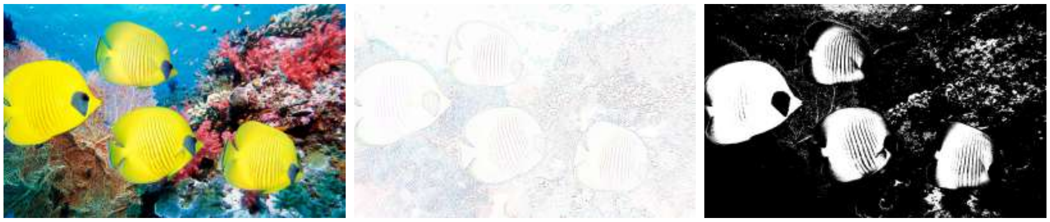
Figure 2.1.1: Image segmentation levels

2.1 Image Segmentation

Image segmentation can be seen as a strategy that subdivides an image into its constituent parts or objects, in order to find regions of interest. The goal of segmentation is to simplify and/or change the representation of an image in a more significant and easier to analyze. Segmentation is used both to locate objects and to find the limits of these within an image. More precisely, the image segmentation is the process of assigning a label to each pixel of the image, so the pixels that share the same label will also have certain visual similarly. The level at which the subdivision of the image is done depends on the particular application, therefore the segmentation end when the relevant objects to the application have been found, as exemplified in Figure 2.1.1 where the the purpose is to find classes within a gait image; at first segmentation level the Walker object is found, at second level the Upper y Lower Body labels are defined, and at N-level the N-Parts of the body are found, in this case Head, Right and Left Arm, Torso, and Right and Left Leg.

Broadly speaking, the regions of interest can be found through discontinuities in the intensity of gray levels in the image, defining edges and/or lines that delimit interesting objects, or by grouping based on similarities in features between pixels, as we can seen on Figure 2.1.2, where the similarities separate the background of objects of interest, and discontinuities establish the boundaries of objects in the image, such as Fish, Coral and Sea. They could finally be defined by hybrid methods between these two approaches. Techniques to find these regions of interest and make appropriate assignment of labels, are as diverse as the datasets that have been considered in the literature.

There is not a standard technique of image segmentation that works effectively on all kinds of images, this is a result of the ambiguity of the task itself. It is worth to say that segmentation is an ill-defined problem and often unconstrained.



(a) Original Image

(b) Sobel filter defining objetcs

(c) Thresholding filter defining background and objects

Figure 2.1.2: Image analysis approaches

Choosing the appropriate segmentation algorithm for a particular applications is critical, because depending on the context, the definition of semantic objects should be carried out quickly, or in detail, or should mediate between these two conditions. One possible solution is to try all possible algorithms probe to solve the segmentation problem, and choosing the algorithm for the best results, but, due to the large number of algorithms proposed in the literature, which often pose a great implementation complexity, make the exhaustive task is not feasible. In practice, only a limited number algorithms can be tested, these must be based on known characteristics and performance measurements, and implementations independent on applications. So then, the investigation in this area should be focus on optimizing, modifying and generalizing techniques have been shown effectiveness in a wide range of applications, as well as generate new algorithms to be compared against the state of the art results. The segmentation techniques can be classified in different ways, for example, according to their application areas, theirs implementations issues, the way them can be used, or the logical form of the algorithms themselves. Considering the last classification standard, and focusing on whether the techniques take uncertainty into considerations, the technique can be grouped in Probabilistic and Deterministic Approaches[37].

2.1.1 Probabilistic Approaches

In this approach, a probabilistic model is built, as deduced a mathematical representation of a set of assumptions for the dual purpose of studying the results of a random experiment and predict their future behavior, when is performed under the same initial conditions. Thus, in the probabilistic model labels assigned to each pixel represent random variables that we want to predict, while each of the pixels and their features, represent the initial conditions. The probabilistic approaches can be subdivided into two groups Graphical Models based Segmentation and Bayesian Segmentation according to how the probabilistic model is built [37].

1. Graphical Models based Segmentation

The graphical models formulate the image segmentation with a solution based on graphical model theory, in which the task is to describe how the random variables and the observation (labels and pixels, respectively) can interact. This is achieved using structural assumptions as to the form of the joint probability distribution of all the variables, typically corresponding to assumptions of independence of random variables. Each class of graphical model corresponds to a factorization property of the joint distribution. Once the basic assumptions as to how variables interact with each other is formed, all questions of interest are answered by performing inference on the distribution. This can be a computationally non-trivial step so that coupling GMs with accurate inference algorithms is central to successful graphical modelling.

According to the types of graphical models the techniques can be divided into undirected graphical models and the directed acyclic graphical models. The undirected graphical models represent non-casual relationships between the random variables, such as the spatial homogeneity. An example of these are the Markov Random Fields (MRF)[13] and Conditional Random Fields (CRF) [24], they incorporate the spatial relationships among neighboring labels as a Markovian prior. This prior can encourage the adjacent pixels to be classified into the same group. The limitation of these two models is the inability to model causality relationships to solve this problem, which would make its simplest formulation, the directed acyclic graphical models: such as Bayesian Networks (BN)[28] can model the causal relationships between random variables using directed links and conditional probabilities.

Markov Random Fields

MRF has been widely used in image segmentation problem[References]. The basic MRF model includes the formulation of the joint probability distribution of the image observation and labels in a regular 2D lattice. MRF assumes that the image observations are conditionally independent given the label on each site, this condition restricts the analysis could be made of the discontinuities and similarities of the image.

From this basic structure (2D lattice), they have detached models such as Hierarchical MRF (Hidden MRF, binary tree-structured MRF, Couple MRF, Multi scale MRF as quad-tree structure, pyramidal structure and complex structures) that propose a model to analyze the overall inference problem through optimization problem between adjacent levels equivalent to pyramids of segmentations (image decomposition). Thus, there is communication between the different levels of the pyramid of the multiresolution image, so the segmentation problem is being resolved by levels.

Spatial-temporal MRF is obtained by adding to the regular MRF the time dimension. The model combines the spatial and temporal aspects of video sequences, considering image differences between consecutive frames as observations. The flat MRF directly models the interactions between pixels, this is only interesting in cases where no long run interactions are needed, e.g. in images with small structures. In images with larger and, more importantly, scale varying content, the hierarchical nature of the markov cube manages to better model the image contents.

2. Bayesian Segmentation

The Bayesian segmentation formulates the problem directly using Bayesian statistics. Strong independence assumptions must be done among random variables in order to derive the probability distributions. The algorithms can be divided into discriminative and generative, according to the property of the statistical formulation. Discriminative algorithms treat image segmentation as a kind of classification problem that satisfy a selected criteria. Support Vector Machines (SVM)[26], Neural Networks(NN)[15], Decision Tree [57], Probabilistic boosting [51], Parzen Window [48], Log Linear [11] and MAP Models [29] are examples of this kind of approaches. Generative algorithms model the joint distribution of the class labels and the observations. Joint probability can be factored into the product of the likelihood and prior distribution of the labels. The likelihood of the data can be modeled using parametric or non-parametric methods. In the non-parametric approximations use a simple histogram to represent the likelihood, and the parametric one, use Gaussian or Mixtures of Gaussians (MoG)[46] to this task. The prior distribution of the labels can be modeled as Gaussian or multinomial distributions. Having the joint likelihood and the prior distributions a MAP inference can be done through stochastic simulation.

2.1.2 Deterministic Approaches

In the deterministic approaches the problem is set as mathematical model where the same entries invariably produce the same outputs, not contemplating the existence of chance or the uncertainty principle The deterministic approaches can be subdivided into clustering techniques, region growing and region splitting and merging. The clustering techniques have into account two general ideas, group the pixels that belong together because they lie on the same object and/or group them belong together because they are locally coherent. That coherency is measure through small distance in feature space (whatever features is going to having into account). The questions in here are, how many clusters is need to find, and which the better distance measure to use is. Clustering algorithms can be classified into herarchical or partitional [21]. The hierarchical [7] involve the clusters themselves being classified into groups, where the process is repeated at different level to form a tree. Partitional techniques [36] generates clusters by optimizing a clustering criterion where the classes are mutually exclusive, thus forming a partition of data.

On the other hand, the region growing methods [49][47]] aims to join pixels of an image with similar properties to create a region of interest. This task is performed as, first, finding a seed pixels inside the image, then merge similar pixels around the seed domain and, finally, the similar pixels are use as new seed. The process is stopped when no seed found which has same value. The region splitting [3] and merging methods [25] represent the image as a tree with connected graphs without cliques. The root of the tree would be the image itself, later the image is divided into a set of four arbitrary disjoints regions that represents the leaves. If the brothers-leaves are homogeneous (according to a specific measure), they can be merge and are going to represent one node of the tree. The process is cyclic and ends when no further merging is possible.

2.2 Road Object Detection

The object detection problem on road traffic is one promising research area for future intelligent transportation systems. The applications of the object detection ranging from autonomous navigation system until contribution on road safety. To address the problem, one of the approximations is to analyze the traffic image sequences from roadside camera, as for example, the CamVid dataset [9] describe in section 4.1. Several image processing algorithms have been developed to solve the problem, these involves the traffic theoretical modeling, object detection and monitoring, and the combination of both. Image Segmentation contributes in the object detection task, though the the recognition of all actors in the scene of the road, according to this, all the previous techniques can be used, taking advantage of the unique characteristics of the data, e.g. the geometrical distribution of the classes, the sky and ground differences on intensity value, spatial context, and so on. The recognition of all objects in different sequences in the road dataset, gives way to track objects in time, laying the foundation for predicting traffic conditions over a period of time.

2.3 Human Gait Analysis

Human Motion Analysis is an important research area in several applications such as video surveillance, image retrieval systems, human interaction, and medical diagnostics. The problem that arises in this type of analysis is the segmentation of the human body in different parts of the body including head, limbs, torso and feet. This problem is challenging due to the occlusion of body parts, variability on the appearance on the images such as clothes with similar colors and finally the image perspective. In recent decades, the research works have been carried out in the segmentation of human body in different parts of videos and pictures directly, as we can see on Section 2, but the perpective and appearance problems are still remain.

[8] shows a new way in research to incorporate depth images, acquired at a low computational and monetary cost, in motion analysis. The presented Kinect device, not only provide portability on the image acquisition, but provides the resources to perform a quantitative analysis of the human gait through depth values of the human walkers with respect to the camera, without the need to take into account factors such as sensitivity to the appearance color and texture. They obtained good results through the set of depth invariant features and a Random Forest classifier, applied on millions of frontal plane human images with respect to the camera, but the performance in sagittal gait images, that are the key in the problem of medical diagnosis, is poor because in the lower and upper limbs the distinction between right and left is missing due to their occlusion. For this reason we intend to use the features described in [8], but with a different classification method that can report a better performance on body parts detection in sagittal images, at same way that does not need millions of training images for a good performance.

In this paper we explore a technique based on two main strategies: first, the division of the scene in small segments called superpixels using a oversegmentation technique, and, second, to label these superpixels defining a multiclass Support Vector Machine (SVM) and then finding a labeling that maximizes the probability given a set of classes. Superpixels are characterized using depth invariant features. The proposed strategy was tested on the Depth Gait Dataset and showed a competitive performance.

2.3.1 Related Work

During the past few decades several studies have been conducted through segmenting a human body to its different parts directly from videos. For example [38] proposed to extract and follow contours of every body part, these parts are approached by simple 3D geometric objects (blocks), which 3D position and motion are estimated for the each image of the image sequence. The approach makes use of knowledge about the human body, reference points, optical flow, contour analysis and 3D shape modeling. In [23] the reported method combines hierarchical body pose estimation, a convex hull analysis of the silhouette, and a partial mapping from the body parts to the silhouette segments using a distance transform method that does not violate the topology of the human body. In [42] Gaussian mixture model is used at the pixel level to train and classify individual pixel colors. Markov Random Field (MRF) framework is used at the blob level to merge the pixels into coherent blobs and to register inter-blob relations. A coarse model of the human body is applied at the object level as empirical domain knowledge to resolve ambiguity due to occlusion and to recover from intermittent tracking failures. [41] Introduce the approach that uses segmentation to guide an recognition algorithm to salient bits of the image, using this segmentation approach to build limb and torso detectors, the outputs of which are assembled into human figures. In [6] is employed a hierarchical visual-hull algorithm which segments only the most interesting regions of the images and includes colour information. The tracking step uses blobs attached to a kinematic model to recover joint angles in an expectation-maximization framework. In [?] the body parts detection is performed in two steps. First, joints candidates are extracted from the silhouette's contour. Then the model is used to apply constraints to isolate each limb.

Part II

Semantic Segmentation and Analysis

Chapter 3

Overall Learning Strategy

3.1 Introduction

The semantic segmentation problem is defined as follows: given an input image (usually representing a scene), assign a label to each one of the pixels; the labels are associated to high-level concepts that give a semantic interpretation to the scene; adjacent components with the same label constitute the "semantic segments" and are associated to the real world objects indicated by the label.

The problem can be approached from different perspectives. One alternative is to directly assign labels to the pixels and, after this, find the connected components that constitute the semantic segments. Another alternative, is to find first a segmentation of the image and then assign a label to each segment.

We follow an intermediate approach, first we found an oversegmentation of the image, then labels are assigned to each small segment, called here superpixel. Later contiguous superpixels with the same label can be merged to form the final segments. The result is illustrated in Fig. **??**. It is important to said that each superpixel can be defined as a set of one pixel, and then, we can handle a per pixel classification.

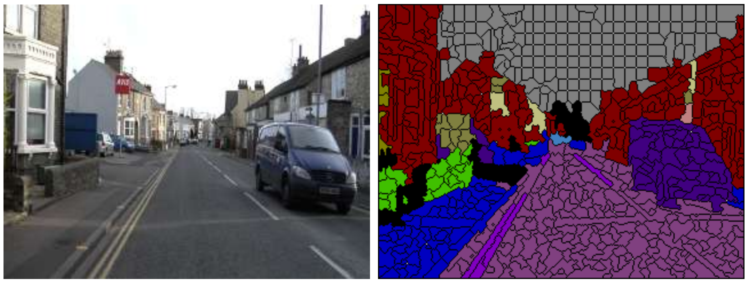


Figure 3.1.1: Semantic segmentation of an image using superpixels. The original image is shown at the left. The right image shows the superpixel-over-segmented image where each superpixel has been assigned a semantic label indicated by a color.

The algorithms to assign labels to the superpixels (Learning Methods) can be as diverse as we want; In this thesis we chose to implement a Markov Random Fields (MRF) and a Support Vector

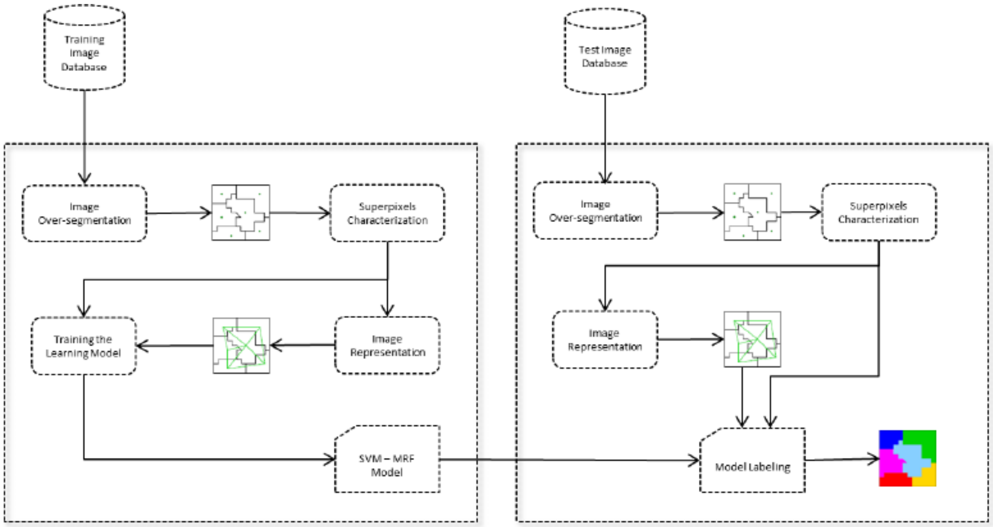


Figure 3.2.1: Semantic segmentation process

Machine (SVM) method.

3.2 Methodology

The strategy we chose to solver the segmentation problem is divided in two main phases: training and testing. During training, the Learning Method is trained using a set of labeled images. During testing the Learning Method is used to assign label to new images. The overall process is illustrated in Figure 3.2.1.

The training process works as follows:

1. For each image, a superpixel extraction algorithm is applied to find an oversegmentation. The algorithm uses watershed segmentation applied on the image Laplacian based on uniformly distributed seeds, or the SLIC method. This step is described in Chapter 4.

2. For each superpixel in each image, a feature vector that represents the visual, geometrical or depth information of the superpixel is calculated. An optional step inside the image characterization is the following: The set of all feature vectors, from all training images, is used to build a Bag-of-Features (BOF) codebook. This is done by applying a quantization algorithm that could be unsupervised (using k-means) or supervised (using Learning Vector Quantization LVQ). All the images of the training data set are represented by the corresponding codewords for each superpixel, additionally the coordinate and color of each superpixel is stored. This step is described in Chapter 5.

3. A superpixel neighborhood graph is calculated. Two superpixels are said to be neighbors if they share one or more boundary pixels. Fig. **??** shows a graph for an example image.

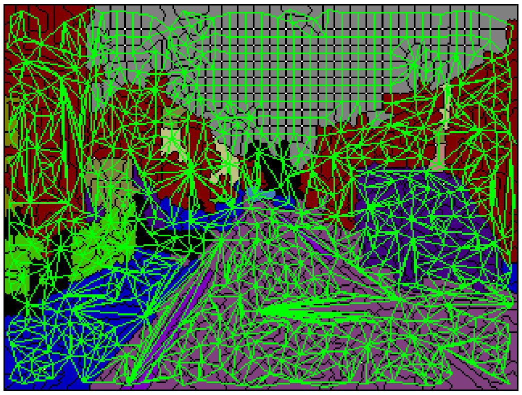


Figure 3.2.2: Superpixel neighborhood graph for an example image.

4. An Learning model is trained by calculating the probability distributions that correspond to the parameters of the model. This step is described in Chapter 6.

In same way the test process is as follows:

1. For a particular image, the superpixel extraction and image representation processes are applied as described in the training process.

2. The Learning Model is applied, using the parameters learned during the training phase, to find the superpixels labels.

Chapter 4

Image oversegmentation

It is need to define a strategy to handle the amount of entities to have into account on the training of learning model, due to the computational complexity involved in classification of all pixels in each image. The datasets is composed by hundreds of images, then we are going to get millions of pixels to characterize and classify. The oversegmentation, which is the process by which the objects being segmented from the background are themselves segmented or fractured into subcomponents, is therefore a good strategy to reduce the number of example to train. We are going to call this subcomponents Superpixels.

The superpixels must capture the image redundancy and greatly reduce the complexity of the image processing task. So, we have two problems, the first if we extract a few number of superpixels the representation will lose redundancy, second, if we extract many of them, the cost computing classification is comparable with the handling pixels.

Besides this, we want superpixels [2]:

1. They should adhere well to image boundaries to reduce the possibility of misclassification.
2. When used to reduce computational complexity as a pre-processing step, superpixels should be fast to compute, memory efficient, and simple to use.
3. They should improve the quality of the results.

There are different strategies to get the superpixels, them can be categorized as either graph-base or gradient ascent methods. Some of the most important algorithms are: Inside the Graph-based methods, we found the Normalize cuts algorithm [35]which builds and cuts a graph based representation of the image, according the contour and texture cues. The cuts are performed through a globally minimizing a cost function defined on the edges at the partition boundaries. Felzenszwalb and Huttenlocher [20] method performs an agglomerative clustering of pixels as nodes on a graph, such that each superpixel is the minimun spanning tree. Moore et al [40] proposed generate the segments finding optimal paths, or seams, that split the image into smaller vertical or horizontal regions.

On gradient-ascent based algorithms, we found methods based on Mean Shift that find local and maxima point on the feature color space. Watershed approach[52], that performs a gradient ascent starting from local minima to produce watersheds that define the superpixels border, and finally, Turbopixels method [31]that progressively dilates a set of seed locations using level-set based geometric flow.

On [2], a comparative of these methods are performed, resulting in the conclusion that the SLIC method proposed by them, produces a better oversegmentations in a shorter time that all the algorithms outlined in the previous paragraph. Watershed method, which has been widely used producing good results, uses an opposite strategy to SLIC extraction. Therefore, as superpixels extraction methods for this thesis, it was decided to use Watershed and SLIC, to conduct a comparative study on the use of the two algorithms.

4.1 Introduction

The watershed concept comes from the field of topography: in a topographic relief, watershed lines are the boundaries separating the basins of alluvial rivers and lakes, each basin is associated with a local minimum of relief. The basic idea of the algorithm is to extend the basins by simulating a process of flooding from local minima. In our case the reliefs represent edges in the image and the result of flooding defines the superpixels to work on. The watershed transformation can be applied to grayscale images, taking into account that the intensity of a point represents a height in a topographic relief associated.

On the other hand, the SLIC algorithm generates oversegments through grouping of pixels according to their color similarity and the proximity in the image plane. This is done in the five-dimensional space $[l\,a\,b\,xy]$, where $[l\,a\,b]$ is the color vector of the pixel in the CIELAB color space, which is considered to be perceptually uniform color for smaller distances, and xy is the position of pixel.

Some examples of the produced oversegmentation are shown in Figure 4.1.1.

In the context of the Human Sagital Gait and CamVid object recognition problems, the superpixels strategy has completely sense, it is due to the computational complexity involved in classification of all pixels in every image on different sequences: Depth Gait dataset is composed by 368 images, about of 95 millions of pixels , meanwhile CamVid has about 701 images that is 484 millions of pixels, all of them need to be characterized and handled, is therefore a good strategy to reduce the number of samples to train.

4.2 Watershed Oversegmentation

The watershed algorithm strategy is to convert the RGB image to grayscale one, because this can be seen as a topographic surface where each point's altitude is given by its gray level, in which each local minimum or maximum represents an edge inside the image.

The surface is flooded from below by allowing water to rise from each regional minimun at a uniform rate across the image. When the water level coming from two distinct minimum point is about to merge, a barrier is lifted to prevent the merging of the sources. Eventually the flooding covers all the surface, and the barriers that were lifted would be the watershed lines. The overall process is illustrated in Figure 4.2.1.

The image can be visualized on three dimensions, two spatial coordinates and one depending on the grayscale. In this topological representation, all image points can be seen as points belonging to global minimum/maxima (red and blue areas in Fig. 4.2.2), points placed in the range of the

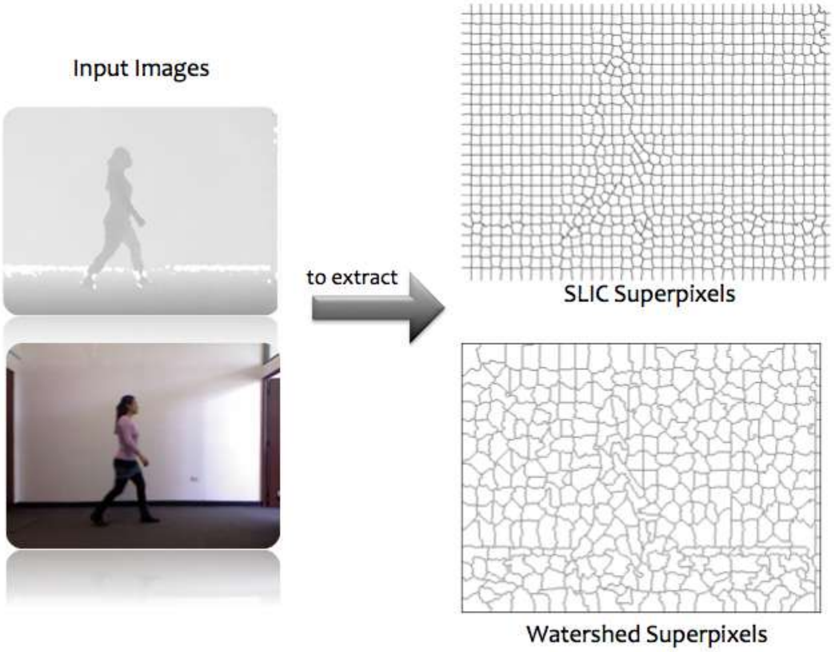


Figure 4.1.1: Image oversegmentation examples

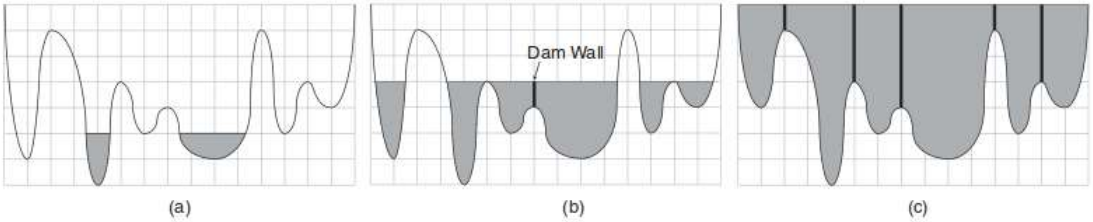


Figure 4.2.1: Three different stages of watershed construction by flooding on a function of one variable. The final watershed lines are given by the thick vertical "dam walls" in (c). [22]

local minimum/maxima and points that can belongs to more than one local minimum (yellow areas in Fig. 4.2.2).

The phases of the oversegmentation strategy can be summarize as flows:

1. Piercing holes in each regional minimum of the image. Initially, the set of pixels with minimum gray level are 1, others 0.

2. The 3D topography is flooded from below gradually. In each subsequent step, we flood the 3D topography from below and the pixels covered by the rising water are 1s and others 0s.

3. When the rising water in distinct catchment basins is about to merge, a barrier is built to prevent the merging, the barrier boundaries correspond to the watershed lines to be extracted by a watershed segmentation algorithm.

However in inhomogeneous images the noise generates a large number of local minima, resulting on oversegmentation in small regions, where not important objects are located, or not represent any object in the original image. The oversegmentation can be reduced using improved methods and morphological filters. If these methods do not improve, one of the best known strategies is to define unambiguous markers for each of the objects of interest.

Markers or seeds replace local minima and initiate flooding algorithms, indicating sectors will result in barriers. In this case, the success of the technique Watershed, depends mainly on an appropriate selection of markers. Depending on the problem, is it need to define a different heuristics to choose the markers which will have to start the process [30].

This segmentation technique is recommended for images with homogeneous textures and weak intensity gradient. Finally, the objects resulting from segmentation corresponding to the minimum gradient of morphological and contours gridlines water gradient.

4.2.1 Algorithm

There are several implementations of the Watershed algorithm, our methodology takes into account the Vincent and Soille proposal [52], due to efficiency in the process of flooding and lower computational cost compared to other technical implementations.

The proposal is presented in five stages ash shown in Algorithm 4.1.

4.3 Simple Linear Iterative Clustering (SLIC) Superpixels

The Simple Linear Iterative Clustering SLIC [2], adapts k-means clustering to generate superpixels. It is an algorithm with a simple design, but it has advantages such a the optimization in the search space to a region proportional to the superpixel size, reducing the complexity to be linear in the number of pixels N and independent of the number of superpixels k. And on the other hand, a weighted distance measure combines color and spatial proximity, while simultaneously providing control over the size and compactness of the superpixels.

Algorithm

The algorithm has into account only one parameter, k. It is the number of equally-size superpixels that we want to get. As described in [2], the algorithm handles color images in the CIELAB space, according to these it begins a clustering process over the pixels as shown in Algorithm 4.2.

Algorithm 4.1 Watershed oversegmentation algorithm

1. The pixels are sorted according to the gray level, then access them through a FIFO data structure(First Input First Output), to start the process of flood barriers image.

2. Each of the local minima is assigned a different label, which spreads to all adjacent pixels within a given level l. It begins by analyzing the binarized image with a threshold l, equal to the minimum gray value of the image to the binarized image with a threshold equal to the maximum value of gray.

3. At each step the components connected to the binarization l and $l+1$ are analyzed. At the end of the flood, all pixels under the Watershed lines have a label indicating they belong barrier or region. The negative flooded basins form the Watershed lines between the different regions of the image, that grew from the regional minimum.

4. The flood process is done by comparing the image f, binarized with a threshold i, that is denoted $Z_i(f)$, and the image at a higher level $Z_{i+1}(f)$, for $i = 0$ to $i = N$, where $N$is the maximum gray level of the image is represented. At each level i of the image f it must be present a regional minimum $m_i(f)$. The vessel or barrier image f at i-level, is name $W_i(f)$, which are initially denoted regional minimum. $W_{i+1}(f)$ is the result of the flooding of the barrier $W_i(f)$.

5. During flooding, three cases are possible: (1) Growth of an existing barrier in $Z_i(f)$. In this case, if the pixel value of the images being compared, ie $Z_{i+1}(f)$ is greater than $Z_i(f)$, occurs growth (flooding) of the barrier $W_i(f)$, ie $W_{i+1}(f)$ is generated, if remains equal the barrier stays still. (2) Emergence of a new barrier, if the value of the pixel of the images being compared, ie $Z_{i+1}(f)$ is less than $Z_i(f)$, new areas of flooding appear, ie new barriers. (3) Determination of zones of influence, if the flood level $i+1$ binds flooded barriers of level i, it is need to separate regions using the zones of influence of each connected component. In this case, the barries of $i+1$, that is W_{i+1} are zones of influence W_i basin.

Algorithm 4.2 SLIC oversegmentation algorithm

1. The first step is the definition of the initial k cluster centers defined as $C_i = [l_i a_i b_i x_i y_i]^T$ and are sampled on a regular grid spaced S pixels apart. The grid defines the superpixels that need to be roughly equally sized, for this reason the grid interval is $S = \sqrt{\frac{N}{k}}$. Each one of these centroids need to be relocated to to the lowest gradient position, because we need to avoid centering a superpixel on an edge or a noise area . The centroid can move in a length invariant searching neighborhood, in this case 3 x3 pixels. The centers are moved to seed locations corresponding to the lowest gradient position in a 3×3 neighborhood. This is done to avoid centering a superpixel on an edge, and to reduce the chance of seeding a superpixel with a noisy pixel.

2. In the second step each pixel i is associated with the nearest cluster center whose search region overlaps its location in a neighborhood defined by a distance measure D. This reduces the search space compared to the traditional K-means clustering approach, that compare every entity against every cluster. The expected spatial extent of a superpixel is a region of approximate size $S \times S$, the search for similar pixels is done in a region $2S \times 2S$ around the superpixel center.

3. The third step is about the adjustment of the cluster centers to be the mean $[l\,a\,b\,x\,y]^T$ vector of all the pixels belonging to the cluster. A residual error must be calculated, between the new cluster center locations and previous cluster center locations, this can be performed through the L_2 norm. This task can be be repeated iteratively until the error converges.

4. The final step enforces connectivity by re-assigning disjoint pixels to nearby superpixels.

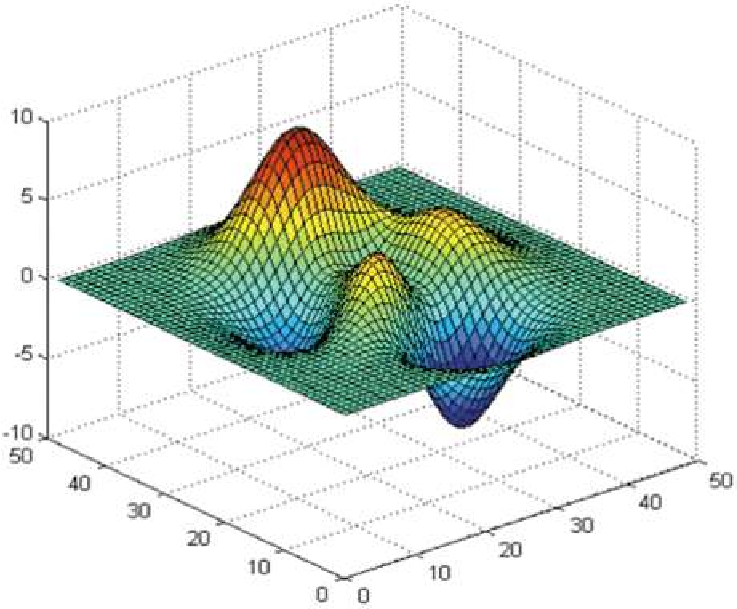


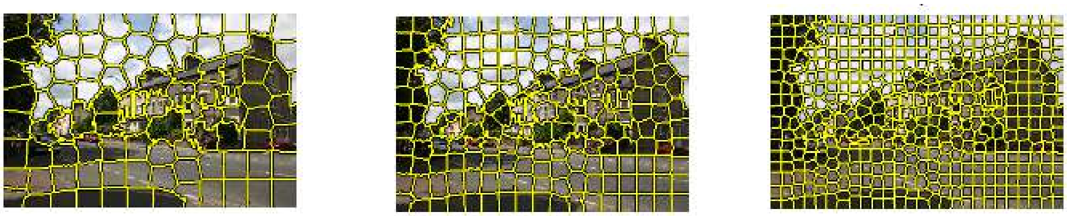
Figure 4.2.2: Maxima and minima representations of a grayscale image

An example of SLIC oversegmentation varying the number of superpixels can be shown in Figure 4.3.1, the irregular segments size and shape are clearly visible. The compactness constraint of SLIC makes the segmentation much more regular. This improves undersegmentation error and motion discontinuity error but comes at the cost of lower boundary recall.

Distance Measurement D

The overlapping of the neighborhood region of the pixel i against its nearest clusters center is defined by the distance measure D, between the pixel and the cluster in the CIELAB color space.

The pixel i is represented by $[l\,a\,b\,x\,y]^T$, whose range of possible values is known, and a position $[xy]^T$, whose values are defined by the image size. To combine the two distances into a single measure, it is necessary to normalize color proximity and spatial proximity by their respective maximum distances within a cluster. Doing so, D is written.



100 superpixels

200 superpixels

500 superpixels

Figure 4.3.1: SLIC oversegmentation example

Chapter 5

Superpixels Characterization and Representation

Once we get the superpixels that will summarize the content of the image, the next step is to characterize the content of each of superpixel, in other words, to represent their characteristics such as low-level property or relationships involving local semantic concepts. In this work were used low level features such as SIFT for the Road Detection Problem and Depth Invariant Descriptors for the Human Gait Analysis.

Once the superpixels are describe, a Bag of Features (BOF) codebook can be used to simply the classification task.

5.1 Scale-Invariant Feature Transfor (SIFT)

The Scale-Invariant Feature Transform (SIFT) allow us to detect and describe local features in images. It was proposed by [33], from them it has been used wildly in image search, object recognition, video tracking and gesture recognition, etc.

The algorithm is popular because it detects stable feature points of an object such that the same object can be recognized with invariance to illumination, scale, rotation and related transformations.

We are going to describe the SIFT algorithm according with its four stages.

1. Scale-space extrema detection

Interest points for SIFT features correspond to local extrema of difference of Gaussian filters at different scales, this is:

- The interest points location are determined as the local extrema of Difference of Gaussians (DoG pyramid). To build the DoG pyramid the input image is convolved iteratively with a Gaussian kernels.

- The DoG filter provides an approximation to the scale-normalized Laplacian of Gaussian. The DoG filter is in effect a tunable bandpass filter.

2. Keypoint localization

Interest points (called keypoints in the SIFT framework) are identified as local maxima or minima of the DoG pyramid images across scales. For each candidate keypoint we need to evaluate:

- Some keypoints are not good enough or their locations may be not accurate, so we should interpolate nearby data to accurately determine its position.
- We need to eliminating edge points. Such a point has large principal curvature across the edge, but a small one in the perpendicular direction.
- It is need to assign an orientation to the keypoint, the principal curvatures can be calculated from a Hessian function H .
- The eigenvalues of H are proportional to the principal curvatures, so two eigenvalues shouldn't differ too much.

3. Orientation Assignment

To determine the keypoint orientation, a gradient orientation histogram is computed in the neighborhood of the keypoint, so:

- The keypoint descriptor can be represented relative to this orientation and therefore achieve invariance to image rotation.
- It is need to compute the magnitude and orientation on the Gaussian smoothed images.
- The weighted gradient magnitudes are used to establish an orientation histogram, which has 36 bins covering the 360 degree range of orientations.
- The highest orientation histogram peaks correspond to dominant orientations. A separate keypoint is created for the direction corresponding to the histogram maximum and any other direction within 80% of the maximum value.

4. SIFT Feature Representation

Once a keypoint orientation has been selected, the feature descriptor is computed as a set of orientation histograms on 4 x 4 pixel neighborhoods. The orientation histograms are relative to the keypoint orientation, the orientation data comes from the Gaussian image closest in scale to the keypoint's scale.

Histograms contain 8 bins each, and each descriptor contains an array of 4 histograms around the keypoint. This leads to a SIFT feature vector with 4 x 4 x 8 = 128 elements. This vector is normalized to enhance invariance to changes in illumination.

5.2 Depth Features

The 3D information, together with the image low-level content, can give us valuable information about the object location, avoiding problems such as physically similar superpixels but semantically different, grouped at the same label. This information could represent the depth as a geometric feature from each superpixel on the oversegmentation model.

In the case of the Human Gait Database (Section 8.1), where the depth information of the scene is given, to characterize every single superpixel on the dataset is needed to choose a set of depth features to describe them, irrespective of the distance of the object, in this case the human walkers to the camera.

[?] proposed a set of depth invariant features, that adapting to our problem will be described as: through a pair of offset vectors of a defined length, and located in the center of superpixel that we are going to describe, we can establish the spatial configuration of it. This is performed through the comparison the depth values between a set of superpixels on a neighborhood localized using pairs of vectors described above. In this way the differences in depth will have a specific configuration for superpixels located near the border of the image, near the delimitation of the space between the walker and background, and so on.

For every single superpixel x we compute its features with the equation 5.2.1

$$f_\theta = d(x + \frac{u}{d(x)}) - d(x + \frac{v}{d(x)}) \tag{5.2.1}$$

where $d(x)$ is the depth at superpixel x,$\theta = (u, v)$ represents the pairs of offset vectors u and v and the normalization factor $\frac{1}{d(x)}$ guarantees that the features are depth invariant.

To ensure depth invariance must be chosen a set of pairs of offset vectors that cover a sufficient space range of the image, for this purpose a distance neighborhood is established and according to this a set of random or fixed vector are chosen. The neighborhood distance is estimated according to the average distance the superpixels and the background (distance in pixels), to ensure the discrimination provided by the definition of the features.

For this work we defined a set of fixed offset vector in a squared neighboor, as is illustrated in Figure []. For each neighboor level, a set of eight vectors are taken, and their combinatorics in pairs in given to the depth feature algorithm to calculate the corresponding characteristics.

Thus, if we choose to use the first neighboor level (8 offset vectors), we will get 28 feature values, if we take the second level (16 vectors) we will get 120 features, if we take the third level (32 vectors) we will get 496 features, and so on.

5.3 Bag of Features (BoF)

The oversegmentation process, for training images, give us several feature vectors classified in one semantic category, therefore for performing any superpixel classification we need to take into account every superpixel achieve at training stage, thus the possible values of any latent variable on the classification model would be unmanageable. An alternative to improve it is to build a codebook containing visual codeword that represent the output of a clustering algorithm over the training superpixels, in that way we have a set of representative superpixel to handle in a classification process. With regard to the superpixel's physical representation, representative algorithms have proven an efficient description of the image, these include the SIFT strategy described above, among others, at same way, there is a set of algorithms for performing the BoF clustering process, the most popular and efficient are K-mean, LVQ, Neuronal Networks among others. According to these possibilities, we need to run some preliminary experiments to calculate the segmentation quality values and choose the best option for complete our BoF phase.

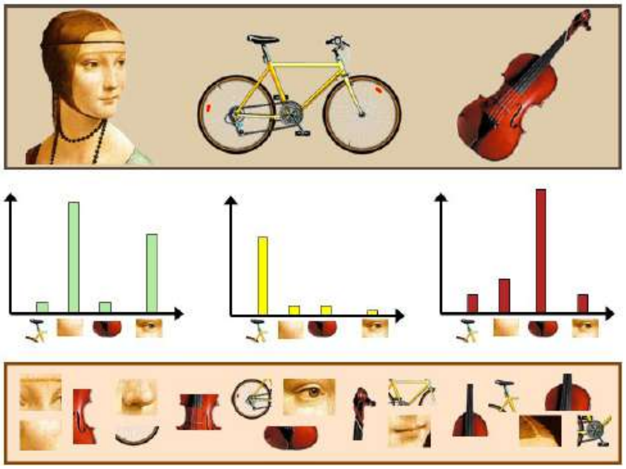


Figure 5.3.1: Bag of words representation. [?]

5.3.1 Introduction

To represent an image using BoF model, an image can be treated as a document. The document is composed by words; in images is need to be defined those words, as significant segments inside it. We can use the BoF model for object categorization by constructing a large vocabulary of many visual words and represent each image as a histogram of the frequency words that are in the image. The Figure 5.3.1 illustrates this idea. The motivation to represent the images as a document, comes from the fact that image features like texture depicts spatially repeating patterns, and many natural phenomena are textures.

To build the BoF model it is need to follow two steps: (1) feature detection and description, and (2) codebook generation. In our case, the features described each superpixel. At (1) the meaningful superpixel's features are explore, the different algorithm that we take into account are describe on previous section. The codebook generation (2) can be performed in several ways, but we describe two approximations that show competitive results in the state of the art, K-means and LVQ, clustering algorithms to summarized the all detected words.

Given a new image, we represent it using the BoF model in the following manner: first, extract descriptors from the image on a grid on the superpixels' center. Next, for each descriptor extracted compute its nearest neighbor in the dictionary. Finally, build a histogram of length k where the i^{th} value is the frequency of the i^{th} dictionary word, as is illustrated in Figure 5.3.2.

5.3.2 BoF Generation

Unsupervised K-means Approach

The clustering strategy to summarize the visual words detected, represented by a superpixels and their corresponding vector of features, divides and segments the superpixels in a predefined number of groups (clusters).

The approximation is describe in Algorithm 5.1.

K-means is a nice method to quickly sort data into clusters, all that is need to know are the

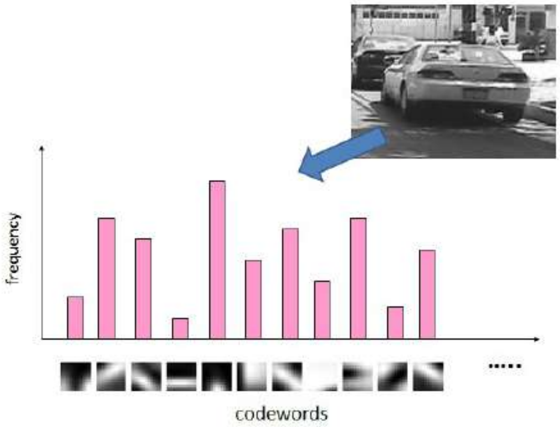


Figure 5.3.2: Bag of words example. [?]

Algorithm 5.1 K-means algorithm

1. Initial cluster seeds are chosen (at random). These represent the "temporary" means of the clusters c.
2. The squared Euclidean distance from each object x_j, that represents the superpixels centroid, to each cluster c_i is computed, and each object is assigned to the closest cluster c_i. There is the implicit assumption that the data should have roughly the same scale to use such distances.
3. For each cluster, the new centroid is computed and each seed value is now replaced by the respective cluster centroid c_i.
4. The squared Euclidean distance from an object to each cluster is computed, and the object is assigned to the cluster with the smallest squared Euclidean distance.
5. The cluster centroids are recalculated based on the new membership assignment.
6. Steps 4 and 5 are repeated until no object moves clusters.

number of clusters are sought to find. Local optima in K-means can derail the clustering results, when de process is not running many times with differing starting values[14].

Supervised LVQ

In contrast with k-means, which selects prototypes without using class labels, LVQ adjusts the position of the prototypes using information given by class labels of each superpixel, taking into account that the training dataset must be labeled. The Learning Vector Quantization algorithm is related to the Self-Organizing Map which is in turn inspired by the self-organizing capabilities of neurons in the visual cortex [27]. The goal here is to have the network "discover" structure in the data by finding how the data is clustered. In vector quantization, we assume there is a codebook which is defined by a set of M prototype vectors (amount of superpixels).

An input belongs to cluster c_i if i is the index of the closest prototype (closest in the sense of the normal euclidean distance).

The algorithm is consisted by 3 basic steps. The algorithm's input is: how many neurons the system will have what weight each neuron has for how fast the neurons are learning . and an input list containing vectors to train the neurons. The algorithm is describe in Algorithm 5.2.

Algorithm 5.2 LVQ algorithm

```
1. Define the number of clusters $N$.
2. Initialize the centroids $C_i$ according with the superpixel labels.
3. Initialize learning rate , epochs counter and repetitions counter.
4. For every epoch do the following steps for set vector as the Neural Network's input.
    (a) Select the winner neuron.
    (b) Update the weight vector for the winner neuron.
5. Check for termination. If not set and return to step 4.
```

In a nutshell, LVQ moves a prototype closer to the training sample points which have the same class as the class assigned to this prototype, and move away from sample points from different classes.

Chapter 6

Learning Algorithms

At this stage we have a set superpixels for training and testing databases, we have already characterize them, and summarized them if it is need. Now we are going to built a learning method to optimize a performance criterion using example data and past experience (labels by hand) to classified our superpixels. This strategy is called supervised learning.

Supervised methods are methods that try to discover the relationship between the input attributes (in this case superpixels) and a target attribute (labels). The discovered relationship is represented in a structure referred to as model. Usually models describe and explain the phenomena, which is hidden in the dataset and can be used to predict the target attribute value.

Supervised classification is one of the tasks performed most often by so-called intelligent systems. Therefore, a large number of techniques have been developed based on Artificial Intelligence (logical techniques / symbolic), Perceptron based techniques and Statistics (Bayesian Networks, techniques based Instance). In following section, we will focus on two of most important techniques of supervised machine learning, Markov Random Fields (MRF) ans Support Vector Machines (SVM).

6.1 Markov Random Fields

Markov Random Fields provides a way of modeling mutual influences among entities like superpixels and their correlated features, through conditional MRF distributions. The MRF is trained calculating the probability distributions that correspond to the parameters of the model. These distributions are:

- $P(x_i^{app}|l_i)$: the appearance conditional probability.
- $P(x_i^{geom}|l_i)$: the geometry conditional probability.
- $P(l_i, l_j)$: the neighboring label join probability.
- $P(l_i)$: the a priori label probability.

6.1.1 Markov Random Field Model Definition

A Markov Random Field (MRF) is a graph, (V, E), where each graph node, $l_i \in V$, corresponds to a latent variable on a lattice structure, and each edge $e_i \in E$, corresponds to edges between those

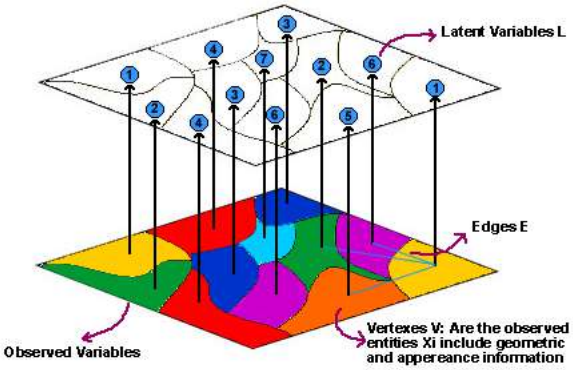

Latent Variables L
Edges E
Observed Variables
Vertexes V: Are the observed entities Xi include geometric and appereance information

Figure 6.1.1: MRF Structure?

variables (An example of this relationship is shown in 6.1.1).

The MRF satisfies the following property:

$$P(l_i|V\backslash l_i) = P(l_i|\mathcal{N}_i), \forall i \in V, \tag{6.1.1}$$

where $P(l_i|V\backslash l_i)$ represents the probability of the presence of the node l_i on the graph given all the vertexes V but l_i and$\mathcal{N}_i$ the set of neighbors of l_i. This is called the *locality property* and basically stands that the random variable l_i is conditionally independent of the rest of variables given its neighbors.

Usually the MRF's variables take values in a discrete set of labels, $\Lambda = \{\lambda_1, \dots, \lambda_m\}$. Also, it is common to associate each variable l_i, with a variable x_i. In this case, the variable l_i is called a latent variable, that means that it cannot be measured directly, and it has to be inferred by the values of the observed variables, x_i. In this particular framework, the problem is: given a set of observations to infer the most probable assignations for the latent variables, which can be stated as:

$$\max_L P(L|X) = \max_L P(l_1, \dots, l_n|x_1, \dots, x_n) \tag{6.1.2}$$

This problem is in principle, a hard problem to solve since the space of possible assignments for L grows exponentially with the size n of the graph. However, there are efficient algorithms that exploit the particular structure of MRF to find optimal or close to optimal solutions to this problem. In general, all the algorithms exploit the so called *factorization property* of the joint probability:

$$p(V) = \frac{1}{Z} \prod_C \psi_C(V_C), \tag{6.1.3}$$

where$p(V)$ represents the probability density function of the set of vertexes V, Z is a normalization constant, C runs over the maximum cliques of the graph.

A clique is defined as any non-adjacent variables that are conditionally independent given all other variables. In case of graphs, the clique is a set of vertex which are all neighbors of each other, it is for every two vertices, there exists an edge connecting the two. As an illustration turn to Figure 6.1.2.

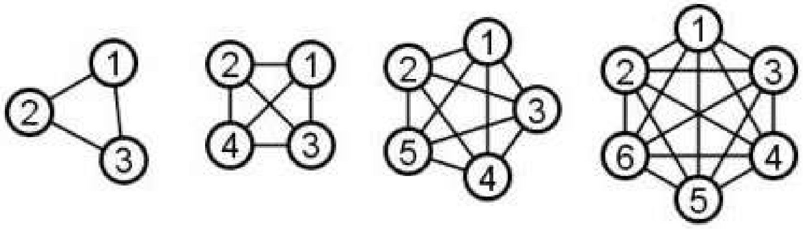
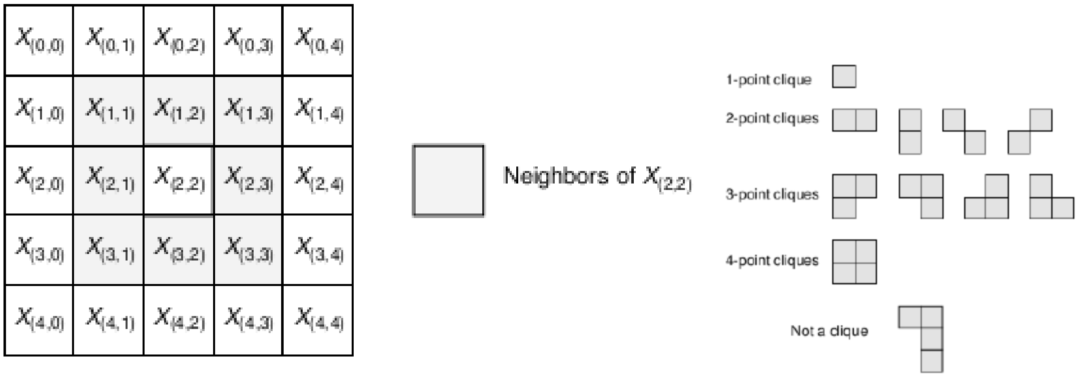


Figure 6.1.2: Examples of cliques with three, four, five and six vertices


$X_{(0,0)}$ $X_{(0,1)}$ $X_{(0,2)}$ $X_{(0,3)}$ $X_{(0,4)}$
$X_{(1,0)}$ $X_{(1,1)}$ $X_{(1,2)}$ $X_{(1,3)}$ $X_{(1,4)}$
$X_{(2,0)}$ $X_{(2,1)}$ $X_{(2,2)}$ $X_{(2,3)}$ $X_{(2,4)}$
$X_{(3,0)}$ $X_{(3,1)}$ $X_{(3,2)}$ $X_{(3,3)}$ $X_{(3,4)}$
$X_{(4,0)}$ $X_{(4,1)}$ $X_{(4,2)}$ $X_{(4,3)}$ $X_{(4,4)}$
Neighbors of $X_{(2,2)}$
1-point clique
2-point cliques
3-point cliques
4-point cliques
Not a clique

(a) 8 point neighborhood

(b) Examples of clique for 8 point neighborhood

Figure 6.1.3: Neighborhood system and clique

In case of images, the graph can be seen as a matrix of points in which each pixel represents a vertex. For an 5x5 pixel image a possible configuration is illustrated in Figure 6.1.3a; the set of all possible clique configurations, for example on vertex $x_{\{2,2\}}$ is seen on Figure 6.1.3b.

ψ_C is function over the variables of the corresponding clique called a potential function. Usually, the potential functions take the form:

$$\psi_C(C) = e^{-E_C(V_C)}, \tag{6.1.4}$$

where E is an energy function. In this case the joint probability can be expressed as:

$$p(V) = \frac{1}{Z}e^{-E(V)} = \frac{1}{Z}e^{-\sum_C E_C(V_c)}, \tag{6.1.5}$$

as a result, the MRF probability distribution is determined by specifying the energy function, and minimizing it is equivalent to maximizing the joint probability.

The problem of semantic segmentation is modeled using a MRF as follows:

1. The vertexes of the graph, V, correspond to the set of superpixels extracted from one image, the edges, E, are determined by the superpixel adjacency relationship.

2. The labels of the superpixels are modeled by the l_i latent variables. The observed variables x_i are broken in two variables x_i^{geom} and x_i^{app} that correspond to the geometric and appearance information of the superpixel respectively.

3. The MRF energy function is defined as follows:

$$E(L) = \alpha E_{app}(L) + \delta E_{geom}(L) + \beta E_{edge}(L) + \gamma E_{prior}(L), \tag{6.1.6}$$

   where:

   - $E_{app}(L) = -\sum_{l_i \in V} \log P(x_i^{app}|l_i)$
   - $E_{geom}(L) = -\sum_{l_i \in V} \log P(x_i^{geom}|l_i)$
   - $E_{edge}(L) = -\sum_{(i,j) \in Ed} \log P(l_i, l_j)$
   - $E_{prior}(L) = -\sum_{l_i \in V} \log P(l_i)$

   This definition is motivated by an expression of the conditional probability of the labeling given by:

$$P(L|X) =$$

$$\frac{P(X|L)P(L)}{P(X)} = \frac{P(X^{app}|L)P(X^{geom}|L)P(L)}{P(X)} \simeq$$

$$P(X^{app}|L)P(X^{geom}|L)P(L) \tag{6.1.7}$$

   This expression is given by the fact that the evidence probability $P(X)$ is same for all the different labels and also explains the linear operator at 6.1.6. Since we are interested on the maximum a posteriori estimation, it is enough to take into account only the numerator.

4. The MRF model the energy function, particularly the edge energy $E_{edge}(L)$ is defined as follow:

$$E_{edge}(L) = -\sum_{(i,j) \in Ed} \log P(l_i, l_j) \tag{6.1.8}$$

$$= -\sum_{(i,j) \in Ed} \log \left[P(l_i, l_j | C_{x_i^{app}} = C_{x_j^{app}}) P(C_{x_i^{app}} = C_{x_j^{app}}) \right]$$

$$-\sum_{(i,j) \in Ed} \log \left[P(l_i, l_j | C_{x_i^{app}} \neq C_{x_j^{app}}) P(C_{x_i^{app}} \neq C_{x_j^{app}}) \right]$$

Where $C_{x_i^{app}}$ is a particular metacluster for the x_i^{app} superpixel appearance variable. The joint probability of latent variables is given according to their co-occurence (in adjacent superpixels) in a training data set. The appearance information of superpixels may be an important factor when deciding wether to assign the same label to two adjacent superpixels, i.e., two similar superpixels are more likely to have the same label than two superpixels with different appearance. To achieve this goal, the BoF codewords are clustered once again into metaclusters, two codewords in the same metaclusters are considered to be similar, conversely, two codewords in different metaclusters are considered to be different.

6.1.2 Integrating the MRF model with 3D information

The 3D information of the images can give us valuable information about the object location, avoiding problems such as physically similar superpixels but semantically different, grouped at the same label, this information could represent the depth as a geometric feature from each superpixel on the oversegmentation model.

The new features on the MRF model can improve the label assignment through the redefinition the energy function $E_{edge}(L)$ as follow:

$$E_{edge}(L) = -\sum_{(i,j)\in Ed} \log P(l_i, l_j) = -\sum_{(i,j)\in Ed} \log P(l_i, l_j | x_{i,j}) \tag{6.1.9}$$

$$= -\textstyle\sum_{(i,j)\in Ed} \log \frac{P(x_{i,j}|l_i,l_j)P(l_i,l_j)}{P(x_{i,j})}$$

where $x_{i,j}$ represents the difference in depth between the i and j superpixels.

One problem in this model modification represents the availability of the data, therefore the data construction is one important key for the evaluation. The depth values as well as the RGB image can be obtained through a kinect camera, the ground truth labels must be done manually. Once we achieve the data set, the corresponding experiments would be performed.

6.1.3 MRF Implementation

The computational problem is to find the labels that maximize the posterior probability (Eq. 6.1.2). Recently, different efficient algorithms have been proposed to solve this problem including: graph cuts, loopy belief propagation and tree-re-weighted message passing [50]. In our implementation we used a general algorithm to solve the max-sum problem in graphs based on linear programming [55] which implementation is available at `http://cmp.felk.cvut.cz/cmp/software/maxsum/`.

6.2 Support Vector Machines

6.2.1 Classical SVM

Support Vector Machines is a technique developed by Vapnik and his group at AT&T BELL Laboratories [**?**] that has proven effective classifier in computer vision tasks. For given observations x and classes$y$, we need to find a optimal approximation in order to separate the classes with a set of hyperplanes so as to maximize the margin among them, but as the SVM was designed for binary problems, we must to use the associated strategy one-against-all decomposition to find the most probable classification for each one of the observations x.

The thesis problem consists in the classification of each one of the superpixels, or observable variables x (superpixels), into a set of N labels, or classes y.

The one-against-all strategy decomposed the problem a N-binary decision F_m where $m = 1,..,N$, in which one hyperplane separates one class from all the rest, therefore, the training observation x must be separated according to their corresponding class. Then, in the test phase the classification of a superpixel x is performed according to maximal value of functions $F_m(x)$.

The optimization method for each $F_m(x)$ function consists in defining a hyperplane in the feature space described as the equation $wTx + b = 0$, where b is a scalar. If the training samples are linearly separable, the optimal hyperplanes with no errors and maximum margin are the result of the optimization problem in the equation (6.2.1).

In the other hand nonseparable samples cannot ensure error-free classification, so, the optimization idea can be generalized by introducing the concept of soft margin described in the equation (6.2.2), where ξ_i are called slack variables that are related to the soft margin, and C is the tuning parameter used to balance the margin and the training error.

$$minimize: \ L(w) = \frac{1}{2} \parallel w \parallel^2$$

$$subject\,to: \ y_i(w^T x_i + b) \geq 1, \ i = 1, ..., N. \tag{6.2.1}$$

$$minimize: \ L(w, \xi_i) = \frac{1}{2} \parallel w \parallel^2 + C \sum_{i=1}^{N} \xi_i$$

$$subject\,to: \ y_i(w^T x_i + b) \geq 1 - \xi_i, \ i = 1, ..., N. \tag{6.2.2}$$

To solve both optimization problems can be posed as a constrained quadratic programming (QP) problem. The solution of problem (6.2.1) gives rise to a decision function of the form (6.2.3). The corresponding pairs of x_i entries are known as support vectors and they fully define the decision function. The support vectors are geometrically the points lying near the class boundaries. The problem (6.2.2) has the decision function (6.2.4) where $k(x; x_i)$ is a nonlinear kernel function.

$$f(x) = sgn \left[\sum_{i=1}^{N} y_i \alpha_i (x \cdot x_i) + b \right] \tag{6.2.3}$$

$$f(x) = sgn \left[\sum_{i=1}^{N} y_i \alpha_i k(x \cdot x_i) + b \right] \tag{6.2.4}$$

6.2.2 Sthocastic SVM

The optimization problem can be solved on a iterative manner, it is, finding the optimal solution at each step on the training phase in which just one example is having into account. The optimization process in this case depends on the examples randomly picked at each iteration. Stochastic Gradient Descent (SGD) is an example of this iterative optimization, instead of finding the gradient or optimum for the function (6.2.2), estimates his gradient on the basis of a single randomly picked example x_t in this manner

$$w_t + 1 = w_t - C_t \nabla_w Q(x_t, w_t). \tag{6.2.5}$$

where x_t is a random picked superpixel at the t time. This optimization depends directly on the stochastic process that determines the order of random chosen examples. Due to the large number of handle examples, is expected that the optimization behaves like (6.2.2).

6.2.3 SVM Implementations

In this work we used the classical SVM implementation from 3.20 LibSVM[?], that has into this features:

- Different SVM formulations Efficient multi-class classification.
- Cross validation for model selection Probability estimates.
- Various kernels (including precomputed kernel matrix).
- Weighted SVM for unbalanced data.
- Both C++ and Java sources GUI demonstrating.
- SVM classification and regression.
- Python, R, MATLAB, Perl, Ruby, Weka, Common LISP, CLISP, Haskell, OCaml, LabVIEW, and PHP interfaces. C# .NET code and CUDA extension is available.
- It's also included in some data mining environments: RapidMiner, PCP, and LIONsolver.
- Automatic model selection which can generate contour of cross validation accuracy.

Also, we prove the Stochastic Gradient Descent applied to SVM through the SGD library [?] that:

- Implements a straightforward stochastic gradient descent algorithm.
- The learning rate has the form $\eta_0/(1 + \lambda\eta_0 t)$ where λ is the regularization constant.
- The constant η_0 is determined by performing preliminary experiments on a data subsample.
- The learning rate for the bias is multiplied by 0.01 because this frequently improves the condition number.
- The weights are represented as the product of a scalar and a vector in order to easily implement the weight decay operation that results from the regularization term.

Part III

Experiments and Results

Chapter 7

Road Object Detection

7.1 Cambrige-driving Labeled Video Database

One of the key elements that makes its way to solving the problem of semantic image segmentation is a set of images containing a physical description of the objects that make up the different scenes and the respective association of semantic concepts of objects in each contexts, for that reason the Cambridge-driving Labeled Video Database (CamVid) [9]became one of the datasets candidate for using in our proposal, also it is a online available dataset, each and every one of the containing images are labeled, and one important element to be added, is that the labeled dataset can be useful to finally evaluate existing algorithms quantitatively.

CamVid addresses the problem of video-based object analysis providing data that is labeled with ground truth, it allows to train algorithms that leverage motion cues for recognition, detection, and segmentation. The dataset is a collection of 701 images (with 960×720 pixels) taken with a camera mounted inside a car and filmed over two hours of video footage. The CamVid dataset published is the resulting subset, lasting 22 min, 14 s. A high-definition 3CCD Panasonic HVX200 digital camera was used, capturing 960 x 720 pixel frames at 30 fps (frames per second). The resulting subset were group into three daytime sequences, contains special physical features such as variations in the intensity of light and different locations (roads), and were shot and selected because they contain a variety of semantic object like cars, pedestrians, cyclists and events as moving and stationary cars, cyclists ahead and along side the car, pedestrians crossing, driving through a supermarket parking lot, accelerating and decelerating, left and right turns, navigating roundabouts.

At the end the dataset associates each pixel of each image with one of 32 semantic classes that were identified in the sequences and that could have interest to drivers. A sample of those images with the labels can be seen in Figure 7.1.1, and the color code that were use to label them in Figure 7.1.2.

7.2 Superpixels Extraction

One of the most important issues in the model construction we propose is, what oversegmentation method to use. To to answer the question a test performance evaluation for both oversegmentation methods is done.

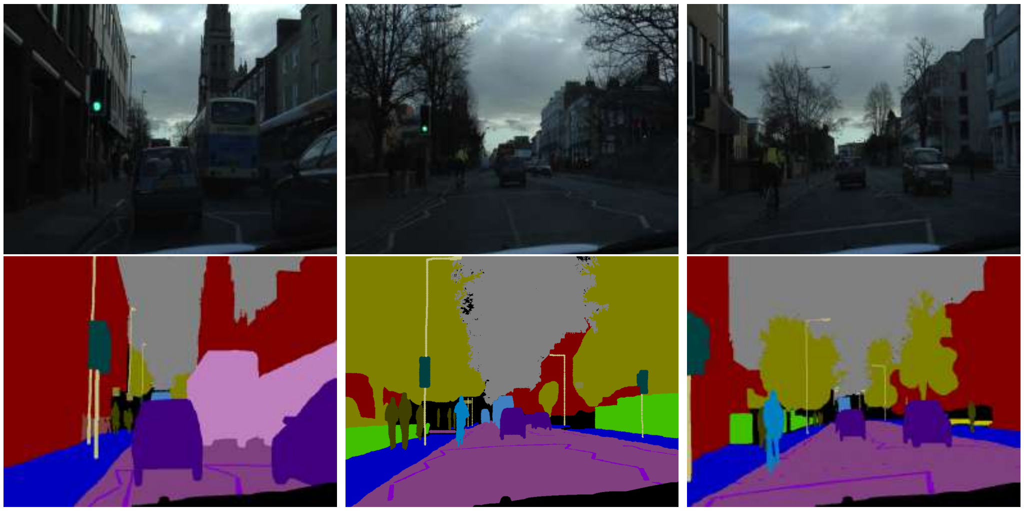
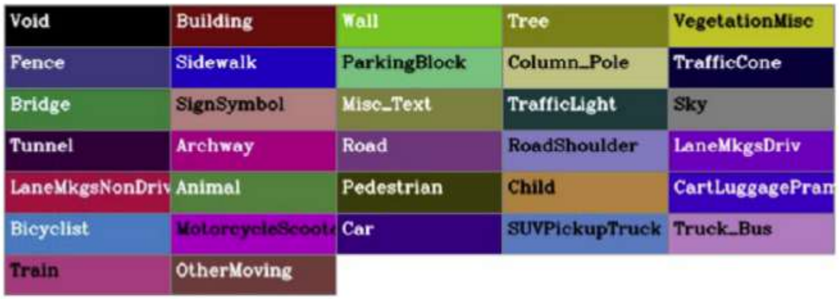


Figure 7.1.1: CamVid aamples



Void	Building	Wall	Tree	VegetationMisc
Fence	Sidewalk	ParkingBlock	Column_Pole	TrafficCone
Bridge	SignSymbol	Misc_Text	TrafficLight	Sky
Tunnel	Archway	Road	RoadShoulder	LaneMkgsDriv
LaneMkgsNonDriv	Animal	Pedestrian	Child	CartLuggagePram
Bicyclist	MotorcycleScoot	Car	SUVPickupTruck	Truck_Bus
Train	OtherMoving			

Figure 7.1.2: CamVid label code color

We are going to compared the two oversegmentation strategies, watershed and SLIC superpixels, this is accomplished by running the algorithms on a subset of images evaluating the output using the following metrics: number of superpixels, execution time, average entropy and average purity of the superpixels.

7.2.1 Performance Measures

Entropy We define entropy as a measure of disorder (inhomogeneity) within each superpixel, so if a superpixel covers only one labeled class, the lowest value of entropy is achieved.

In this context x_{ij} is defined as the number of pixels of class i within the superpixel j, therefore the entropy of the$j$ superpixel is defined as

$$E_j = \frac{1}{N_j} \sum \frac{X_{ij}}{N_j} Log(\frac{X_j}{N_j})$$

where N_j is the total number of pixels inside the j superpixel.

In this vein, the total image entropy is define as the average of the $E_j$values

$$E_T = \frac{1}{N} \sum E_j$$

where N represents the number of superpixels. A more accurate measure could be reached weighting each superpixel entropy according to the size of the corresponding superpixel.

Purity Purity makes referral to the domain percentage of the main class inside a superpixel. The optimal purity is 100%. This measure is highly correlated with entropy. Its main advantage is that it is easier to interpret.

Comparison

7.2.2 Previous Watershed Exploration

The Watershed algorithm has two parameters, the threshold and the number of steps to execute, therefore, to compare the algorithm to SLIC method, we need to explore this to parameters to know how many superpixels are extracted and how them behave.

Test Setup

The design for both tests to be performed consists of a refinement phase, in which experiments are done over one image, in order to evaluate the most significant parameter range, this is the limit in which the model presents the best entropy without a critical increase on the superpixels number or runtime. Once the range is known, we proceed to execute the experiments on 100 images from the database (randomly chosen). Finally, with the quantity results we will make an analysis of dependence between the parameters and performance in order to establish the optimal values for a subsequent use in the MRF building.

Parameter	Interval	Values
Threshold	0-50	0,5,10-. . . 45,50
Steps	10-200	10,20. . . 190,200

Table 7.1: One image experiment parameters

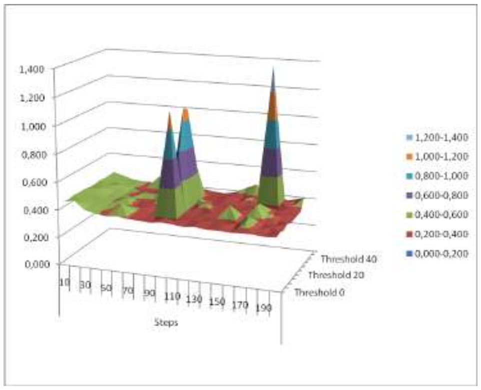


Figure 7.2.1: One image - Laplacian execution time

One Image Analysis

For this experiments we use a randomly chosen image and the interval of the parameters is specified in the Table 7.1.

We considered therefore 11 different values of threshold and 20 different number of steps, leading to 220 results, or oversegmented images. In preliminary execution tests, the entropy time evaluation was defined within 10 and 20 seconds, according to this, the experimental test were planned to run between 36 to 73 minutes.

Watershed Tuning Performance

Oversegmentation Time This test is related to the Watershed execution time. The graphical results are show in the Figure 7.2.1. As we can see, the execution time for every value of the parameters is close to the interval of 0.2 to 0.4 seconds. Some specifically values draw some peaks in the graphic, but these can be seen as outliers.

Number of Superpixels This test is related with the number of entities that the Watershed algorithm provides. The graphical results are show in the Figure 7.2.2. This is a useful result, because give is an concrete idea of the number of superpixels that we are going to get according with the threshold and steps parameters.

Superpixels Entropy This test calculates the average entropy of the superpixels according to the parameters value. The graphical results are show in the Figure 7.2.3. Insofar as we reduce the number of steps, the entropy is dramatically minimized. Meanwhile, the threshold seems to have no relevance on results.

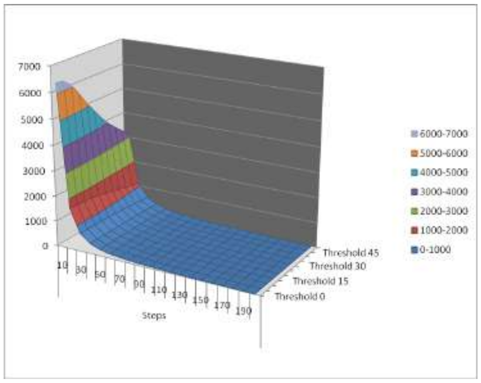


Figure 7.2.2: One image superpixels according with threshold and steps

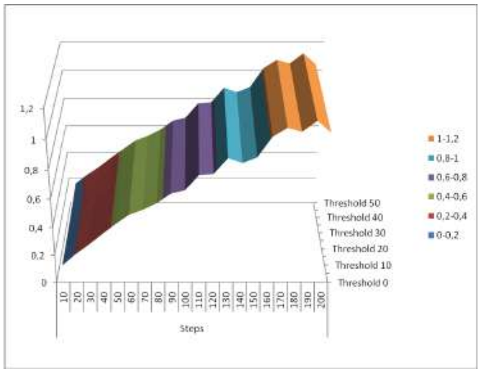


Figure 7.2.3: One image entropy for different threshold and number-of-steps values

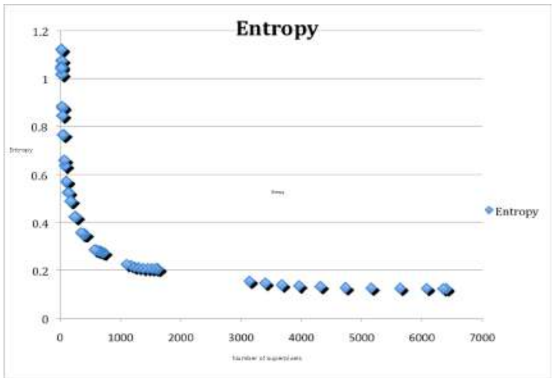


Figure 7.2.4: One image entropy. Each point corresponds to a run of the algorithm with a particular set of parameters.

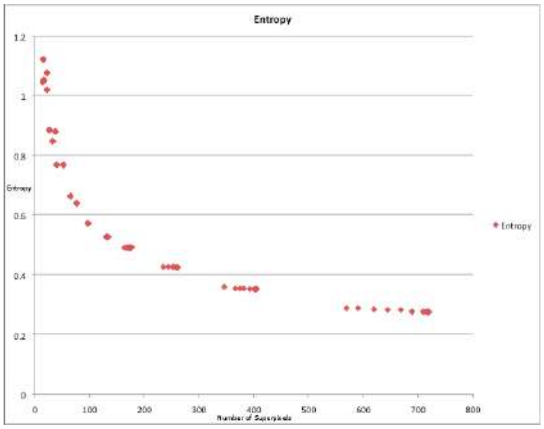


Figure 7.2.5: One image entropy

Relationship between Number of Superpixels and Entropy With the results obtained in previously tests, we can draw the relationship between the number of superpixels and the entropy in Figure 7.2.4. As it was expected, when the number of superpixels grows the value of the entropy decreases. The idea is to find a good compromise between a low entropy value and a low number of pixels. This is accomplished by identifying the graph elbow. In this case, the target points are in the range of 200 to 400 superpixels, which correspond to results generated by running the algorithm with a threshold value of 30 to 50 and 20 to 50 steps.

An overview of the critical points for the previous entropy graphs, define the relationship or qualitative similarity between the appearance of different images. We chose the critical point (16, 1.12), (235, 0.42) and (719, 0.27) that generates the over-segmented images in the Figure 7.2.6.

Is evident, that the mid-point chosen, which is included within the optimal range, better defines the objects inside the image without generating a significant increase of entities, therefore, we can ensure that the chosen interval is appropriate.

The second test highlights the dependency of the number of superpixels on threshold parameter,

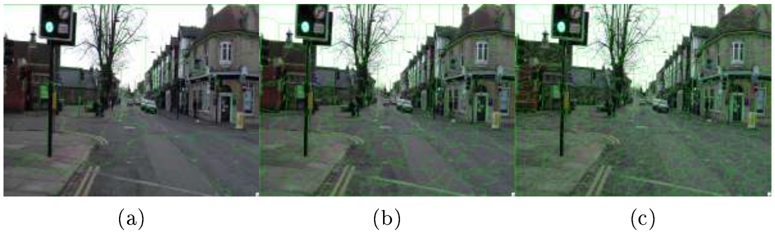


Figure 7.2.6: One image qualitative analysis. The Images were generated with a) 190 steps and 0 as threshold b)50 steps and 50 as threshold c) 30 steps and 10 as threshold

because for large values of the steps parameter, the number of superpixels becomes stable. Thus the test validity establishing the range for 0 to 50 steps. The entropy evaluated in the third experiment shows the steps parameter dependence, because the threshold shift in the axis are constant. Another notable conclusion in the graphic is that as smaller number of steps, smaller is the entropy value obtained, in this way, the steps becomes a defining parameter.

One Hundred Image Analysis

According to the one-image analysis, for this experiment we take 3 threshold values (30, 35 and 40), and 7 different values for the number-of-steps (20, 30, 40, 50, 80, 90 and 100). These combination of parameter values produces 12 different output for each image. Because we provide 100 images to the algorithm, the procedure will generate 12,000 results equivalent to 3.3 to 6.6 hours of total run time.

Characteristics Performance The characteristics to evaluate are the same than in the previous experiment. The summary of each of these instances is given in Table 7.1.

Experimental Analysis The overall behavior of the experiments shows the same trend as for a single image test, upholding the approach to global optimal algorithm parameters.

Performing the analysis of the entropy versus superpixels graph in Figure 7.2.7 a optimal elbow of the performance function is specified, which generates a gain in entropy without critically increasing the complexity of the model. This point is near to 600 superpixels and an entropy value of 0.35 that is equivalent to 20 steps and 30 or 35 threshold value.

Conclusions

The choice of suitable parameters for the Watershed algorithm optimization requires an analysis that includes the value of the computational time, the model complexity (number of superpixels) and generally good performance of the model, in a such way that the balance between these aspects becomes essential in order to establish the appropriate environment for the construction of a probabilistic segmentation model.

Steps	Threshold	Entropy	Purity	# Superpixels
20	30	0.35	0.93	603.75
20	35	0.40	0.93	538.91
20	40	0.45	0.93	482.91
30	30	0.39	0.92	333.95
30	35	0.44	0.92	301.71
30	40	0.48	0.91	274.59
40	30	0.43	0.90	215.84
40	35	0.47	0.90	197.46
40	40	0.50	0.90	181.28
50	30	0.44	0.89	167.59
50	35	0.47	0.89	154.59
50	40	0.50	0.89	143.36
80	30	0.58	0.85	75.55
80	35	0.61	0.86	70.80
80	40	0.65	0.86	66.32
90	30	0.62	0.85	63.5
90	35	0.65	0.85	60.00
90	40	0.68	0.85	56.66
100	30	0.60	0.86	60.37
100	35	0.63	0.86	56.99
100	40	0.65	0.86	53.80

Table 7.2: 100 images segmentation results for different parameter values. Each metric corresponds to the average over the 100 different results.

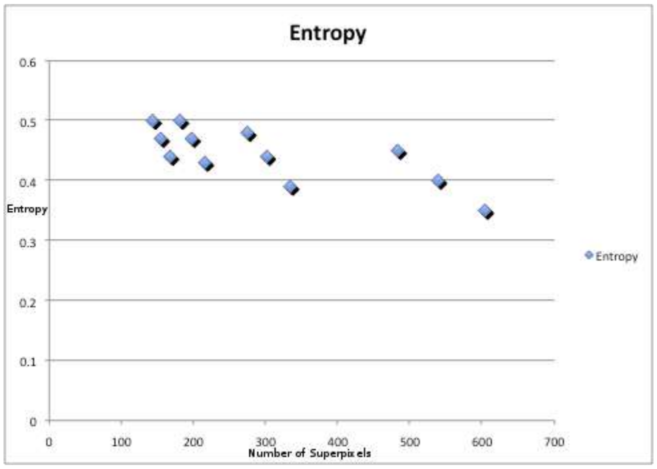


Figure 7.2.7: All images entropy

# Superpixels	Watershed			SLIC		
	Entropy	Purity	Execution Time	Entropy	Purity	Execution Time
300	0.44	0.92	3.0seg	0.56	0.92	2.5 seg
400	0.45	0.93	3.25seg	0.48	0.94	3.0 seg
600	0.35	0.93	5.0seg	0.40	0.91	4.0 seg

Table 7.3: Oversegmentation comparison

7.2.3 Oversegmentation Comparison

Now that we have a Watershed analysis of a good number of superpixels to take into account, we compare the two oversegmentation strategies on a 10 randomly chosen dataset. The results can the seen in Table 7.3.

As the results show, the entropy and purity of the superpixels are comparable, but SLIC superpixels outperforms the extraction time of the Watershed method. This may become a critical feature when we are dealing with large data set, or applications requiring data processing in real time.

7.3 Experimental Results

7.3.1 Experimental Setup

The images of the Cambridge-driving Labeled Database were divided in a training data set with 367 images and a test dataset with 233 images (101 images were not used), following the same experimental setup as the one used in [9].

The experimental results are separate into two phases, at first we perform a parameter tuning over a test data set, achieving the best values for alpha, betha, gamma and delta parameters, later we run a final evaluation experiment over a validation image set for achieving the accuracy for each one of the semantic classes, obtaining in this way a conclusive global accuracy value. This process is applied for the three proposed MRF models, MRF with appearance information, MRF with additional geometrical values, and MRF with a modified label join probability.

7.3.2 Results

Superpixel extraction and representation

Approximately, 3×10^5 watershed superpixels were extracted from the training image data set. This set of superpixels was broken again in training and test sets with 10^5 and 2×10^5 superpixels respectively. These sets were used to tune-up the relative weight of the color components in the superpixel descriptor (3 color components vs. 128 SIFT features). The impact of the color weight in the codebook construction is evaluated using the average codebook word entropy and accuracy. The entropy of a codebook word w corresponds to the entropy of the label distribution given the codebook word: $P(l = \lambda_i|w)$. The ideal case correspond to a word with entropy equal to 0, i.e., a word that uniquely determines a label. To calculate the accuracy, each word is used as a predictor of the superpixel label, assigning the label with the highest conditional probability. According

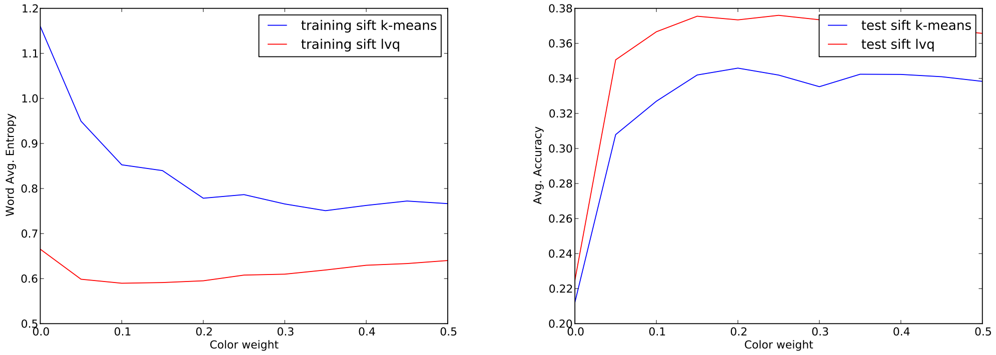


Figure 7.3.1: Performance of LVQ and k-means for BOF codebook construction. Entropy (left) and accuracy (right).

to a parameter analysis on the codebook construction, the color information contributes to the discriminative power of words.

BOF codebook learning

An important part of the codebook construction is the algorithm used to find the most characteristic codewords from a set of superpixels descriptors. The most common alternative in BOF representation is k-means. However, it ignores the label information that may help to build a more discriminative codebook. We used Learning Vector Quantization (LVQ), which uses the label information to guide the codebook learning process, as an alternative to k-means. Figure 7.3.1 shows that LVQ performs better reducing the entropy and increasing the accuracy of the resulting codebook. Entropy was evaluated on the training data set and accuracy on the test data set.

These tuning results show the importance for leaning concepts (classes) inside the semantic segmentation task e.g. learning of labels and using the split and merge procedure to defined representative clusters and therefore decreasing the distortion through the LVQ algorithm. K-means results by the way are relative to initial centroids and can be skewed towards at not semantic sense clusters.

Semantic segmentation using appearance

The first set of segmentation experiments were performed without taking into account the geometric information of the superpixels, i.e., the δ coefficient in Eq. 6.1.6 was set to zero (actually the first version of the segmentation program didn't take into account geometric information). The α parameter was kept equal to 1 and the other parameters were varied. Figures 7.3.2 and 7.3.3 show the performance, in terms of annotation accuracy, for training and test data sets respectively. In training data, the maximum performance is reached when only the appearance component of the energy is taken into account ($\alpha = 1, \beta = \gamma = \delta = 0$), this has to do with the fact that the appearance conditional probability was learned using the training data set, so the performance on this data set is expected to be good, but it doesn't has to be the case for test data. In fact, for test

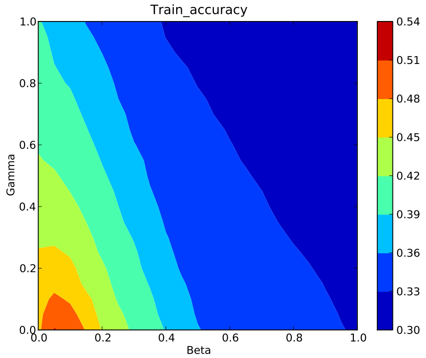


Figure 7.3.2: Parameter tuning on training images. The beta and gamma parameters are varied while keeping alpha=1.

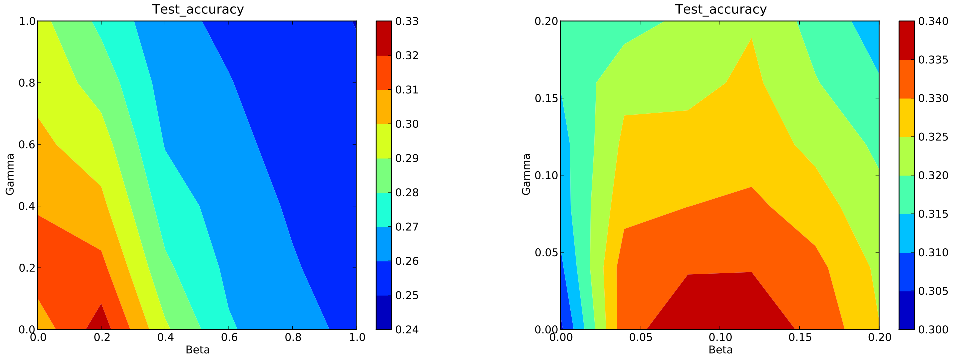


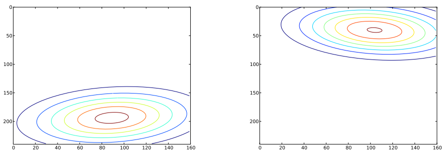
Figure 7.3.3: Parameter tuning on test images. The beta and gamma parameters are varied while keeping alpha=1. The right image shows a detail of the region where the maximum accuracy is.

data the joint label probability associated to labels (controlled by β) is important to increase the performance from 30% of accuracy to 34%, Figure 7.3.3.

Semantic segmentation using appearance and geometrical information

The proposed algorithm involved geometric and appearance information, therefore the coordinate of the morphological center of each superpixel[1] , exactly where the superpixel SIFT descriptor is calculated, was recorded and a probability distribution was estimated independently for each label class. The distribution used was a bivariate Gaussian. The vertical axis symmetry of the images was exploited to make the estimation problem easier. The original x coordinate that was in the

[1] The 'morphological center' is defined as the maximum of a distance map calculated in the superpixel with respect to the superpixel boundary. It is possible that there are more than one pixel with the maximum distance, in this case the tie is broken arbitrarily choosing on of the maximum. The ' morphological center' is the point of the superpixel which is at maximum distance of any boundary pixel.



(a) Class Road (b) Class Sky

Figure 7.3.4: Geometric distributions for different label classes.

range $[0, 320)$ was mapped to the range $[0, 159)$ by applying the transformation $x' = \min(x, 319 - x)$.

Figure 7.3.4 shows a plot of the estimated probability density function for two sample semantic classes, the sky and road. It is easy to see that those are classes that are well localized, i.e., some classes are more likely to appear on some particular areas of the images. Figure 7.3.5 shows how increasing the importance of the geometric information improve the performance of the segmentation system.

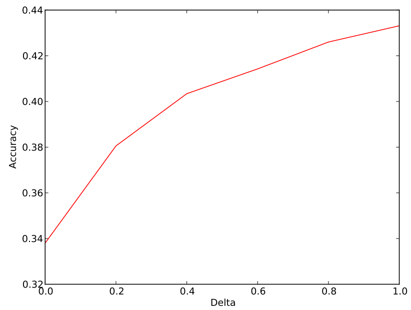


Figure 7.3.5: Accuracy in the test set for a MRF model that takes into account geometry in addition to appearance. The Delta (δ) parameter controls the importance of the geometrical information.

Same as before, the parameters of the model are tuned up using the test data set. The best values found were $\alpha = 0.6$, $\beta = 0.04$, $\gamma = 0$, and $\delta = 0.8$.

At same way the label joint probability was re-defined and put to the test for tuning, taking into acount metacluster sizes of 250, 500, 750, 1000 and 1500, and specific exploration of 600 and 850.The best results achieved on the test data set were $\alpha = 0.8$, $\beta = 0.04$, $\gamma = 0$, and $\delta = 1$ for a metacluster size of 750 as can be showed at Figure 7.3.6. This result implying an improvement in the accuracy values, highlighting the importance of the label co-occurence.

Final Evaluation

Performing the experiments into the validation test we obtained the following results. The confusion matrix of the semantic categories of the data set are given in Table7.4 where MRF with appearence and geometrical information were characterized. All these values are summarized in Table7.5, where accuracy comparison is made against the representative algorithms in the literature: a baseline

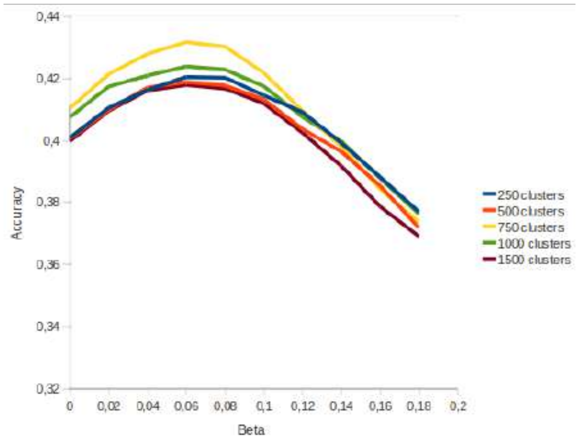


Figure 7.3.6: Accuracy in the test set for a MRF varying the Beta (β) parameter which controls the importance of the edge energy, which is related to the join label probability, and defining some metacluster sizes.

based on SVM, co-occurence labels and superpixels [?], among others.

7.3.3 Discussion

The first important result is that no one of the methods achieve a significant accuracy for all classes. This indicate that the problem is hard and that further research is required. Second, the experiment reflects the importance of geometric information, obtaining an improvement of up to 33% for the method in [1], 10% in [?] and 24% in our model. Some of the results are biased by the kind of geometric descriptors, Micusik, for example, makes use of 3D information provided inferred from the video stream, while our model focuses on 2D information, therefore one of the improvements to follow is the integration of 3D and even 4D information in continuous video, as is the case for the CamVid data set. Third, we can see that in some specific classes the precision is high, and all methods perform competitively, such as sky, road and sidewalk. In some others, the performance varies disproportionately as fence, sign and column, and in the last ones the achieved accuracy is considerably low as in bicyclist. This behavior is derived from the way each one of the entities of the problem is described and how they are related. Finally, although our model does not produce the best results in all classes, actually only in sky, the proposed method produced encouraging results, which are worthy of further research.

	Real										
	bicyclist	building	car	column	fence	pedestrian	road	sidewalk	sign	sky	tree
bicyclist	23,00%	4,10%	16,80%	2,20%	5,10%	24,10%	8,90%	12,50%	2,90%	0,00%	0,40%
building	2,00%	55,40%	4,20%	4,40%	6,00%	5,60%	0,30%	1,80%	2,60%	4,70%	13,00%
car	7,70%	10,10%	40,00%	3,00%	5,60%	12,90%	7,10%	10,60%	2,10%	0,10%	0,80%
Column-pole	5,90%	36,00%	4,60%	7,80%	8,80%	12,50%	1,40%	5,40%	2,70%	8,60%	6,30%
fence	6,00%	17,10%	12,10%	3,60%	26,70%	16,00%	2,30%	10,90%	3,50%	0,00%	1,70%
pedestrian	14,60%	8,90%	11,30%	5,70%	11,10%	35,10%	2,50%	9,30%	1,30%	0,00%	0,20%
road	1,30%	0,30%	3,30%	0,30%	0,40%	0,80%	79,80%	13,70%	0,30%	0,00%	0,00%
sidewalk	3,30%	2,10%	2,50%	1,00%	1,80%	3,40%	11,50%	74,20%	0,20%	0,00%	0,10%
Sign-symbol	2,80%	42,80%	5,90%	9,20%	8,70%	10,70%	0,20%	1,60%	7,70%	1,10%	9,40%
sky	0,00%	0,90%	0,30%	1,10%	0,00%	0,00%	0,00%	0,00%	0,30%	94,60%	2,80%
tree	3,00%	27,50%	3,00%	11,60%	7,50%	6,70%	0,10%	0,60%	4,70%	9,60%	25,70%

Table 7.4: Confusion matrix for extended MRF model using both appearance and geometric information on the test set.

	bicyclist	building	car	column	fence	pedestrian	road	sidewalk	sign	sky	[illegible]
Appearance + Geometry											
Our MRF Geom+App	23%	55.4%	40%	7,8%	26,7%	35,1%	79,8%	74,2%	7,7%	94,6%	25,7%
Textoonboost + SIM	22.5%	46.1%	68.6%	0.7%	46.6%	53.6%	89.5%	60.5%	42.9%	89.7%	61.9%
Micusik co-occ & wsshedLoG splxs	28.8%	71.71%	76.5%	11.4%	4.8%	59.1%	88.4%	84.7%	12.5%	89.5%	56.1%
Appearance only											
Our MRF Appearance	13,4%	58%	27,1%	0,9%	3,2%	13,8%	77%	50,3%	5,7%	94,5%	27%
Textonboost Model	23.6%	38.7%	71.1%	1.1%	40.1%	54.6%	88.6%	55.5%	51.4%	90.1%	60.7%
Micusik co-occ & wsshedLoG spxls	32.3%	66.1%	70.8%	18.1%	3.1%	49.3%	84%	79.2%	9.4%	88.2%	62.6%

Table 7.5: Per-class and average accuracy for different methods. The first two lines correspond to the proposed method. Th third one, is a baseline method that ignores structure and classifies each superpixel independently using a SVM. The last five lines, correspond to methods and results reported by [?].

Chapter 8

Human Gait Analysis

The overall body part classification process is described in Figure 8.3.1. The sequential diagram describes the phases that composed the classification task; The first phase makes allusion of the image pre-processing that is discussed at Section8.1. This step provide us a set of superpixels from the training dataset, we can chose to have not into account the void class inside the classificator, through the background substraction, establishing the superpixels within the human silhouette as the only entities to classify. The background substraction is performing by the arithmetic subtraction of the depth image with respect to a generic image from the background plus and slack variable.

The second phase, performs the superpixels characterization. With this information the third phase proceeds to the superpixels dataset building, that assigns to each characterized superpixel a semantic class, in this case on of the eight body parts, in this manner we have a consistent and usable training dataset.

The final phase is about the leaning model construction, whereby, new image samples without labeling manual will be segmented.

8.1 Data Acquisition and Processing: Human Body Parts Database

Human Motion Analysis is an important research area in several applications such as video surveillance, image retrieval systems, human interaction, and medical diagnostics. The problem that arises in this type of analysis is the segmentation of the human body in different parts of the body including head, limbs, torso and feet. This problem is challenging due to the occlusion of body parts, variability on the appearance on the images such as clothes with similar colors and finally the image perspective.

To test the robustness of our image segmentation framework, we decided to apply it to the human body parts recognition problem. To this end, we constructed a dataset comprising image sequences of frontal human gait, which describe the natural movement of different persons over time. The sequences were acquired through a Kinect device, that con only provides RGB images but depth mapping, that can be useful to the extent that allows us to get more information on the scene to assess.

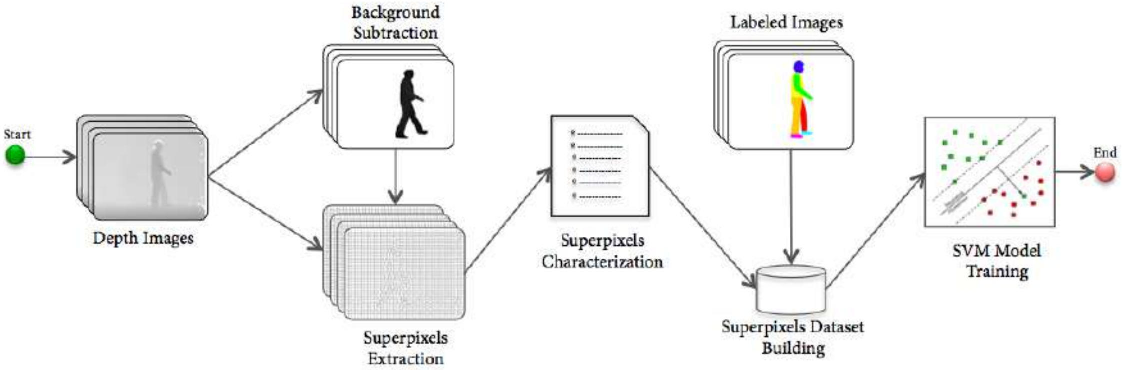


Figure 8.0.1: Semantic segmentation process.

8.1.1 Kinect Sensor Device

The Microsoft Kinect device was originally realeased for allowing to users play video games through only the movement of their body, this is achieved through an RGB camera, an infrared depth sensor, an accelerometer, four microphones and a motor to adjust the tilt.

This gaming device not only attracted the gaming and commercial attention but the scientific one, because it integrates different acquisition features, its low price and its shelf availability.

The infrared sensor helps to build the depth map by analyzing a speckle pattern of infrared laser light combined with the monochrome CMOS sensor, it is done through structured light general principle that project a known pattern onto the scene and infer depth from the deformation of that pattern, so, the calibration between the projector and the camera has to be known.

Each one of the depth values of the images can be represented as a 3D coordinate or a grayscale value. Kinect software is capable of automatically calibrating the sensor based the physical environment, accommodating for the presence of furniture or other obstacles.

The Kinect sensor outputs the RGB video at a frame rate of 30 Hz. with VGA resolution (640 × 480 pixels) with a Bayer color filter, meanwhile the depth sensing video stream with same resolution and 11-bit depth.

8.1.2 Image Acquisition

The use of kinect, device composed by two depth sensor and an RGB camera, not only reduces the image acquisition cost on the specific gait analysis problem, but it ensures the portability of a working tool that intends to provide gait laboratory service.

The Depth Gait Dataset we build is composed by 368 diverse gait movements of eight different persons, and was acquired using the Kinect device. The dataset is composed by RGB images and the corresponding depth scanning, this last represented by a grayscale image. Both images have a resolution of 640x480 pixels, but because of Kinect device has an intrinsic problem of calibration between the RGB image and its corresponding depth map, it is the appropriate the data preprocessing, setting a single 589x442 pixels resolution.

The setup of the image acquisition is illustrated in Figure 8.1.1. Each one of the nine persons walked across a runway without obstacles, sagital side from the camera, on a gait cycle of approxi-

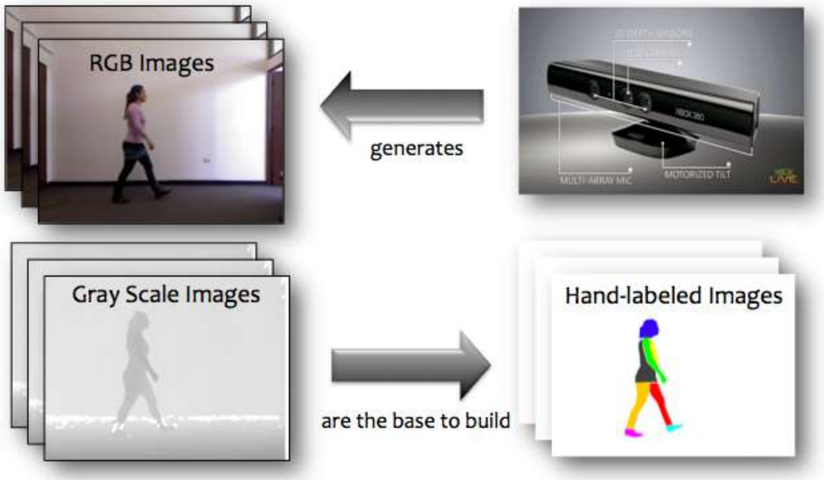


Figure 8.1.1: Image Acquisition

mately four minutes in duration. The persons that were chosen have different characteristics such as weight, body volume, texture and color of the clothes. In total were taken five women and four men gaits rightward and leftward.

The rightward sequences labeled into nine different body parts, among these are the head, limbs, arms, front and back torso and feet. The choice of these body parts is due to the support they provide in the prediction of the human body joints. The segments together with the coordinates of the joints give way to human gait analysis.

8.1.3 Hand-Labeling Process

Based on each pair of images was carried out the labeling process, sets out twelve labels for specific body parts (head, torso, left and right arm, left leg and right, left and right foot) that define target points for gait analysis. The labeling process was performed manually with the guidance of Layer segmentation tool [?]. Some examples of the data set are found in Figure 8.1.2.

8.1.4 Joints Labeling

We use The Human Annotation Tool (HAT) [Reference http://www.eecs.berkeley.edu/~lbourdev/hat/] that allows us to annotate the body structure, for example joints and their spacial relationship, thereby defining an outline 3D position of the person in a given image. HAT was developed in java and provides an applet interface in which the user can annotate each one of the joints, building in real time the 3D skeletal position. One example of the process can be seen in Figure 8.1.3.

The result of these annotation process are a list of XML files, one for each image, containing the coordinates of 13 joints, these are head, shoulders, elbows, wrists, hips, knees, ankle and joint of the big toe.

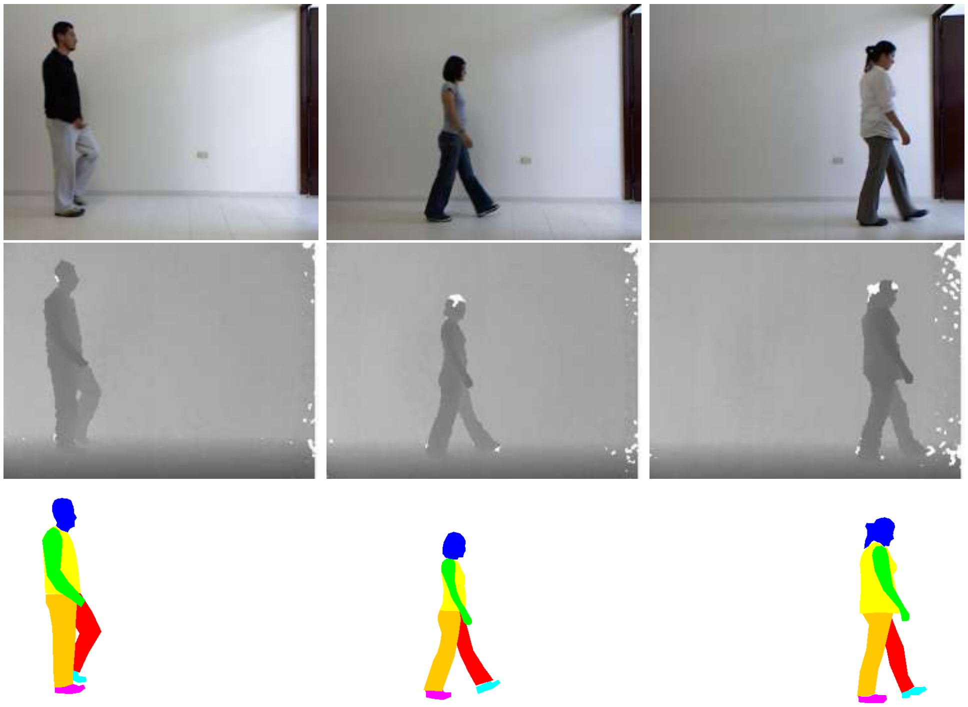
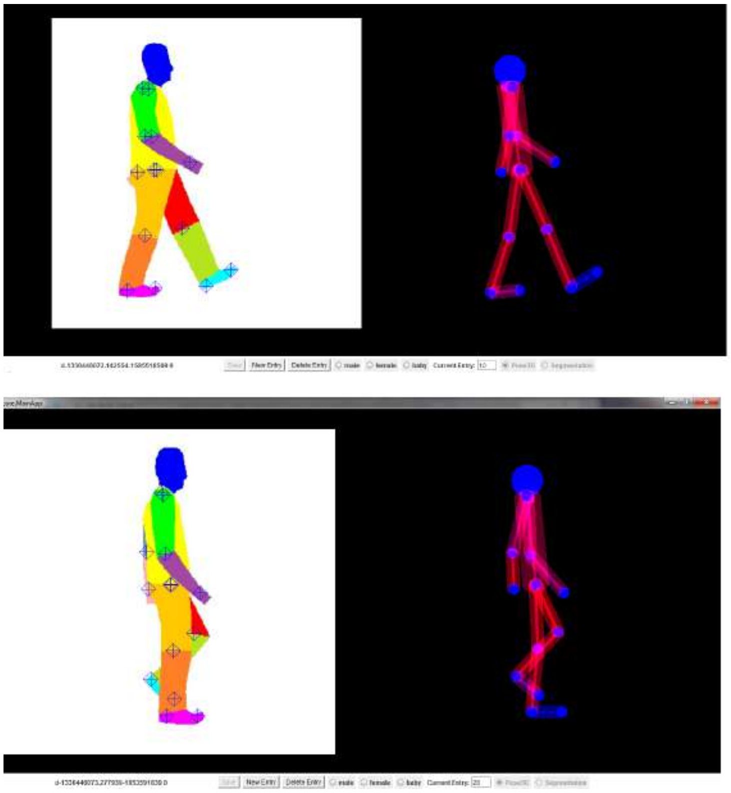


Figure 8.1.2: Image dataset samples



Current Entry: 10 ● Pose3D ○ Segmentation

Current Entry: 20 ● Pose3D ○ Segmentation

Figure 8.1.3: Human Annotation Tool on Gait Images

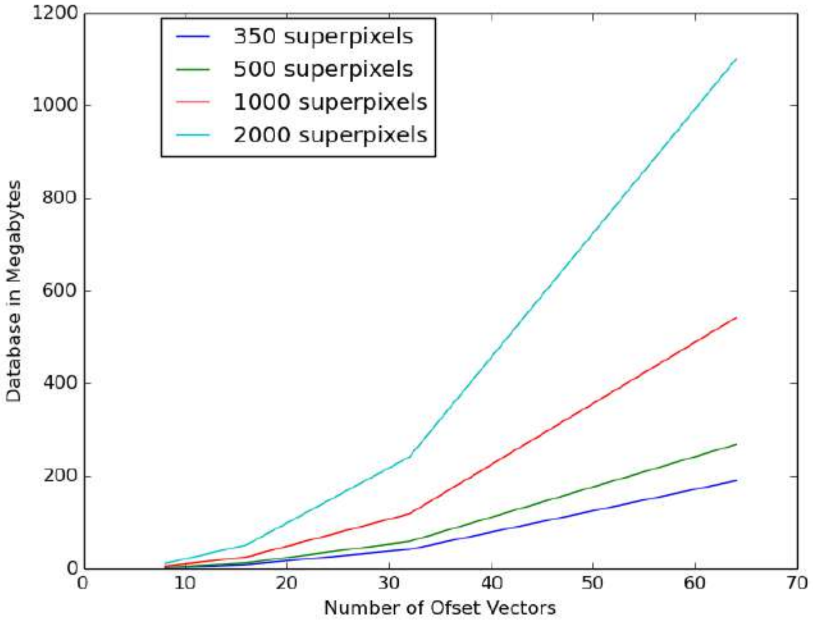


Figure 8.2.1: Growth of database

8.2 Database Size

Once the images that compose the dataset are organized and labeled, the next steps in our methodology is, first to set the number of SLIC superpixels to extract per image, and second, to explore the parameters of the feature extraction algorithm.

By choosing as superpixels descriptors the characteristics described in Section 5.2, the parameters to consider are the number of offset vectors and their possible combinations.

Variations on the number of offset vectors and superpixels to extract, lead to an increase or reduce the size of train and test of the database. Thus, it is need to perform a comparative analysis of the number of vectors and superpixels against the size of the database.

Taking into account the analysis made Section 7.2, which identifies that a number of between 350 and 2000 superpixels, guarantee stable values of entropy within the dataset, so it is explored within this range. Regarding offset vectors, and considering the restriction to be made about these in a neighborhood of 8, the possible values evaluated are 8, 16, 32 and 64, a large value produces a big number of features per superpixels, this is not well handled by the SVM classifying to use.

The exponential growth in the size of the data set is shown in figure 8.2.1.

8.3 Experimental Results

8.3.1 Preliminary Tuning

As was said in previously section, we have to take into consideration the number of pixels, the numbers of offset vectors, and additionally, the length of the vectors. So it is necessary to make a prior exploration of these parameters, to establish a set of the most successful experiment.

Therefore we made the parameter exploration over the training dataset (described below), it is done performing 10-fold cross validation, for each of 10 experiments, use 9 folds for training and the remaining one for testing . We take into account 100.000 samples per class. The results are presented in Table X.

8.3.2 Experimental Setup

For the experimental setup the dataset was divided into training and testing items, selecting 326 images for training and the remaining 41 images for testing. The parameters taken into account in the experiment are the amount of superpixels by image and the number of features or offset vectors. As performance metrics of the classification we use the accuracy of each of the parts of the human body through the confusion matrix between the ground truth labeling and the most probable label assignment given by the classifier.

The number of offset vectors that we take into account are 32 and 64, in a neighborhood of about 10 pixels, that defines 496 and 2016 features. The length of the vector is set in 3000 and 5000. At same way we extracted 350 superpixels per image.

8.3.3 Results

The accuracy obtained for each of the explored classification methods is summarized on the Table 8.2. This performance measure can be described in detail by the per class accuracy as shown in Table 8.4. The confusion matrix of the results are show in Table8.5.

Finally the performance of the methods can be measure through the computation time of the training and testing phase, this values are listed in Table 8.3.

8.3.4 Discussion

According to the images on Figure 8.3.1, we can conclude that the segments are correctly placed, and in a qualitative manner the right and left differentiation, missing on[?], is being correctlyaddressed, since the limbs are not confused.

#offset vectors	8						16						32			
#sp \ vector length	1000	2000	3000	5000	7000	10000	1000	2000	3000	5000	7000	10000	1000	2000	3000	5000
350	91.47%	86.23%	87.43%	86.11%	90.08%	93.06%	87.20%	89.69%	91.40%	94.69%	95.29%	95.05%	92.99%	95.70%	96.59%	96.84%
500	88.54%	81.45%	84.17%	82.11%	86.35%	90.72%	83.73%	87.09%	89.14%	92.81%	93.62%	93.31%	90.83%	94.22%	95.43%	95.74%
1000	87.24%	79.70%	85.78%	80.15%	79.05%	70.38%	75.44%	79.25%	82.78%	88.01%	89.27%	88.63%	84.68%	89.97%	92.27%	92.49%
2000	85.11%	76.70%	83.48%	81.15%	77.13%	75.87%	66.42%	70.00%	73.50%	81.61%	84.04%	82.89%	77.29%	84.81%	88.19%	88.73%

Table 8.1: Global Class Accuracy - Parameter Tuning

		# Offset Vectors			
		32		64	
Vector Length		3000	5000	3000	5000
Method	Stochastic SVM	96.6795%	97.0723%	97.9914%	97.9247%

Table 8.2: Performance comparison of methods through accuracy measurement on the testing phase, according to the number of superpixels per image

	Maximun Training Samples per Class							
Method	1000		5000		10000		100000	
	Training	Testing	Training	Testing	Training	Testing	Training	Testing
Stochastic SVM	0.5min	3seg	1.6min	5seg	3min	10seg	4min	15seg

Table 8.3: Performance comparison of methods through time measurement on the training phase.

	Classes								
Method	Head	Right Arm	Left Arm	Front Torso	Right Leg	Left Leg	Right Foot	Left Foot	Void
Stochastic SVM	89%	75%	33%	86%	81%	69%	71%	52%	100%

Table 8.4: Per class accuracy for the best average testing accuracy

	Real								
	Head	Right Arm	Left Arm	Front Torso	Right Leg	Left Leg	Right Foot	Left Foot	Void
Head	89.0%	1.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	10.0%
Right Arm	0.0%	75.0%	0.0%	9.0%	5.0%	1.0%	0.0%	0.0%	11.0%
Left Arm	0.0%	0.0%	33.0%	0.0%	0.0%	0.0%	0.0%	0.0%	67.0%
Front Torso	1.0%	6.0%	0.0%	86.0%	1.0%	0.0%	0.0%	0.0%	7.0%
Right Leg	0.0%	0.0%	0.0%	1.0%	81.0%	7.0%	0.0%	0.0%	11.0%
Left Leg	0.0%	0.0%	0.0%	1.0%	13.0%	69.0%	0.0%	1.0%	17.0%
Right Foot	0.0%	0.0%	0.0%	0.0%	4.0%	0.0%	71.0%	0.0%	25.0%
Left Foot	0.0%	0.0%	0.0%	0.0%	4.0%	4.0%	4.0%	52.0%	37.0%
Void	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	100%

Table 8.5: Image Gait Analysis Confusion Matrix

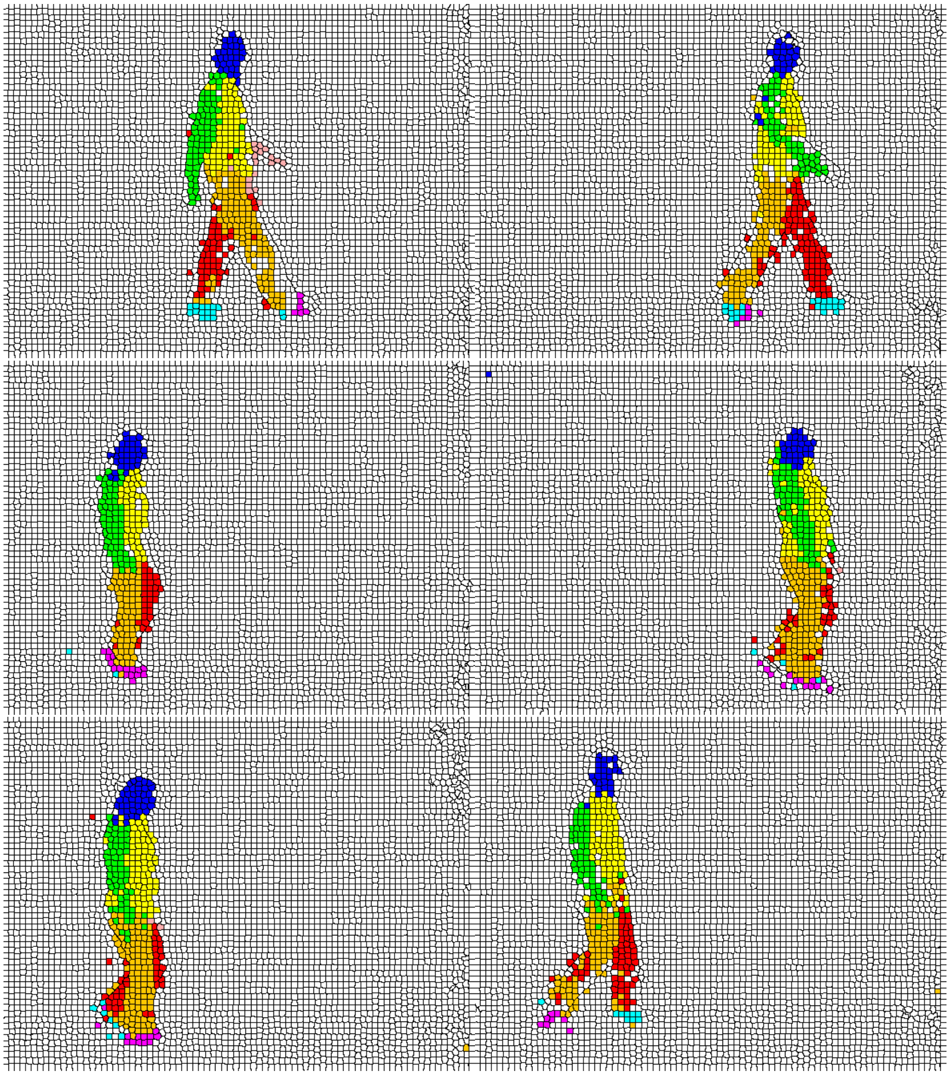


Figure 8.3.1: Segmented images

Chapter 9

Conclusions

This thesis has presented a strategy to address the problem of image understanding. This thesis focused on image segmentation by using machine learning and image processing techniques. The proposed methods were evaluated with performance measures, which allowed to determinate the performance of such methods in an objective way. This thesis evaluated the proposed strategy in different contexts such as Road Object Detection and Human Body Parts Identification.

Results show that depth invariant features extraction that allows the SVM model construction, show very good results in Body Parts Identification. At same way the MRF model outperforms the road object identification according to the SVM base line. The main contribution of this research work is a the MRF model construction, and the body identification strategy. The following subsections discuss different aspects of the addressed problems and of the strategies used to tackle them.

9.1 Road Detection

A well defined joint label probability along with a quality geometrical information, has been shown be the first step in a competitive segmentation method definition, at same way that the representation and description of the image not only facilitate the modeling of the problem but to establish an efficient labeling process extensible to several data sets.

We have presented a MRF model that achieve promising results and offers a robust framework to include some more specifically information, that derives an effective way to segment an image. As part of future work are exploration of alternative oversegmentation strategies, geometrical information improvement, through 3D information integration (and even 4D), analysis for new ways to define the neighborhood graph and probability definitions inside our MRF Model.

9.2 Human Gait Analysis

The work has proposed a SVM and SGDSVM methodologies for human body segmentation into different body parts, that have been shown be the a competitive segmentation methods. We obtained average accuracy values over 97%, for this reason any of the two methods solves the semantic segmentation problem satisfactorily. If we need to achieve high computational performance on training and testing phase, it is better to use the SGDSVM approximation. On the other hand if we need to take the highest values of accuracy without taking into account the computational cost, SVM provides better support.

Bibliography

[1] *Semantic texton forests for image categorization and segmentation*, 2008.

[2] Radhakrishna Achanta, Appu Shaji, Kevin Smith, Aurelien Lucchi, Pascal Fua, and Sabine Süsstrunk. SLIC superpixels compared to state-of-the-art superpixel methods. *IEEE transactions on pattern analysis and machine intelligence*, 34(11):2274–82, November 2012.

[3] J.T. Allen and T. Huntsberger. Comparing color edge detection and segmentation methods. In *Proceedings. IEEE Energy and Information Technologies in the Southeast'*, pages 722–728. IEEE, 1989.

[4] Amina Asghar and Naveed Iqbal Rao. Semantics sensitive segmentation and annotation of natural images. In *SITIS 2008 - Proceedings of the 4th International Conference on Signal Image Technology and Internet Based Systems*, pages 387–394, Washington, DC, USA, 2008. IEEE Computer Society.

[5] Thanos Athanasiadis, Phivos Mylonas, Yannis Avrithis, and Stefanos Kollias. Semantic image segmentation and object labeling. *IEEE Transactions on Circuits and Systems for Video Technology*, 17(3):298–311, 2007.

[6] Omar Attum, Perri Eason, Gary Cobbs, and Sherif M. Baha El Din. Response of a desert lizard community to habitat degradation: Do ideas about habitat specialists/generalists hold? *Biological Conservation*, 133(1):52–62, 2006.

[7] J.-M. Beaulieu. Versatile And Efficient Hierarchical Clustering For Picture Segmentation. In *10th Annual International Symposium on Geoscience and Remote Sensing*, pages 1663–1663. IEEE, 1990.

[8] a Blake. Real-time human pose recognitiom in parts from single depth images. *ï¿œï¿œIEEE Conf. on Computer Vision and Pattern Recognition (CVPR)ï¿œï¿œ*, 2:1297–1304, 2011.

[9] Gabriel J. Brostow, Julien Fauqueur, and Roberto Cipolla. Semantic object classes in video: A high-definition ground truth database. *Pattern Recognition Letters*, 30(2):88–97, January 2009.

[10] Nicoletta Calzolari, Charles J Fillmore, Ralph Grishman, Nancy Ide, Alessandro Lenci, Catherine Macleod, Antonio Zampolli, Linguistica Computazionale, Università Pisa, and Dipartimento Linguistica. Towards Best Practice for Multiword Expressions in Computational Lexicons. *Lrec'02*, pages 1934–1940, 2002.

[11] a. Cardinali and G.P. Nason. A statistical multiscale approach to image segmentation and fusion. In *2005 7th International Conference on Information Fusion*, volume 1, page 8 pp. IEEE, 2005.

[12] Chuan-Yu Chang, Hung-Jen Wang, and Chi-Fang Li. Semantic analysis of real-world images using support vector machine. *Expert Systems with Applications*, 36(7):10560–10569, 2009.

[13] F. Cohen. Adaptive hierarchical algorithm for accurate image segmentation. In *ICASSP '85. IEEE International Conference on Acoustics, Speech, and Signal Processing*, volume 10, pages 897–900. Institute of Electrical and Electronics Engineers, April 1985.

[14] G.B. Coleman and H.C. Andrews. Image segmentation by clustering. *Proceedings of the IEEE*, 67(5):773–785, 1979.

[15] J. G. Daugman. Complete discrete 2-D gabor transforms by neural networks for image analysis and compression. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, 36(7):1169–1179, July 1988.

[16] Dejan Depalov, Thrasyvoulos Pappas, Dongge Li, and Bhavan Gandhi. Perceptually based techniques for semantic image classification and retrieval. In *Proceedings of SPIE - The International Society for Optical Engineering*, volume 6057, pages Society for Imaging Science and Technology, IS and, 2006.

[17] Jundi Ding, Runing Ma, and Songcan Chen. A scale-based connected coherence tree algorithm for image segmentation. *IEEE Transactions on Image Processing*, 17(2):204–216, 2008.

[18] M. a. El Saban and B. S. Manjunath. Interactive segmentation using curve evolution and relevance feedback. In *Proceedings - International Conference on Image Processing, ICIP*, volume 4, pages 2725–2728. IEEE, 2004.

[19] Jianping Fan, Yuli Gao, and Hangzai Luo. Multi-level annotation of natural scenes using dominant image components and semantic concepts. In *Proceedings of the 12th annual ACM international conference on Multimedia - MULTIMEDIA '04*, page 540, New York, NY, USA, 2004. ACM Press.

[20] Pedro F. Felzenszwalb, Gyula Pap, Eva Tardos, and Ramin Zabih. Globally optimal pixel labeling algorithms for tree metrics. In *2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 3153–3160. IEEE, June 2010.

[21] Bernhard Flury. Algorithms for clustering data. *Journal of Statistical Planning and Inference*, 21(1):137–138, January 1989.

[22] Allan Hanbury and Julian Stottinger. On segmentation evaluation metrics and region counts. In *2008 19th International Conference on Pattern Recognition*, pages 1–4. IEEE, December 2008.

[23] I. Haritaoglu, D. Harwood, and L.S. Davis. Ghost: a human body part labeling system using silhouettes. *Proceedings. Fourteenth International Conference on Pattern Recognition (Cat. No.98EX170)*, 1:77–82, 1998.

[24] Xuming He Xuming He, R.S. Zemel, and M.a. Carreira-Perpinan. Multiscale conditional random fields for image labeling. In *Proceedings of the 2004 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2004. CVPR 2004.*, volume 2, pages 695–702. IEEE, 2004.

[25] M.V. Kharinov and M.M. Nesterov. Intelligent program for automatic image recognition based on compact object-fitting hierarchical image representation in terms of dynamic irregular branchy trees. In *MELECON '98. 9th Mediterranean Electrotechnical Conference. Proceedings (Cat. No.98CH36056)*, volume 1, pages 58–62. IEEE, 1998.

[26] K.I. Kim, K. Jung, S.H. Park, and H.J. Kim. Supervised texture segmentation using support vector machines. *Electronics Letters*, 35(22):1935, 1999.

[27] T. Kohonen. Exploration of very large databases by self-organizing maps. In *Proceedings of International Conference on Neural Networks (ICNN'97)*, volume 1, pages PL1–PL6. IEEE, 1997.

[28] V.P. Kumar and U.B. Desai. Image interpretation using Bayesian networks. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 18(1):74–77, 1996.

[29] Seung-Hyun Lee Seung-Hyun Lee, Jaekyoung Moon Jaekyoung Moon, and Minho Lee Minho Lee. A Region of Interest Based Image Segmentation Method using a Biologically Motivated Selective Attention Model. In *The 2006 IEEE International Joint Conference on Neural Network Proceedings*, pages 1413–1420. IEEE, 2006.

[30] Sébastien Lefèvre. Knowledge from markers in watershed segmentation. pages 579–586, August 2007.

[31] Alex Levinshtein, Adrian Stere, Kiriakos N Kutulakos, David J Fleet, Sven J Dickinson, and Kaleem Siddiqi. TurboPixels: fast superpixels using geometric flows. *IEEE transactions on pattern analysis and machine intelligence*, 31(12):2290–7, December 2009.

[32] Jing Li, Xuelong Li, and Dacheng Tao. KPCA for semantic object extraction in images. *Pattern Recognition*, 41(10):3244–3250, 2008.

[33] D.G. Lowe. Object recognition from local scale-invariant features. In *Proceedings of the Seventh IEEE International Conference on Computer Vision*, volume 2, pages 1150–1157 vol.2. IEEE, 1999.

[34] Jiebo Luo, Andreas E. Savakis, and Amit Singhal. A Bayesian network-based framework for semantic image understanding. *Pattern Recognition*, 38(6):919–934, 2005.

[35] J. Malik. Normalized cuts and image segmentation. In *Proceedings of IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 731–737. IEEE Comput. Soc, 1997.

[36] Jianchang Mao and Anil K. Jain. A self-organizing network for hyperellipsoidal clustering (HEC). *IEEE Transactions on Neural Networks*, 7(1):16–29, January 1996.

[37] Kevin McGuinness. Image segmentation, evaluation, and applications. *Electronic Engineering*, 2009.

[38] D. Meyer, J. Denzler, and H. Niemann. Model based extraction of articulated objects in image sequences for gait analysis. *Proceedings of International Conference on Image Processing*, 3(Informatik 5):78–81, 1997.

[39] Branislav Mičušík and Jana Košecká. Semantic segmentation of street scenes by superpixel co-occurrence and 3D geometry. In *2009 IEEE 12th International Conference on Computer Vision Workshops, ICCV Workshops 2009*, pages 625–632. IEEE, 2009.

[40] Alastair P. Moore, Simon J. D. Prince, and Jonathan Warrell. Lattice Cut: Constructing superpixels using layer constraints. In *2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 2117–2124. IEEE, June 2010.

[41] G. Mori, Xiaofeng Ren Xiaofeng Ren, a.a. Efros, and J. Malik. Recovering human body configurations: combining segmentation and recognition. *Proceedings of the 2004 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2004. CVPR 2004.*, 2(c):326–333, 2004.

[42] Sangho Park Sangho Park and J.K. Aggarwal. Segmentation and tracking of interacting human body parts under occlusion and shadowing. *Workshop on Motion and Video Computing, 2002. Proceedings.*, 0(December):105–111, 2002.

[43] Giuseppe Passino, Ioannis Patrasfnm, and Ebroul Izquierdo. Context awareness in graph-based image semantic segmentation via visual word distributions. *2009 10th International Workshop on Image Analysis for Multimedia Interactive Services, WIAMIS 2009*, 0:33–36, 2009.

[44] a. K. Qin and David a. Clausi. Multivariate image segmentation using semantic region growing with adaptive edge penalty. *IEEE Transactions on Image Processing*, 19(8):2157–2170, 2010.

[45] Md Mahmudur Rahman, Prabir Bhattacharya, and Bipin C. Desai. A unified image retrieval framework on local visual and semantic concept-based feature spaces. *Journal of Visual Communication and Image Representation*, 20(7):450–462, 2009.

[46] S. Sanjay-Gopal and Thomas J. Hebert. Bayesian pixel classification using spatially variant finite mixtures and the generalized EM algorithm. *IEEE Transactions on Image Processing*, 7(7):1014–1028, January 1998.

[47] M. Shridhar, a. S. Sethi, and M. Ahmadi. Image segmentation: A comparative study. *Canadian Electrical Engineering Journal*, 11(4):172–183, October 1986.

[48] M. Singh and N. Ahuja. Regression based bandwidth selection for segmentation using Parzen windows. In *Proceedings Ninth IEEE International Conference on Computer Vision*, pages 2–9 vol.1. IEEE, 2003.

[49] W E Snyder and a Cowart. An iterative approach to region growing using associative memories. *IEEE transactions on pattern analysis and machine intelligence*, 5(3):349–352, May 1983.

[50] Richard Szeliski, Ramin Zabih, Daniel Scharstein, Olga Veksler, Vladimir Kolmogorov, Aseem Agarwala, Marshall Tappen, and Carsten Rother. A comparative study of energy minimization methods for Markov random fields with smoothness-based priors. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 30(6):1068–1080, 2008.

[51] Zhuowen Tu. Probabilistic boosting-tree: Learning discriminative models for classification, recognition, and clustering. In *Proceedings of the IEEE International Conference on Computer Vision*, volume II, pages 1589–1596. IEEE, 2005.

[52] L. Vincent and P. Soille. Watersheds in digital spaces: an efficient algorithm based on immersion simulations. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 13(6):583–598, June 1991.

[53] Liang Wang, Weiming Hu, and Tieniu Tan. Recent developments in human motion analysis. *Pattern Recognition*, 36(3):585–601, 2003.

[54] XiaoFeng Wang and Xiao Ping Zhang. A new localized superpixel Markov random field for image segmentation. In *Proceedings - 2009 IEEE International Conference on Multimedia and Expo, ICME 2009*, pages 642–645, Piscataway, NJ, USA, 2009. IEEE Press.

[55] Tomáš Werner. A linear programming approach to max-sum problem: A review, 2007.

[56] Yaowu Xu, Eli Saber, and a. Murat Tekalp. Dynamic learning from multiple examples for semantic object segmentation and search. *Computer Vision and Image Understanding*, 95(3):334–353, 2004.

[57] M. Yoshikawa, H. Shindo, R. Nishii, and S. Taaaka. A fully automated design of binary decision tree for land cover classification. In *1995 International Geoscience and Remote Sensing Symposium, IGARSS '95. Quantitative Remote Sensing for Science and Applications*, volume 3, pages 1921–1923. IEEE, 1995.

[58] Gui Mei Zhang, Ming Ming Zhou, Jun Chu, and Jun Miao. Labeling watershed algorithm based on morphological reconstruction in color space. In *HAVE 2011 - IEEE International Symposium on Haptic Audio-Visual Environments and Games, Proceedings*, pages 51–55. IEEE, October 2011.

[59] Ruofei Zhang Ruofei Zhang and Zhongfei Zhang Zhongfei Zhang. Hidden semantic concept discovery in region based image retrieval. In *Proceedings of the 2004 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2004. CVPR 2004.*, volume 2, pages 996–1001. IEEE, 2004.

[60] Thomas Zöller and Joachim M. Buhmann. Robust image segmentation using resampling and shape constraints. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 29(7):1147–1164, 2007.