Universitat Autònoma de Barcelona

Refining Pedestrian Detection via non Explicit Shape Models

Hilda Caballero Barbosa HCABALLERO@CVC.UAB.ES
Computer Vision Center
Campus UAB, Edifici O
Bellaterra, Barcelona, Spain

Advisors: Antonio López and David Gerónimo

Abstract

Pedestrian Protection Systems are aimed at detecting pedestrians to provide information to the driver (e.g., warnings in dangerous situations) and perform active actions on the host vehicle (e.g., braking). Research on this topic has been typically focused on classification techniques that output bounding boxes around the detected targets. In this Master Thesis we develop a target segmentation technique that refines the raw bounding boxes into shape-like detections by exploiting edge segment features and a bag of words based representation. The experimental results demonstrate that the proposed algorithm provides a rough silhouette of the pedestrians, which compared to bounding boxes, can potentially provide vital information for tracking or verification tasks.

Keywords: Segmentation, Bag of Words, Canny Edge Detector, Connected Component Labeling, Bilateral filtering

1. Introduction

The problem of deaths and injuries as a result of road accidents is now acknowledged to be a global phenomenon. According to a World Health Organization report (WHO, 2009) approximately 1.3 million people die each year on the world's roads, and between 20 and 50 million suffer non-fatal injuries. The statistics show that road traffic injuries remain an important public health problem, particularly for low and middle-income countries. Over 90% of the world's fatalities on the roads occur in these countries, which in fact only have 48% of the world's registered vehicles (WHO, 2009). In high-income countries, pedestrian fatalities are relatively lower but still represent large societal and economic costs to the

nations. For instance, in 2003 the United Nations reported almost 150000 injured and 7000 killed in vehicle-topedestrian accidents just in the European Union alone (UNECE, 2005).

Aimed at improving road safety, both the scientific community and the automobile industry have contributed to the development of different types of safety systems. As a result, research has moved towards innovative information technology systems, which rely on digital maps and sensors that enable vehicles to understand the environment around them, assist the driver in various ways, and significantly contribute to a safer driving. These systems are referred to as *advanced driver assistance systems* (ADAS).

The algorithm proposed in this project is focused in a particular type of ADAS, called *pedestrian protection systems* (PPS). The objective of a PPS is to detect the pedestrians (stationary and moving) and prevent accidents by warning the driver or triggering autonomous actions. According to (Gerónimo et al., 2009), a typical PPS can be split in six different modules, each one with its own specific objectives in the general task of pedestrian detection. The modules, ordered in a pipeline scheme, are: prepocessing, which performs camera adjustments previous to the processing; foreground segmentation, which generates windows likely to contain pedestrians; object classification, which labels these windows as pedestrian or non-pedestrian; verification, which filters false positives, and refinement, which provides a more accurate localization of the pedestrian than the raw output of the classifier; tracking, which takes time into advantage; and application, which takes the high level decisions based on the previous modules.

The research presented in this paper is focused on the verification/refinement module, specifically on the task of refining the typical bounding boxes that frame the classified pedestrians after the classification and verification steps. A silhouette-like output (or approximate shape) has many benefits compared to raw windows. As an example, the latter tracking step can make use of a more specific region of the image (defined by the refinement) in order to construct a color or edge based model. In addition, the verification stage can potentially make use of the extracted shape. Different proposals exploiting these approaches can be found in the literature (Gavrila and Munder, 2007).

There are some well-known papers that perform shape extraction of pedestrians. In Gavrila and Munder (2007) pedestrians are modelled by a set of annotated pedestrian shapes. These annotated training shapes or (also called templates) try to cover the wide range of different pedestrian poses in a hierarchy scheme that is built offline. The Chamfer Distance is used to measure the similarity between an instance of the static shape model and an observed image, so the proposal is named The Chamfer System. An object class detection approach is introduced in (Ferrari et al., 2008). An explicit shape model is built based on continuous connected curves, which represent the prototype shapes of different object categories. In order to learn and detect shapes, the edge points (edgels) are found by the Berkeley edge detector[1] (Martin et al., 2004). Then, the edgels are grouped into pairs of connected, approximately straight segments, which are called Pair of Adjacent Segments (PAS). The shape model is learnt by constructing a PAS codebook using example instances of each category. At the same time, there exist some algorithms that work with image appearance cues and perform object detection based on them. Such algorithms produce a

1. The Berkeley edge detector formulates boundary-detection as a classification problem of discriminating nonboundary from boundary pixels, using human-marked boundaries from the Berkeley segmentation dataset as groundtruth.

likelihood of the object presence either as a a function of a bounding box or even in the form of per-pixel soft segmentation masks. Such appearance-based detectors can be integrated with shape prior and edge-contrast cues. An example of such an integration is the detector proposed in (Leibe et al., 2008), which is capable of performing object segmentation. Similarly to the aforementioned approaches, this system also requires the annotation of a shape mask for each training example.

Up to our knowledge, the issue of pedestrian segmentation without explicitly annotating shape models has so far not been covered. Our proposal introduces a novel algorithm to segment pedestrians based on segment extraction and bag of words representation without using an explicit annotated model, to be used after any classifier. In order to obtain the representation a training data set is used. Firstly, a set of extracted edge segments are described (as 4-dimensional vectors) and used to generate a visual vocabulary. Then, the training samples are represented by a global histogram of the visual words in the vocabulary. This bag of words representation is used to obtain the pedestrian shape model, which is utilized to generate the pedestrian segmentation. The parameters involved in the algorithm are tuned using a validation data set. Finally, the parameter configurations obtained are used as the final models, and tested using a test data set, which allows to evaluate the system performance.

The outline of the paper is as follows. The basic ingredients of the algorithm developed in this research project are introduced in Section 2. Section 3 details the parameters optimization procedure and additional ideas to improve the basic algorithm, which are evaluated using a validation set. The different image datasets used are described in Section 4. In Section 5 experimental tests to assess the performance of the system are reported. Section 6 discusses the conclusions of this research and summarizes the main extensions left for future research.

2. Pedestrian Shape Extraction via Non-Explicit Models

The proposed algorithm is aimed at extracting the pedestrians' shape based on segment extraction and a bag of words representation without using explicit annotated shape models. The system is divided into two-steps:

1. Model Construction, in which a pedestrian shape model is generated.

2. Shape Retrieval, which obtains the pedestrian segmentation based on the pedestrian shape model from above.

The following subsections describe the structure of the algorithm and its main ingredients.

2.1 Model Construction

The aim of this step is to obtain a pedestrian shape model by using a bag of (visual) words representation.

The bag of words model (also known as bag-of-features or bag-of-visterms) consists of the following four steps (Rué, 2008):

1. Feature extraction: automatically detect regions/points of interest and compute local descriptors over these regions/points.
2. Visual vocabulary creation: quantize the descriptors into words to build the visual vocabulary or codebook.
3. Represent images by the frequencies of visual words: Find the occurrences of each specific word in the vocabulary (the closest) in order to build the bag of words (histogram of words).

In our project, the features of interest are edge segments, each described by four measurements. A visual codebook is constructed by applying K-means clustering on the features. The cluster centers are considered as the visual words of the codebook, so each feature in the training images is matched to the closest center and accumulated in a histogram of occurrences (the so-called bag-of-words). The size of the histogram is equal to the number of words in the codebook, i.e., to the number of clusters. Figure 1 shows an overview of this step.

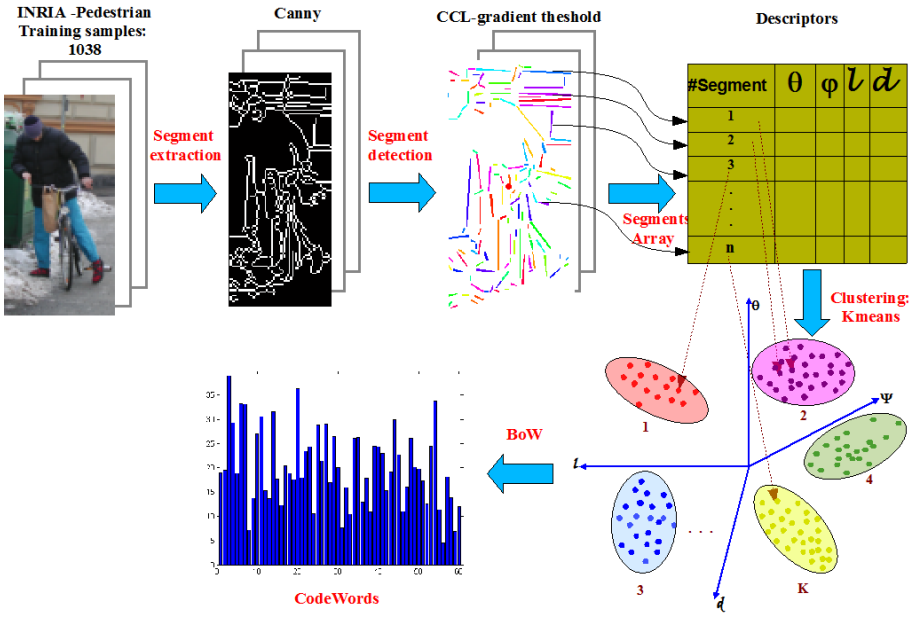


Figure 1: Overview of the Model Construction step of the proposed system.

The following sections detail the pedestrian shape model. Notice that in our proposal the bag of words is used as a representation of the pedestrian shape and not as a classification method.

2.1.1 Segments Extraction

Firstly, in order to obtain the segments, an edge detection process is performed. The edge detection uses the traditional Canny detector (for more details about Canny detector, refer to Jain et al., 2001). Then, in order to divide the edges into segments, we propose

a Connected Component Labeling (CCL)-like algorithm. The proposed algoritm, named Gradient-CCL, is based on the original CCL, which finds all connected pixels in an image attending to a neighborhood criterion and assigns a unique label to all pixels in the same component (CC), but makes use of both orientation and spatial neighborhood to compute the components.

The original CCL works on binary or graylevel images and makes use of different measures of connectivity: usually 4- and 8-connectivity. All pixels in a CC share similar pixel intensity values and are connected with each other. The sequential CCL algorithm requires two passes over the image: one pass to record equivalences and assign temporary labels and the another to replace each temporary label by the label of its equivalence class (Jain et al., 2001).

The proposed Gradient-CCL algorithm scans binary edge images and groups pixels into components based on 8-connectivity. As a measure of similarity, our algorithm makes use of spatial neighborhood, gradient orientation and gradient magnitude. Specifically, one pixel p is connected to a specific component C if the difference between the gradient orientation of p and the average gradient orientation of C is smaller than a certain Gradient-CCL threshold. Figure 2 illustrates one example of this criterion. In this example the edge pixel analyzed appears in a shadowed red square and there are two labeled pixels in the 8-neighborhood. These neighbors belong to two different CCs, which are labeled as 55 and 56 respectively. Moreover, the average gradient orientation of the CC labeled as 55 is 47.8 degrees and is 80.9 degrees for the CC labeled as 56. Since the Gradient-CCL threshold is set to 26 degrees, the analyzed pixel is assigned the label 56.

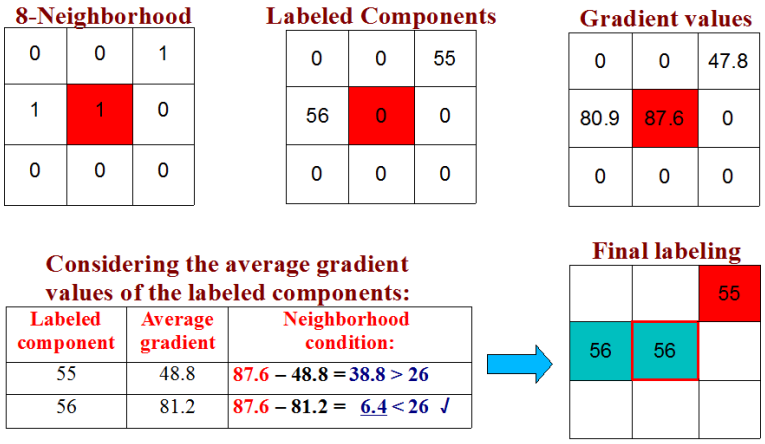


Figure 2: Example of the Gradient-CCL algorithm criteria.

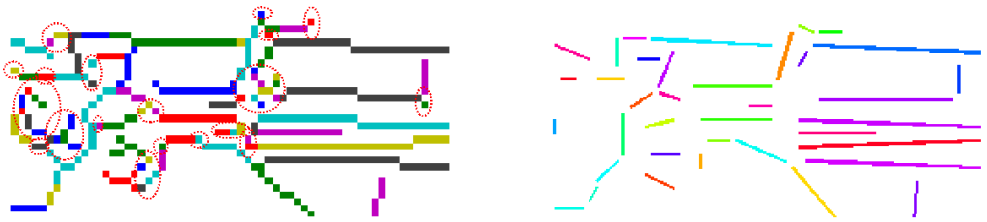
In case that a pixel has neighbors with different labels but similar gradient orientation (i.e., the average gradient orientation of the CCs labeled passed the threshold), the edge pixel is labeled to the smallest label and an equivalence table is prepared to keep track of all the labels that are equivalent. Similarly to the original CCL, this table is used in the second pass to assign a unique label to all the pixels of a component.

Finally, the edge segments obtained with the Gradient-CCL algorithm are prepared to be used in the next stages of the algorithm, i.e., the clustering. This involves the following steps:

1. Eliminating noisy segments: small segments can lead the algorithm to error, so the segments with less than 3 pixels are removed.

2. Obtaining line segments: a line segment is computed by joining the ending points of a edge segment. This dimensionality reduction of the original edge segment is aimed at easing computation of the descriptor without the danger of losing information, given that edge components are typically straight lines thanks to the CCL algorithm. The central edge pixel within a 8-neighbourhood is an ending point if it only has one edge pixel neighbor.

Figure 3 illustrates these two steps. The noisy segments appear inside the red ellipses in Figure 3(a), while final line segments are shown in Figure 3(b). The result of applying the Gradient-CCL algorithm to one of the training images is illustrated in Figure 4. The CCs appear in different colors, and each one defines an edge segment.



(a) Noisy segments (b) Line segments after the elimination

Figure 3: Example of (a) elimination of noisy segments and (b) the final line segments obtained.

2.1.2 Segment Description

Each one of the line segments extracted is represented by a descriptor that consists of:

- l, the length of the line segment.
- d, the distance from the center of the line segment to the center of the sample.
- θ, the angle of the line segment with respect to the positive x-axis.
- Ψ, the relative angle between the center of the line segment and the center of the sample.

2.1.3 Bag of Words Construction

The steps involved in the Bag of Words (BoW) construction in our proposal, which correspond to the generic steps described in Section 2.1, are the following:

Figure 4: Result of the Gradient-CCL algorithm (Gradient-CCL threshold = 26).

1. **Segments array**. Firstly, the n line segments obtained from the training samples (as described in 2.1.1) are stored in a segment array structure.

2. **Creation of the Visual Vocabulary**. The vocabulary is constructed by applying the K-means clustering algorithm on the segments represented in the segments array. K-means (Macqueen, 1967) is an unsupervised clustering algorithm that classifies a given data set through a certain number of clusters (K) fixed a priori.

   The input data of K-means is the segments array[2] and the output data are the K centroids of the clusters, which are designated as visual words in order to build the vocabulary. Figure 5 illustrates one example in which 60 clusters are obtained. As it can be seen, each element of the segments array is a member of one of the clusters, and each centroid is one element in the vocabulary (one visual word). The visual vocabulary is a set of line segments that are deemed to be representative of the descriptors retrieved from the training set.

3. Finally, the visual vocabulary is used to construct a **histogram based representation** of the training samples, i.e., the bag of words. Each descriptor (segment line) is assigned to the closest visual word in the vocabulary by using a distance measure (two different distance measures are used, as it will be explained in detail in Section 3). Then all training samples are represented as a global histogram counting the number of occurrences of each visual word. The number of occurences, however, is not

2. Note that values of the descriptors are normalized to a range between 0 and 1. This is calculated applying a linear scaling transformation to the descriptors θ and Ψ. l is divided by the height of the image. The descriptor d is divided by the distance of the image center to the image corner.

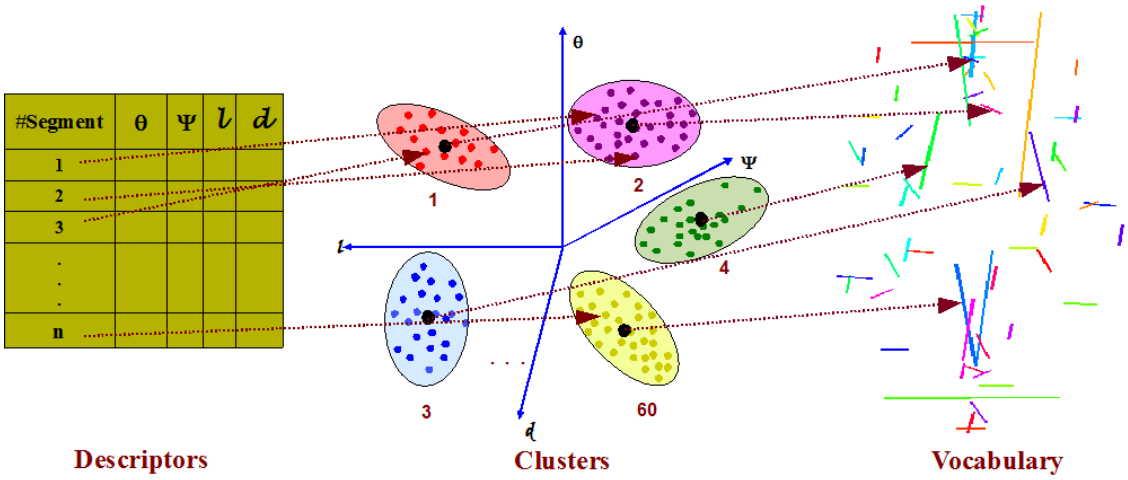


Figure 5: Visual vocabulary creation.

enough to have a measurement of importance for each visual word. The smaller the minimum distance for each visual word, the bigger is its importance. Therefore, the number of occurences is normalized dividing it by the sum of the minimum distances of the descriptors.

2.1.4 Pedestrian Shape Model

The next step is to determine which of the visual words in the vocabulary better describe the segments belonging to people. This is achieved with a thresholding process over the bag of words representation. Given that the bag of words representation has the number of occurrences of each visual word over all the training samples and that segments that correspond to pedestrians will appear with a higher frequency than the segments that belong to the background (clutter segments), a specific threshold to filter clutter can be applied to the histogram. After the thresholding process, a new bag of words representation in which the visual word occurrences that are lower than the threshold are set to zero is obtained.

2.2 Shape Retrieval

Once the model has been constructed, it can then be used to extract the shape of a test pedestrian sample. Figure 6 illustrates this step[3].

2.2.1 Segment Matching with Model

Similar to the procedure explained in Section 2.1.1 and Section 2.1.2, the segments of a test image are extracted and described. Then, each of the line segments is assigned (matched) to the closest visual word in the pedestrian shape model (two different distance criteria are used, which are detailed in Section 3). All the segments not matched (because each line segment is assigned to one visual word only) are eliminated in this step.

3. Notice that colors not necessarily match between steps and they are only aimed at enhacing their visualization in the step itself.

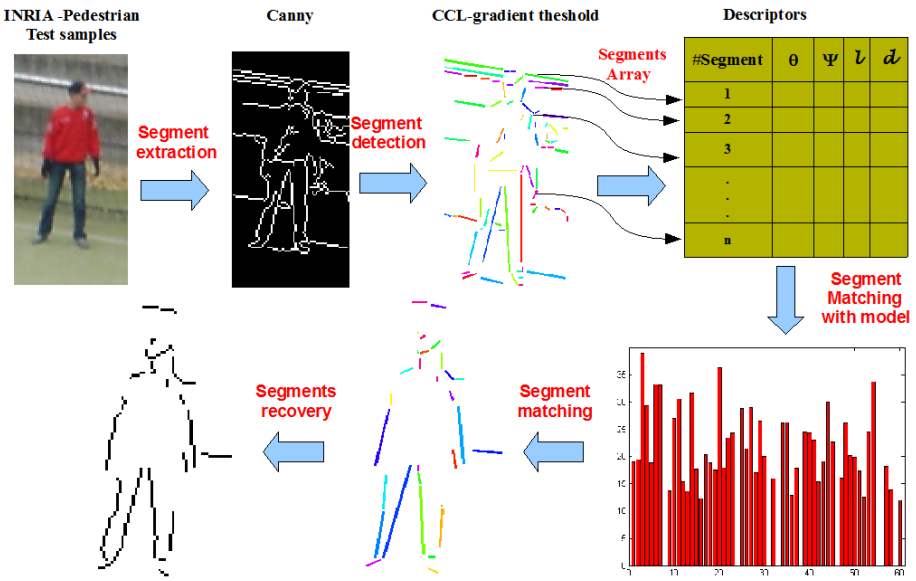


Figure 6: Overview of the Shape Retrieval step of the proposed system

2.2.2 Segments Recovery

Finally, the line segments that matched with the pedestrian shape model are used to recover the original edge segments in the sample test.

3. Algorithm Details, Parameter Tuning and Improvements

This section focuses on describing: 1) the experiments developed in order to find the best configuration of the parameters involved in the algorithm together with 2)the introduction of some ideas that improve the proposed basic algorithm. The experiments presented demonstrate how the algorithm has been developed and gradually refined. The tuned parameters are the following:

In the Model Construction steps:

1. *Segments extraction:* The Gradient-CCL threshold used in the Gradient-CCL algorithm (Section 2.1.1).

2. *Bag of words construction:*

   - The number of clusters used to create the visual vocabulary, i.e. the adequate size of vocabulary.
   - The distances measure used in the assignation of each descriptor (segment line) for each training sample to the closest visual word in the vocabulary: Euclidean or Mahalanobis distance.

3. *Pedestrian shape model:* The threshold applied over the bag of words representation (in order to filter the clutter segments, Section 2.1.4), which will be referred to as BoW-threshold.

In the Shape Retrieval step:

1. Each line segment (from a validation or test sample) is matched with a visual word in the pedestrian shape model using one of the two following distances:

   - To the closest visual word using the Euclidean distance.
   - To a specific cluster if its Mahalanobis radius is below a certain threshold. This threshold is set to two times the standard deviation (σ) of the cluster. The Mahalanobis radius is the distance from the centroid, scaled in each dimension by σ_i, the standard deviation in that dimension. More precisely, if μ_i is the mean in dimension i, then the radius of point $y = [y_1, y_2, ...]$ is (Rajaraman and Ullman, 2009):

$$\sqrt{\sum_i \left(\frac{y_i - \mu_i}{\sigma_i}\right)^2} \tag{1}$$

In addition, the proposed improvements to the basic algorithm consist in:

- Using a background model to eliminate the clutter segments.
- Applying bilateral filtering over the training samples as a preprocessing method.

3.1 Evaluation Methodology

The experiments are performed on a subset of the validation samples (Section 4). In each experiment the evaluation methodology is as follows:

- Attending to the literature, the performance measure employed used to compute the similarity between two silhouette images is the Chamfer Distance (Barrow et al., 1977). The Chamfer Distance from a template contour or groundtruth (T=$\{t\}$) to the segmented contour (Q=$\{q\}$) is measured as the mean value of distances between points in the template to their closest points in the segmented contour:

$$d_{cham}^{(T,Q)} = \frac{1}{N_T} \sum_{t \in T} min||t - q||$$

  Where N_T is the number of points in T and $||.||$ can be any norm (the Euclidean distance is used in our evaluation). However, the use of this measure could lead to problems in our evaluation given that the number of pixels in the retrieved shape and the groundtruth is not necessarily the same. Accordingly, in order to provide a fair and correct evaluation, we make use of the Symmetric Chamfer Distance, which takes into account the sum of the distance from the retrieved shape to the groundtruth and vice versa (Zhang et al., 2004) (Cao et al., 2006):

$$D_{cham}^{(Q,T)} = d_{cham}^{(T,Q)} + d_{cham}^{(Q,T)}$$

The smaller the value of $D_{cham}^{(Q,T)}$, i.e., the error, the better the approximation to the groundtruth, i.e., the desired result.

- A baseline is calculated for each one of the 100 validation samples (see Section 4). The baseline serves as an anchor point for measuring performance. The baseline is the Symmetric Chamfer distance between the Canny detector result and the handlabeled groundtruth silhouette. The baseline can be considered as the result given by a naïve shape extractor.

- The performance measure is obtained for each one of the 100 validation samples: first, the algorithm is applied to each validation sample and the segmented contour is obtained. After this, the Symmetric Chamfer distance from the segmented contour to the handlabeled groundtruth silhouette is calculated.

- The mean and standard deviation of the baseline and the performance measures are compared and analyzed to prove that the results are statistically significant. A one-way analysis of variance (ANOVA[4]) test is used over these groups (the baseline and the performance measures). A significant p−level resulting from a 1-way ANOVA test indicates the probability of getting a mean difference between the groups as higher as what is observed by chance. The lower the p−level, the more significant the difference between the groups (standard meaningful differences are for p−level < 0.05, as is p−level < 0.01). Moreover, ANOVA involves calculating a statistic called the "F ratio" (the between-groups variance / the within-groups variance). The F ratio, which gets larger as the distribution overlap gets smaller (i.e., a large F indicates a difference in the group means).

3.2 Baseline

Firstly, the baseline is obtained, and its mean and standard deviation (stdev) are calculated in pixels:

baseline mean=8.6336, baseline stdev = 1.238

The baseline remains without changes for all the experiments because, as it is mentioned before, it is the Symmetric Chamfer distance from the Canny result to the groundtruth silhouettes.

3.3 Segments Extraction

Canny parameters
As it was mentioned in subsection 2.1.1, the Canny detector is used for obtaining the edges in the image. The Canny method uses two thresholds, one to detect strong edges and another one to detect weak ones. The Sigma parameter is used as the standard deviation of

4. StatPlus 2007 Professional Build 4.9.0.0 software is used for this purpose.

the Gaussian filter. Therefore, it is necessary to determine the value of these three parameters. In order to determine the value of the parameters both an automatic and a manual way have been employed:

Manual Canny parameters:
Low threshold = 0.03
High threshold = 0.38
Sigma = 0.39

Automatic Canny parameters
Low and High thresholds are calculated automatically. Their value is relative to the highest value of the gradient magnitude of the image.
Sigma = 1

The manual selection of the Canny parameters tries to reduce the clutter edges while the edges of interest remain. Figure 7 illustrates the result of Canny (with manual and automatic parameters) over two different validation samples. As it can be seen, manual parameters can cause the elimination of relevant edges (such as the head edges, Figure 7(b)) or produce more clutter edges (Figure 7(e)). Therefore, after this qualitative analysis, the automatic parameters are selected to be used to detect the edges.

(a) (b) (c) (d) (e) (f)

Figure 7: Edge detection results. Original images (a) and (d). Canny results: manual parameters (b) and (e), automatic parameters (c) and (f)

Gradient-CCL algorithm threshold tuning
In order to determine the Gradient-CCL threshold, the Gradient-CCL algorithm (Subsection 2.1.1) is applied over a representative selection of the validation samples. Firstly, their corresponding binary images are obtained using the Canny edge detector (with automatic parameters). After this, the value of the Gradient-CCL threshold is varied within the range [18, 35] (based on explorative qualitative tests). In order to get the final line segments. Segments with less than 3 pixels are considered as noisy and have been eliminated.

Figure 8 illustrates the results using the Gradient-CCL threshold 18, 23, 26 and 35, which are the representative values that have lead us to the following observations:

- A weak threshold (Gradient-CCL threshold=35) results in long line segments. This can cause that meaningful line segments for describing the person to be missed (see Figure 8(d)).

- A strong threshold (Gradient-CCL threshold=18) provides too many small segments, which causes that meaningful line segments for describing the person to be lost (see Figure 8(a), less than 3 pixels).
- A Gradient-CCL threshold of 23 and 26 respectively shows to better preserve the line segments that define the person's contour (Figure 8(b) and Figure 8(c)).

The following subsection explains the experiments using both the Gradient-CCL threshold of 23 and 26. The experiments are evaluated quantitatively to decide which Gradient-CCL threshold value improves best the performance of the algorithm. The ranges of the different variables described in the following Section have been defined by explorative qualitative tests.

(a) 18 (b) 23 (c) 26 (d) 35

Figure 8: Results for different Gradient-CCL thresholds.

3.4 Bag of Words Construction

Here we will study the optimun configuration of the following parameters involved in the bag of words construction:

- **The Gradient-CCL threshold.** Two different values are evaluated, Gradient-CCL threshold of 23 and Gradient-CCL threshold of 26.
- **The number of clusters used to create the visual vocabulary.** The vocabulary size was varied within to this range: $\{60, 80, 100, 120, 140, 160, 180\}$.
- **The BoW-threshold.** The range that is tested includes $\{10, 8, 6, 4, 2\}$.

Additionally, the rest of parameters involved in the algorithm are set in the following way:

- The edges are detected using Canny with automatic parameters.
- The Euclidean distance was used in the bag of words creation.
- The Euclidean distance was applied in the segments matching step.

In the following experiments, the Gradient-CCL threshold was fixed, while the number of clusters and the value of the BoW-threshold are varied according to the ranges explained above. For each combination of the parameters, the algorithm was evaluated as described in Section 3.1.

Table 1 shows the mean and stdev of the performance measures when the Gradient-CCL threshold is set to 23. Additionally, the One-way analysis of variance (ANOVA test) evaluation with respect to the baseline is shown (p-level and F ratio). Table 2 illustrates the mean and stdev of the performance measures when Gradient-CCL threshold is set to 26. Likewise, the ANOVA test results (F y p-level) are shown. Based on the Tables mentioned above we can draw the following conclusions:

1. In order to identify the optimum combination of parameters, it is necessary to find the smallest mean of Symmetric Chamfer Distances in the performance measures. The trends of the error for the different parameters combinations is illustrated in Figure 9. As it can be seen, the smallest mean (for both Gradient-CCL thresholds) results for:

   - Number of clusters = 80,
   - BoW-threshold = 6

2. Comparing the performance measures of this parameter combination with respect to the baseline, it turns out that the performance measures error is lower than the baseline error. On the other hand, the performance measures have a higher variation (stdev). In addition, according to the ANOVA test, the p-level of less than 0.05 indicates that the probability that the differences of the means is random, is much less than 5%. Therefore, the difference in the mean of Symmetric Chamfer Distances is significant.

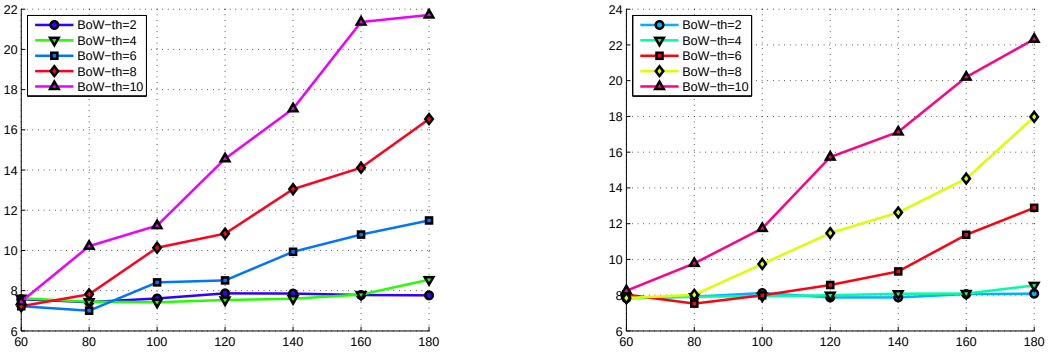
Number of clusters	**BoW-threshold**									
	10		**8**		**6**		**4**		**2**	
	Mean	**stdev**	**Mean**	**stdev**	**Mean**	**stdev**	**Mean**	**stdev**	**Mean**	**stdev**
60	7.4554	1.5766	7.2303	1.5526	7.2303	1.5526	7.6208	1.5071	7.5751	1.6024
	F=34.54		F=49.94		F=49.94		F=26.96		F=27.32	
80	10.2132	2.0597	7.8217	1.7256	**7.0069**	**1.5816**	7.45	1.5465	7.433	1.5812
	F=43.21		F=14.61		**F=65.59**		F=35.69		F=35.74	
100	11.2376	2.4023	10.1378	2.0463	8.4112	1.7925	7.4263	1.4748	7.6137	1.4745
	F=92.84		F=39.55		$F = 1.04^*$		F=39.31		F=28.06	
120	14.5624	2.8803	10.8378	2.2964	8.509	1.8588	7.5354	1.4898	7.8652	1.4642
	F=357.63		F=71.39		$F = 0.31^*$		F=32.14		F=16.06	
140	17.0509	3.1317	13.0509	2.3975	9.9345	1.9724	7.603	1.5144	7.859	1.485
	F=624.77		F=268		F=31.21		F=27.76		F=16.05	
160	21.359	3.6172	14.1166	2.6632	10.7927	2.1751	7.8028	1.5283	7.7843	1.4642
	F=1107.88		F=348.54		F=74.42		F=17.84		F=19.62	
180	21.7136	3.323	16.5363	2.6021	11.49	2.2952	8.5474	1.7889	7.7657	1.4888
	F=1360.54		F=752.12		F=119.98		$F = 0.16^*$		F=20.09	

Table 1: Quantitative results for setting Gradient-CCL threshold to 23. The symbol * indicates that p-level>0.05, for the other cases p-level<0.0001. Optimum parameters are marked in bold.

Number of clusters	BoW-threshold									
	10		8		6		4		2	
	Mean	stdev	Mean	stdev	Mean	stdev	Mean	stdev	Mean	stdev
60	8.2491	1.649	7.8398	1.6113	8.0311	1.5135	7.8308	1.452	7.8308	1.452
	$F = 3.48^*$		F=15.26		F=9.49		F=17.7		F=17.7	
80	9.7771	2.0505	8.0166	1.7248	**7.5272**	**1.5153**	7.9159	1.4795	7.9159	1.4795
	F=22.79		F=8.44		**F=31.97**		F=13.84		F=13.84	
100	11.7317	2.4349	9.7424	2.0808	7.9891	1.6458	7.9398	1.4267	8.1207	1.4293
	F=128.65		F=20.97		F=9.79		F=13.49		F=7.36	
120	15.7243	2.3686	11.4692	2.3999	8.568	1.6756	7.9946	1.417	7.8753	1.4546
	F=703.87		F=110.26		$F = 0.1^*$		F=11.53		F=15.76	
140	17.1338	2.7501	12.6207	2.7176	9.3243	1.9443	8.0775	1.6139	7.8734	1.4299
	F=794.36		F=178.25		F=8.98		F=7.47		F=16.15	
160	20.2013	3.128	14.5229	2.2545	11.3756	2.1758	8.0881	1.4993	8.0621	1.3659
	F=1182.38		F=524.28		F=119.98		F=7.87		F=9.61	
180	22.3211	3.8574	17.9766	3.1182	12.8864	2.6508	8.544	1.7087	8.0831	1.4314
	F=1141.5		F=775.53		F=211.31		$F = 0.18^*$		F=8.46	

Table 2: Quantitative results for setting Gradient-CCL threshold to 26. The symbol * indicates that p-level>0.05, for the other cases p-level<0.01. Optimum parameters are marked in bold.

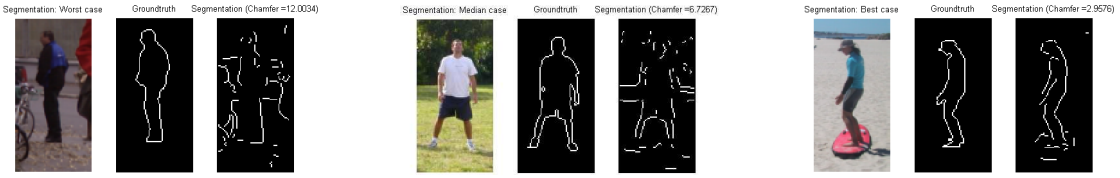
As conclusion of these experiments, the following parameters configuration improves the algorithm performance: Gradient-CCL threshold: 23; vocabulary size: 80; BoW-threshold: 6; edges detected using Canny with automatic parameters; Euclidean distance used in the bag of words creation and in the segments matching step. This parameter configuration will be referred to as configuration A. The segmentation results obtained with the algorithm developed in this project are presented in Figure 10 (using Gradient-CCL threshold of 23).



(a) Gradient-CCL threshold = 23

(b) Gradient-CCL threshold = 26

Figure 9: The function of the error for different parameter configuration (x-axis: Number of clusters, y-axis: Symmetric Chamfer Distance mean).



(a) Worst case (b) Median case (c) Best case

Figure 10: The worst, median and best case in the segmentation result (Gradient-CCL threshold = 23).

Using Mahalanobis distance as the distance measure

Here the experimental results are presented using a different distance measure: the Mahalanobis distance. As it was explained at the begining of this section, a distance measure is required in the bag of words creation and in the shape retrieval step. Therefore, this new distance measure is applied in the following way:

- In the bag of words creation: assigning a specific segment in the validation sample to the closest visual word in the vocabulary.
- In the Shape Retrieval step: matching the line segments (from a validation or test sample) to a specific visual word in the pedestrian shape model using the Mahalanobis radius.

Firstly, in order to improve the algorithm performance given by the parameter configuration A, the Mahalanobis distance is used in the shape retrieval step to decide whether a given segment belongs or not to a cluster. The rest of the parameters of the best configuration, configuration A, obtained above are kept.

Table 3 shows the mean and stdev of the performance measures of the parameter configuration above. This error is still smaller than the baseline error. Furthermore, p-level is much less than 0.05, which means that the results obtained with this parameter combination actually are better than the baseline, even though the variance is bigger. However, the error of this performance measures is bigger than the error obtained in configuration A. Therefore, an improvement in the final results using the Mahalanobis radius in the Shape retrieval is not shown.

Number of clusters	**BoW-threshold = 6**			
	Mean	**stdev**	**F**	**p-level**
80	7.7365	1.3972	23.09	< 0.001

Table 3: Quantitative results, Mahalanobis radius is used in the Shape Retrieval step.

Secondly, the Mahalanobis distance is used in both during the bag of words creation and also the shape retrieval step. Edge detection and CCL-Gradient threshold are the same

as configuration A. Then, the following parameters are varied to find out the optimum configuration in the bag of words construction:

- The number of clusters used to create the visual vocabulary. The vocabulary size is varied according to this range: {75, 80, 85, 90}
- The BoW-threshold. The range experimented of this threshold is { 12, 10, 8, 6}

Table 4 lead us to the following conlcusions:

1. The trends of the error for the different parameters combinations is illustrated in Figure 11. As wee can see, the smallest error results for:
   - Number of clusters = 85
   - BoW-threshold = 10
2. The value of this smallest error of the performance measures is lower than the baseline error with a p-level<0.02. This means that the results obtained with this parameters combination actually are better than the baseline, even though the variance is bigger. However, it does not improve the algorithm performance with respect to the configuration A. From now on, this parameter configuration will be referred to as configuration B.

Number of clusters	**BoW-threshold**							
	12		**10**		**8**		**6**	
	Mean	**stdev**	**Mean**	**stdev**	**Mean**	**stdev**	**Mean**	**stdev**
75	7.645	1.6525	8.1536	1.5471	8.4924	1.5072	8.4924	1.5072
	F=22.92		F=5.87		$F = 0.52^*$		$F = 0.52^*$	
80	8.3002	1.7196	7.6529	1.5155	7.9603	1.4136	8.1618	1.3934
	$F = 2.47^*$		F=25.12		F=12.84		F=6.41	
85	7.8701	1.8414	**7.3182**	**1.5933**	7.7824	1.4693	8.0212	1.4304
	F=11.84		**F=42.5**		F=19.62		F=10.48	
90	8.1973	2.1578	7.9395	1.6	7.7813	1.4433	8.0538	1.4062
	$F = 3.08^*$		F=11.77		F=20.09		F=9.57	

Table 4: Quantitative results using Mahalanobis distance (bag of words creation) and Mahalanobis radius (shape retrieval). The symbol * indicates that p-level>0.05, for the other cases p-level<0.02

. Optimum parameters are marked in bold.

Background model

In order to identify where the clutter segments in the segmentation results come from, the visual words that define the pedestrian shape model are analyzed. This pedestrian shape model was obtained with the best parameter configuration found until now: the configuration A. Figure 12(a) depicts as line segments such visual words.

The visual words that describe the pedestrian shape model are the cluster means obtained with K-means (Section 2.1.3). Figure 12 exemplifies the content of two different line segment clusters. One example of clutter segments are the elements of the cluster labelled

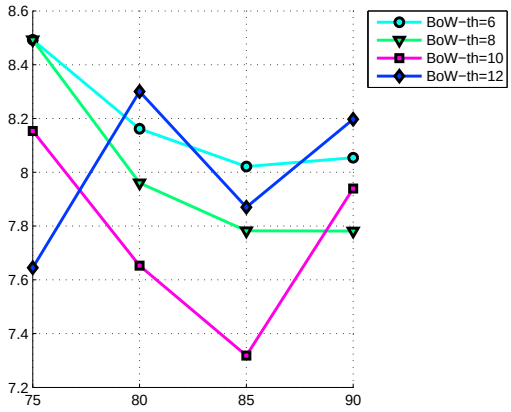


Figure 11: The trends of the error for the different parameters combinations using Mahalanobis distance (bag of words creation) and Mahalanobis radius (shape retrieval). x-axis: Number of clusters, y-axis: Symmetric Chamfer Distance mean

as 1 (first visual word in the vocabulary that defines the pedestrian shape model). The visual word 1 is shown in Figure 12(b) as the black line segment. The elements in this cluster are the colored line segments. Figure 12(d) demonstrates some images where such line segments come from. This visual word is one example of clutter segments. Similarly, Figure 12(c) exhibits the visual word labeled as 8 (black line segment) and the elements in this cluster (colored line segments). Figure 12(e) shows images, where such line segments come from. As can be seen, most of these line segments come from the pedestrian legs.

The cluster analysis above is useful to find a solution to eliminate the clutter segments. The proposed solution is to build a background histogram and subtract this model from the pedestrian histogram. This subtration aims to obtain a new pedestrian shape model, in which visual words that represent clutter line segments are eliminated.

The background model uses the pedestrian visual vocabulary to construct a histogram based representation of the background samples (henceforth referred to as BoW-background). This model is obtained following a similar procedure to the pedestrian shape model. For each background sample, line segments are extracted (Section 2.1.1) and described (Section 2.1.2). Then, each descriptor (line segment) for each background sample is assigned to the closest visual word in the pedestrian vocabulary. Therefore, each descriptor is designated to the pedestrian visual word with the minimum distance to it. Then each background sample is represented as a histogram counting the number of occurrences of each pedestrian visual word. Figure 13 illustrates this process.

The experiments using the BoW-background model are done as follows:

- Four different BoW-background models have been built from four different sets of background samples (from the background dataset) that have been selected randomly. The number of samples in each set was the same as the training set.

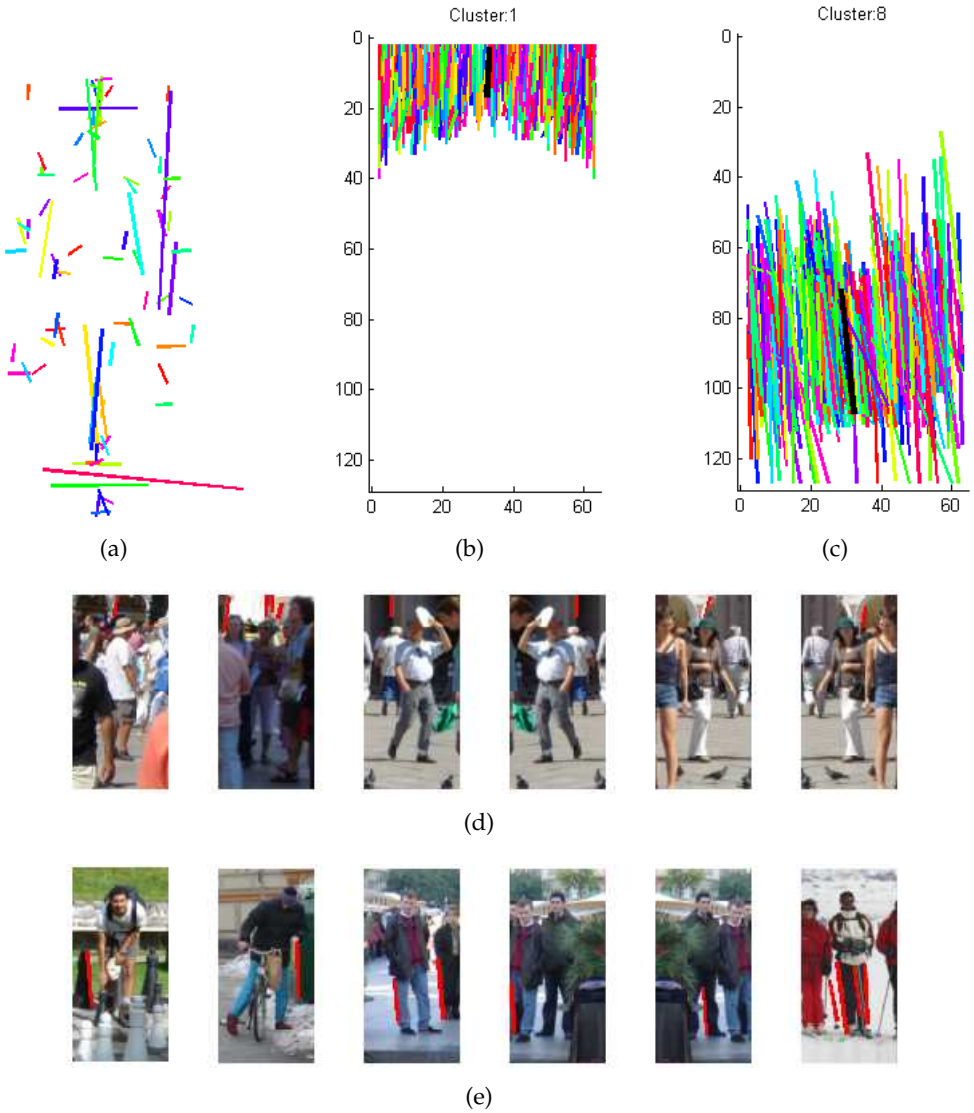


Figure 12: (a) Pedestrian shape model obtained with the parameter configuration A. Line segments in clusters 1 and 8 (visual vocabulary obtained with the configuration A). (b) and (c) Elements of the cluster. (d) and (e) Images from where these elements come (the line segmentes are marked in red).

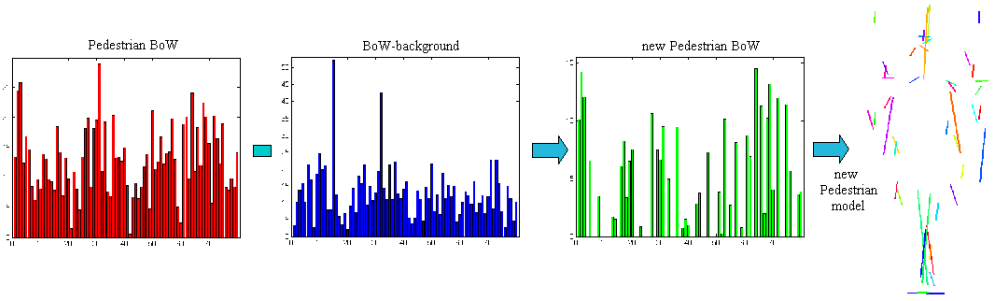


Figure 13: New pedestrian shape model obtained using a BoW-background model

- The pedestrian visual vocabulary used to construct the histogram representation is obtained with the parameters configuration *A*.
- The assignation of the background descriptors to the closest visual word in the vocabulary was done using two different distance measures: Euclidean and Mahalanobis.

Distance (BoW-Background creation)	**BoW-Background models**							
	set_1		**set_2**		**set_3**		**set_4**	
	Mean	**stdev**	**Mean**	**stdev**	**Mean**	**stdev**	**Mean**	**stdev**
Euclidean	6.5023	1.9007	6.4183	1.9285	6.573	1.9425	6.7723	1.9211
	F=88.28		F=93.45		F=80.02		F=66.32	
Mahalanobis	**6.4028**	**1.8928**	6.5399	1.9325	6.415	1.8337	6.5348	1.8344
	F=97.28		F=83.22		F=100.55		F=89.94	

Table 5: Quantitative results using Euclidean and Mahalanobis distances in the BoW-background creation. p-level<0.001. Optimum parameters are marked in bold.

Table 5 shows us the following:

1. The trends of the error for the different parameters combinations is illustrated in Figure 14. As wee can see, the smallest error results for:
   - BoW-background model from set_1.
   - Mahalanobis distance in BoW-background creation.
2. The value of this smallest error of the performance measures is lower than the baseline error with a p-level<0.001. Therefore, the results obtained with this parameter combination are better than the baseline, even though the variance is bigger.
3. Finally, the worst, median and best case (according to the Symmetric Chamfer distance measure) of the segmentation results obtained with the previous parameters configuration are presented in Figure 15.

The conclusions of these experiments are the following:

- The parameters configuration using a BoW-background model (set_1), and the Mahalanobis distance in its creation improves the algorithm performance with respect to the baseline and to the configuration *A*. Likewise, the segmentation results in Figure 15 show this qualitative improvement, because clutter segments are reduced. This parameter configuration will be referred forward as configuration *C*.

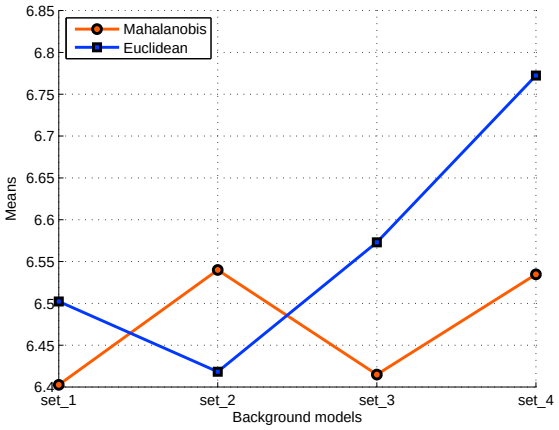
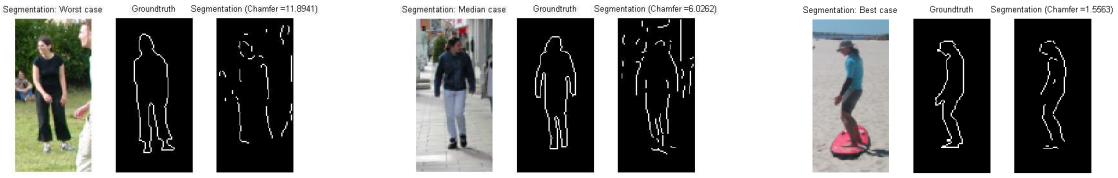


Figure 14: The trends of the error for the different parameters combinations using a BoW-background model (Euclidean and Mahalanobis distances are used in its creation). *x*-axis: Number of clusters, *y*-axis: Symmetric Chamfer Distance mean.



(a) Worst case (b) Median case (c) Best case

Figure 15: The worst, median and best case in the segmentation result. Using a BoW-background (set_1, Mahalanobis distance in its creation).

Bilateral filtering

In order to reduce the clutter segments even more, bilateral filtering[5] is applied over the training samples as a preprocessing method. Bilateral filtering smooths images while preserving edges, by means of a nonlinear combination of nearby image values. The method combines gray levels or colors based on both their geometric closeness and their photometric similarity, and prefers near values to distant values in both domain and range (for more details about bilateral filtering, see (Tomasi and Manduchi, 1998)).

5. For this purpose the bilateral filtering implentation developed by Douglas Lanman is used (Lanman, 2006).

Bilateral filtering uses three parameters: the half-size of the Gaussian bilateral filter window is defined by W_G. The spatial-domain standard deviation is given by σ_d and the intensity-domain standard deviation is given by σ_r.

Firstly, some experiments were developed, setting the bilateral filtering parameters manually according to the suggested values by Tomasi in (Tomasi and Manduchi, 1998):

- $W_G = 5$
- $\sigma_d = \{3, 10\}$
- $\sigma_r = \{0.1, 0.2\}$

In these experiments, range filtering is done over gray level values (with normalized values in the closed interval [0,1]). Figure 16 exemplifies the results of the Gradient-CCL algorithm over one validation sample. Figure 16(a) shows the result using Canny with automatic parameters. Figures 16(b), 16(c), 16(d) and 16(e) depict the result using bilateral filtering with the manual parameters above. As it can be seen, the bilateral filtering reduces the clutter segments. However, it could also cause the elimination of relevant segments (Figure 16(e)). After a qualitative analysis, the bilateral filtering result in Figure 16(b) seems to eliminate clutter segments, while keeping the relevant ones. Therefore, the following manual parameters were choosen to experiment with: $W_G = 5$, $\sigma_d = 3$, $\sigma_r = 0.1$. Further analysis with different bilateral filtering parameters will be left to future work.

(a) Canny automatic parameters (b) σ_d=3, σ_r=0.1 (c) σ_d=3, σ_r=0.2 (d) σ_d=10, σ_r=0.1 (e) σ_d=10, σ_r=0.2

Figure 16: Results for different bilateral filtering thresholds.

The experiments using bilateral filtering as preprocessing method allow to study the effect of the following parameters in the algorithm performance:

- The number of clusters used to create the visual vocabulary. The vocabulary size varied according to this range: {80, 100, 120}.
- Bilateral filtering parameters: $W_G = 5$, $\sigma_d = 3$, $\sigma_r = 0.1$.

The rest of parameters involved in the algorithm are the same parameters used in configuration *C*.

Based on Table 6 mentioned above we can draw the following conclusions:

1. As Table 6 shows, the smallest error is obtained by setting the vocabulary size equal to 100.

2. The value of this smallest error of the performance measures is lower than the baseline error with a the p-level<0.001 (less than 0.05). Therefore the results obtained with this parameters combination are better than the baseline, even though the variance is bigger.

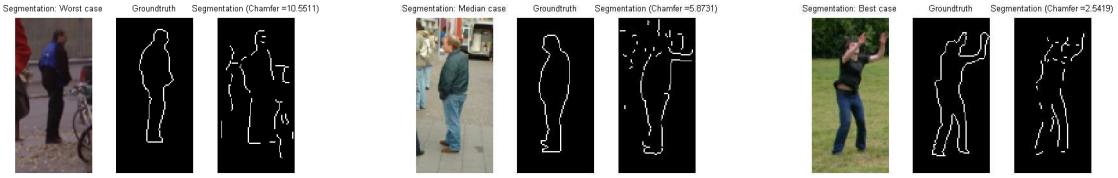
3. Finally, the worst, median and best case (according to the Symmetric Chamfer distance measure) of the segmentation results obtained with a vocabulary size equal to 100 are presented in Figure 17. Henceforth, this configuration will be referred to as configuration *D*.

The conclusions of these experiments are the following:

- The parameter configuration *D* improves the algorithm performance.
- Likewise, the segmentation results in Figure 17 show a qualitative improvement, because clutter segments are reduced. Despite the fact that the Chamfer distance (error) for the best case of the parameters configuration above is bigger than the error for the best case of the configuration *C*, the error of the median and the worst case are smaller.

Number of clusters	**BoW-Background model (set_1)**		
	Mean	**stdev**	**F ratio**
80	6.4744	1.9473	F=87.55
100	**6.1033**	**1.965**	F=118.69
120	6.5582	1.9155	F=82.8

Table 6: Quantitative results using bilateral filtering to preprocess training samples. p-level<0.001. Optimum parameters are marked in bold.



(a) Worst case (b) Median case (c) Best case

Figure 17: The worst, median and best case in the segmentation result. Using a bilateral filering, 100 clusters and BoW-background set_1.

4. Datasets

As introduced in Section 2, the developed system uses three different image datasets for: training, validating and testing. The first two are aimed at constructing and tuning a model and the latter one to test the model at extracting shapes. All the sets are composed of positive samples, and as previously mentioned, in one of the proposals we also make use of background images.

The image datasets are taken from the INRIA person dataset. This dataset was presented by Dalal and Triggs (Dalal and Triggs, 2005) and it is one of the most used datasets to evaluate the performance of human detection algorithms.

The datasets are detailed as follows:

- Training set. A set of 1038 images (519 positive training samples and their mirrors) used for obtaining the pedestrian shape model. Samples from the original INRIA-TRAIN dataset containing padding pixels and being very similar to others have been discarded.
- Validation set. A set of 50 samples (25 and their mirrors) from INRIA-TRAIN, which are used to tune the parameters of the training step (Section 3).
- Test set. 200 testing samples (100 testing samples and their mirrors) taken from INRIA-TEST are used to assess the performance of the system (Section 5).
- Background set. A set of 4152 images (negative training samples, taken from INRIA-TRAIN) is used to generate a background model. This set is divided in four subsets of 1038 negative training samples to create four background models (Section 3).

Figure 18 shows some examples of each of the datasets used in this project.

5. Experimental Results

The proposed algorithm is tested by using the pedestrian shape models from the configurations A, B, C and D (described in Section 3) and evaluated following the methodology explained in Section 3.1. The proposed algorithm has been implemented in MATLAB Ver. 7.6.0.324. The experiments have been done on an AMD Turion(tm) 64 X2 TL-62 2.10 GHz system (2 Gb of RAM).

Table 7 shows the mean and standard deviation of the performance measures for pedestrian shape models A, B, C and D. Additionally, the mean and standard deviation of the baseline and the one-way analysis of variance (ANOVA test) evaluation with respect to the respective baseline is shown (p-level and F ratio). A selection of qualitative results are also shown in Figure 19.

At the view of these quantitative and qualitative results some insights about the algorithm can be highlighted:

Comparing the performance measures of these models with respect to the baseline (presented in Section 3.2), it turns out that their error is lower than the baseline one. The

Figure 18: Images from the INRIA person data set used in the different datasets of this report. Each row shows three samples an its mirrors per each of the datasets used in this project: (a) Training, (b) validation, (c) testing and (d) background samples (these do not have mirrors).

pedestrian shape models							
A		B		C		D	
Mean	**stdev**	**Mean**	**stdev**	**Mean**	**stdev**	**Mean**	**stdev**
7.6079	1.6671	7.8719	1.7256	7.1826	1.8611	**6.8943**	**1.9082**
F=68.56		F=41.12		F=106.82		**F=141.72**	

baseline	
Mean	**stdev**
8.8698	1.3661

Table 7: Baseline and quantitative results for pedestrian shape models A, B, C and D. p-level<0.001. Optimum parameters are marked in bold.

p-level of less than 0.05 indicates that the probability that the differences of the means is random, is much less than 5%. Therefore, the difference in the means is significant taking into account the differences of the variance. As expected from the analysis detailed in Section 3, the smallest error is achieved by configuration D (the best parameters configuration during the tuning).

As Figure 19 illustrates,the final results performed using the model D provide a rough silhouette of the pedestrian in which most of the clutter is removed. As can be seen, the algorithm is still subject to improvement, so a deeper study on the nature of outliers would for sure improve the results. In the case of the missing shape, the errors correspond to either a clusters mismatching or poor contrast. In spite of these errors, it is clear that the information provided by this algorithm at this very stage of development is already promising for a number of algorithms. For instance, the major boundaries of pedestrians are clearly identified, which can be used to a finer color/edge analysis for tracking. In addition, since a large proportion of the provided edges really correspond to the pedestrians' silhouette, the verification carried out in the same module than refinement could also take advantage of our proposal.

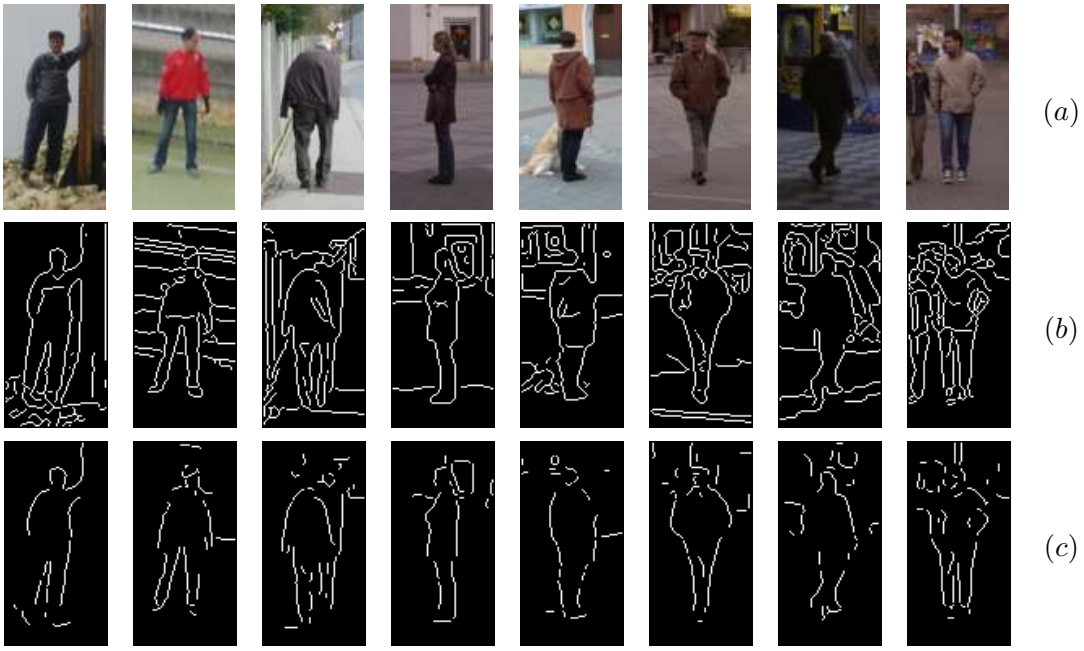


Figure 19: Selection of test results using pedestrian model D with minimum errors. (a) Original images. (b) Baseline results. (c) Segmentation results.

6. Conclusions and Future Work

In this report we have overviewed the state of the art in pedestrian shape extractions for pedestrian protection systems. As has been seen, all the proposals are subordinated to a shape annotation stage, which in most of the cases is time consuming and subject to errors. Our proposal introduces an algorithm that extracts pedestrian shapes from classification bounding boxes without the requirement of an explicit shape mask annotation step. We can summarize the outcomes of the Master Thesis in three main points:

1. The proposed algorithm builds a pedestrian shape model using a bag of words representation. Additionally, a background model has been utilized and a bilateral preprocessing step have been proved to be helpful in the improvement of the performance. Furthermore, the influence of the parameters involved in the algorithm has been studied and one optimal parameter configuration has been found.

2. We have made use of the state of the art tools, such as the bag of words representation. Furthermore, in order to eliminate clutter, the bilateral filtering is applied to smooth images while preserving edges. In addition, the Gradient-CCL algorithm has been developed.

3. The experimental results demonstrate that the proposed technique provides a shape likely to be exploited by e.g., tracking or verification algorithms, as it was suggested in Section 1.

In order to refine and enhance the outcome of the proposed algorithm, the following variations and extensions are proposed for future work:

1. Regarding the model construction, the possible variations of the algorithm include:

    - Segments extraction: as a result of illumination and contrast problems in the image, some edges are not detected by Canny. This could cause edges that belong to the pedestrian silhouette to be missing. As a possible solution, some methods, such as Meanshift, region growing, or watershed, could be applied to obtaind the regions present in the images and afterwards the correspondent contours.
    - Line segments definition: in order to better identify the segments that describe the pedestrian silhouette a method to localize and detect junctions can be applied, such as the algorithms proposed in (Maire et al., 2008) or (Haron et al., 2003). Likewise, a connected pair-wise segments method could be used. For instance, using pairwise constraints between the human body parts (Ren et al., 2005).
    - Segments descriptors: a different representation such as B-splines might be used to describe the identified segments.
    - Vocabulary construction: a new clustering algorithms could be utilized to obtain the vocabulary. Some of this algorithms are Spectral Clustering (Luxburg, 2006) or Agglomerative Clustering (Leibe et al., 2008).

- Bag of words representation: employ a different distance measure to assign each descriptor (segment line) to the closest visual word in the vocabulary. For instance, applying the Chi-square distance instead of the Euclidean or Mahalanobis distances. Partitionate the training samples into diferent clusters of distinct body poses.

2. Shape Retrieval

   - Segment matching with model: probe the use of a Gaussian classifier, such as the proposed in (Wölfel and Ekenel, 2005), to match the line segments (extracted from a validation or test sample) with the closest visual word in the pedestrian shape model.

In addition to the experiments in Section 3 that were used to tune the parameters involved in the proposed algorithm, variations of the following parameters could be taken into account: experiment with a wider range of Gradient-CCL thresholds, test with diferent vocabulary sizes, widen the range of the bilateral fitering parameters. Furthermore, in order to evaluate the proposed algorithm, a comparative study could be done based on dissimilary measures, such as recommended by (Chabrier et al., 2008).

Finally, it is suggested to conduct a comparison with existing state of the art shape extraction algorithms such as (Martin et al., 2004), (Gavrila and Munder, 2007), (Ferrari et al., 2008) or (Leibe et al., 2008).

We are looking forward to further explore the approach presented here within our future work.

References

H. G. Barrow, J. M. Tenenbaum, R. C. Bolles, and H. C. Wolf. Parametric correspondence and chamfer matching: Two new techniques for image matching. In *Proc. Int'l Joint Conf. Artificial Intelligence (IJCAI77)*, pages 659–663, 1977.

H. Cao, N. Ohnishi, Y. Takeuchi, T. Matsumoto, and H. Kudo. Fast human pose retrieval using approximate chamfer distance. *Transactions of the Institute of Electrical Engineers of Japan. C(2006)*, 126(12):1490–1496, 2006.

S. Chabrier, H. Laurent, C. Rosenberger, and B. Emile. Comparative study of contour detection evaluation criteria based on dissimilarity measures. *EURASIP Journal on Image and Video Processing*, 2008:1–13, 2008. Available at `http://www.hindawi.com/journals/ivp/2008/693053.abs.html` (Visited on July 14, 2009).

N. Dalal and B. Triggs. Histograms of oriented gradients for human detection. In *Proceedings of the 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*, volume 1, pages 886–893, San Diego, CA, USA, 2005. IEEE Computer Society.

V. Ferrari, L. Fevrier, F. Jurie, and C. Schmid. Groups of adjacent contour segments for object detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 30 (1):36–51, 2008.

D. M. Gavrila and S. Munder. Multi-cue pedestrian detection and tracking from a moving vehicle. *International Journal of Computer Vision*, 73(1):41–59, June 2007.

D. Gerónimo, A. M. López, A. D. Sappa, and T. Graf. Survey of pedestrian detection for advanced driver assistance systems. *IEEE Transactions on Pattern Analysis and Machine Intelligence (in press)*, 2009.

H. Haron, D. Mohamed, and S. Mariyam Hj. Shamsuddin. Extraction of junctions, lines and regions of irregular line drawing: The chain code processing algorithm. *Teknologi D.*, 38(D):1–28, 2003.

R. Jain, R. Kasturi, and B. G. Schunck. *Machine Vision*. McGraw-Hill College, 2001.

D. Lanman. Bilateral filtering, September 2006. Code available at `http://www.mathworks.com/matlabcentral/fileexchange/12191` (Visited on July 20, 2009).

B. Leibe, A. Leonardis, and B. Schiele. Robust object detection with interleaved categorization and segmentation. *International Journal of Computer Vision Special Issue on Learning for Recognition and Recognition for Learning*, 77(1-3):259–289, 2008.

U. V. Luxburg. A tutorial on spectral clustering. Technical Report TR-149, Max Planck Institute for Biological Cybernetics, Department for Empirical Inference, August 2006.

J. B. Macqueen. Some methods of classification and analysis of multivariate observations. In *Proceedings of the Fifth Berkeley Symposium on Mathematical Statistics and Probability*, volume 1, pages 281–297, Los Angeles, California, USA, 1967. University of California Press.

M. Maire, P. Arbelaez, C. Fowlkes, and J. Malik. Using contours to detect and localize junctions in natural images. *Computer Vision and Pattern Recognition (CVPR)*, pages 1–8, June 2008.

D. R. Martin, C. C. Fowlkes, and J. Malik. Learning to detect natural image boundaries using local brightness, color, and texture cues. *IEEE TRANSACTIONS ON PATTERN ANALYSIS AND MACHINE INTELLIGENCE*, 26(5):530–549, MAY 2004.

A. Rajaraman and J. D. Ullman. Data mining lecture notes (cs345a, winter 2009): Topic clustering, January-March 2009. Available at `http://www.stanford.edu/class/cs345a/` (Visited on July 29, 2009).

X. Ren, A. C. Berg, and J. Malik. Recovering human body configurations using pairwise constraints between parts. In *Proc. 10th Int'l. Conf. Computer Vision*, volume 1, pages 824–831, Beijing, 2005.

A. Bosch Rué. *Image classification for a large number of object categories*. PhD thesis, Universitat de Girona, 2008. Supervisors: Andrew Zisserman and Xavier Muoz Pujol.

C. Tomasi and R. Manduchi. Bilateral filtering for gray and color images. In *Proceedings of IEEE International Conference on Computer Vision (ICCV 98)*, pages 836–846, New Delhi, January 1998. IEEE Computer Society.

UNECE. Statistics of road accidents in europe and north america, 2005. Geneva, Switzerland.

WHO. Global status report on road safety: Time for action. World Health Organization, 2009. 1211 Geneva 27, Switzerland.

M. Wölfel and H. K. Ekenel. Feature weighted mahalanobis distance: Improved robustness for gaussian classifiers. *13th European Signal Processing Conference (EUSIPCO 2005)*, September 2005.

J. Zhang, R. Collins, and Y. Liu. Representation and matching of articulated shapes. In *Proceedings of the 2004 IEEE Conference on Computer Vision and Pattern Recognition (CVPR'04)*, volume II, pages 342–349, June 2004.