Object Fingerprinting

Stefan Renkema

September 2009

Master Thesis
Artificial Intelligence
Department of Artificial Intelligence
University of Groningen, The Netherlands

Internal Supervisor
Dr. Marco Wiering (Artificial Intelligence, University of Groningen)

External Supervisor
Dr. Ir. Sander van der Meer (Computer Vision & Statistics, TNO Delft)

university of groningen / faculty of mathematics and natural sciences / artificial intelligence

Contents

Abstract

To allow tracking of vehicles across large stretches of highway, it is necessary to use multiple cameras. The usage of such camera networks poses the problem of accurately reacquiring each individual vehicle, as it leaves the receptive field of the one camera, and enters the other. The usage of low cost, low resolution cameras do not allow for license plate recognition. Therefore, a purely vision based solution to this problem is to extract visual features from each vehicle to create an object fingerprint. This fingerprint can then be used to reidentify vehicles as they enter the next camera's image. In this thesis several computer vision methods are explored for their ability to tackle the object fingerprinting problem. Moreover, an ensemble of those methods is created that surpasses the performance of the best individual technique.

Acknowledgements

Firstly I would like to thank my supervisors. Marco Wiering for his feedback for the creation of this thesis, ideas to further explore possible techniques and near instantaneous e-mail responses. Sander van der Meer for his patience in showing me around at TNO and familiarizing me with the systems used and practical advice. I would also like to thank the people at Computer Vision & Statistics department, for their help in complicated problems and welcoming atmosphere. To Maaike I also owe a great deal of thanks, for her love, understanding and patience. I would also like to thank my friends for their comradery and comic relief. Last but most certainly not least I wish to thank my parents, for the relentless support I have received from them over my many years of studying and always believing in me.

“Als het leven geen zin heeft, dan máákt het maar zin!”

-Gummbah

Chapter 1

Introduction

1.1 Object Fingerprinting

Increases in computing power have enabled computers to aid in automating video based monitoring. Whereas traditionally human operators would have to observe a large array of displays, computers step in more and more to assist and automate. An example of computer assistance can be drawing attention of a human operator to certain events, such as aggression [1], to aid in security tasks. Other applications may be more large scale and integrated, such as tracking vehicles in camera networks[1]. Such tracking may be done for several reasons. With a complete track of all vehicles that passed through a camera network an analysis can be made on how traffic flows through a road net. With accurate information about traffic flows, adjustments can be made to the road net to combat congestion. The effect such adjustments have can then again be monitored by the same camera network, making it an integral part of the process.

In order to come to an accurate model of traffic flows, vehicles have to be reliably tracked between cameras. This can be done by location and time information alone. However, this means that as the distance between cameras increases, the accuracy of the tracks will deteriorate. Unrecorded changes in the vehicles' trajectories lead to inaccuracies with passing on the correct identity of vehicles from one camera to the next. One solution is to place as many cameras as needed to reduce the problem to an acceptable level, another is to use visual features to re-identify vehicles. An obvious approach would be to use the obligated license plates as distinctive features, but this puts quite severe demands upon the specifications of the cameras used. When the cameras used do not yield images from which the license plates can be read automatically, another set of visual features has to be used. To overcome the loss of distinctiveness of the license plate, as much of the vehicle's visual features have to be incorporated into a distinctive descriptor for each vehicle; an object fingerprint. For reasons of practicality the object fingerprints should be easily comparable, for example through a simple Euclidean distance measure, without any further processing.

This thesis will explore several possible techniques for object fingerprinting in the domain of vehicles on a highway. The starting point for this thesis lays with ongoing research into the mobility domain at the Computer Vision and Statistics department of TNO Science and Industry. Since this project has been in progress for some time, there is an architecture in place which does a lot of the pre-processing of the raw video data, most important of which for the fingerprinting task is the detection of vehicles in videos. These result in a detection window, containing the detected vehicle. From this window the features are extracted that make up the fingerprint. Although object fingerprinting is the task at hand in this paper, the techniques used are also common in other computer vision applications; object detection and object recognition. With object detection the job is to either report whether a given object is present or not, or to report the likelihood of presence. With object recognition the goal is to report the nature of an object given that it is present in some input, usually for objects of different classes. Object fingerprinting is defined here as object recognition when considering only objects of the same class. This may seem as an insignificant distinction, but it may very well be that to distinguish objects of different classes, such as bicycles from cows, other parameters and techniques work better than when comparing vehicles with other vehicles.

[1] www.tno.nl/vbm

1.2 Earlier Work

Earlier work on object fingerprinting in the vehicle domain has focussed on aerial footage and employed mostly holistic recognition approaches [2]. The novelty of the contribution made in [2] lays in the introduction of using local features; SIFT [3] and PCA-SIFT [4] to achieve reacquisition in roadside smart camera networks. There each of the smart cameras performs its own processing and then communicates the results. Because of communication bandwidth limitations it was preferable to have as small fingerprints as possible to reduce broadcast needs, maximizing the ratio of reacquisition scores to transmitted bits. In our system central processing is assumed therefore the latter restriction is not taken into consideration.

This thesis continues research on object fingerprinting with what is suggested as future work in [2]; the search for further features and the inclusion of colour information. Moreover, an ensemble method will be used to derive a combination of features that should surpass the best individual method's performance. As of yet there is no consensus on how to represent colour in computer vision, as testified by the large number of colour spaces available, a wide selection of which are described in [5].

1.3 Research Questions

Recommendations to further the state of the art in object fingerprinting as offered in [2] include the search for further features and a manner in which to include colour information into object fingerprinting. Therefore two research questions can be formulated:

Can employment of previously unexplored and perhaps colour based features generate a significant increase in reacquisition scores for object fingerprinting?

When provided with an array of techniques, each aimed at the same task, it is interesting to explore the possibility of combining the outputs of each individual method to further boost performance. Therefore the second research question is:

Is it possible to create an ensemble of object fingerprinting features, that together surpass performance of the best individual feature?

1.4 Structure of this thesis

The remainder of this thesis is structured as follows; Chapter 2 describes the features used and the manner in which the fingerprints they produce are compared. Also the ensemble technique used to combine the feature outputs is presented there. Chapter. 3 describes the data and corrections applied to it to improve fingerprinting performance. The results of the experiments with the different techniques are reported in Chapter 4. In Chapter 5 conclusions are drawn based on the findings in this thesis, and recommendations are made for future work.

Chapter 2

Methods

2.1 Scale Invariant Feature Transform (SIFT)

A very popular computer vision technique that has greatly influenced its field since it was first devised is the Scale Invariant Feature Transform, SIFT [3]. In short it does two things, localising positions, keypoints, in object images that can consistently be re-localised when the same object is presented again with some transformation, and describing these keypoints in such a way that it can be reliably separated from other keypoint descriptors.

2.1.1 Scale Space

SIFT variations all have one thing in common, the keypoint detection mechanism. Using Scale Space Theory, the image at hand is halved in size several times, each resulting in a so called octave, making up a pyramid with the original sized image as its base. By blurring each octave several times through convolutions with increasingly large Gaussian kernels levels within each octave are created. A fixed number of levels go into an octave. This way a space is created ranging from small to large scale kernels, hence the name Scale Space Theory. By subtracting adjacent levels in scale space Difference of Gaussian (DoG) levels are created, as illustrated in Figure 1. By comparing every pixel value in a DoG level with its eight neighbours in the same level and the eighteen neighbours in adjacent levels extrema in DoGs are detected, where the descriptors will be calculated. For a more accurate localisation of the extreme valued pixels a second order Taylor expansion is applied to neighbouring pixels. This step is important since as octaves become smaller, the area each pixel covers in the original image increases, and without this sub-pixel localisation, inaccuracies in extrema locations would be introduced.

Keypoints located on edges are filtered, as are low contrast keypoints which are dubbed not stable enough for reliable redetection. Finally the major orientations of the keypoint are determined. There is the dominant direction in which the strongest gradient in pixel values is present in the local image patch and besides that there are the orientations that are close in strength to the dominant orientation, which are also taken into account. The SIFT procedure now continues to compute its descriptors for all of the keypoints that passed the filtering stage.

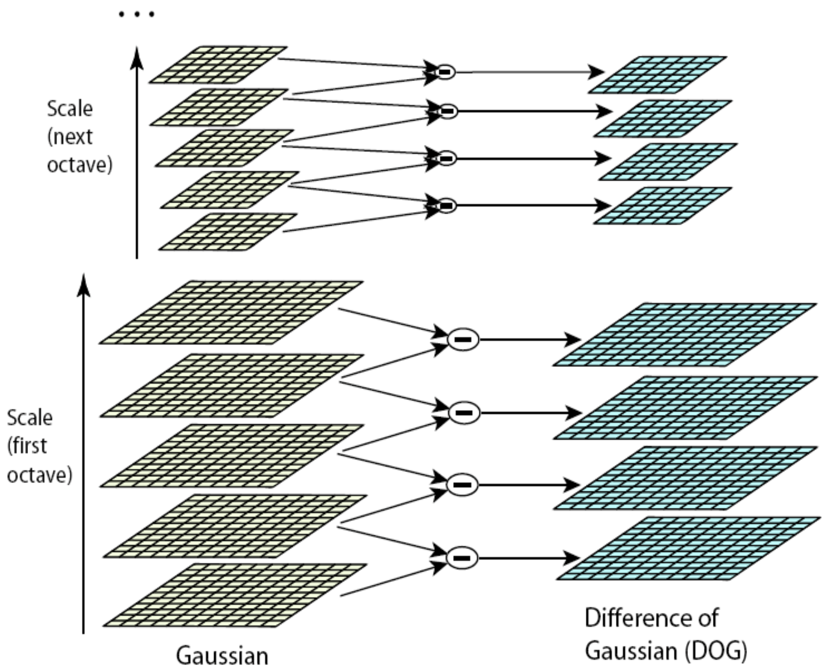

Figure 1: Gaussian Scale space, consisting of the original grey value image at the base, increasingly blurred with Gaussian kernel convolution between levels, and halved in size between octaves. Difference of Gaussian (DoG) maps, are constructed by taking the difference of two adjacent levels. [3]

2.1.2 Descriptor

Given a previously detected keypoint, a descriptor is computed according to the keypoint's location, orientation and scale. The size of the local image patch to be used for feature computation is determined by the scale at which the keypoint was detected and a magnification factor, 3.0 by default. The square patch is divided into 4 x 4 square regions, within which for each of the regions the gradient magnitude and orientation is determined. The gradient magnitudes are weighted by a Gaussian window with standard deviation of half the size of the local image patch, favouring the gradients that lie nearest to the centre of the patch. The final descriptor is computed by accumulating the weighted gradient magnitudes per groups of four squares and categorizing the gradient directions into 8 bins. This results in a vector of length 128 when using the parameters as proposed in [6].

2.1.3 Matching

With SIFT, in order to determine whether or not two images contain the same object, matching on the descriptors is performed. In [6] it is proposed to use Euclidean distance without the usage of a threshold, instead a relative distance between the nearest match and the second nearest match has to meet a certain ratio, 0.8 by default.

The philosophy behind this ratio based matching is that a true matching descriptor will be significantly closer in Euclidean distance than a non-match. In the case where no matches exist, all non-matches will be at non-discriminating distances; the distance to the best match lies within 0.8 times the distance to the second best match. Lowering the ratio typically results in a decrease of matches, while raising the ratio to 1.0 results in all points being matches.

Images which yield the most matched descriptors are typically linked as depicting the same scene or for the Object Fingerprinting task, containing the same object.

2.2 HueSatSIFT

An elegant approach to extend SIFT with colour information is provided by [7]. Here an additional 128 sized descriptor is concatenated to the regular SIFT descriptor by considering HSV colour space [8] along with the regular intensity channel. This results in a 256 sized descriptor. The detection of the keypoints is the same as with regular SIFT. For lack of a better name, the proposed SIFT extension will be named HueSatSIFT here.

According to the authors, HueSatSIFT contributes mostly by reducing false matches when compared to regular SIFT. Due to the lack of colour information two objects may appear to be very similar in grey value images, but when displayed in colour appear nothing alike. This could be the case for vehicles of the same make and model, but with a different paint job.

2.2.1 Hue and Saturation

In the HSV colour space, the H or hue channel describes the wavelength of a pixel, S describes how saturated a colour is; how much grey present. Lastly the V channel gives the intensity of the colour. Contrary to S and V, H is a cyclic value. Starting at 0 degrees with a full red, the hue passes through all colours to end up with a very similar red at 359 degrees.

The angular representation of colour allows to use it to replace the standard SIFT gradient direction in the HueSatSIFT descriptor. The saturation by the same analogy is used to replace the gradient magnitude, since it indicated how strongly a colour is present. Here the intensity channel is not present in the colour descriptor, otherwise intensity would be included twice due to the concatenation with the regular SIFT descriptor, which is of course intensity based.

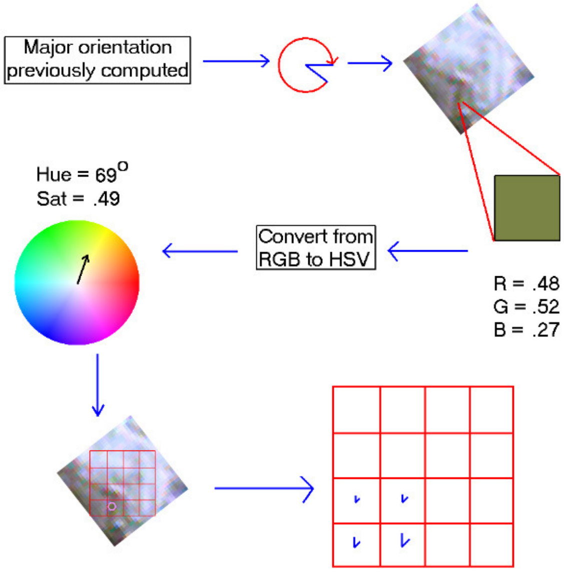

Figure 2: Schematic construction of the HueSatSIFT descriptor [7]

2.2.2 Descriptor

In order to achieve a 128 element colour descriptor the Hue orientations are divided into 8 bins and the patch is divided into 4x4 square regions, the same as with regular SIFT. Figure 2 illustrates the global approach to computing the colour based descriptor. A local image patch as used by SIFT is generated at the location, orientation and scale at which the keypoint was detected, but of course for this purpose the local patch does contain colour information. Once the patch is converted into HSV colour space it is processed pixel wise, according to the hue and saturation of the pixel the appropriate bins of the descriptor are updated.

Because hue and saturation can be influenced slightly by changes in intensity, the neighbouring directional bin is also updated through linear interpolation. A similar linear interpolation is applied to the location of the given pixel, such that the saturation is shared with three neighbouring squares. The resulting concatenated descriptor of length 256 can be matched by the same technique using relative Euclidean distances as with standard SIFT.

2.3 Colour Invariant SIFT

Instead of creating an alternative descriptor it is also possible to perform SIFT keypoint detection on alternative colour spaces, besides just the intensity channel, such as in [9] where on each channel of the HSV colour space SIFT is applied. Another more advanced technique is through the usage of invariant colour models [10, 11] each with specific invariance to shadows, illumination, highlights and noise. In [12] a number of local descriptors based on invariant colour models are compared, with the SIFT descriptor based on chromatic invariants as overall winner. Therefore this method is investigated here for its applicability to object fingerprinting.

2.3.1 Colour Invariance

Invariance to illumination is achieved through converting the acquired RGB colours into the Gaussian opponent colour model. Equation 1 gives the linear transformation matrix, by Gaussian differentiation with respect to the image axis x and y. Gradients independent of the intensity distribution are obtained. This Gaussian differentiation is achieved through convolution with a Gaussian Kernel differentiated with respect to x or y respectively. This independence of intensity means shadows and shading do not affect the colour descriptor. Figure 3 displays a toy image in RGB colours and its representation in the chromatic invariant model.

$$\begin{bmatrix} \hat{E}(x,y) \\ \hat{E}_{\lambda}(x,y) \\ \hat{E}_{\lambda\lambda}(x,y) \end{bmatrix} = \begin{pmatrix} 0.06 & 0.63 & 0.27 \\ 0.30 & 0.04 & -0.35 \\ 0.34 & -0.60 & 0.17 \end{pmatrix} \begin{bmatrix} R(x,y) \\ G(x,y) \\ B(x,y) \end{bmatrix}$$

$R(x,y)$, $G(x,y)$ and $B(x,y)$ denote the red, green and blue pixel values at input image position (x,y) . The output image is produced in the opponent colour space; $\hat{E}(x,y)$ is the intensity channel, $\hat{E}_{\lambda}(x,y)$ is the yellow-blue channel and the red-green channel is given by $\hat{E}_{\lambda\lambda}(x,y)$

Equation 1: Transformation of RGB into opponent colour space [10].

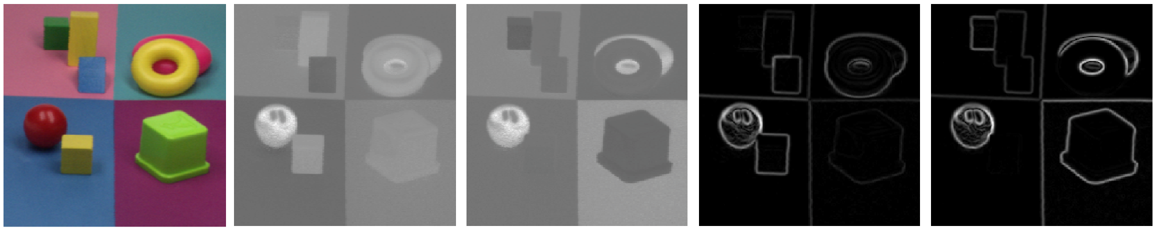

Figure 3: Left to right, original image, $\hat{C}_{\lambda}$ first spectral derivative, $\hat{C}_{\lambda\lambda}$ second spectral derivative and their magnitudes $\hat{C}_{\lambda w}$ and $\hat{C}_{\lambda\lambda w}$.

2.3.2 Invariant Descriptor

To achieve the extraction of SIFT descriptors the same scale space extrema search as described in Subsection 2.1.1 is performed, firstly on the greyscale channel. If an extreme position is found, but fails to pass a minimal contrast threshold, the position is inspected another two times for the yellow-blue and red-green opposite colour channels. This gives the position two more chances as being included, and therefore, on average, will result in more keypoints compared to regular SIFT. For all of the keypoints that pass through the filtering process a descriptor as described in Subsection 2.1.2 is computed. This results in three 128 sized descriptors, one for the local greyscale gradient, and an additional two for the colour gradients as computed in Equation 2.

$$\hat{C}_{\lambda j} = \frac{\hat{E}_{\lambda j}\hat{E} - \hat{E}_{\lambda}\hat{E}_{j}}{\hat{E}^2}, \hat{C}_{\lambda\lambda j} = \frac{\hat{E}_{\lambda\lambda j}\hat{E} - \hat{E}_{\lambda\lambda}\hat{E}_{j}}{\hat{E}^2}, \text{ with } j = \{x, y\}$$

Gradients $\hat{C}_{\lambda j}$ and $\hat{C}_{\lambda\lambda j}$ for the horizontal (x) and vertical (y) directions. $\hat{E}$ is the intensity channel of the opposite colour space, $\hat{E}_{\lambda j}$ the directional Gaussian derivative of the red-green channel, as is $\hat{E}_{\lambda\lambda j}$ for the blue-yellow channel. The gradients are normalized by the squared pixel intensities $\hat{E}^2$

Equation 2: Differentiation producing the invariant colour gradients [12]

The magnitudes of the colour gradients are given by Equation 3. Together the three descriptors result in a descriptor of length 384 per keypoint. Despite the difference in size the same technique, based on relative Euclidean distances for matching descriptors can be used.

$$\hat{C}_{\lambda w} = \sqrt{\hat{C}^2{}_{\lambda x} + \hat{C}^2{}_{\lambda y}}, \ \hat{C}_{\lambda\lambda w} = \sqrt{\hat{C}^2{}_{\lambda\lambda x} + \hat{C}^2{}_{\lambda\lambda y}}$$

Absolute value of the gradient magnitudes obtained from Eq. 2 produce the gradient magnitudes for the invariant colour SIFT descriptor

Equation 3: Gradient magnitudes for both colour channels [12]

2.4 PCA-SIFT

Yet another variation on SIFT comes in the shape of PCA-SIFT [4], again the same method for detecting keypoints is employed as with regular SIFT. However the descriptor which is computed at the given location, orientation and scale is quite different.

2.4.1 Principal Components Analysis

Principal Components Analysis (PCA) is a commonly used technique to reduce the dimensionality of data, with minimal loss of information [13]. By computing the covariance matrix of some acquired data, normalized by the data mean, and then performing Eigen value decomposition on the resulting matrix, an Eigen space is obtained. Then onto this Eigen space a projection can be made with a new data instance which reduces the dimensionality of the given data to the number of components specified in the construction of the Eigen space.

Inherent to this approach lays the assumption that the data is a Gaussian distribution in which mean and standard deviation are meaningful, and moreover, large covariances actually yield important information. Also PCA is restricted to orthogonal linear combinations. Despite this, PCA is found to be useful combined with SIFT in [2] for object fingerprinting.

2.4.2 Descriptor

After keypoint detection, which supplies PCA-SIFT with local image patches at the detected scale and rotated to the dominant gradient orientation, a descriptor is computed by means of PCA.

In [4] a patch of 41 by 41 pixels is proposed, from which two gradient patches, horizontal and vertical, can be extracted sized 39 by 39 pixels. Despite the fixed size, the information within the patch may have been obtained from a far larger or smaller region in the original image, since the local image patch obtained at a certain scale is resized to the desired patch size. Combining both 39x39 patches from the vertical and horizontal gradients produces a feature of length 39x39x2= 3042. To minimize the effect of intensity changes the vector is normalized to unit length. The goal is to reduce the 3042 values to the ones containing the most information, choosing 36 of these principal components is the default number, as it was found by the authors of [4] to be of equal distinctiveness as the original sift descriptor. Custom patch sizes can be used of course and also the number of principal components can be altered. Any change in either the number of principal components or the patch size requires recomputation of the Eigen space. Figure 4 shows the first 36 principal components for the local patches. The Eigen space is obtained through the application of PCA on the above mentioned 3042 sized feature vectors as collected from the training set. Matching descriptors from different images is proposed in [4] to be achieved through a simple thresholding function. The level of the threshold is to be chosen through analysis of the recognition scores. Here the same relative distance matching as with SIFT will be used, since this allows for a more direct comparison to the other SIFT-based approaches.

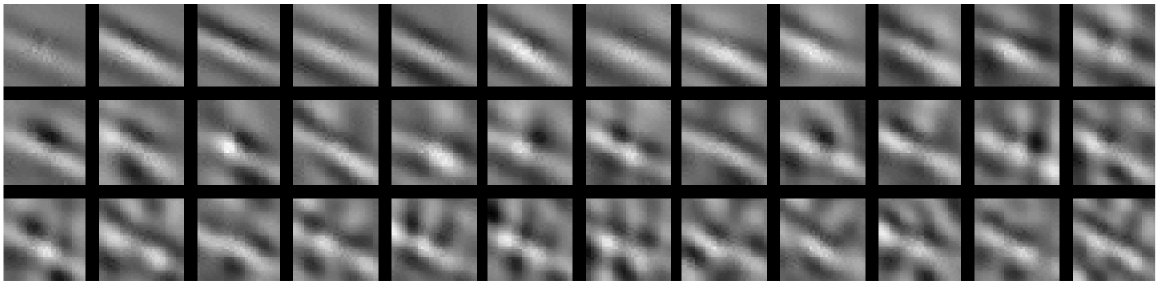

Figure 4: Left to right, top to bottom, the first 36 principal components obtained from PCA applied to the first 10,000 keypoint patches of the 09.45 dataset.

2.5 Bag of Visual Words

The bag of words approach is quite common in document classification techniques, such as [14] where it is used for detection of spam in weblogs. Instead of parsing the grammar of the sentences in a document, a holistic approach is used that simply considers all words present regardless of their location. To adopt the bag of words approach in machine vision it is required to formulate a vocabulary of visual words and a means of detecting these words in an image, [15] describes an example of this approach.

The first problem that arises is the question of what constitutes a visual word in the first place. In [2] visual words take the shape of clustered PCA-SIFT descriptors, used as a means for reduction in communication overhead between smart-cameras. Here, instead of clustered PCA-SIFT descriptors, regular SIFT descriptors will be used, since the latter yielded far better results in initial testing. The approach used here remains largely the same as in [2], only the nature and the length of the descriptor used, 128 instead of 36 is different. The principle is to cluster a large collection of SIFT descriptors and assign unique index numbers to the leaves. This results in a reduction in information transport from 128 values to just a single number per descriptor. A vector of unique descriptor indexes then make up the fingerprint for a specific object, the number of matching indexes between two images determines the strength of the match. Clusters are acquired through hierarchical k-means clustering, with k=3 and choosing random descriptors as the initial cluster centroids. Clustering is continued until the number of clusters equals the size of the descriptor set; every cluster at the final depth is then occupied by a single descriptor. By selecting different levels in the cluster hierarchy the optimal size of the vocabulary can now be chosen. Figure 5 illustrates the principle of hierarchical k-means clustering.

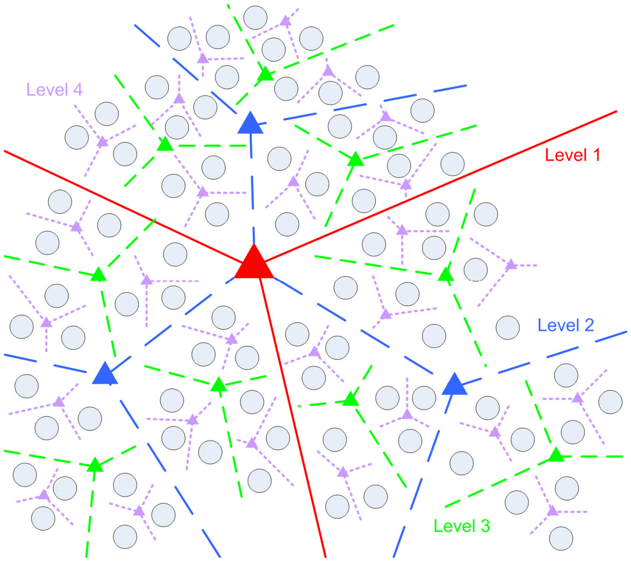

Figure 5: 2D representation of hierarchical k-means clustering, with k=3 at levels 1 to 4 [2]

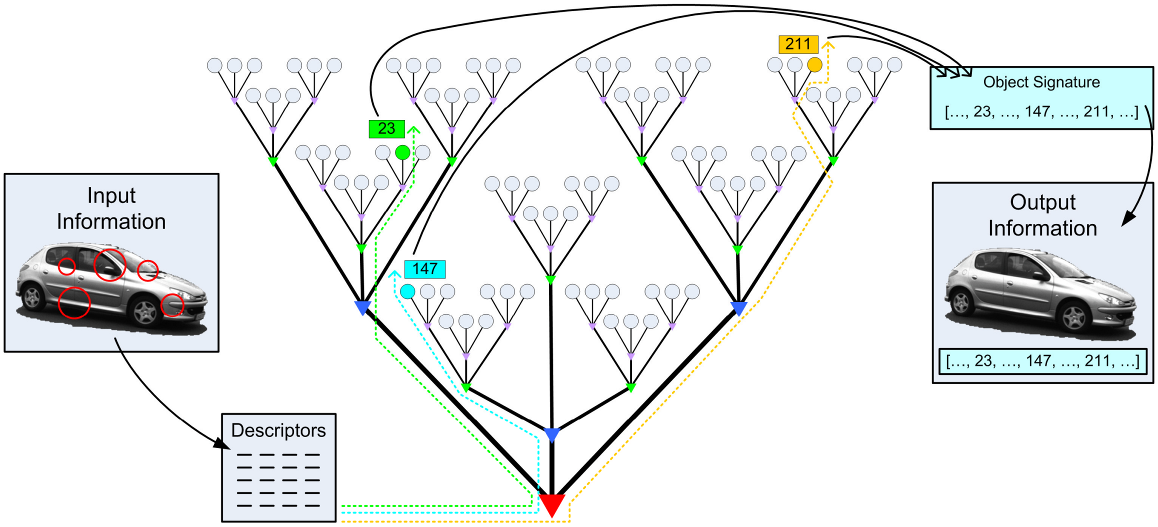

Figure 6: Fingerprint creation from hierarchically clustered SIFT descriptors [2]

The detection of visual words is achieved firstly through scale space analysis as described in Subsection 2.1.1, computing the SIFT descriptor as in Subsection 2.1.2. The unique index number is assigned by matching the descriptor to the ones in the cluster, and adopting the index number of the best match. Although the vocabulary can be very large when the choice is made to use one of the final levels in the cluster hierarchy, the tree representation still allows for efficient retrieval of descriptor indexes. The match between two fingerprints is determined by the intersection between the two vectors of descriptor indexes, the larger the intersection, the more likely the two fingerprints belong to the same instance. Figure 6 gives an example of how a fingerprint is constructed with the Bag of Visual Words approach.

A potentially powerful extension to the Bag of Words approach using SIFT is to use Invariant Colour SIFT descriptors, despite the larger size of the descriptors the same method of clustering can be applied. The reduction in communication overhead is even larger when colour SIFT is used, since the reduction there is from 384 values to one value, helpful for smart camera approaches to employ the expressiveness of colour SIFT without increasing the demand on communication bandwidth. As much descriptors as possible will be put into the vocabularies, while ensuring equal amounts for both the regular and colour invariant SIFT conditions. Also the additional colour keypoint positions may prove beneficial for performance.

2.6 Spatial Pyramid Matching

Whereas holistic approaches by definition discard spatial information, in [16] a method is introduced that includes spatial information into otherwise holistic methods. By dividing an image into several layers of a spatial pyramid, features from portions of images can be compared. Instead of doubling the number of regions per level as in [16], the approach as demonstrated in Figure 7 is used. This causes regions in higher levels to sometimes overlap borders between regions of preceding levels. This is done to reduce any effect of the arbitrary location of region boundaries. The number of levels in the pyramid can be varied to explore the optimal number.

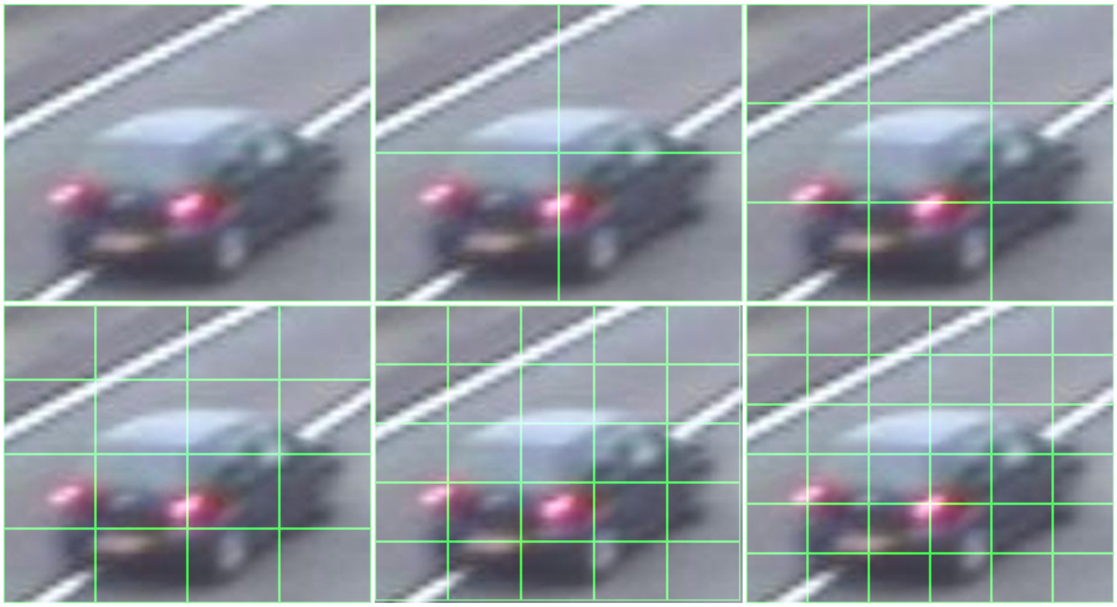

Figure 7: Pyramid levels 0 to 5 and the increasing number of regions per level

Inspiration for the two features to use in the Spatial Pyramid Matching scheme here is derived from the work in [9] where the Spatial Pyramid Matching approach is used for image classification. First a Bag of Visual Words approach is used similar to that of Section 2.5, except SIFT descriptors are acquired in the images not through scale space analysis, but by computing descriptors at fixed interval positions. And instead of only counting unique visual words just once, a histogram of occurrences is constructed. The same will be done here for fingerprinting, albeit on square image patches instead of circular ones. The size of the vocabulary was chosen at 300 words by the authors of [9], but through similar clustering as in Section 2.5 any vocabulary size can be used. The second feature is based on discrete intensity gradient directions. Through convolution with Sobel kernels, the orientation of the gradient at every pixel location within a region is calculated. The orientations as previously computed are matched to the bins of a histogram corresponding to a range of directions from 0 to 2 Pi. The number of bins, and thus the size of the angle they cover, of course can be varied to find the optimal value. Both features result into a histogram each, which after normalizing to unit length are matched through Equation 4. Complete Spatial Pyramid representations are matched by the matching kernel in Equation 5, here with equal weights for all levels.

$$\chi^2(Q,V) = \sum_i \frac{(q_i - v_i)^2}{q_i + v_i}$$

The distance between the two histograms Q and V is determined by their respective elements q_i and v_i for all i.

Equation 4: Matching of histograms through chi-square

$$K(Q,V) = \sum_{l \in L} \exp\left(\chi^2(Q_l, V_l)\right)$$

The distance between two Spatial Pyramid representations Q and V is determined by computing the χ^2 distance between the histograms Q_l and V_l for all pyramid levels l

Equation 5: Matching kernel for Spatial Pyramids

2.7 Colour Co-occurrence Histograms

Colour Co-occurrence Histograms or CCHs are a clever way of combining the colour and spatial layout of an image into a single holistic descriptor. Applications for CCHs are usually found in object recognition domains, such as in [17] and [18]. CCHs combine colour and spatial information by inspecting an input image pixel wise and within a certain radius count what colours co-occur along with the central pixel. This raises the matter of how to represent colour. The simple answer is to use RGB information, cameras perceive colour this way, and monitors use it for projection. However, in the RGB colour space intensity and colour are intertwined. The commonly used HSV colour space however has no such interdependencies among its channels [19, 20], and incidentally it more closely represents how humans perceive colour. For those reasons we will use the HSV colour space here.

Having established a colour space to work with there is still the need for a manner in which to quantize the colours present in an image. Without quantizing the colour space the resulting CCH would most likely exceed the input image in size for the data in the current domain. Also, quantizing inherently increases the tolerance level to re-identify a particular colour. This tolerance is important since between cameras there will be at least slight differences in perceived colour. One possibility is to divide the three dimensional colour space into smaller cubes. These cubes each span a certain range of colours, colours in the input images are then labelled by the index number of the cube that contains them. Although this is an effective way to quantize a colour space there is a drawback; colours that do not occur within a domain are equally well represented as colours that do occur. In practice this means that increasing the size of the CCH, which grows quadratically as more categories of colours are included, yields a smaller effective growth in the number of colours in the domain that are represented.

A way to overcome the waste in colour representation is to use clustering on colours actually present in a training set, instead of quantizing the entire colour space by dividing it into cubes. From a video track the colours present in vehicle detection windows can be gathered, and then be clustered into a desired number of colour templates, each with a unique label. By assigning the unique template labels to colours as found in input images a CCH can be constructed in the usual manner. Clustering here is performed using k-means clustering, which continues until there is no change in cluster center membership for data points. Figure 8 shows all unique colours from the detection windows present in a single 15 minute video, and the trajectory cluster centers follow between iterations. Notice how sparsely the colour space is occupied by the colours actually present in a 15 minute interval. Since the detection windows are not all of the same size, normalization to unit length is applied before comparison using Equation 4. Figure 9 shows an example of a CCH and the corresponding vehicle.

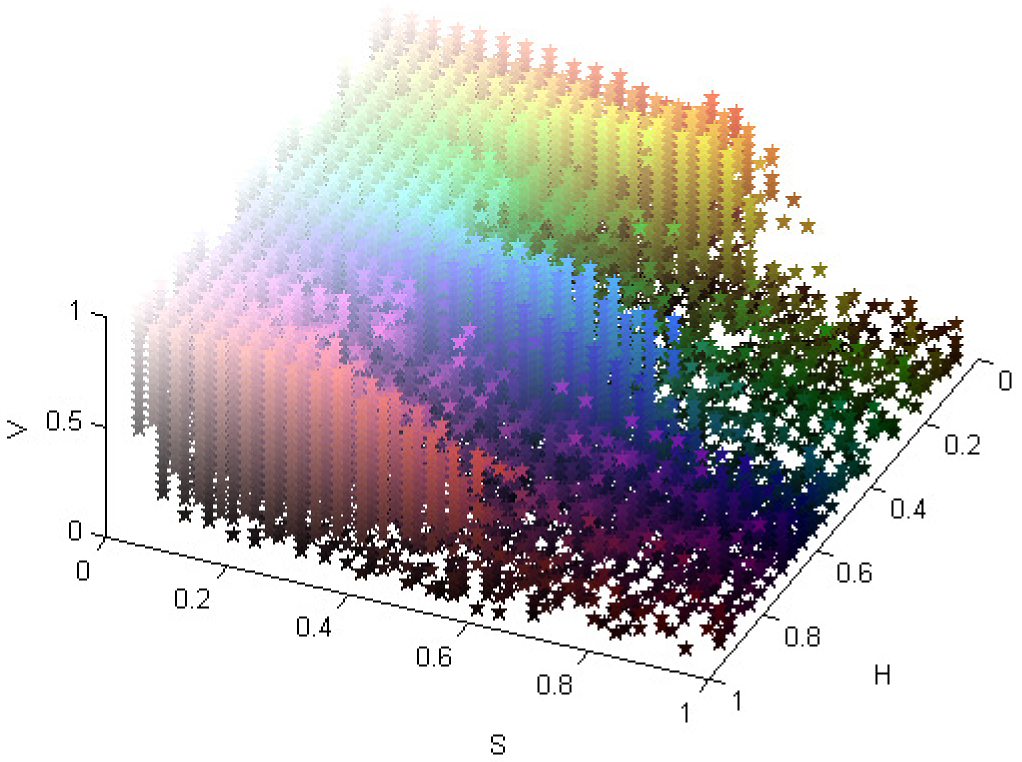

Figure 8: Unique colour instances in detection windows during 15 minutes of video

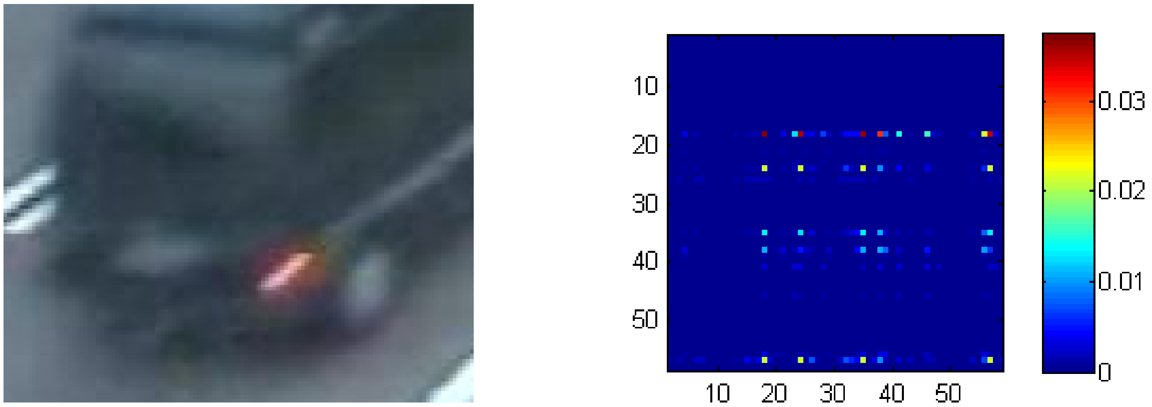

Figure 9: Colour Co-occurrence histogram, for a vehicle image, with radius 20 pixels and 59 colour clusters

2.8 Cortex-Like Mechanisms

A biologically inspired approach to computer vision presents itself in the form of [21] in which the ventral stream of the primate visual cortex is modelled to be used for object categorization. Besides object categorization [21], other examples of successful applications based on this research are presented in [22] and [23] in the context of handwriting recognition, so perhaps for object fingerprinting the method will also prove viable.

The model consists of four layers, S1, C1, S2 and finally C2, which together after processing results in a scale, orientation and position invariant description of an input image, Figure 10 gives an overview.

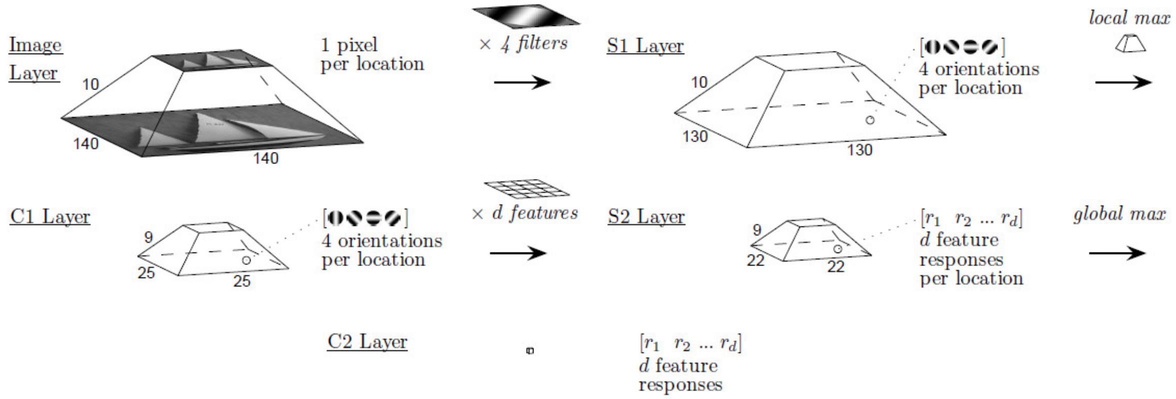

Figure 10: Overview of how an input image is converted into a d-sized descriptor with d previously stored patches through the visual cortex model. Modified from [23]

The first level S1 contains units that simulate the receptive fields of cells in the primate primary visual cortex area V1 with Gabor functions, Equation 6.

$$F(x, y) = \exp\left(-\frac{\left(x_0^2 + \gamma^2 y_0^2\right)}{2\sigma^2}\right)\cos\left(\frac{2\pi}{\lambda}x_0\right),$$

$$with: x_0 = x\cos\theta + y\sin\theta$$

$$and: y_0 = -x\sin\theta + y\cos\theta$$

The orientation of the filter is governed by θ, γ controls the aspect ratio, σ determines the width and λ specifies the bandwidth. Together these variables control the size and shape of the Gabor Filter

Equation 6: Gabor Function [21]

The parameters of the Gabor functions are tuned to fit the results obtained from neurological research. To achieve scale invariance a range of sizes are used, but to keep the total number of Gabor functions tractable for each scale just four orientations are considered, with 16 scales and 4 orientations this results in 64 Gabor functions. At the next level C1, so called complex cells are simulated. These cells display some tolerance to shift and size. This invariance is achieved in the model by gathering the local maxima over position and scale in the S1 level. S2 is the next level, here C1 inputs are compared at corresponding orientations to previously seen C1 patches, as stored in a dictionary. The matching of patches is done by an Euclidean distance based approach, named a Gaussian-like radial basis function, Equation 7.

$$r = \exp\left(-\left\|\mathrm{X} - P_i\ \right\|^2\right)$$

The response r is determined per S2 unit by the input image X and P_i denotes the current prototype.

Equation 7: Gaussian-like radius basis function [21]

The output of the model is obtained at the final level C2; the best match associated with each patch over all scales and positions as computed in S2 is stored in the output vector, which is of the same dimension as the number of patches. The number of samples extracted from the images for both dictionary creation and for recognition at runtime can be varied to find an optimal value. The scale at which C1 features are selected is chosen randomly until the specified number of samples is reached. For construction of the dictionary the input images can be acquired from the relevant domain, or the choice can be made to try and create a universal dictionary. The intuitive option of creating a relevant dictionary also proves to be the most successful in the experiment performed in [21], therefore this approach will be explored here.

The usage of a single image for each vehicle per camera means that the model cannot benefit from multiple training samples, as is the case in [22, 23]. Contrary to the publicly available model[2] that uses a Support Vector Machine for classification, here the C2 fingerprints are matched with a simple 1-NN algorithm. This is done for simplicity, since 1-NN does not require retraining when new instances are introduced, and for a fair comparison to the other methods. All input images are first converted to greyscale and resized to the same size before the model is applied.

2.9 Boosting

Boosting [24, 25] is a process which is know as a meta-algorithm. Instead of actually performing machine learning tasks such as classification, recognition or as is the case here, object fingerprinting it manages the training process to optimize performance of such tasks. Provided with a large number of preferably computationally inexpensive features, boosting combines a collection of so called weak classifiers into a single strong classifier.

The combination of weak classifiers is chosen through several rounds, where in each the best performing weak classifier is added to the set of best performing classifiers from earlier rounds. After every round the weights of the training samples are adjusted, according to the performance of the set of best features; samples which were incorrectly processed are given a higher weight with respect to the other samples. This way, the most difficult samples are used to compile the final strong feature set, intended to provide the strongest possible combination of weak features.

2.9.1 Multiclass Boosting

Typically boosting is applied to two class problems, such as in [26] for the task of face detection or as in [27] to detect pedestrians; either the target class is present or it is not. The object fingerprinting task however, is a multiclass problem, since given the fact that an object is present, it can be any of all identities. For multiclass problems it is possible to conduct a number of 1 vs. all classifications, and then choose the strongest response [28]. But for an unknown number of classes, each time a new class presents itself, the whole boosting process would have to be started from scratch for the new class. Therefore, for object fingerprinting the choice was made to boost weak classifiers for a nearest neighbour classification, similar to what is explored in [29] for handwritten digits. Algorithm 1 describes how nearest neighbour boosting is achieved here. To reduce the time needed for the boosting process, instead of comparing all vehicles against each other, a random selection of thirty vehicles is used, and the correct match of course is also included.

[2] http://www.mit.edu/~jmutch/fhlib

```
Input:  $M$ training samples $(x_i, y_i)$, $i = 1,\dots,M$, image $x_i \in \Re^2$, label $y_i \in \aleph$
        $T$ specifying the number of rounds
        $N \geq T$ Haar-like features $h_j$ for nearest neighbour classification
        $w_1(i) = 1/M$ weights per training sample
For $t = 1,\dots,T$
    1) Calculate feature $h_j$ for each vehicle, perform fingerprinting on selection of vehicles
       with 1-NN using $h_j$ output as fingerprint for all $j$
    2) Determine classification error $\varepsilon_j = \sum_i w_t(i)\{O_j(x_i)\}$ for all $j$ with
       matching function $O_j(x_i) = 0 \; if \; h_j(x_i) = y_i \; or \; 1 \; if \; h_j(x_i) \neq y_i$
    3) Select $h_j$ with lowest $\varepsilon_j$ to add to final classifier
    4) Calculate $\alpha_t = \frac{1}{2}\ln\left(\frac{1-\varepsilon_t}{\varepsilon_t}\right)$, Quality of classifier $h_t$
    5) Update weights $w_{t+1}(i) = \frac{w_t(i)\exp(\alpha_t h_t(x_i))}{Z_t}$, where $Z_t$ normalises to unit length
```

Algorithm 1: Boosting with nearest neighbour classification

So each round among 31 vehicles, the correct one has to be sought out by the nearest neighbour algorithm. Once boosting is completed the optimal number of features can be determined starting with the most successful ones as determined by the quality measure. The used features are so called Haar-like features [2], which are very simple in nature, pixel values in the positive areas are summed together, whereas pixel values within the negative areas count are subtracted, Figure 11 gives some examples. Each feature is resized to span the full sub image of individual vehicles. The sum of the negative and positive areas is the output value of a Haar-like feature, and the values of all features in the final boosted classifier make up the feature vector to be used for fingerprinting by 1-NN matching. Through the usage of integral images [2] Haar-like features can be computed very quickly, in constant time. This speedy computation of Haar-like features is a great benefit, since due to the weak nature of individual features large numbers of them are required. The approach of boosting Haar-like features is commonly applied to detection tasks. This means capturing properties that all objects in the class have in common. The goal here is to try and see if it is possible to use the same kind of features, to capture those properties which vary as much as possible between vehicles, and use it to discriminate one from the others.

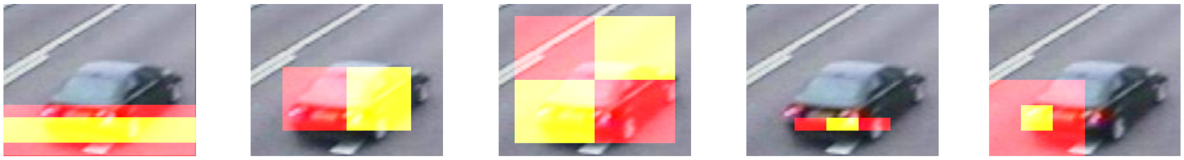

Figure 11: A few Haar-like features projected onto a vehicle, yellow denotes positive areas and red negative ones.

2.10 Feature Ensemble

The methods described in the previous sections can be used on their own to match vehicle identities in different cameras to each other. But combining the outputs of each individual method into a single verdict may potentially be more accurate than the highest individual score. A condition for this however is that there is a spread in the errors made between each classifier involved [30]. If every classifier would produce the same results, it is impossible to come to a higher score than that of a single one. The class of techniques that combine multiple outputs into a single one are know as ensemble techniques. A range of such techniques exists; Bagging [31], Bayes Optimal Classifiers [32] and Stacking [33] are some common examples. With Bagging a bootstrapping procedure is used to improve classifier performance on random combinations of training data. Bayes Optimal Classifiers uses the Bayesian rule to compute the most likely label for an instance based on multiple hypothesis. With Stacking a machine learning technique, such as a neural network [33], is trained with the outcomes of individual classifiers as its input. A very popular ensemble technique is Adaboost, a variant of Boosting which also served as a basis to the technique discussed in Section 2.9. An advantage of Adaboost over the other ensemble techniques is that it shifts the influence training instances have on the outcome; more difficult samples have greater influence. Adaboost returns a quality value alpha for each input methods' classification. This quality can then be used for weighing the votes of the boosted qualifiers. The most successful method will have the largest vote in the final output. Naturally for the final classifier to be any better than the strongest classifier by itself, the sum of the weights of the other classifiers has to exceed the weight of the strongest one. If this were not the case, the ensemble classifier would always produce the same result as the strongest one on its own. Since many of the methods used for fingerprinting here are SIFT-based, it is naive to assume a large spread in errors between classifiers. Adaboost with its scheme to shift weights to more difficult training samples is therefore the one used here, since it aids to ensure diversity in the final classifier.

The algorithm used here to ensemble the different fingerprinting methods is very similar to Algorithm 1. However, instead of Haar-like features, classifications from each of the described methods are used as input. And also instead of nearest neighbour classification, for each of the vehicles in the training set for boosting, the identity each method returns is compared to the ground truth of the data set, which was created manually. Once training is done fingerprinting can be performed with the feature ensemble. Every method has a vote on which vehicle in the second camera it considers to be the best match to the vehicle under consideration. This vote is weighted using the quality α , the identity that gathers the largest amount of α is the one cast by the feature ensemble. Should any of the α values be negative, then all α values are linearly scaled such that the lowest value becomes 1.

Chapter 3

Data

The dataset used was obtained from roadside cameras situated at the national highway A67 near Venlo, The Netherlands in May of 2007. Figure 12 gives an overview of the situation in which the four cameras were positioned. Despite traffic travelling in the other direction is visible, only vehicles on the right side of the road are considered here. Although there is some variation in traffic density, none of the videos used in this thesis display any form of traffic jams. Weather conditions were dry for all videos.

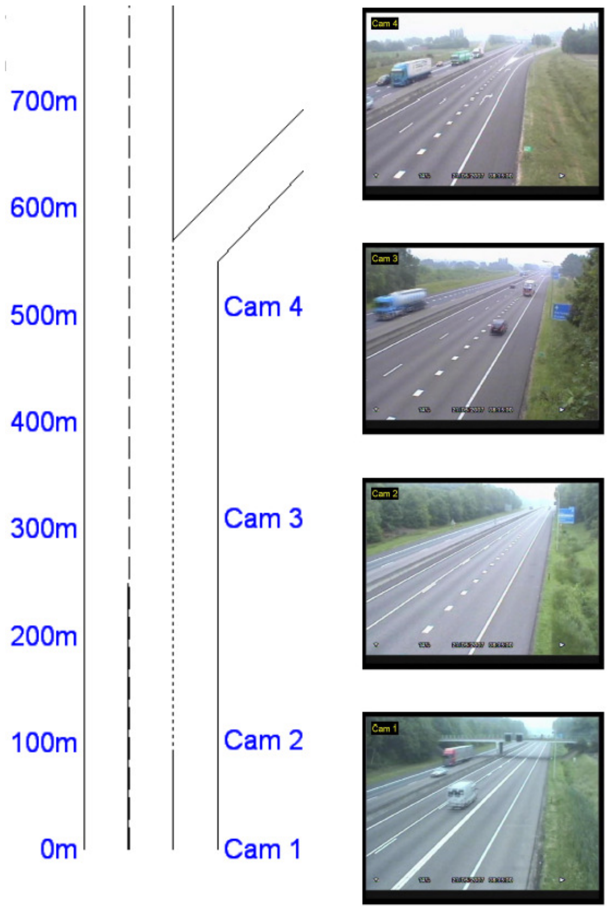

Figure 12: Schematic representation of recording situation

Of the four cameras just the first two are used for experiments with a distance of 105 meters between them, although of course the distance between cameras does not matter for object fingerprinting, as long as both camera positions have only slight differences in lighting conditions. The cameras operated at a resolution 768 by 576 pixels, with automatic white balancing engaged. Figure 13 shows an example of good lighting conditions whereas Figure 14 shows the same camera position during backlight. Clearly computer vision systems benefit from lightly clouded conditions, since these clouds diffuse light more evenly across the sky.

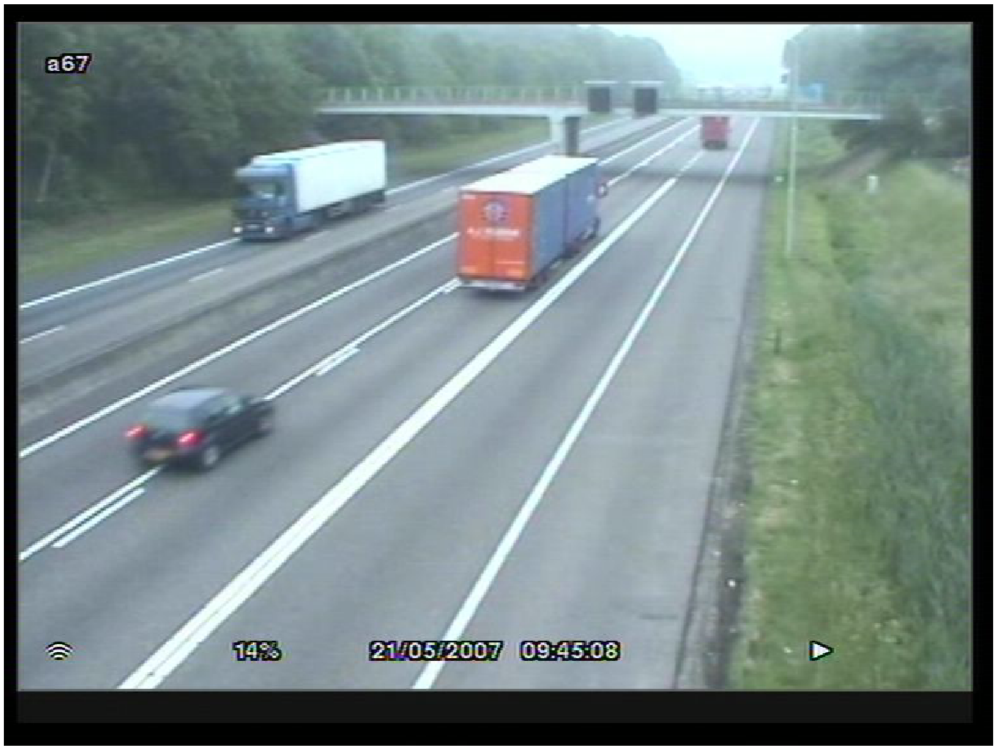

Figure 13: A video frame during good lighting conditions, from the 09.45 data set.

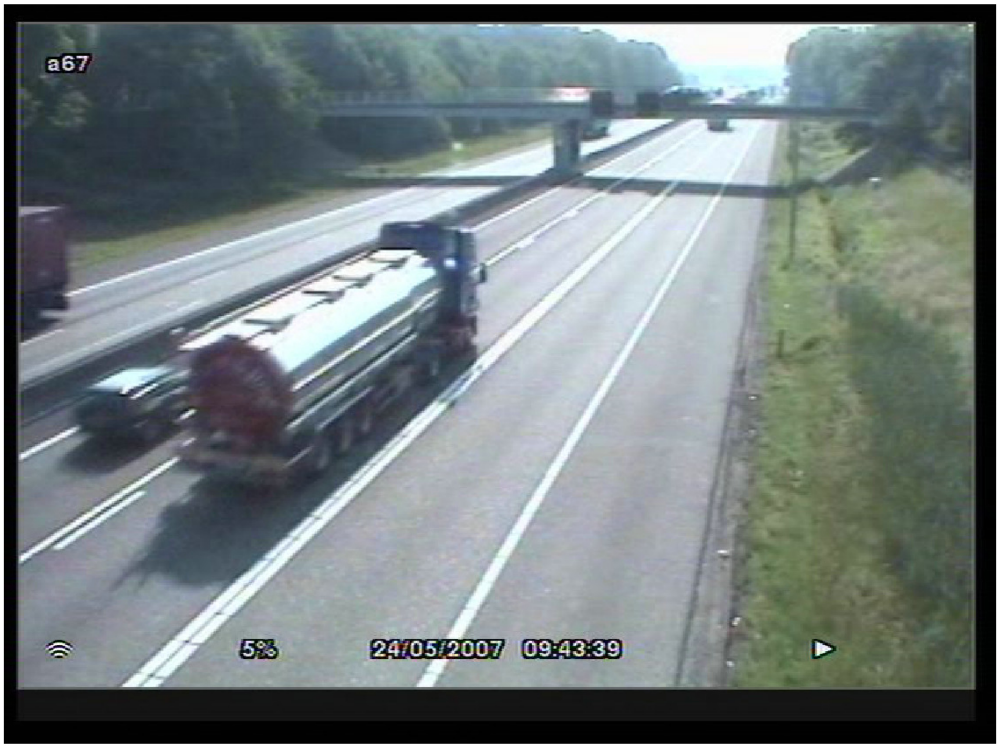

Figure 14: A video frame from the backlight 09.30 dataset, notice the difference in contract when compared to Figure 13

Vehicle fingerprints are extracted from detection windows. Despite the right side of the vehicles are visible due to the camera angle at which was recorded, not as much of the vehicle as possible is included in the detection windows. This is due to the tendency of the detector to focus on the rear of vehicles. To capture as much of the sides as possible the original detection windows are widened, if the size of the window falls below some threshold. This helps to include as much of the vehicle as possible, while attempting not to introduce a lot of background noise. Figure 15 shows a widened and an original detection window alongside each other.

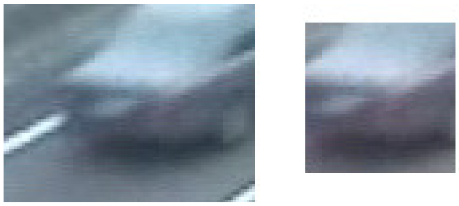

Figure 15: Widened and original detection window of the same vehicle

Because of an unavoidable analogue conversion from one video format to another, wrongly interlaced video frames were introduced. This is most likely due to an asynchrony between the internal clocks in the recording devices. To attempt to correct the introduced interlacing errors a method was devised; based on the observation that correctly interlaced images display lower changes in intensity in the vertical direction than incorrectly interlaced ones. By comparing the given detection window to a window constructed with the even image rows of the original detection and the uneven rows from the preceding and following video frames, the best combination can automatically be selected by choosing the one with the lowest contrast between adjacent rows. This method of deinterlacing maintains the original resolution of the input image. This is a great advantage, since the size of the detection windows are very small to begin with. Figure 16 displays an example of corrected interlacing on a detection window containing a vehicle.

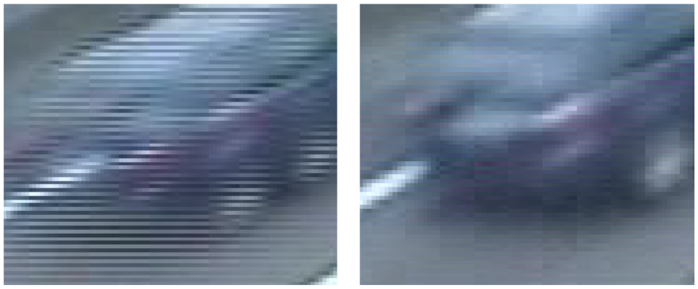

Figure 16: Detection window, before and after correcting the faulted interlacing

Another correction applied to the data before processing is a correction in colours, to try and normalize images between cameras. Each camera appears to have a somewhat different colour calibration, which may influence the accuracy of vehicle reacquisition by introducing variation in the fingerprints. To calibrate the cameras the detection windows of vehicles are used, since the appearance of the vehicles is the only constant shared by all cameras, save for small changes in viewpoint. For each of the cameras in a dataset with good lighting conditions, the median of the RGB channels in the detection windows was computed. To normalize the colours for the detection windows from all cameras the colour channels are adjusted to match the median value. Figure 17 shows the difference between both conditions.

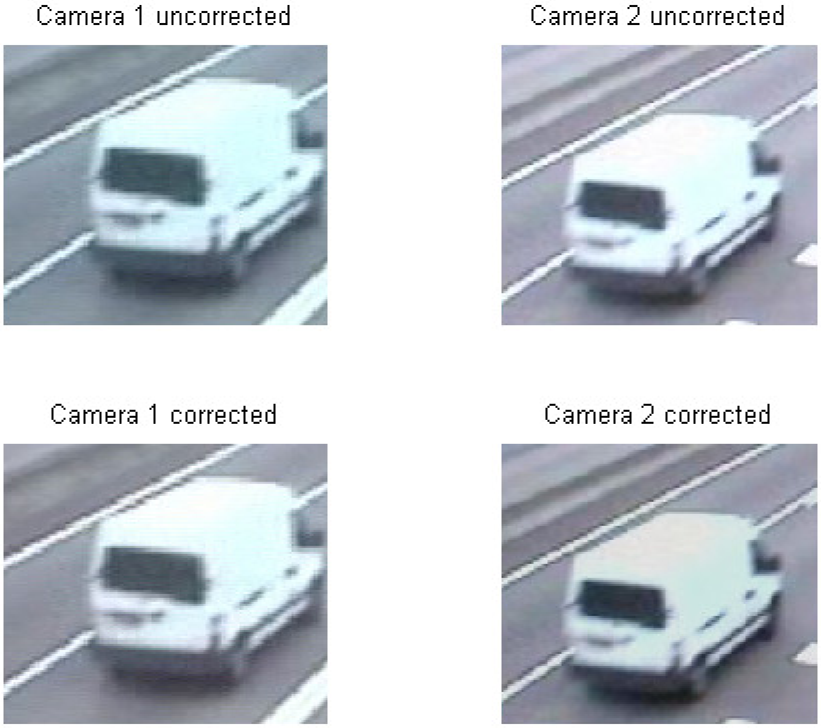

Figure 17: The same vehicle as seen in cameras 1 and 2, before and after colour correction. Notice how the uncorrected image in camera 1 has a blue mist about it that is reduced after the correction.

Chapter 4

Results

In this Chapter the results obtained from each of the methods individually, and lastly the result from the ensemble of all features is reported in terms of precision scores. A precision score represents the fraction of vehicles that are correctly linked between cameras. This measure therefore expresses how reliable a fingerprinting method is. For the vehicle data the 15 minute clips are processed through a tracking scheme that keeps track of individual vehicles. From these tracks various features such as speed, acceleration, lateral position and distance travelled are determined. For the fingerprinting task under consideration here, the speed and acceleration are only of importance. By predicting the velocity of the vehicle over the distance to the next camera, an estimated time of arrival is computed. This allows for considering only a fraction of the vehicles in each data set by selecting only vehicles within a time interval centred on the predicted arrival time for the next camera. A 30 second time interval was selected which is wide enough to ensure that the correct vehicle is present and narrow enough to keep processing times reasonable. The usage of a subset of the total data also makes the job somewhat easier, since there are less non-target vehicles to be mistaken for the correct one. But as will become clear, the methods used are still faced with a challenge since the results do not display any ceiling effect.

For each 15 minute data set the average number of comparisons that are made differ somewhat due to varying traffic density. Therefore the number of vehicles overall and the number of non-target vehicles in the mentioned time interval will be different per set. Based on these numbers alone, the data set with the most vehicles within a 30 second time window will be the most difficult since there is a greater chance of picking the wrong vehicle. Table 1 gives an overview of the specifics for each data set. Once all vehicles in a 15 minute clip are processed, a global optimization is applied, matching best matches first. This helps to fully explore the potential of each method.

Start time	**No. Vehicles**	**Avg. No. Non-targets**	**Lighting**	**Remarks**
08.00	**297**	**20.9**	**Normal**	**Ensemble set***
08.15	**333**	**22.9**	**Normal**	
09.30	**275**	**19.8**	**Backlight**	
09.45	**268**	**18.7**	**Normal**	**Training set***

Table 1: Specifications of the 15 minute data sets used. *Where applicable

4.1 SIFT

SIFT has set the bar in [3] as the best performing method, therefore it will be used here as the benchmark by which the other methods will be judged. The extracted SIFT descriptor based fingerprints are matched through the relative distance approach as described in Subsection 2.1.2. In [7] a relative distance of 0.8 was found to achieve good results. To discover whether or not the provided relative distance is also best for the fingerprinting task a range of relative distances will be explored here. Starting at the hyper-specific 0.1 all the way up to the all matching 1.0 with increments of 0.05. Figure 18 shows the accuracy scores for each of the relative distances.

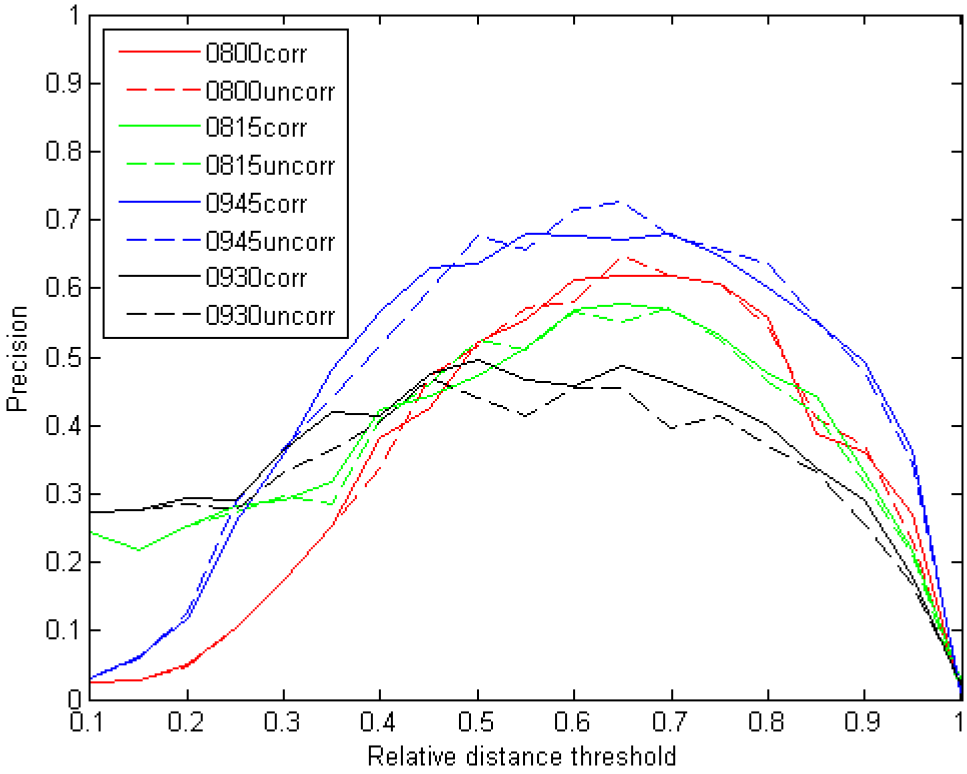

Figure 18: Precision with SIFT, see text for the explanation for the 08.15 and 09.30 high initial scores

Peak performance is found at the relative distance 0.65 with an accuracy of 72.7% for the 09.45 set, for the other sets the greatest precision is also found at the same distance. This is a lower value than found in [7], but quite close to the value of 0.6 found in the public SIFT implementation[3] provided by the inventor of SIFT, Lowe himself. Apparently the poor lighting conditions weigh heavier on the SIFT fingerprint than the number of non-matches that have to be considered, since the 09.30 set yields the lowest precision. The colour correction seems to have little effect on the precision scores, and is only slightly beneficial for the 08.15 and 09.30 sets. Overall relative distance 0.65 and uncorrected colours achieve best performance. It appears the intensity gradients used in SIFT do not benefit from the colour correction. Two, at first glance, abnormalities appear at the beginning of the graphs for the 08.15 and the 09.30 sets. There, the graph starts off with a rather high precision compared to the other two sets. The source of the abnormalities lays in the optimization step which occurs as a final step. The input to the optimisation algorithm is the number of matches found between two fingerprints from the different cameras. At the very specific range of the relative distances the number of matches found is all zeros, when the optimizer is provided with all zero values, it returns an identity matrix; the Nth vehicle in the first camera is linked to the Nth vehicle in the second camera. With the 08.15 and 09.30 sets for a large portion of the vehicles it is the case that the index numbers correspond between cameras. As soon as matches start to be accepted by the relative distance threshold, this effect rapidly disappears as non-zero values are then available for a meaningful optimization. This effect will be present in all SIFT based approaches that use the relative distance matching technique; SIFT, Colour Invariant SIFT, HueSatSIFT and PCA-SIFT.

[3] http://www.cs.ubc.ca/~lowe/keypoints/

Altogether the performance of SIFT does not reach the accuracy scores achieved in [3], 72.7% compared to 90.6%. This is an indication that the data used here is harder to distinguish by a combination of lower resolution images, and more vehicles to make mistakes with.

4.2 HueSatSIFT

Exploring the various relative distances for the HueSatSIFT method yields the results as displayed in Figure 19. Peak precision is reached at a plateau of the two data points at 0.6 and 0.65 relative distance ratios with an accuracy of 74.0%. Although peak performance is found on the not colour corrected 09.45 data set, the difference between corrected and uncorrected colours scores is very small. Especially when compared to regular SIFT. It appears the apparent disruption in the intensity gradient due to the colour correction that distracts the regular SIFT descriptor is largely compensated for by the colour extension, in which no gradient information is present. Compared to the scores of regular SIFT the HueSatSIFT extension only delivers a fractional increase in precision for the normal lighting conditions. On the 09.45 which scored the highest accuracy there is an increase from 72.7% to 74.0%. The backlight 09.30 set is an exception to this. Here an increase from 49.5%, as achieved with regular SIFT on corrected colours, to 55.9% on the uncorrected set with HueSatSIFT. This means that because of the backlight, despite a decrease in intensity gradient information, enough colour information remains to help discriminate between vehicles. Overall best performance is achieved at relative distance 0.65 using corrected colours. A single sided paired T-test with significance level 0.05 results in a p-value of 0.130, therefore the significance offered by HueSatSIFT is not statistically significant compared to regular SIFT.

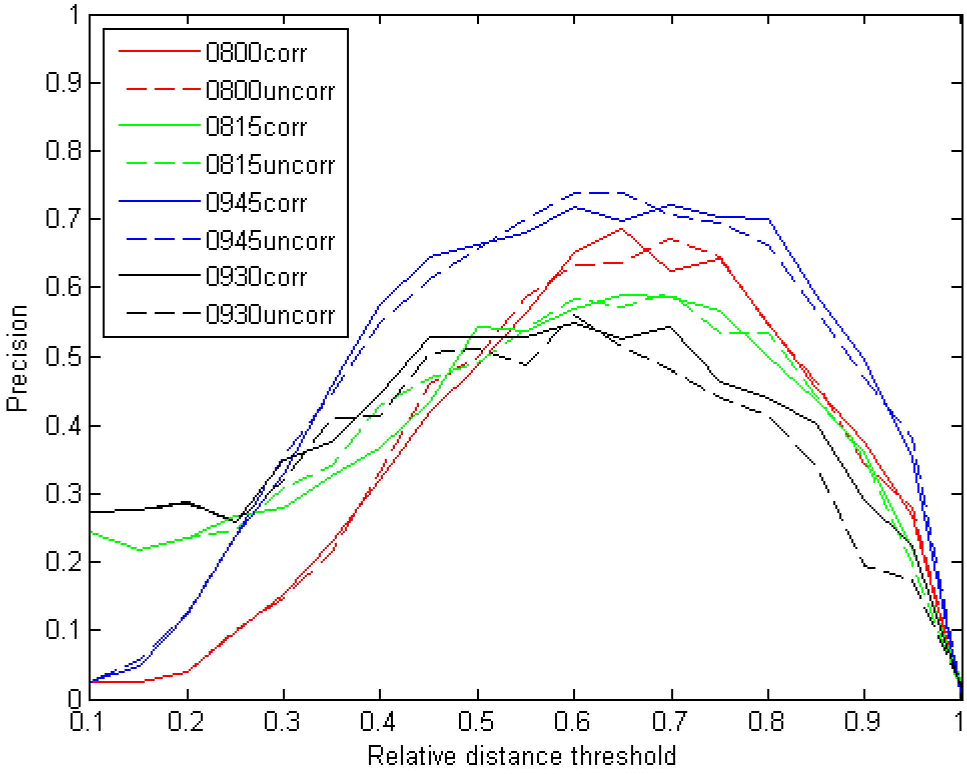

Figure 19: Precision scores for the HueSatSIFT method

4.3 Colour Invariant SIFT

Since other SIFT-based approaches cannot benefit from additional keypoints extracted from the colour channels, Colour Invariant SIFT is examined with and without additional keypoints. This ensures that a direct comparison can be made between other SIFT-based methods, where only the descriptors vary in type and not in number. As before matching the fingerprints for the Colour Invariant SIFT descriptor is done through the usage of relative distances between the best and second best matches, Figures 20 & 21 show the results for the condition without and with addition keypoints respectively. Without extra keypoints peak performance is found at relative distance 0.85 for the not colour corrected 09.45 set, with an accuracy of 83.6% which is an increase of 10.9% over the peak of regular SIFT, although for the other sets 0.8 is a better choice. The backlight 09.30 data set, also scores higher with an accuracy of 56.5% at relative distance 0.8.

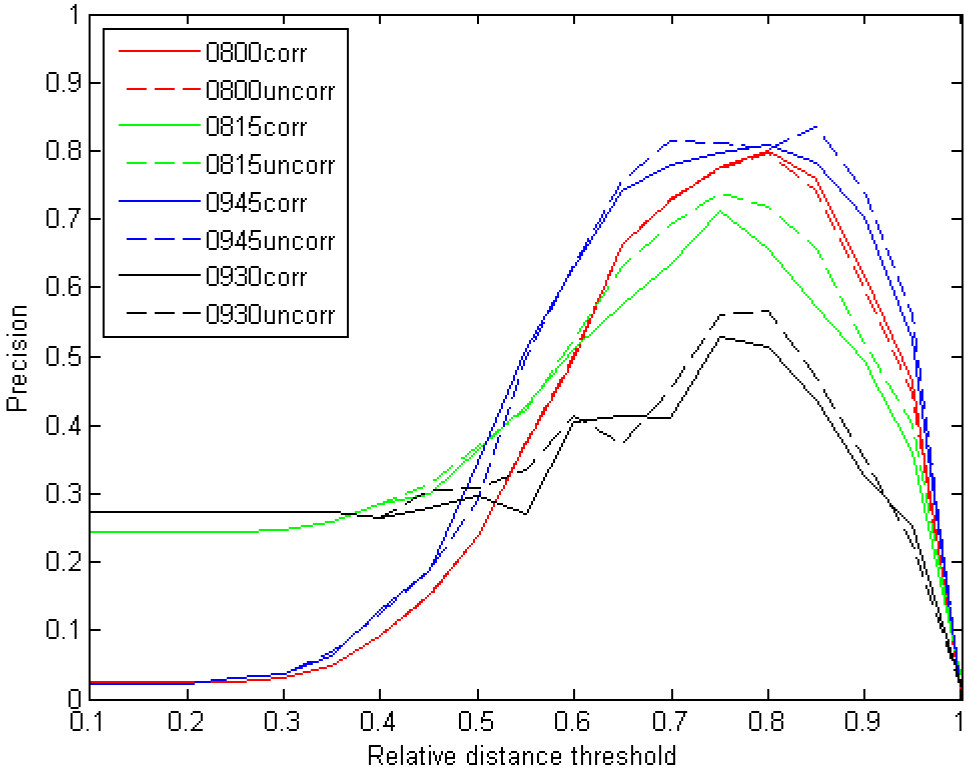

Figure 20: Precision scores obtained with Colour Invariant SIFT, without additional keys

Unexpectedly, the colour correction is of as little influence as with regular SIFT, and again the uncorrected condition scores higher most of the time. Intuition would dictate that a colour based approach should benefit from a more constant colour representation, but as it turns out colour gradient based SIFT descriptors do not benefit from this correction. Contrary to regular SIFT and HueSatSIFT the envelope of the Colour Invariant SIFT graph is more compact with the graph starting to climb much later, the only explanation for this is the difference in descriptors length. Apparently somewhere between the length 256 HueSatSIFT descriptor and the 384 sized Invariant Colour SIFT descriptor, the significance of Euclidian distances has changed. This effect is probably akin to the Curse of Dimensionality problem as described in [34, 35]. Despite this shift in the significance of the Euclidian distances, the longer descriptor does result in higher precision scores.

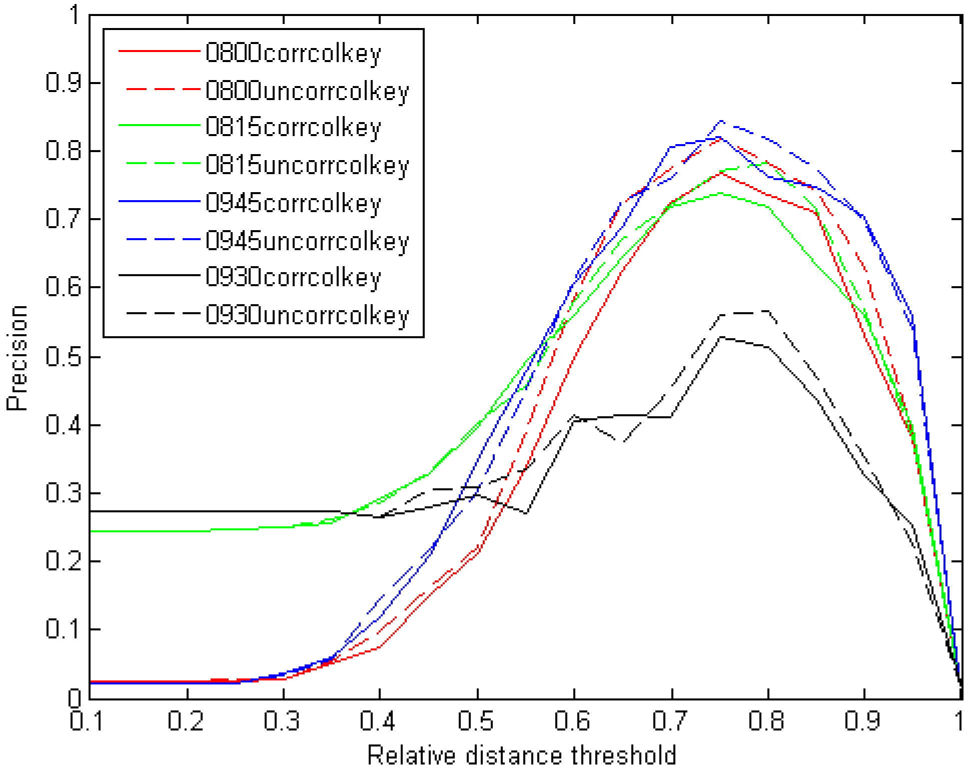

Figure 21: Precision scores obtained with Colour Invariant SIFT with additional keys

Peak performance with additional keys is still found at the 09.45 data set with a slightly higher accuracy of 84.4%. More remarkable is the increase in precision for the 08.15 data set, for the uncorrected colour condition the additional keys increase performance from 73.7% to 78.3%. For the backlight data set no change in performance is present. This is no surprise since there are very few additional colour keypoints. Table 2 lists the number of keypoints per data set for the case of only grey value keypoints and additional colour keypoints. Surprisingly the 09.30 condition has the highest number of keypoints in total, and on average per vehicle. This is most likely due to the high contrast present in that video, which allows a larger number of candidate keypoints to pass through the minimal contrast filter. Overall relative distance 0.75 with uncorrected colours works best. A single sided paired T-test at the 0.05 significance level indicated that the increase in accuracy offered by Colour Invariant SIFT with additional colour keypoints is indeed significant, with a p-value of 0.0054 when compared to regular SIFT.

	Standard SIFT Keypoints		Additional Colour Keypoints	
Track	**Total keypoints**	**Average per vehicle**	**Total keypoints**	**Average per vehicle**
0800	11529	38.82	11687	39.35
0815	13610	40.87	13647	40.98
0930	16350	59.45	16354	59.47
0945	14336	53.49	14410	53.77

Table 2: Number of keypoints as detected in the first camera, without and with additional colour keypoints derived from colour channels

4.4 PCA-SIFT

PCA-SIFT is the last of the methods which uses the relative distance matching technique. Since the authors of [5] found 36 principal components to be of equal expressiveness as regular SIFT, this number shall be used here. It is also the default setting for the public source code[4] of PCA-SIFT. To make a direct comparison with regular SIFT 128 principal components are also considered here. Figures 22&23 show the precision vs. relative distance graphs for the 36 and 128 components conditions respectively.

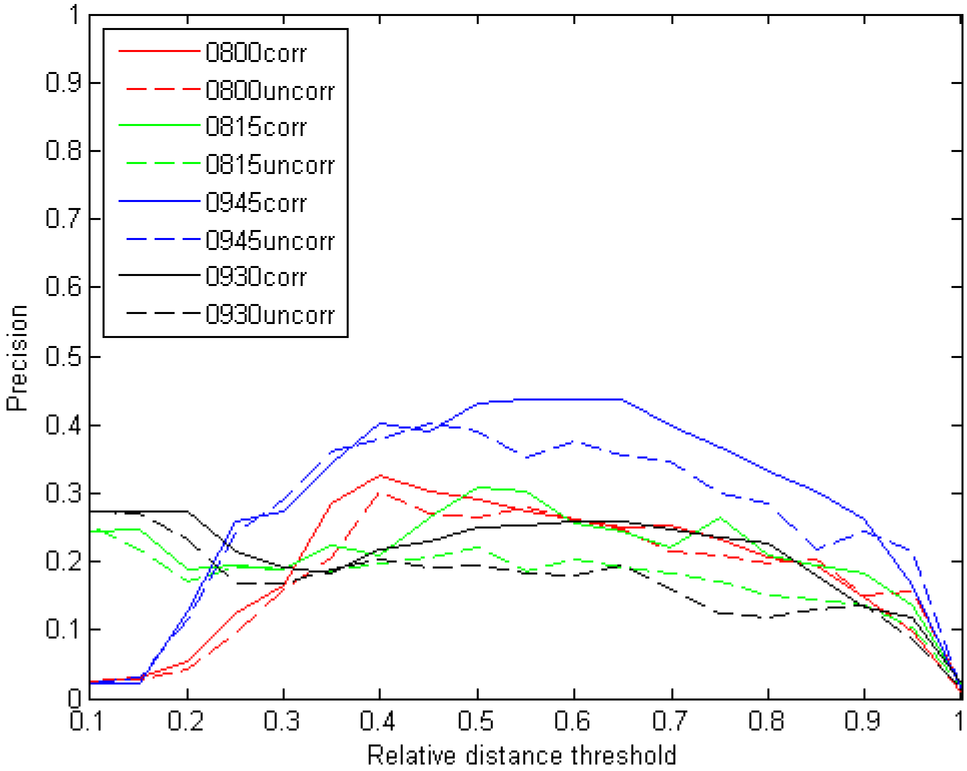

Figure 22: Precision scores for PCA-SIFT with 36 principal components

When compared to regular SIFT, contrary to experiments performed by the authors of [5], performance is much lower. When using 36 principal components, peak performance is achieved on the set from which the Eigen space was obtained, an accuracy of 43.7% for the 09.45 set. The best score on a true test set is the 08.00 corrected colours one, with an accuracy of 32.7%. Overall the best precision scores are found at relative distance 0.5 for the 36 principal components case. This is the lowest relative distance value of all SIFT-based methods. For the case with 128 principal components the highest score is once again achieved on the training set, 47.0% with the 09.45 set at relative distance 0.6. Performances on the rest of the sets are very closely matched with accuracy scores between 30.8% and 32.3%, but remain far from competitive compared to regular SIFT. Overall, the best performance is achieved when using corrected colours at relative distance 0.6. It appears PCA-SIFT requires images of higher resolution and more sharpness to work with in order to be successful.

[4] http://www.cs.cmu.edu/~yke/pcasift

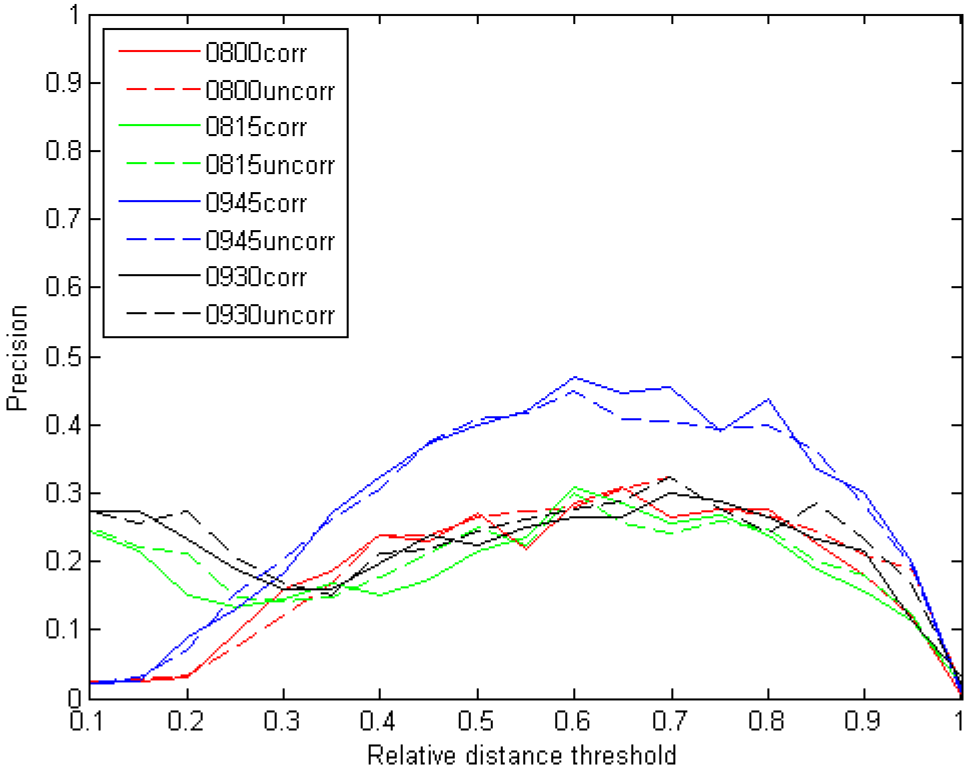

Figure 23: Precision scores for PCA-SIFT with 128 principal components

4.5 Bag of Visual Words

Accuracy scores for the Bag of Visual Words approach are presented in Figure 24 with regular SIFT words, and the results for Invariant Colour SIFT words are shown in Figure 25. For the regular SIFT the colour correction hardly makes any difference for the precision scores. Without doubt the training set achieves the highest performance by far, with a maximum accuracy of 60.2% at clusters level 10, which contains 13,657 visual words. This score is the same for both corrected and uncorrected colours. Right after the optimal cluster level is reached, for the training set a rapid descent towards very low precision scores sets in. This effect is easily explained when considering the nature of the training set. From the first two cameras all SIFT descriptors, computed on not colour corrected detection windows, all the detected keypoint are encompassed in the training set. 50,000 SIFT descriptors served as the input for the hierarchical clustering algorithm, and were extracted from all four cameras. This vocabulary size was chosen for memory considerations when using the 384 sized invariant colour SIFT descriptors. All of the descriptors from the first and second camera are present at the final level of clustering. This means that for each extracted SIFT descriptor a unique identifier exists, unless it was exactly equal to another one. The chance of computing two exactly similar SIFT descriptors in two different images is of course very small, given that there are 128 (or 384) 8 bit values per descriptor. Therefore, as the level of clustering progresses, the more perfect matches there are between the detected keypoints and the vocabulary, the less matches between visual words. And with the decrease in correspondence between visual words the precision scores decrease as well. Since almost the exact same drop in performance is present for the corrected colours condition, this means that with regular SIFT the colour correction only slightly changes the descriptors. For the test sets, where no perfect matches in the vocabulary are to be expected, the drop in performance towards the end does not exist.

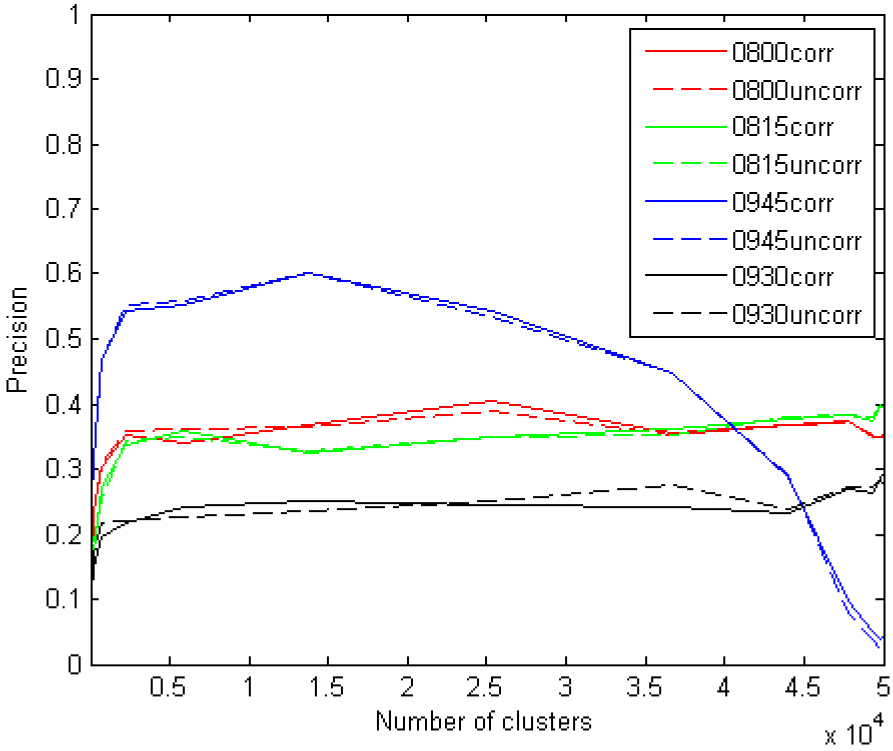

Figure 24: Precision scores for the Bag of Visual Words approach using SIFT descriptors

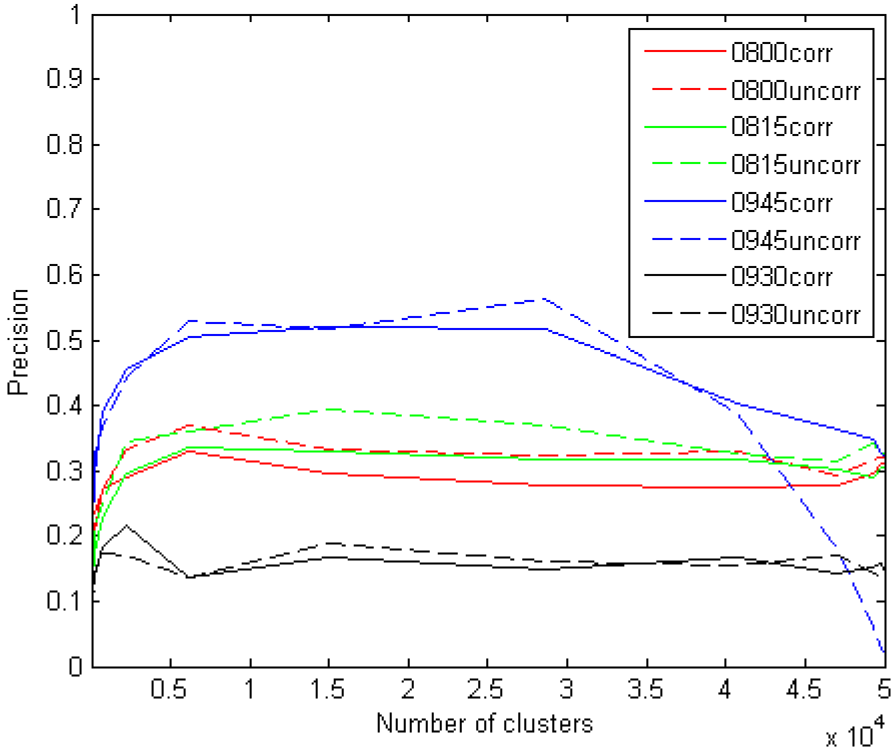

Figure 25: Precision scores for the Bag of Visual Words approach using Invariant Colour SIFT words

The best performing genuine test set is the 08.00 one, with an accuracy of 40.3% at cluster level 11 which has 25,257 visual words on the corrected colours set. For the 08.15 and 09.30 sets there is a boost in performance at the later stages of clustering, whereas the 08.00 set displays a slight dip towards the end. On the whole for the Bag of SIFT based Visual Words, a cluster level of 10 is optimal, producing a 13,657 sized vocabulary.

For the visual words derived from invariant colour SIFT, once again the training set scores the highest performance; an accuracy of 56.3% at cluster level 11, with 28,562 words for the uncorrected colours condition. The best scoring test set is the 08.15 one, with an accuracy of 39.6%, at cluster level 10 which has 15,021 words. Contrary to regular SIFT words, with colour invariant SIFT words colour correction does have a large influence on performance. The drop in performance on the training set is much lower for the colour correction condition, this means that the colour correction disrupts the colour based parts of the descriptor to such extend that perfect matches with the vocabulary are far less common and thus performance is maintained for longer. Uncorrected colours achieve higher scores with invariant colour SIFT, a similar observation as with the direct matching approach as discussed in Section 4.3. Despite the advantage of having more keypoint locations to extract visual words from, the invariant colour SIFT words do not reach the performance of the regular SIFT words. The increased size of the descriptor is most likely responsible for this. As was observed in Section 4.3 the significance of the relative distances between descriptors had changed between 128 and 384 values. A similar effect is most likely present here. Because the Euclidean distance measure becomes less reliable for the longer descriptors it is more likely that descriptors originating from corresponding keypoint locations in two images, are each matched to different vocabulary words. This explains the lower performance compared to the regular SIFT words. Overall for the for invariant colour SIFT words, uncorrected colours and a cluster level of 10, which contains 15,021 words is optimal for this method.

4.6 Spatial Pyramid Matching

Results for the Spatial Pyramid Matching method are two-fold. Firstly there is the Pyramid Histograms of Oriented Gradients, Figure 26. Secondly the results for the Pyramid Histograms of Visual Words approach are shown in Figures 27 and 28 for the uncorrected and colour corrected conditions respectively. Since for the Histograms of Oriented Gradients approach the precision scores are exactly the same for the corrected and uncorrected colour conditions only one graph per data set is provided. The fact that the results are identical regardless of colour correction indicates that only the gradient magnitudes are influenced and the orientations remain unchanged. Since SIFT descriptors contrary to the orientation histograms incorporate both, the first does change and the second does not, at least not at the used pyramid levels. Looking at the results it becomes obvious that between the levels 3 to 5 there is hardly any change in precision, with absolutely no change between the two final levels 4 and 5. Peak performance is obtained with the 09.45 data set with an accuracy of 49.2% at levels 3 to 5 and with 10 degrees per bin; 36 bins to cover the full 360 degrees. The 10 degrees per bin value is the optimum, except for the 08.00 set which peaks at 5 degrees per bin. This indicates that the true optimal value may lie at a range per bin of somewhere between 5 and 10 degrees. Although the results are underwhelming when compared to previously discussed SIFT based approaches, an interesting result is obtained with the backlight set, here performance exceeds the precision achieved with the 08.15 set. This means that despite the adverse lighting conditions the gradient directions at least are maintained, explaining the competitive scores on the backlight set. Overall for the Histograms of Oriented Gradients approach a bin size spanning 10 degrees and a maximum pyramid level of 4 is optimal. Results achieved with the Pyramid Histograms of Visual Words approach achieve about the same level of performance as did the previous Pyramid based method. A noticeable difference to the Bag of Visual Words approaches is the optimal number of visual words.

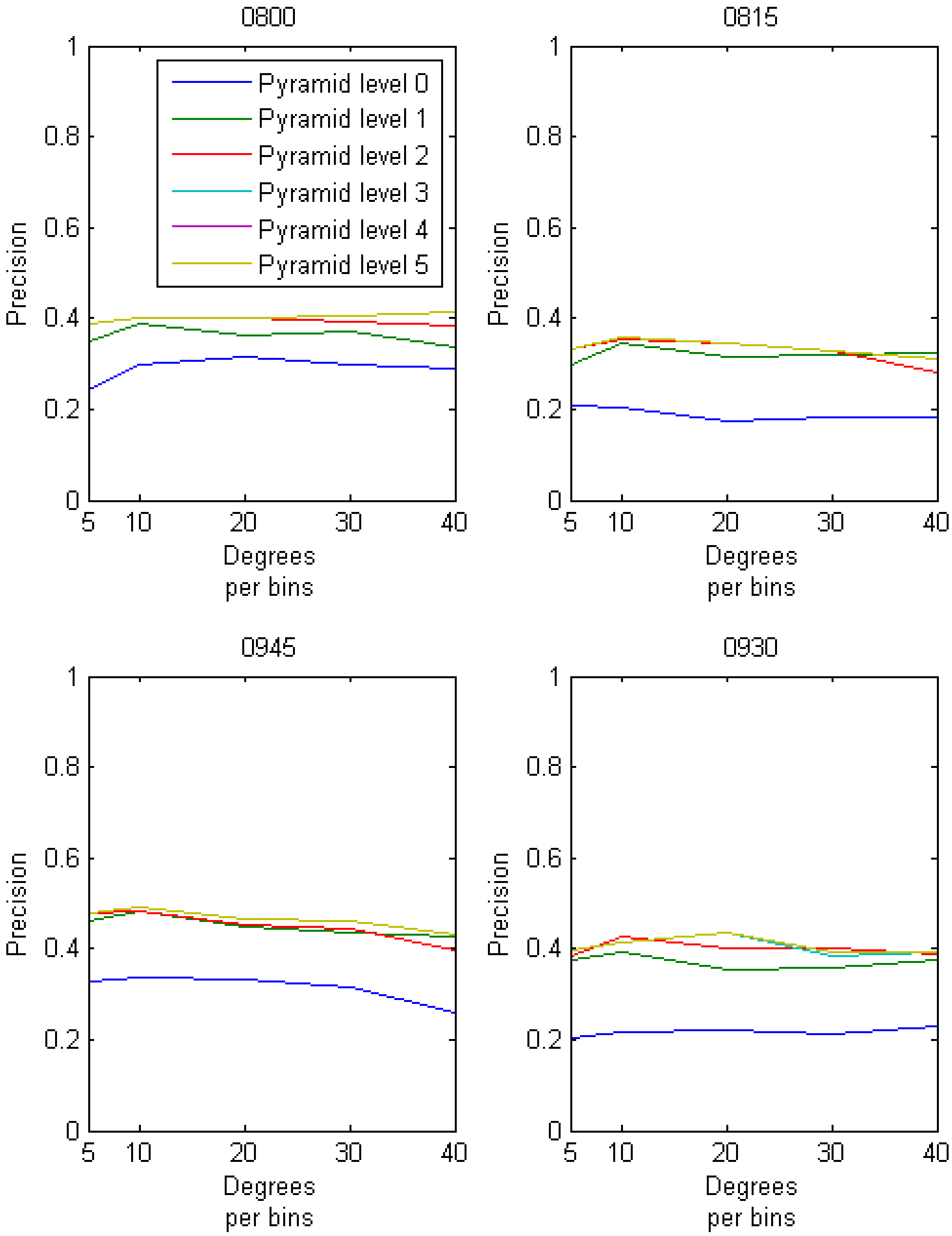

Figure 26: Results for the Pyramid Histograms of Oriented Gradients approach

For Pyramid Histograms of Visual Words, a vocabulary size of 27 words works best, a number far lower than the 13,657 words for Bag of Visual Words approach. This difference must be caused by the difference in the method of matching; word frequencies instead of corresponding word labels. Best performance is achieved on the 09.45 training set, with a precision of 43.1% at pyramid level 5 and cluster level 4, with 27 words. Again, as with the Pyramid Histograms of Oriented Gradients approach the backlight 09.30 set is once again the best scoring test set, with an accuracy of 41.5% at pyramid level 4 and 81 words at cluster level 5. Overall the best performance is achieved at pyramid level 4 and a vocabulary of 27 words at cluster level 4 for the colour corrected condition.

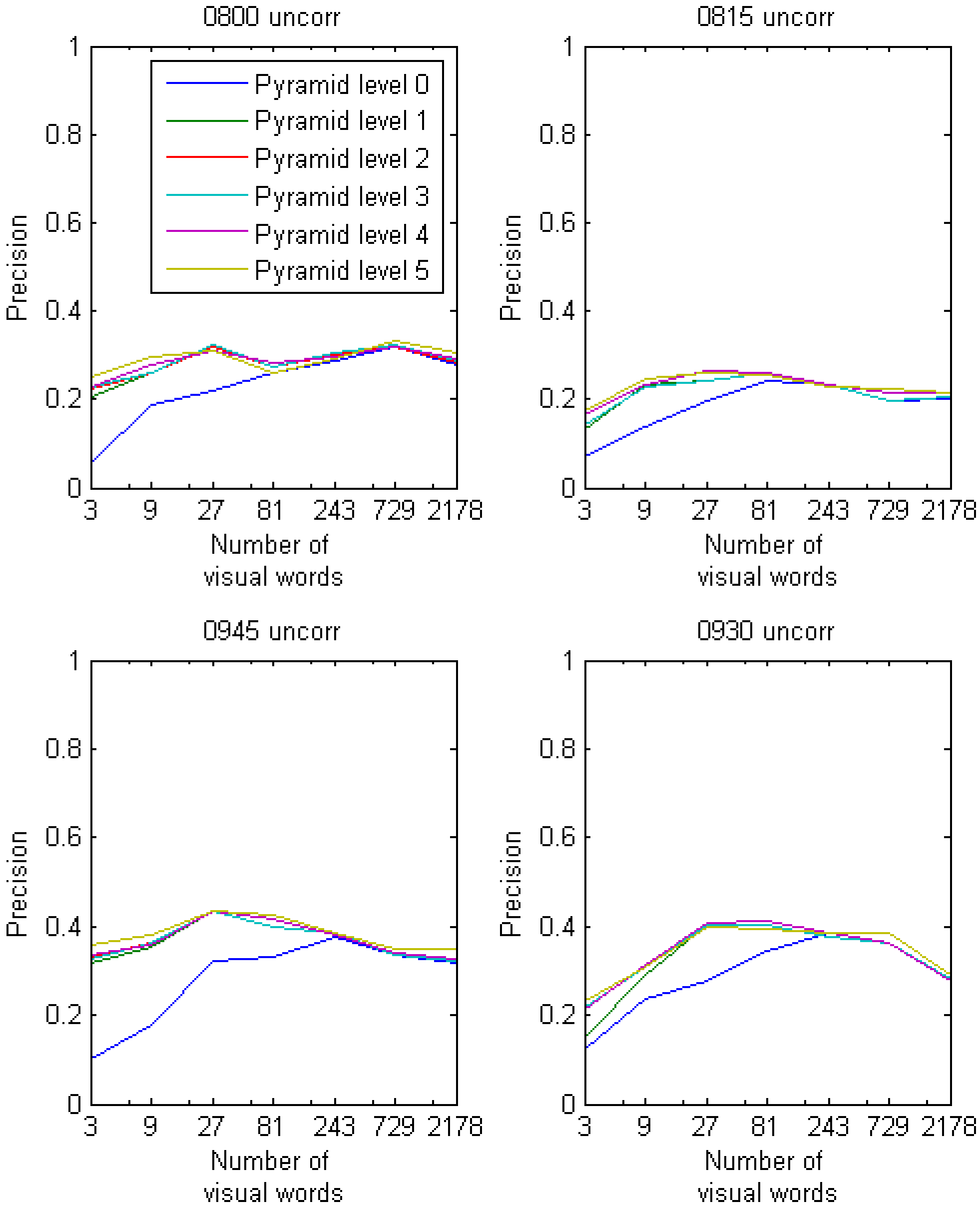

Figure 27: Results for the Pyramid Histograms of Visual Words approach without colour correction

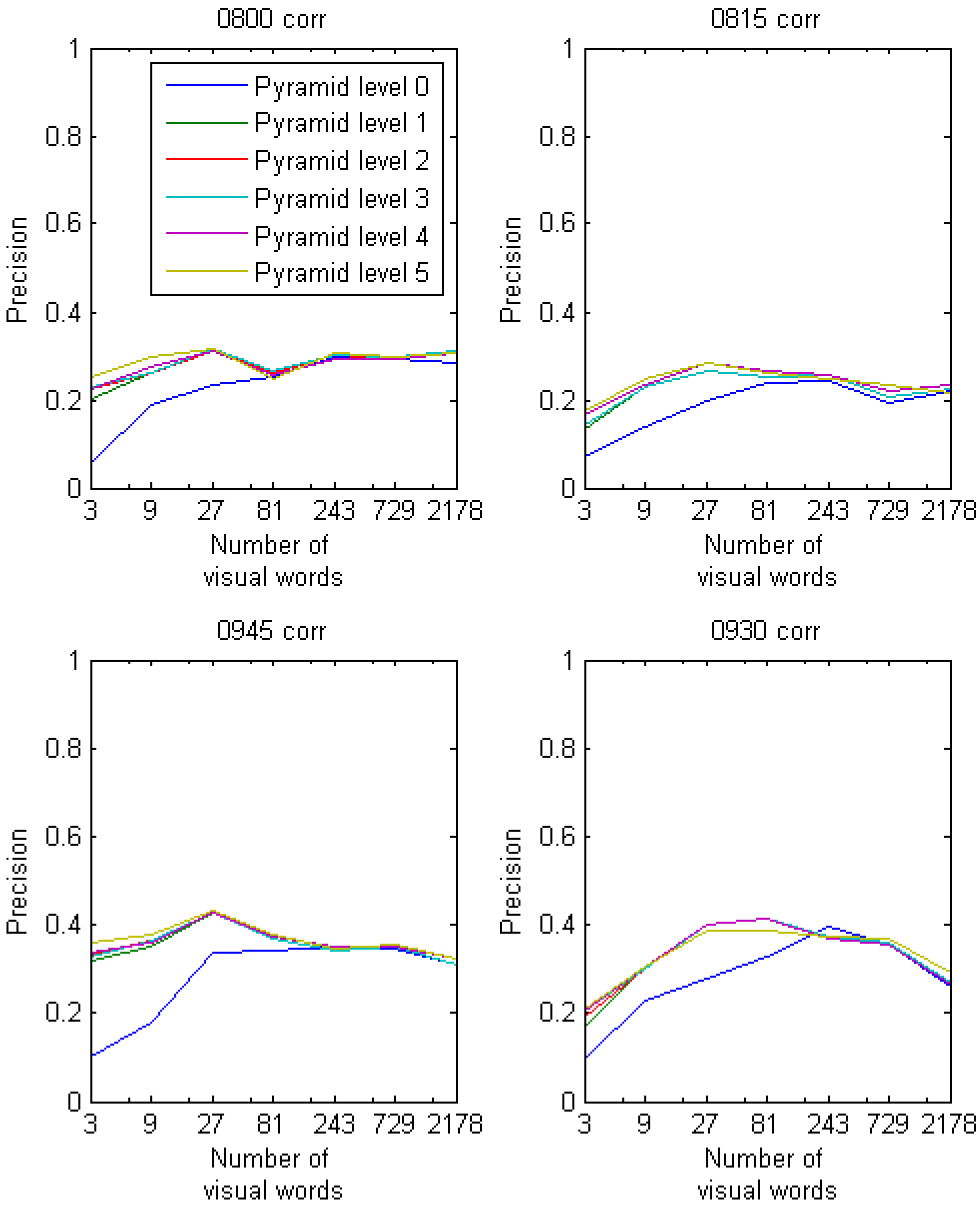

Figure 28: Results for the Pyramid Histograms of Visual Words approach with colour correction

4.7 Colour Co-occurrence Histograms

With the Colour Co-occurrence Histograms approach the radius for co-occurrence is varied, Figure 29 presents the accuracy scores that are obtained in this manner. As can be seen the 09.45 set achieves the highest accuracy with a radius of 40 pixels, however it must be noted that the 09.45 set was used for training; gathering the colours for clustering. The next highest score is then achieved by an actual test set, the 08.00 set, 57.2% for the colour corrected set when using a radius of 29 pixels. The backlight 09.30 set reaches peak performance much earlier with 35.1% at a 9 pixel radius, but is it clear that the backlight disrupts the colour representation and therefore creates poorly distinguishable CCHs, even when colour correction is applied. The CCH method benefits most from the colour correction, almost doubling scores on the training set, and also vastly increasing precision for the test sets with around 50%. The vast increase in performance as a result of the colour correction at first glance may seem surprising. This increase can be explained by the fact that the Colour Co-occurrence Histograms method uses direct matching of colours, instead of gradients. Therefore it is logical that a more consistent representation of colours between cameras, as produced by the colour correction, yields better results. Overall colour correction in combination with a radius of 23 pixels is found to be optimal.

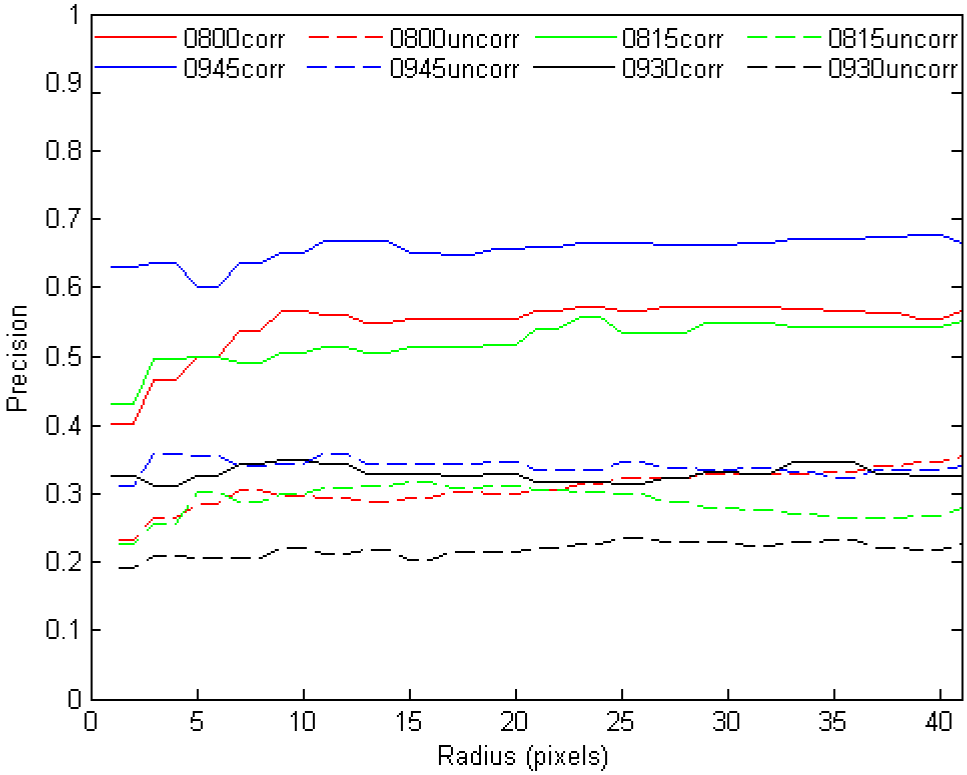

Figure 29: Precision scores for Colour Co-occurrence Histograms

4.8 Cortex-Like Mechanism

The independent variable for which the effects are explored for the visual cortex model is the number of C2 features. As discussed in Section 3.8, the visual cortex model introduces stochasticity into the results during feature selection. The effect this has on performance can be observed in the jaggedness of the resulting precision graphs, Figures 30 and 31 for the uncorrected and corrected colours data sets respectively. Since the 09.45 data set was used to extract the dictionary of images patches from, it is no surprise this set scores the highest.

However, accuracy scores of 46.9% on the uncorrected set with 760 C2 features and 36.2% with 720 C2 features are still inferior to most other methods. Surprisingly the highest scoring true test set is the backlight 09.30 set, achieving an accuracy of 36.8% with uncorrected colours and 780 features and with corrected colours 25.0% with 330 features. When only considering the total number of vehicles and the average number within a 30 second interval, the 09.30 is the easiest set of the four. This means that the visual cortex model is capable of overcoming, at least to some extent, the difficulties posed by the adverse lighting conditions. On the whole the visual cortex model does not yield competitive results. In typical applications where the model does thrive many examples per item are available, but here for object fingerprinting there is just one from the initial detection. Overall the best scores are achieved with 830 C2 features on uncorrected colours.

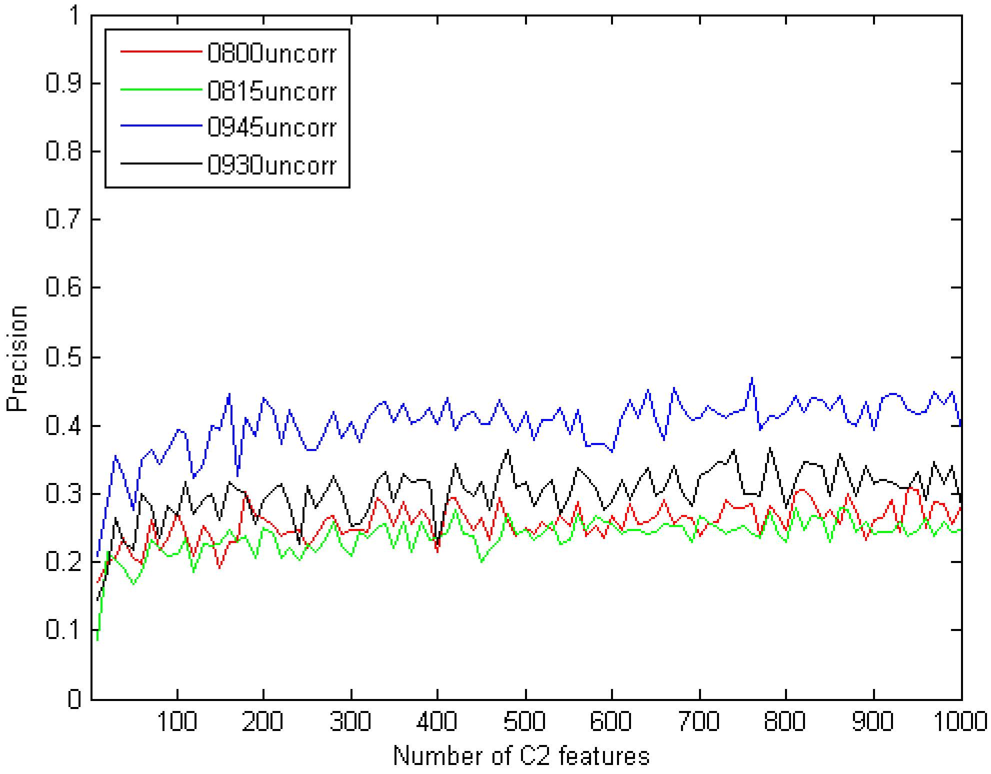

Figure 30: Precision scores for the visual cortex model, on the not colour corrected condition

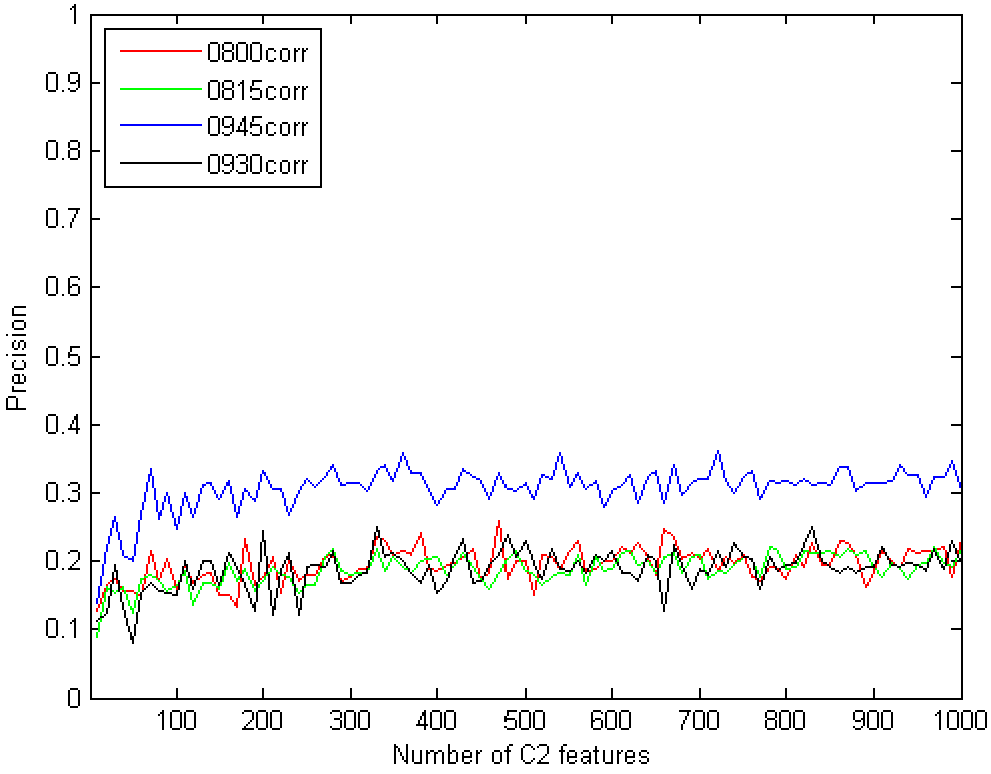

Figure 31: Precision scores for the visual cortex model, on the colour corrected condition

4.9 Boosting

For the method of boosting the number of Haar-like features can be varied to find an optimal number. A very similar approach for the detection of the vehicles in the video frames was used to acquire the detection windows to extract the fingerprint from. As mentioned in Chapter 3, to include as much of the vehicles as possible the detection windows are widened a little bit, exposing more of the vehicle. This was beneficial for the other features used in this thesis. But for boosting the narrow original detection windows might work better since the features would be computed on the exact same location as was used for detection. Therefore after experiments with the same input as the other methods, the boosting method was tested again on the more narrow images. Since the not colour corrected condition scored the highest, the narrow images colours are left unchanged. To create the cascade of Haar-like features, training was performed on the 09.45 dataset. Figure 32 shows the results for the widened images, as used with the other methods and Figure 33 shows results obtained with the original narrower and not colour corrected detection windows. When comparing Figures 32 and 33, it can be seen that for each of the data sets except the 09.30 set, the original, narrow detection window works best. Peak performance is achieved as expected on the 09.45 training set with an accuracy of 41.1% with 90 Haar-like features. The best test set is the 08.00 one with an accuracy of 28.4% at 80 Haar-like features. A closer investigation into what causes the poor results with the narrow images on the 09.30 set shows that the initial detections are to blame. What happens very often is that in the video frames the detector mistakes the shadow of a vehicle as part of the vehicle. Figure 34 gives an example. When extending the detection window, more of the vehicle is included, allowing for better performance. Altogether it can be said that the hoped for discriminating ability of Haar-like features did not materialize.

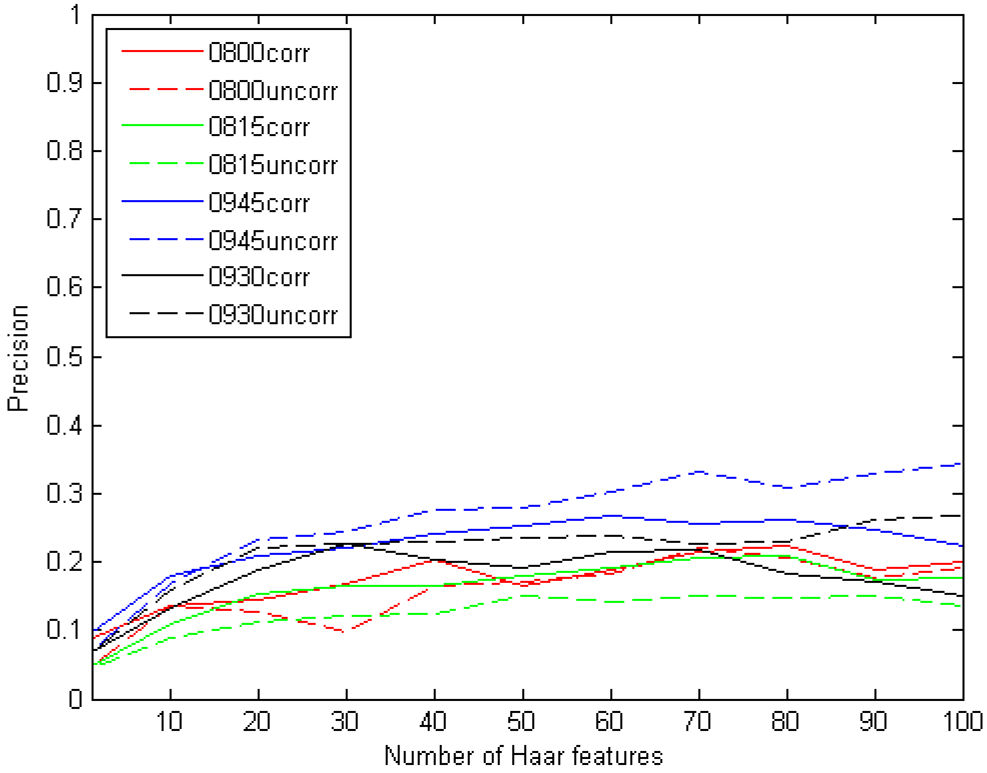

Figure 32: Precision scores for the method of Boosting on widened detection windows

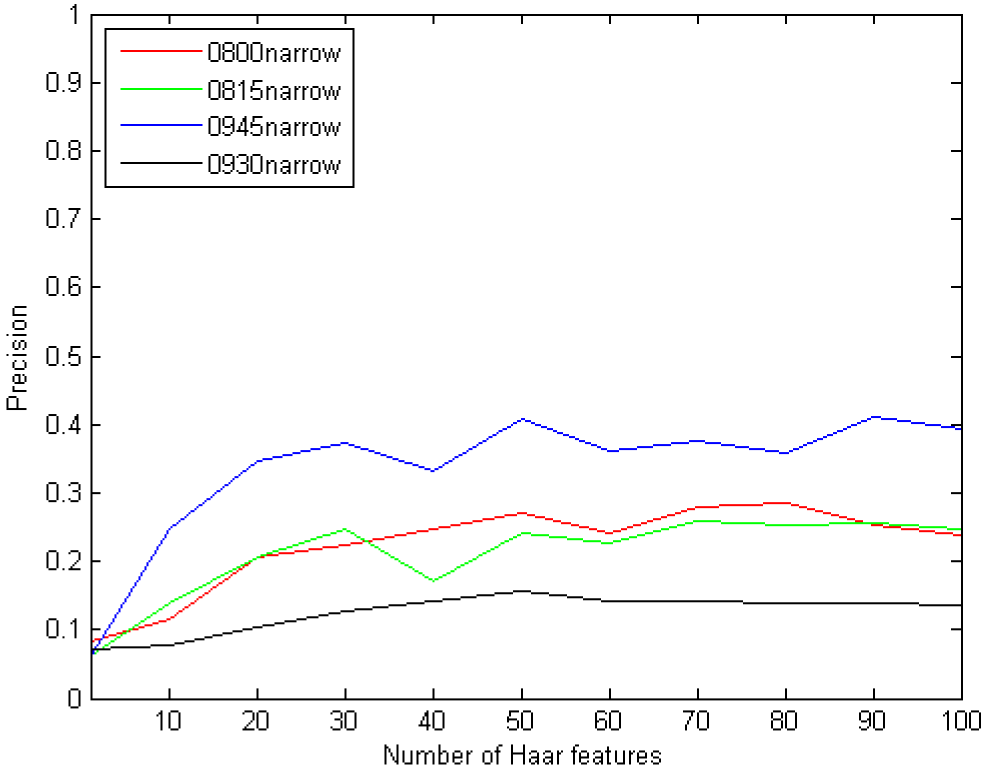

Figure 33: Precision scores for boosting the boosting approach on not colour corrected detection windows.

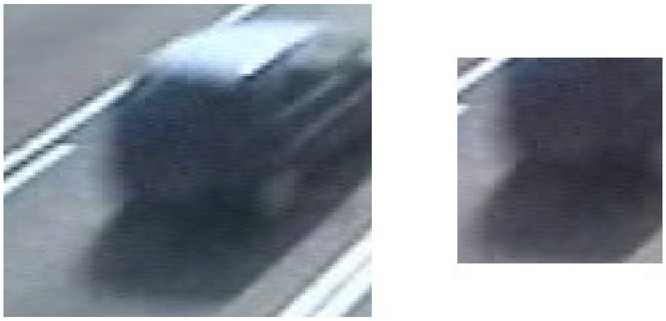

Figure 34: An instance where the vehicle detector mistakes the shadow for part of the vehicle, but the widened window contains almost the entire vehicle.

4.10 Feature Ensemble

Having established the optimal parameters and colour conditions for each of the individual methods, feature ensembling can be performed. Since many of the used features have used the 09.45 data set for training, the feature ensemble will be created using the 08.00 data set for training. Since this is the easiest data set that remains, when considering the number of vehicles on average in a 30 second time window and lighting conditions, the most difficult sets remain available for testing. The result of the feature ensemble algorithm is given in Table 3, which lists the used parameters for each method, ordered by the quality measure alpha, from which the weights are derived. The higher alpha the more accurate the method becomes. Since the alpha scores range from positive to negative values, in addition to the full set of combined features, the positive and negative alpha methods will be considered. The weights as reported in Table 3 are used because weighing votes with negative alphas is meaningless. Therefore alpha values are scaled linearly such that the lowest performing method is assigned a weight of 1. Figure 35 shows the results achieved on each data set with the three combinations of features. Although there are more methods in the group with negative alphas, the top four methods with positive alphas achieve higher performance. Overall, the ensemble of all features combined results in the highest performance, except for the 09.45 set. On the 09.45 data set the positive alphas group of methods achieves the highest accuracy, this is due to the fact that the fourth best method, Colour Co-occurrence Histograms, was trained on this set, and as a result achieved a performance far greater than on the other sets. This means the highest accuracy is achieved on the 09.45 set, with 87.3% on the positive alphas group. Of more interest are the true test sets, 08.15 and 09.30. On the 08.15 set the combination of all methods together scores highest with an accuracy of 79.6%. This is an increase of 5.9% over just the best performing individual method on its own. The largest benefit of the feature ensemble is witnessed with the 09.30 set. Where Invariant Colour SIFT with additional colour keypoints achieved an accuracy of 56.5%, all features combined yield an accuracy of 69.6%, which is an increase of 13.1%. A single sided paired T-test at the 0.05 significance level indicated that the increase in accuracy offered by the complete feature ensemble is indeed significant, with a p-value of 0.00017 when compared to regular SIFT. The same applies to the positive alphas feature ensemble, with a p-value of 0.00015. The T-test to discover whether the feature ensemble of all features performs significantly better than just invariant colour SIFT with additional colour keypoints fails. A p-value of 0.0603 however indicates that perhaps the addition of a single extra data set would be enough to reach significance.

Method	Alpha	Weight	Colours	Parameters
Invariant Colour SIFT	0.7408	2.3166	Uncorr.	Relative. Dist. 0.75
HueSatSIFT	0.5308	2.1066	Corr.	Relative. Dist. 0.65
SIFT	0.4607	2.0365	Uncorr.	Relative. Dist. 0.65
Colour Co-occurrence Histograms	0.2834	1.8593	Corr.	23 pixel radius
Pyramid Hist. of Oriented Gradients	-0.0369	1.5389	Uncorr.	Pyramid Level 4, 10 deg.
SIFT Bag of Visual Words	-0.0597	1.5161	Corr.	Cluster level 10
PCA-SIFT	-0.1069	1.4689	Corr.	Rel. Dist. 0.6, 128 Comp.
Invariant Colour SIFT Bag of Words	-0.2299	1.3459	Uncorr.	Cluster level 10
Pyramid Histogram of Visual Words	-0.3285	1.2473	Corr.	P. Level 4, Cluster level 4
Haar-like feature Boosting	-0.5023	1.0735	Uncorr.	90 Haar-like features
Cortex-like mechanism	-0.5758	1.0000	Uncorr.	720 C2 Feat.

Table 3: Overview of the used colour conditions, method parameters and quality measure alpha.

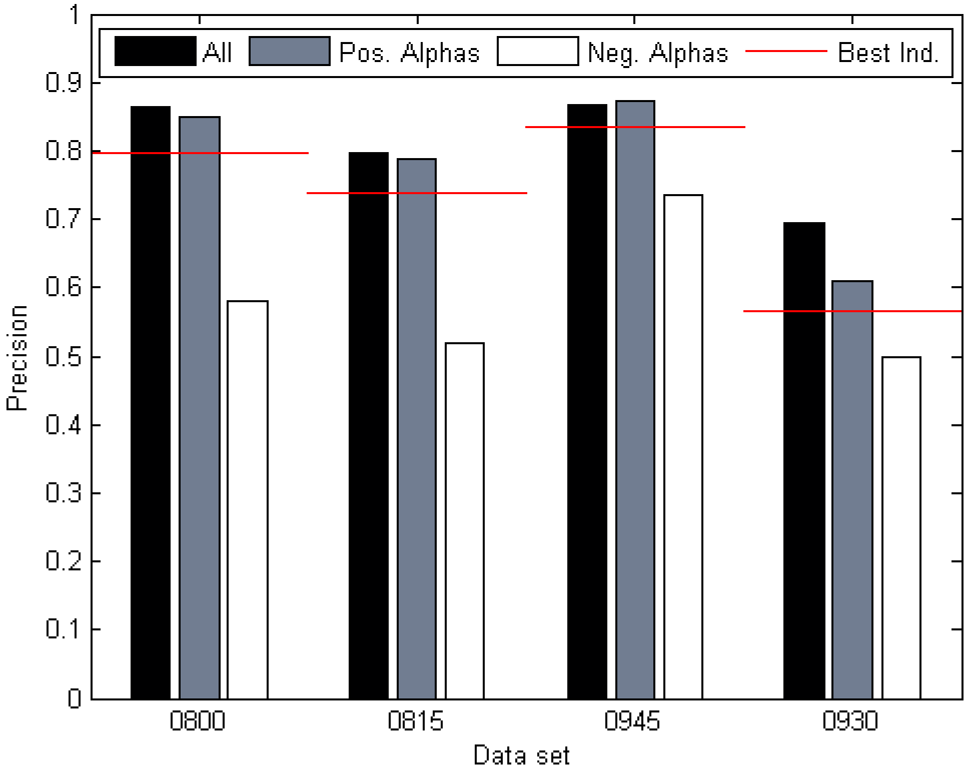

Figure 35: Precision scores achieved with feature ensembles, for all features combined, and the positive and negative alpha grouped together. Horizontal red lines indicate the highest score achieved on the respective data set with the best performing individual method; Invariant Colour SIFT with colour keypoints.

Chapter 5

Conclusion and future work

The research questions posed at the beginning of this thesis can now be answered. The answer to the first question *"Can employment of previously unexplored and perhaps colour based features generate a significant increase in reacquisition scores for object fingerprinting?"* is positive and more specifically, it takes the shape of invariant colour SIFT. When combined with additional colour keypoint detection capability, this yields the highest accuracy score achieved by an individual method. This increase furthermore is statistically significant. Also the ensemble of all features, and the subset of only the methods that achieved alpha scores higher than 0, also provide a statistically significant increase each.

The second research question *"Is it possible to create an ensemble of object fingerprinting features, that together surpass performance of the best individual feature?"* however has to be answered negatively. Although there was still enough room for improvement left by the best performing individual feature, and indeed an increase in accuracy was present, this increase did not pass the significance test. But it must be said that just one additional data set most likely is enough to achieve significance.

For future work, the following considerations are of importance. Although in theory object fingerprinting is a task that can be considered an extreme overfit of detection, practical testing offers no evidence for such applicability. As demonstrated by the low precision scores with the boosting of Haar-like features approach, additional viable features for object fingerprinting are not likely to be found in the domain of detection methods.

The pyramid histogram based approaches, originally used for object classification, do not guarantee to transfer competitive performance to the fingerprinting task, although the results of the experiments performed here may not have fully exploited the full potential of pyramid based approaches. The detection windows as produced by the vehicle detector are not very consistent when it comes to placement of the vehicle within the windows, also the size of the detected vehicles may vary due to perspective effects. This means that corresponding features may only on occasion lie within the same pyramid regions between detection windows originating from different cameras. A more consistent detector may help overcome this problem for object fingerprinting, provided the objects in the domain that need to be reacquired are rigid in physiology and viewpoint differences between cameras are limited.

The visual cortex model is another example of a successful classification technique that did not thrive on the task of object fingerprinting as performed here; one example object and 1-NN matching. As was said before, multiple images per object should help increase performance, since in literature this leads to higher scores on classification tasks. However, the situation in which object fingerprinting is performed has to lend itself to extraction of multiple examples. Accurate within video frame tracking could be the answer to this; as an object moves through the camera's field of view, multiple examples could be extracted. In such case reliable tracking is an absolute necessity; any error in tracking would result in introducing incorrect examples, and therefore erroneous reacquisition. In addition multiple examples can be created synthetically by inducing changes in perspective through employment of projective geometry. On the data available here this proved not to be possible, since even the slight changes in perspective result in image black edges to be included within the vehicle images.

PCA-SIFT, did not live up to the expectations as created by results obtained in [3]. This is most likely due to the nature of the data as used in the experiments here. Blurry, low resolution images dramatically reduce PCA-SIFT's performance in object fingerprinting. It should not be forgotten that in situations where higher resolution data is available, PCA-SIFT has proved itself to be competitive with regular SIFT [3, 5]. Since PCA-SIFT on its own did not yield results remotely similar to that of regular SIFT, an extension into the colour domain was not undertaken. However, for data on which PCA-SIFT does result in competitive performances to regular SIFT, it is well worth to explore a colour extension similar to that of invariant colour SIFT. By grouping the intensity based descriptor with additional colour based descriptors, results may even exceed that of invariant colour SIFT, while reducing descriptor length. For example, three 36 sized descriptors would result in a single 108 valued invariant colour PCA-SIFT descriptor, which is 16% shorter than the regular SIFT descriptor.

Bag of Visual Words based approaches particularly are of interest for smart camera networks, in which communication overhead is to be kept to a minimum. But as can be seen in the experiments performed here, results are far from competitive compared to direct feature matching approaches. Moreover, there appears to be a limit on descriptor sizes that can reliably be matched in visual word vocabularies, as testified by the reduced performance with Invariant Colour SIFT words.

The colour correction scheme used here is quite simple by nature, and only substantially helped the Colour Co-occurrence Histogram method. It is clear that in order to also benefit methods that incorporate colour gradients, a more advanced technique of colour tuning is required. A method for colour correction in non-overlapping cameras such as presented in [36] might work well for methods that use colour gradients. But an even simpler approach is to use a colour calibration card, containing for example a colour spectrum, to tune each camera's colour sensitivity before placement, so that at least under the same lighting conditions, consistent colours are recorded. More advanced camera and lighting models will then have to be used to try and cope with variable lighting conditions.

Direct matching of SIFT-based features has proved to be superior when compared to other methods in this thesis. Especially the Invariant Colour SIFT approach has distinguished itself. It is therefore recommended as future work to further explore SIFT-based approaches, other colour spaces may be used to derive colour descriptors in addition to the intensity based one. And in addition to colour spaces, different settings for the SIFT algorithm may be explored as well, since it is possible that the optimal descriptor length, as well as the relative distance measure, varies with the task at hand. Colour Co-occurrence Histograms may also prove to be successful in other colour spaces than HSV as was used here. Although there is no consensus on whether or not to use colour information in computer vision, it should be kept in mind that evolution has seen fit to place colour sensitive cells in the eyes of a wide variety of creatures [37].

Despite the successes achieved with methods from the SIFT family of methods, the search for additional features is also strongly recommended to introduce diversity in fingerprinting methods that ensemble techniques require. As demonstrated, feature ensembling can provide an increase in performance over strong features, with the addition of fairly weak performing features. However, it should be noted that the computation of a large number of different features will require more time, putting restraints on the possibilities for real-time processing. For such real-time applications it is worthwhile to focus on additional features that lend themselves to be implemented on GPU's. Such an implementation is available for SIFT [38], and greatly reduces processing time.

Altogether, the key points for recommended future work on object fingerprinting are: to focus on testing alternative colour spaces, exploring additional colour based local image descriptors and to search for methods that are not based upon SIFT. Even holistic approaches can be considered when using feature ensemble techniques.

References

[1] W. Zajdel, J. D. Krijnders, T. C. Andringa, and D. M. Gavrila. CASSANDRA: audio-video fusion for aggression detection. Proceedings of the IEEE Conference on Advances Video and Signal Based Surveillance AVSS, pages 200-205, 2007.

[2] C. Arth, C. Leistner, and H. Bischof, Object Reacquisition and Tracking in Large-Scale Smart Camera Networks, Proceedings of the 1st IEEE International Conference on Distributed Smart Cameras, pages 156-163, 2007.

[3] D. G. Lowe, Object recognition from local scale-invariant features International Conference on Computer Vision, Corfu, Greece, pages 1150-1157. 1999.

[4] Y. Ke and R. Sukthankar. PCA-SIFT: A more distinctive representation for local image descriptors, Proceedings of the Conference on Computer Vision and Pattern Recognition, Washington, USA, pages 511-517, 2004.

[5] M. Tkalcic and J.F. Tasic, Colour spaces: perceptual, historical and applicational background, EUROCON computer as a Tool, pages 304-308, 2003

[6] D. G. Lowe, Distinctive image features from scale-invariant keypoints, International Journal of Computer Vision, 2(60) pages 91-110, 2004

[7] R. H. Luke, J. M. Keller and J. Chamorro-Martinez, Extending the Scale Invariant Feature Transform Descriptor in to the Color Domain, ICGST International Journal on Graphics, Vision and Image Processing, GVIP volume 8, issue IV, pages 27-33, 2008.

[8] S. Sural, G. Qian, and S. Pramanik. Segmentation and Histogram Generation Using the HSV Color Space for Image Retrieval, International Conference on Image Processing (ICIP), pages. 589-592, 2002

[9] A. Bosch, Zisserman A., and X. Munoz, Image classification using ROIs and multiple kernel learning, International Journal of Computer Vision, 2008. submitted.

[10] J.-M. Geusebroek, R. van den Boomgaard, A. W. M. Smeulders and A. Dev, Color and Scale: The Spatial Structure of Color Images, Sixth Europian Conference on Computer Vision (ECCV), pages 331-341, 2000.

[11] J.-M. Geusebroek, R. van den Boomgaard, A.W.M. Smeulders and H. Geerts, Color Invariance, Pattern Analysis and Machine Intelligence, IEEE Transactions on Volume 23, Issue 12, pages 1338-1350, 2001.

[12] G. J. Burghouts and J.-M. Geusebroek, Performance evaluation of local color invariants, Computer Vision and Image Understanding, volume 113, pages 48-62, 2009.

[13] L.I. Smith, A Tutorial on Principal Components Analysis, Cornell University, USA, 2002.

[14] P. Kolari, A. Java, T. Finin, T. Oates and A. Joshi (2006). Detecting Spam Blogs: A Machine Learning Approach, Proceedings of the 21st National Conference on Artificial Intelligence (AAAI 2006), Boston, MA, 2006.

[15] G. Csurka, C. Dance, C. Bray, and L. Fan, Visual categorization with bags of keypoints, Proceedings Workshop on Statistical Learning in Computer Vision, 2004

[16] S. Lazebnik, C. Schmid, and J. Ponce. Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories, CVPR, 2006.

[17] S. Ekvall, F. Hoffmann, and D. Kragic, Object recognition and pose estimation for robotic manipulation using color cooccurrence histograms, IEEE/RSJ International Conference on Intelligent Robots and Systems, IROS, 2003.

[18] P. Chang and J. Krumm, Object Recognition with Color Cooccurrence Histograms, IEEE Conference on Computer Vision and Pattern Recognition, Fort Collins, CO, 1999.

[19] V. Arvis, C. Debain, M. Berducat, and A. Benassi, Generalization of the Co-occurrence Matrix for Color Images: Application to Color Texture Classification Image Anal Stereol, pp. 63-72, 2004.

[20] P. Pérez, C. Hue, J. Vermaak and M. Gangnet, Color-Based Probabilistic Tracking, European Conference on Computer Vision, pages 661–675, 2002.

[21] T. Serre, L. Wolf, S. Bileschi, M. Riesenhuber and T. Poggio, Robust object recognition with cortex-like mechanisms, IEEE Trans. Pattern Anal. Mach. Intell. volume **29**, pages 411–426, 2007.

[22] T. Van der Zant, L. Schomaker, and K. Haak, Handwritten word spotting using biologically inspired features, IEEE Trans. on PAMI, 2008.

[23] A. Borji, M. Hamidi and F. Mahmoudi, Robust handwritten character recognition with features inspired by visual ventral stream, Neural Processing Letters, v. 28 n.2, pages 97-111, 2008.

[24] Y. Freund and R.E. Schapire. A short introduction to boosting. Journal of Japanese Society for Artificial Intelligence, 14(5) pages 771–780, 1999.

[25] R.E. Schapire. The boosting approach to machine learning: An overview, Workshop on Nonlinear Estimation and Classification. MSRI, 2002.

[26] P. Viola and M. Jones, Rapid object detection using a boosted cascade of simple features. In Proc. CVPR, pages 511–518, 2001.

[27] P. Viola, M. J. Jones, and D. Snow. Detecting pedestrians using patterns of motion and appearance, The 9th ICCV, Nice, France, volume 1, pages 734–741, 2003.

[28] R. Rifkin, A. Klautau, In defense of one-versus-all classification. Journal of Machine Learning Research, Vol. 5, pages 101–141, 2004.

[29] Y. Freund and R. E. Schapire, Experiments with a new boosting algorithm, Machine Learning: Proceedings of the Thirteenth International Conference, pages 148-156, 1996.

[30] R. Polikar, Ensemble Based Systems in Decision Making, IEEE Circuits and Systems Magazine, vol.6, no. 3, pages 21-45, 2006.

[31] L. Breiman, Bagging predictors, Machine Learning vol. 24, pages 123-140, 1992.

[32] K. Ali and M. Pazzani, Classification using Bayesian averaging of multiple, relational rule-based models. Learning from Data: Artificial Intelligence and Statistics V, pages 207–17, 1996

[33] D.H. Wolpert, Stacked generalization. Neural Networks 5, pages 241–259, 1992

[34] C. C. Aggarwal, A. Hinneburg and D. A. Keim. On the Surprising Behavior of Distance Metrics in High Dimensional Space. Proceedings of the ICDT Conference, pages 420-434, 2001.

[35] K.S. Beyer, J. Goldstein, R. Ramakrishnan and U. Shaft When is “nearest neighbor” meaningful? Database Theory -ICDT , 7th International Conference, Proceedings. Volume 1540 of Lecture Notes in Computer Science, pages 217-235, 1999.

[36] N. Joshi, B. Wilburn, V. Vaish, M. Levoy and M. Horowitz, Automatic Color Calibration for Large Camera Arrays, UCSD CSE Technical Report CS2005-0821, 2005.

[37] F. Pichaud, A. Briscoe, and C. Desplan, Evolution of color vision, Current Opinion in Neurobiology 9, pages 622-627, 1999

[38] S. Sinha, J.-M. Frahm, and M. Pollefeys, GPU-based Video Feature Tracking and Matching, Tech. Rep. TR06-012, University of North Carolina at Chapel Hill, 2006.