A TOOL FOR QUERY AND ANALYSIS OF MPEG ENCODED VIDEO

Chris Kreucher

Submitted to the Horace Rackham School of Graduate Studies of the University of Michigan in partial fulfillment of the requirements for the degree of

MASTER OF SCIENCE IN ENGINEERING

May 1997

The University of Michigan – Dearborn

Department of Electrical and Computer Engineering

A Tool For Query and Analysis of MPEG Encoded Video

Chris Kreucher

Approved as to the style and content by:

Sridhar Lakshmanan, Chair of Committee

Karl Kluge, Member

K. Venkatesh Prasad, Member

ABSTRACT

This thesis presents a tool for efficient query and analysis of MPEG encoded video. The image sequences for the video are obtained via a vehicle-mounted forward-looking camera. The tool accepts queries that signify both temporal and geometric events as the vehicle traverses common roadways, such as "identify portions of the video where the vehicle is making a lane change maneuver," or "identify portions of the video where the vehicle is going around a tight curve," etc. Both query and analysis are done directly in the encoded domain.

The tool deals with the encoded video stream much like a standard MPEG decoder:

- For inter-coded (I) frames of the MPEG video stream, the shape of the lane/pavement markers present in those frames is estimated by a comprehensive search procedure. This lane shape estimation is accomplished in a Bayesian setting using a set of DCT-based lane edge features, a global shape model for lane edges, and a coarse-to-fine optimization algorithm.
- The estimation of lane shape in P-frames is somewhat different. Using the motion of macroblocks between frames, and the estimate of lane shape in the previous I- or P-frame, the tool quickly identifies the position and geometry of the lane/pavement markers in the new P-frame. This involves a non-linear least squares fit, followed by a local search in the lane shape parameter space. The lane shape estimation procedure for B-frames is very similar to that for P-frames.
- Finally, the lane's geometry in each frame and as it evolves over time is analyzed in order to establish a connection between its signature and the one mandated by the query.

Several experimental results are presented to illustrate the efficacy of the tool.

TABLE OF CONTENTS

1 Introduction **1**

1.1 Background 1

1.2 Review of Related Work 2

1.3 Main Contributions of This Thesis 4

1.4 Thesis Organization 8

2 Extracting Lane Features in the Compressed Domain **9**

2.1 Background 9

2.2 Frequency Domain Features of Lane Edges 10

2.3 Deformable Template Shape Model 13

2.4 Bayesian Lane Detection 14

2.5 Experimental Results 15

2.6 Comparison to Previously Proposed Lane Detection Systems 18

2.7 Summary and Conclusions 25

3 Matching Semantic Queries to Features Extracted from JPEG Encoded Images **26**

3.1 Background 26

3.2 The Matching Method 26

3.3 Summary and Conclusions 31

4 Feature Tracking and Temporal Event Query/Browsing for MPEG Encoded Video Sequences **32**

4.1 Background 32

4.2 Initial Guess of The Lane's Shape Using Motion Vectors 33

4.3 Lane Shape Refinement 34

4.4 Extension to B-frames 35

4.5 Example Sequence 36

4.6 Results of MPEG Encoded Video Query 37

4.7 Summary and Conclusions 41

5 Concluding Remarks **42**

Appendix A: A Summary of the JPEG Standard **43**

Appendix B: A Summary of MPEG Standard **47**

References **50**

CHAPTER 1 – INTRODUCTION

1.1 BACKGROUND

The enormous amount of image and video data that typifies many modern multimedia applications mandates the use of encoding techniques for efficient storage and transmission. The video encoding (compression) standard of choice in many new personal computers, video games, digital video recorders/players/disks, digital television, etc. is the one adopted by the Motion Pictures Experts Group (MPEG)–see [1][2], Appendix A for a brief description of the Joint Photographic Experts Group (JPEG) standard, and Appendix B for the MPEG standard. Since this standard is being so widely accepted, it is important to develop tools that will enable MPEG encoded videos to be easily manipulated. These include tools that directly manipulate MPEG encoded video to achieve commonly desired functions such as overlap, translation, scaling, linear filtering, rotation, pixel multiplication, etc.–see [3].

While such tools are very useful for certain types of popular video editing tasks, they are not so relevant for other important functions such as feature extraction, query, and browsing. Many techniques do exist for feature extraction, query and browsing of spatial domain video–see [4-6] for several different examples. However, they all involve the expensive and inefficient operation of decoding MPEG encoded videos before they can be applied.

This thesis presents a tool for shape-based feature extraction, query, and browsing of MPEG encoded video sequences without the need for any expensive/inefficient decoding. Specifically, the video sequences in question consist of common roadway scenes, as imaged by a forward-looking vehicle-mounted camera. The objective is to develop a tool for retrieving portions of the video using geometric frame-based queries such as “find portions of the video where the lanes are narrow” or “find portions of the video where the lanes are straight,” etc., as well as temporal sequence-based queries such as “find portions of the video where the driver is making a lane change” or "find portions of the video where the driver is going around a tight curve" etc.

There are three main constituents to this thesis:

- A method of extracting features from images in the encoded domain. This entails finding frequency domain signatures that correspond to the spatial domain features of interest, such as lane and pavement edges.

- A way of matching geometric-event queries that describe the desirable shape of the lane in a target frame to these encoded domain features. This involves translating the query into a constraint on the global distribution of the lane features, and computing a goodness-of-fit between the constrained global lane feature distribution and the target image.
- A procedure to track lane features through MPEG encoded video sequences, and interpret their evolvement in the context of retrieving portions of the video that match temporal-event queries of interest. This involves the establishment of a correlation between the query in question and the corresponding plots of the lane's geometry over time.

1.2 REVIEW OF RELATED WORK

This review is broken down in terms of the relationship of the previously published results to each of the three main thesis constituents. First up is a review of previously published methods for extracting features from images in the encoded (frequency) domain. There are many previously published papers that deal with frequency domain counterparts to spatial domain features [7-16]. Some of these papers [7-10] deal with the problems of texture image restoration, segmentation, and classification using frequency domain features. While others [11-15] use frequency domain features to extract edges and also to achieve edge-preserving image coding/compression. References [11], [15], and [16] are the most relevant to this work. Especially, reference [16] deals with a problem similar to one of the components of this thesis: Curve extraction using a multi-dimensional Fourier transform. Curves are represented in a piecewise linear fashion and linked together via a quad tree. At any given node, the image’s intensity profile is used to determine whether or not an edge is present at that node. This is accomplished by using a multi-resolution Fourier transform (MFT). Large regions of the image are first examined in the MFT domain for the presence/absence of edge-like features. If an edge-like feature is deemed present in a certain region, then the region is further subdivided by using the quad tree, and a similar presence/absence decision is made at the lower nodes of the tree. This process is repeated until every pixel in the image has a classification in terms of whether or not it lies on an edge. The MFT is convenient for detecting edge features at multi-resolutions and has been used to detect globally relevant edges in a variety of images. The approach presented in this thesis has some commonality with the one in [16] in the use of frequency domain to detect edge-like features and the interpretation of these features' significance in a global context. However, the methods and models employed here are vastly different than [16].

Next is a review of previously published methods that establish a connection between encoded/compressed images and semantic queries regarding their content. Many methods for querying/browsing encoded or compressed images have been proposed earlier, and they broadly fall into two categories:

1) Those that encode images so that certain spatial domain features are easily identified even upon encoding.
2) Those that work directly with images encoded using standard discrete cosine transform (DCT) based compression techniques such as JPEG, MPEG, H.261, etc.

References such as [17-22] belong to the first category, and they develop procedures that enable spatial domain features such as color, shape, texture, etc. to be easily identified directly in the encoded domain. These methods have the appeal that the image database is created with the queries of potential interest directly influencing the encoding process. However, they suffer from two problems: either (a) they involve non-standard techniques, or (b) they result in compression ratios that are not favorable in comparison to standard JPEG/MPEG. References such as [23][24][6] belong to the second category. Like our system, they are appealing for the simple reason that they work on images encoded using standard techniques. However, there are some important limitations. For example, [23] uses "image keys"–a vector of encoded image-based quantities–to match a query with other encoded images with similar keys. This indeed matches visually similar images. There is, however, no semantic meaning to the keys generated. Reference [24] focuses on low-level visual features such as texture, color, and shape in the spatial domain. By deriving encoded domain analogues of low-level spatial domain features, it computes a similarity measure between a query image and the images in a database based on how close their corresponding encoded domain features are. However, no global significance can be attached to this match. Reference [6] is similar to [24], but only involves color and texture (not shape) matching in the encoded domain.

Finally is a review of previously published methods for efficient browsing of MPEG encoded video. Reference [26] presents a technique for extracting text-related captions from MPEG encoded video sequences. Frames of the encoded video stream are analyzed and segmented in order to separate the text-related regions from the rest. Reference [27] reports a method for determining whether or not an encoded video clip contains a human face, and if so, their number and position within that clip. Within each frame of the encoded video stream face-like regions are detected (if present), by examining the discrete cosine transform (DCT) coefficients for the presence/absence of certain types of color and edge features. Reference [28] uses the difference between successive I-frames and features derived from the motion vectors in P- and B-frames in order to accomplish a direct segmentation of an MPEG encoded video into its constituent shots. The segmented video is then subsequently characterized in terms of motion and camera work. Reference [29] proposes a simple algorithm to determine abrupt scene changes in an MPEG encoded video sequence through analysis using motion information at the macroblock level. Bit rate information and the number of motion predicted blocks are both used to identify potential locations within the encoded video where the difference between successive frames is significant. Reference [30] finds both gradual and abrupt scene changes by analyzing the frame-to-frame evolution of the DCT coefficients corresponding to just the spatial domain DC average alone. A considerable amount of research has also been done in the area of video sequence coding that allow certain spatial domain features of interest (such as color, texture, shape, etc.) to be easily extracted directly from the encoded representation [31][32]. The shortcoming of this strategy is that the resulting encoding techniques are specific to the problem, hence the encoded video is not portable or exchangeable with other systems and users. Therefore, if the video is already in MPEG compressed format, it must be decoded and re-encoded in the new format before these systems can be used.

1.3 Main Contributions Of This Thesis

As described in section 1.1, this thesis is concerned with the problem of querying or browsing MPEG encoded video sequences of common roadway scenes. The main contributions of this thesis are three-fold:

- First is the identification, usage, and systematic evaluation of a set of frequency domain features for the purposes of lane detection in the encoded domain. The identification of these frequency domain features is based on a novel set of DCT coefficients that capture relevant information concerning the strength and orientation of spatial lane edges. These features are used in combination with a deformable template shape to develop an estimate of global lane shape. The efficacy of this method is established, and compared to a spatial domain method.

- Second is the interpretation of the estimated lane shape to allow query by image content. The lane shape estimate developed earlier allows global characteristics of the features to be identified and categorized so that queries such as “find portions of the video containing wide lanes” can be used to find particular instances in a database of JPEG encoded images. This is facilitated by a new posterior density based ratio test for assessing how well an image answers a query.
- Third is the interpretation of the lane feature's motion through an encoded video sequence. This is used to both improve the feature detection process and also to answer temporal queries such as “find portions of the video where the vehicle makes a lane change” that call for a specific signature to be associated with the lane's evolving geometry through time.

We provide below a sample of the results obtained in this thesis–more examples and results are presented in the main body. The first constituent of this thesis is feature extraction from individually encoded frames of MPEG video. This amounts to developing a technique of feature extraction from JPEG encoded still images, as MPEG inter-coded (I-) frames are nearly identical to JPEG encoded images (see Appendices A and B). In order to test the efficacy of this procedure, the algorithm was applied to a widely varying set of roadway images, including images obtained under a variety of lighting and environmental conditions, shadowing, lane occlusion(s), solid and dashed lines, etc. Figure 1 shows the results:

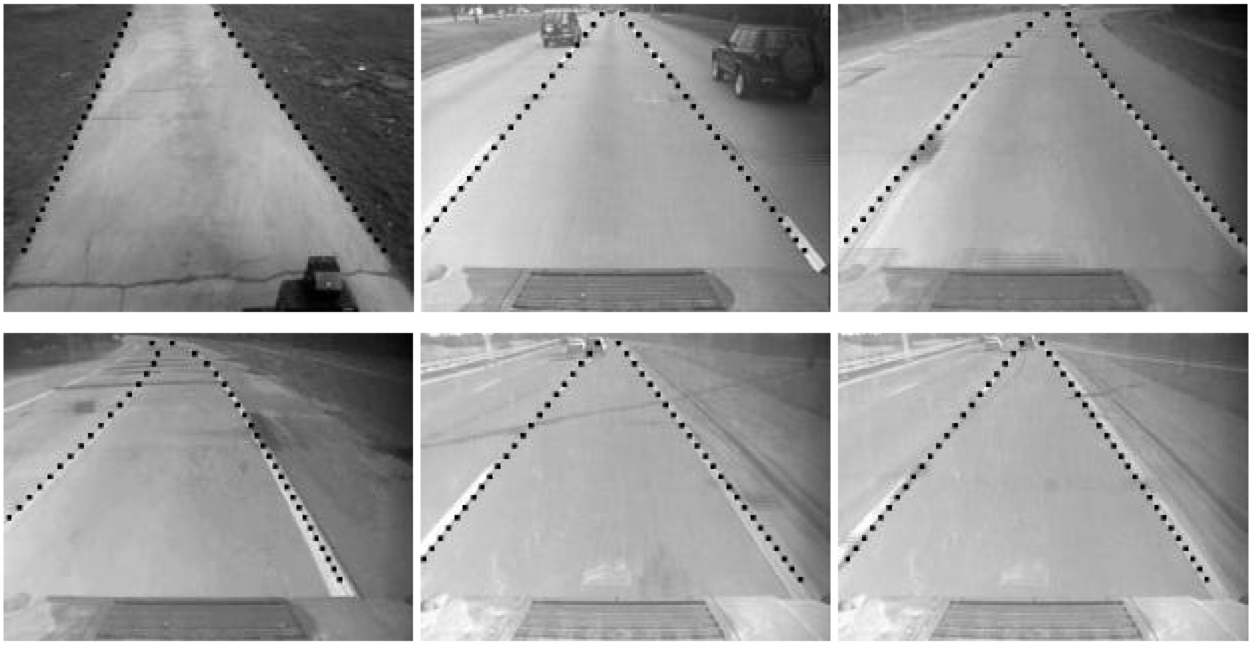


Figure 1. Examples of lane finding in JPEG images.

The second constituent of this thesis is a method for querying still images based on geometric properties of the lane's shape. Figure 2 contains four typical database images. This database was processed and the results for four common queries–a search for roads with straight, curved, wide, and narrow lanes–is given in Table 1. The scores shown in Table 1 are a measure of the goodness-of-fit between the query and the image in question, and as one notices the scores accurately reflect the geometric properties of the lane.

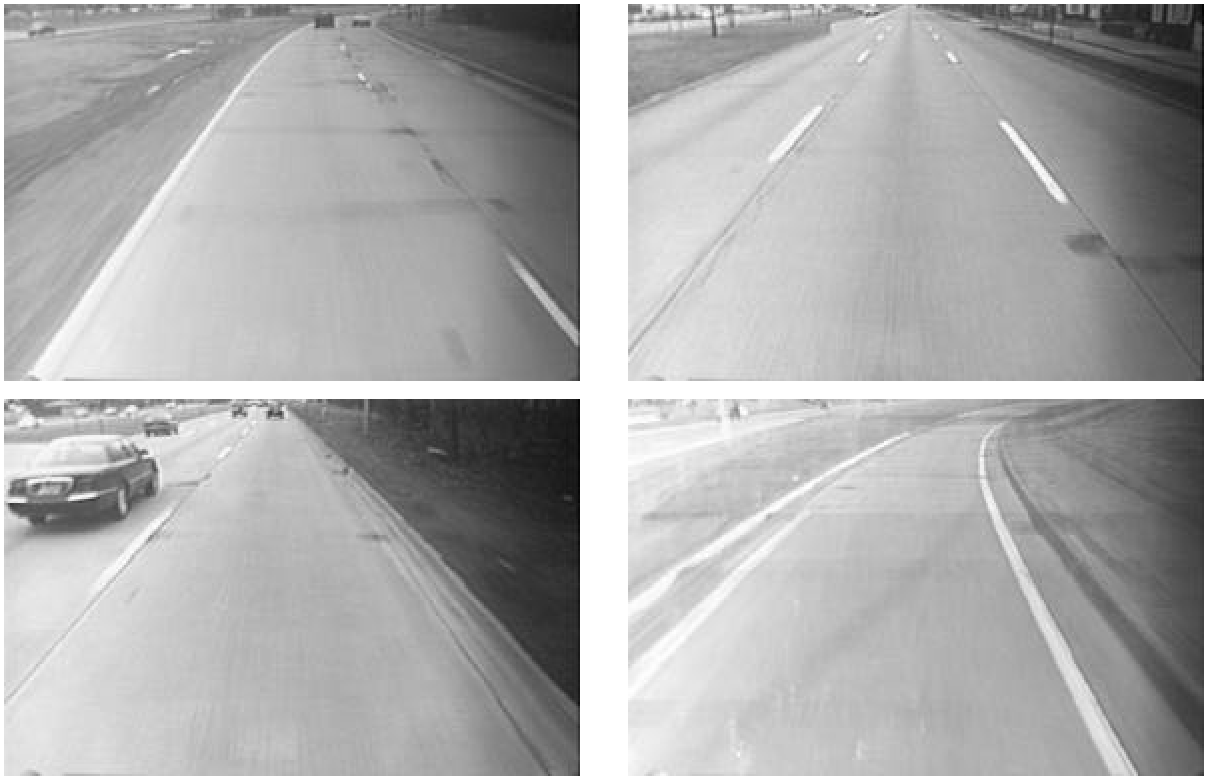


Figure 2. Four images from the database, numbered in raster scan as image 1, 2, 3 and 4.

	"Straight" Score	"Curved" Score	"Narrow" Score	"Wide" Score
Image 1	0	17.04	9.91	0
Image 2	16.85	0	11.48	0
Image 3	12.45	0	0	24.21
Image 4	0	82.82	0	40.27

Table 1. The results of querying the images in Figure 2.

The third and final constituent of this thesis is a motion-based feature tracking method to perform temporal-based queries on MPEG encoded video, based on the feature extraction and image query techniques described earlier. Figure 3 shows a short sequence of 12 frames from an MPEG encoded video, where the query "find portions of the video where the lanes are straight" is used. The sequence given consists of the vehicle traversing a road that is straight at first and then becomes curved. The resulting frames returned by the system are highlighted in red.

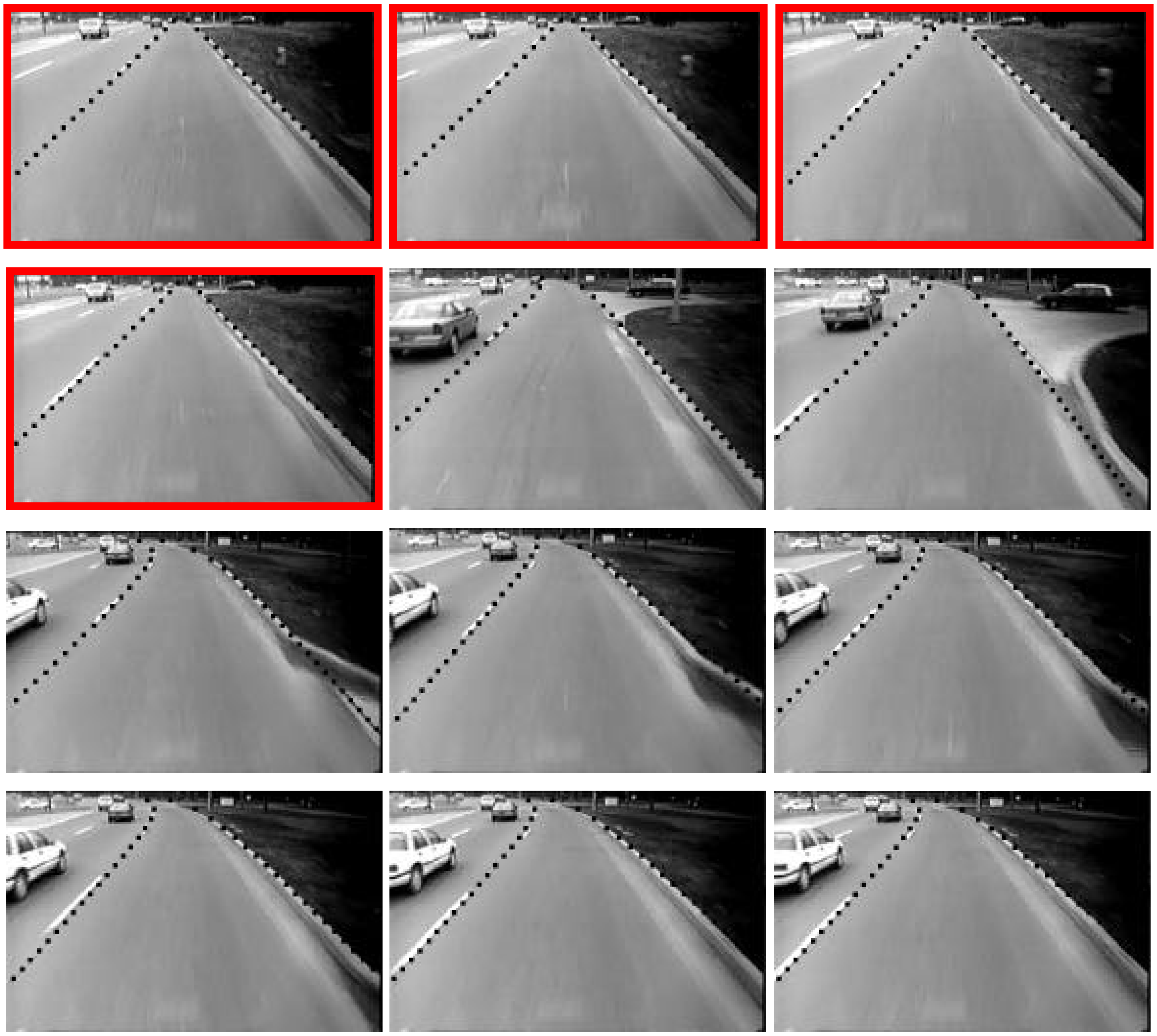


Figure 3. A 12 frame sequence where the result of the query "find the portion of the sequence containing a straight lane" is highlighted in red.

By a proper adaptation of its constituents, the tool developed in this thesis can indeed be made to query/browse MPEG encoded video archives that are more general than those that contain just roadway scenes. The tool can also be made to accept queries that concern other geometric/temporal attributes contained in those video sequences, such as those in [6][25].

However, in its present form the tool cannot be readily adapted to handle queries such as those in [23] and [24] that involve appearance and color, respectively.

1.4 Thesis Organization

This rest of this thesis is organized as follows. Section 2 presents the method of extracting lane features in the encoded/compressed domain as well as the procedure for determining the lane's geometry from the global distribution of these features. Section 3 shows how the frequency-domain feature detection method described in section 2 is used to form a system for querying and browsing JPEG encoded still image databases. Section 4 provides the aforementioned extension to MPEG video query, by developing a method of tracking image features through encoded video sequences and using this tracker to answer temporal event queries. Finally, section 5 concludes with some relevant remarks and some possible extensions of this work.

CHAPTER 2 - LANE FEATURE EXTRACTION IN THE ENCODED DOMAIN

2.1 BACKGROUND

In this chapter, we present the method of extracting lane features from individual images in the encoded domain. This discussion pertains to JPEG encoded still images and MPEG frames that are inter-coded (I-frames). The method is based on a set of frequency domain features that capture relevant information concerning the strength and orientation of spatial lane edges and proceeds as follows.

As specified in the JPEG and MPEG standards [1][2], an encoded image consists of a series of 8×8 pixel blocks. In the technique developed here, a feature vector representing the amount of "perspective lane edge energy" is calculated directly from the DCT coefficients (utilized by JPEG and MPEG) for each 8×8 pixel block. The block feature vectors are then collectively used in combination with a deformable template model of the desired lane markers. This combination is accomplished in a Bayesian setting, where the deformable template model plays the role of a prior probability, and the feature vectors are used to compute a likelihood probability. The lane detection problem is reduced to finding the global maximum of a four-dimensional posterior probability density function, and this is done using a computationally efficient coarse-to-fine search strategy.

The rest of this chapter is organized as follows. The method of lane detection in JPEG encoded images is broken into three subsections. The method of determining the frequency domain signature of lane edge features is developed in section 2.2. The deformable template shape model which is used as a global constraint on the distribution of lane edge features is give in section 2.3. Section 2.4 encompasses the Bayesian combination of the shape model and the frequency domain features. The experimental results are given in the two subsections immediately following the details of the method. Section 2.5 includes a small database of images and the resulting lane shape estimates found for those images. Section 2.6 presents a comparison between this frequency domain method of lane detection and a spatial domain method, as well as a review of previously published vision-based lane detection algorithms.

2.2 FREQUENCY DOMAIN FEATURES OF LANE EDGES

Much work has been done on the manipulation of DCT coefficients to create desired effects in the spatial domain. See, for example, references [33], [34], and [35] for algorithms that perform geometric transforms including rotating, flipping, and shearing in the spatial domain by directly manipulating the DCT coefficients. These techniques are typically based on the mathematical properties of the DCT that allow simple coefficient manipulation to produce different spatial domain effects. Other procedures include image enhancement and downscaling as described in [36].

There have also been several attempts at image analysis and low-level feature extraction through investigation of the DCT coefficient distributions. Specifically, reference [37] presents a method of identification of edges based on an inspection of DCT coefficients. Edges are coarsely identified by comparing the magnitudes and signs of some of the transform coefficients. Furthermore, some techniques for determining the strength and orientation of such coarsely determined edges are also presented. Reference [15] performs a classification of DCT blocks based on three directional activity indices calculated directly from the transform coefficients. These directional activity indices are determined by summing the energies of DCT coefficients in certain positions in the frequency domain. The indices allow determination of edge magnitudes roughly corresponding to horizontal, vertical and diagonal directions in the spatial domain image. In [15], this analysis was used strictly for augmenting JPEG, as different quantization tables can be used for blocks of differing orientations. The same technique has also been used for verification of detected features in a face detection systems–see [27].

Lane edges are the objects of interest in this work. Recall that we are looking for features that discriminate between lane markings and extraneous (non-lane) edges. An examination of roadway scenes obtained from a forward-looking vehicle-mounted camera easily reveals that lane markers tend to have “diagonally dominant” orientations in the image plane due to the perspective transformation inherent in the ground plane imaging process, whereas the extraneous edges have no such preferred orientations (see Figure 4 for examples). We have found the frequency domain to be a convenient vehicle to discriminate between edges that are diagonally dominant and those that are randomly oriented. Details follow.

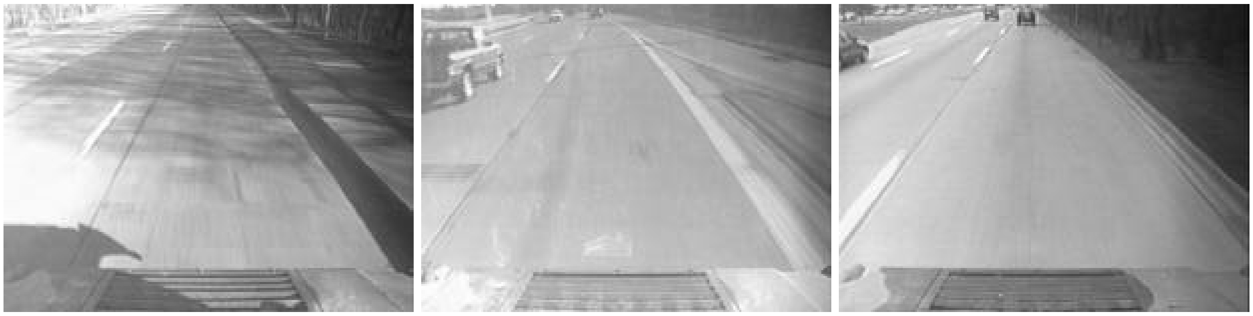
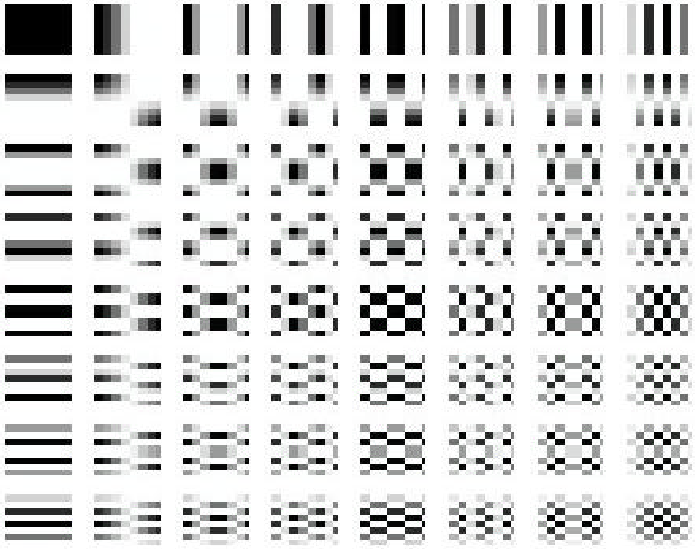


Figure 4. Examples of several roadway scenes to illustrate the fact that lane and pavement edges are oriented diagonally in the image plane

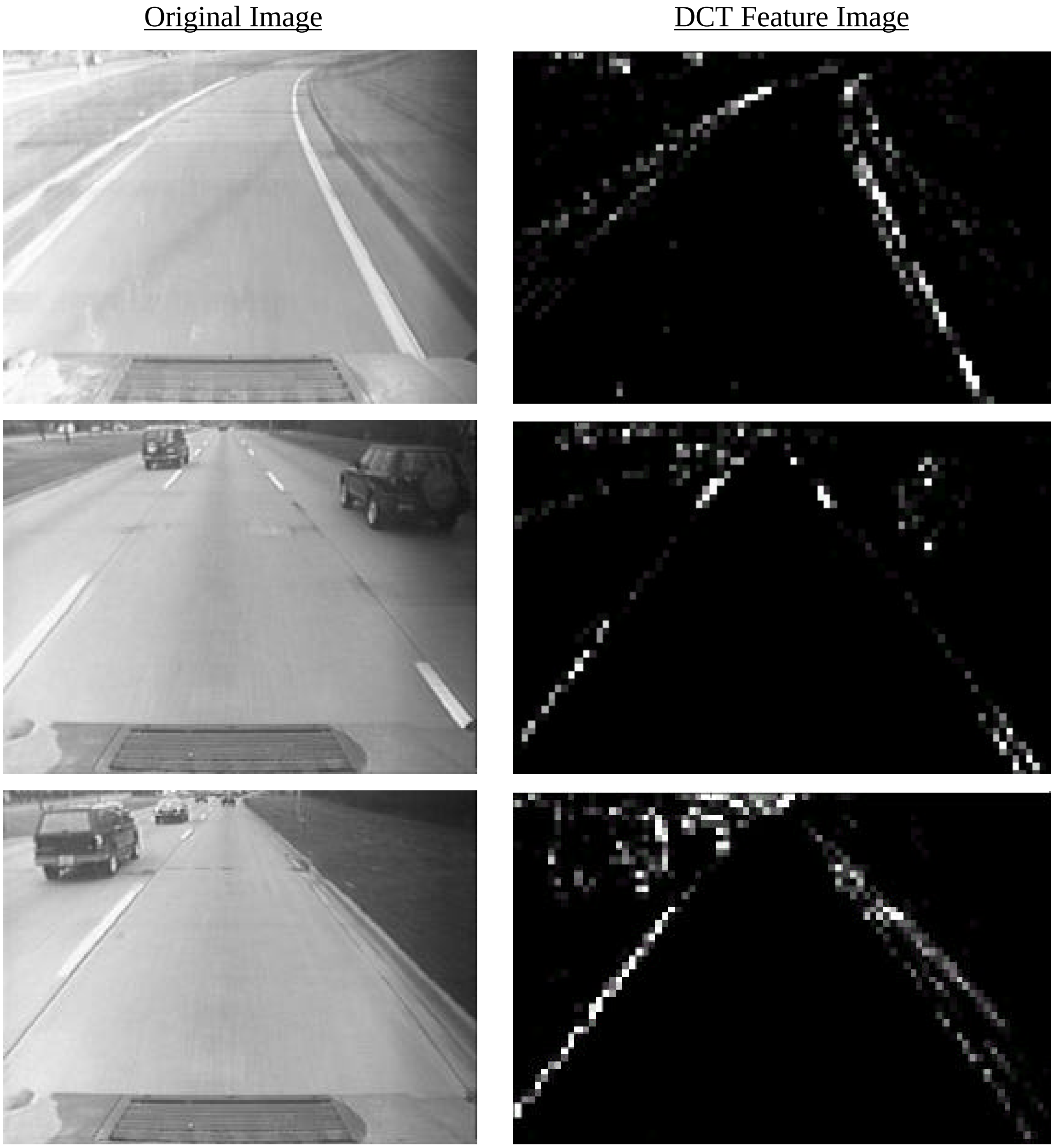
Under the JPEG and MPEG standards for image coding [1][2], a given image is first divided into 8×8 blocks of pixels. Each of the 8×8 pixel blocks are then orthogonally decomposed in terms of a set of 64 discrete cosine transform (DCT) bases. Each of these bases, as seen in Figure 5, correspond to spatial domain edges of a certain spatial frequency and orientation. Out of these 64 bases, it has been found that “diagonally dominant” edges are best represented by a set of 12. The matrix in Figure 5 indicates which 12 bases out of the 64 they are. Figure 6 shows several examples of the “value” of these 12 bases from the standpoint of lane detection. For each of the original images in Figure 6, the corresponding feature images are obtained by just summing the squares of its 12 special DCT decompositions. Note, in each case, despite the original image having features/edges of various strengths and orientations, the corresponding DCT feature images contain only information about diagonally dominant edges.



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

Figure 5. Left: The DCT bases. Right: The matrix that represents which 12 of the 64 bases capture diagonally dominant edges.

Note that the frequency domain features adopted here are similar to the ones presented in [15][27]. In [15], these frequency features were used for code book optimization. Whereas in [27], the objective was to detect faces using these frequency domain features.



Original Image

DCT Feature Image

Figure 6. DCT features of typical roadway scenery.

2.3 Deformable Template Shape Model

As mentioned earlier, the method for lane detection in JPEG encoded images uses a global shape model to predict the manner in which lane markers appear in the image plane. As commonly done [38], we assume that lane markers are circular arcs on a flat ground plane. For small-to-moderate curvatures, a circular arc with curvature *k* can be closely approximated by a parabola of the form:

$$x = \frac{1}{2} k\, y^2 + m\, y + b \tag{1}$$

The derivation of the class of corresponding curves in the image plane is given for the case of an untilted camera, but it can be shown that the same family of curves results when the camera is tilted. Assuming perspective projection, a pixel *(r, c)* in the image plane projects onto the point *(x, y)* on the ground plane according to the equations:

$$x = c\, c_f\ y \tag{2}$$

and

$$y = \frac{H}{r\, r_f} \tag{3}$$

where *H* is the camera height, r_f is the height of a pixel on the focal plane divided by the focal length, and c_f is the width of a pixel on the focal plane divided by the focal length. Substituting eqs. (2) and (3) into eq. (1) and performing some simple algebraic manipulation results in the image plane curve:

$$c = \frac{k\, H}{2\, r_f\ c_f\ r} + \frac{b\, r_f\ r}{H\, c_f} + \frac{m}{c_f} \tag{4}$$

or, combining the ground plane and camera calibration parameters together,

$$c = \frac{k'}{r} + b'\, r + vp \tag{5}$$

In the case of a tilted camera, the same family of curves results if the image coordinate system is defined so that row 0 is the horizon row. For left and right lane edges defined by concentric arcs, the approximation is made that the arcs have equal curvature and equal tangential orientation where they intersect the *X* axis, so k' and *vp* will be equal for the left and right lane edges. While the radius of curvature and tangent orientation of the left and right lane edges will differ slightly, constraining the left and right lane edges to have the same k' and *vp* parameters closely approximates the actual lane edge shapes for all but very small radii of curvature. As a result, the lane shape in an image can be defined by the four parameters k', b'_{LEFT}, b'_{RIGHT}, and vp. In summary, the k' parameter is linearly proportional to the curvature of the arc on the ground plane. The *vp* parameter is a function of the tangential orientation of the arc on the ground plane, with some coupling to the arc curvature as well (depending on the amount of camera tilt). The b'_{LEFT} and b'_{RIGHT} parameters are functions of the offset of the arc from the camera on the ground plane, with couplings to arc curvature and tangential orientation (again, the relative contributions of these couplings depend on the camera tilt) [45].

In the next section, this deformable template shape model is used in conjunction with the DCT-based lane features described in section 2.2 to estimate the exact geometric shape of the lane markers.

2.4 Bayesian Lane Detection

Real world lanes are never too narrow, wide or curved. Accordingly, a prior probability density function (pdf) is constructed over the lane shape parameters k', b'_{LEFT}, b'_{RIGHT}, and vp:

$$\mathrm{P}(k', b'_{LEFT}, b'_{RIGHT}, vp) \propto \{\mathrm{atan}\,\alpha[b'_{RIGHT} - b'_{LEFT} - 1] - \mathrm{atan}\,\alpha[b'_{RIGHT} - b'_{LEFT} - 3]\} \times \{1 - \beta\left(\frac{\mathrm{k'}}{\mathrm{x}}\right)^2\}, \qquad (6)$$

where $\alpha = 10$, $\beta = 0.01$ and $c = 600$ values are chosen to reflect that *a priori* knowledge.

It is also assumed that given the values of k', b'_{LEFT}, b'_{RIGHT}, and vp, the probability of the JPEG encoded image having a certain set of DCT feature values (the ones described in section 2.2) is given by the likelihood pdf:[1]

$$P(DCT\ feature\ values \mid k', b'_{LEFT}, b'_{RIGHT}, vp) \propto \sum_{i,j} \sum_{k,l \in C_{i,j}} (dct\_coeff(k,l))^2, \qquad (7)$$

where the sum over *(i,j)* covers those 8×8 pixel blocks through which the left and right lanes (as dictated by k', b'_{LEFT}, b'_{RIGHT}, and vp) pass, $C_{i,j}$ denotes the set of 12 DCT bases that capture diagonally dominant edge features (see section 2.2 for details), and *dct_coeff(k,l)* denote the *(k,l)*th DCT coefficient of the *(i,j)*th block of 8×8 pixels. This likelihood pdf encodes the

[1] Ideally, the likelihood pdf should also contain a normalizing factor that depends on the lane shape parameters. However, calculating this factor is very difficult, as it involves an integration of the RHS of eq. (7) over all the DCT feature values. Omission of such a factor is ubiquitous to Bayesian methods in image analysis.

knowledge that the true lane markers lie along portions of the image that uniformly have a high amount of perspective (diagonally dominant) edge energy.

These two pdfs are combined using Bayes' rule, and the lane detection problem is reduced to one of finding the global maximum of a posterior pdf (i.e., the MAP estimate):

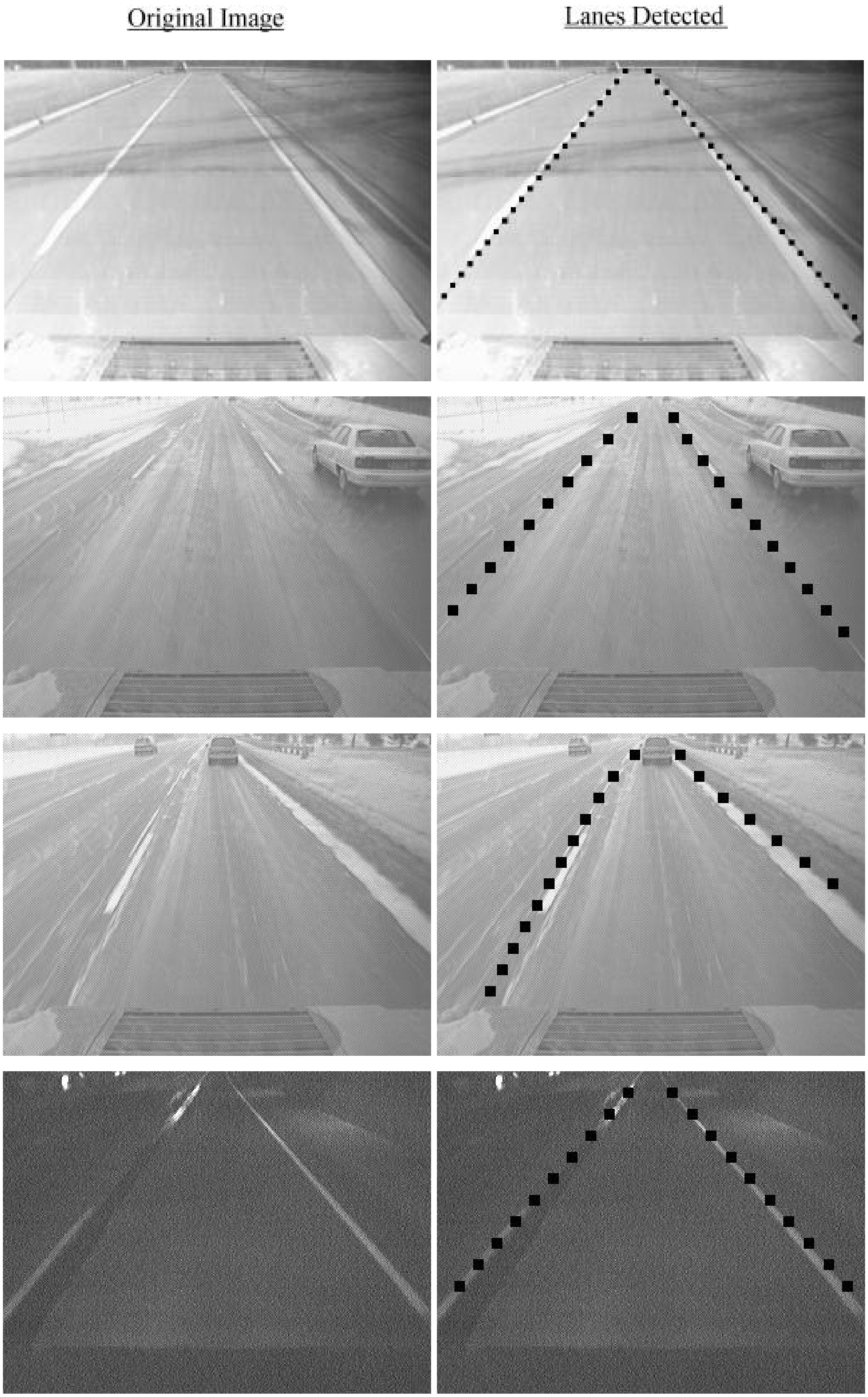
$$
\begin{aligned}
&\underset{k',b'_{LEFT},b'_{RIGHT},vp}{\operatorname{argmax}} \mathrm{P}(k',b'_{LEFT},b'_{RIGHT},vp \mid \text{DCT feature values})\\
&= \underset{k',b'_{LEFT},b'_{RIGHT},vp}{\operatorname{argmax}} \mathrm{P}(k',b'_{LEFT},b'_{RIGHT},vp) \times \mathrm{P}(\text{DCT feature values} \mid k',b'_{LEFT},b'_{RIGHT},vp)\\
&= \underset{k',b'_{LEFT},b'_{RIGHT},vp}{\operatorname{argmax}} \Big(\operatorname{atan} a\,((b'_{RIGHT}-b'_{LEFT})-1) - \operatorname{atan} a\,((b'_{RIGHT}-b'_{LEFT})-3)\\
&\qquad \times (1 - b\left(\frac{k'}{c}\right)^2) \times \sum_{i,j} \sum_{k,l \in C_{i,j}} (dct\_coeff(k,l))^2
\end{aligned}
\tag{8}
$$

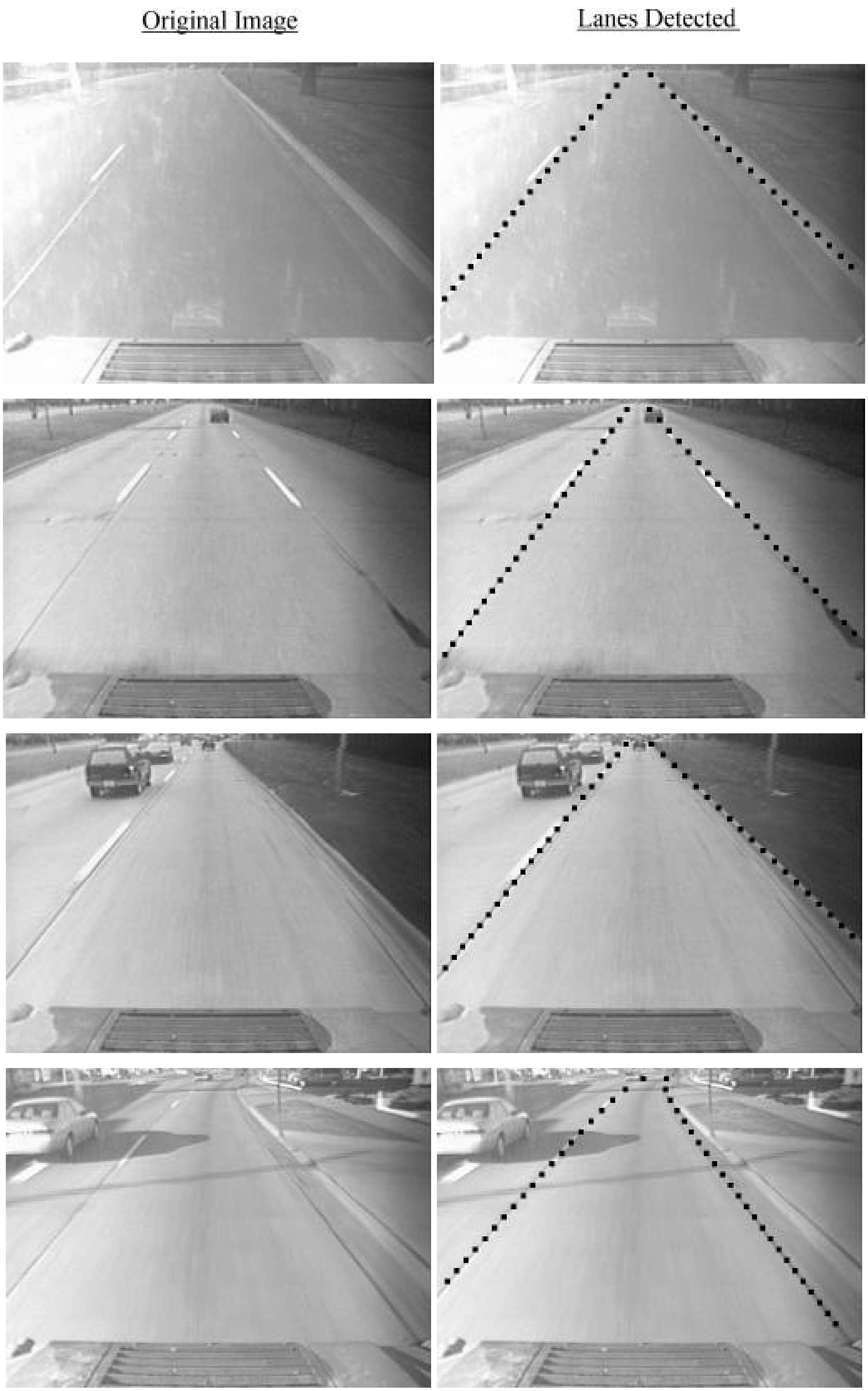
The MAP estimate is found by a coarse-to-fine search over the four-dimensional parameter space of k', b'_{LEFT}, b'_{RIGHT}, and vp. First, the parameter space is searched very coarsely, resulting in a quick maximization of eq. (8). The result of this coarse search is then used as the center of another, finer search. This second search encompasses a much smaller area (half in each dimension) of the parameter space in comparison to the first one, but the area is searched more finely. This process is repeated for a third and final time. The maximum value of this last search is then used as the MAP estimate of the lane shape contained in that target image.

We have found that this coarse-to-fine approach results in superior performance in comparison to that of several other algorithms that were attempted for obtaining the MAP estimate, including conjugate gradient, downhill simplex, and the Metropolis algorithm. In fact, if the number of objective function evaluations in eq. (8) is fixed, then this coarse-to-fine strategy even outperforms the customary exhaustive search over the whole parameter space sampled/divided accordingly.

2.5 EXPERIMENTAL RESULTS

The lane extraction procedure described in the previous sections was applied to a varied set of images. The images include those that were obtained under a variety of lighting and environmental conditions, shadowing, lane occlusion(s), solid and dashed lines, etc.


Original Image
Lanes Detected


Original Image
Lanes Detected

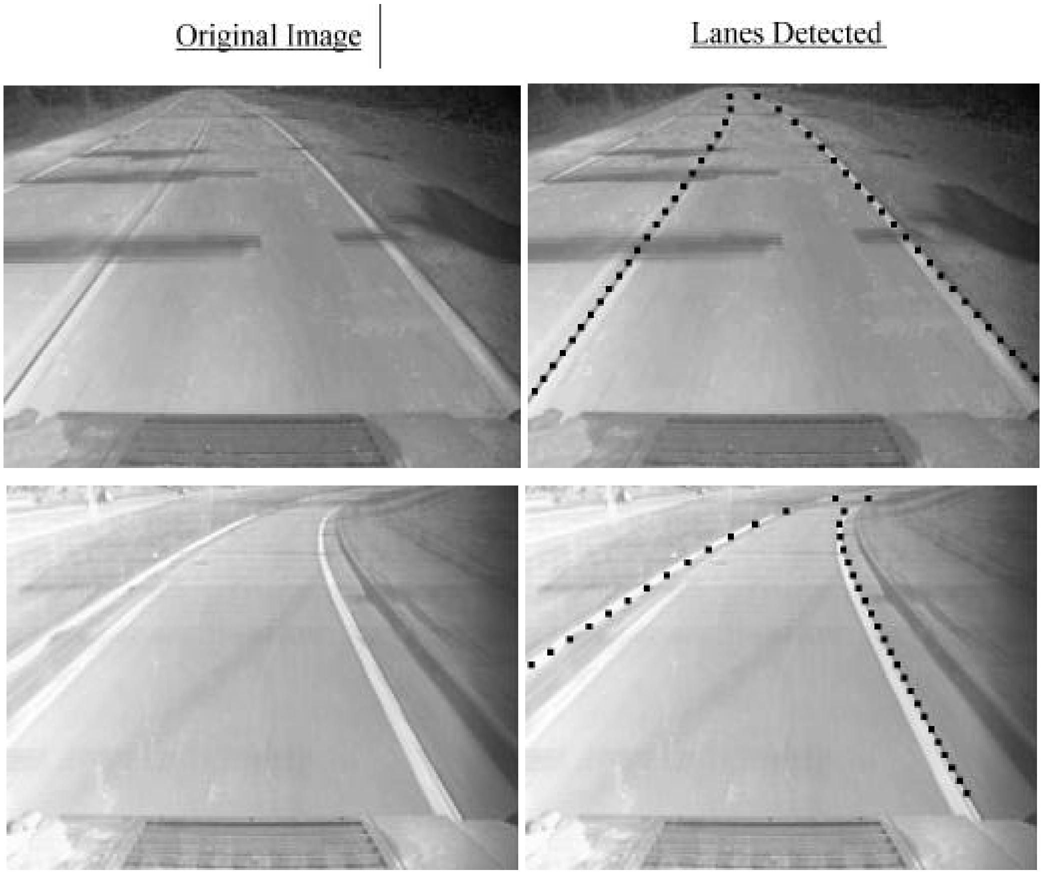


Figure 7. Example results of processing a varied set of images

2.6 Comparison to Previously Proposed Lane Detection Systems

Given the results shown in the proceeding sections, it is natural to make the assertion that the frequency domain based lane detection method developed in this thesis may be useful just by itself. To this end, a survey of standard (spatial domain) lane detection techniques is warranted, and a comparison to such traditional spatial domain techniques is necessary.

Lane detection, the process of locating lanes in an image with no prior estimate to aid the search, is an important enabling or enhancing technology in a number of intelligent vehicle applications, including lane excursion detection and warning, intelligent cruise control, lateral control, and ultimately autonomous driving. Studies such as [40-42] contain a detailed discussion of these applications and their overall impact on the economy, environment, and driver safety.

The first generation of lane detection systems were all edge-based. They relied on thresholding the image intensity to detect potential lane edges, followed by a perceptual grouping of the edge points to detect the lane markers of interest. Also, often times the lanes to be detected were assumed to be straight. See [43-45] and the references therein. The problem with thresholding the intensity is that, in many road scenes, it isn't possible to select a threshold which eliminates the detection of noise edges without also eliminating the detection of true lane edge points. Therefore, these first generation lane detection systems suffered when the images contained extraneous edges due to vehicles, on-off ramps, puddles, cracks, shadows, oil stains, and other imperfections in the road surface. The same deficiency also applied when the lanes were of low contrast, broken, occluded, or totally absent.[2]

The second generation of systems sought to overcome this problem by directly working with the image intensity array, as opposed to separately detected edge points, and using a global model of lane shape. For example, ARCADE [45] uses global road shape constraints derived from an explicit model of how the features defining a road appear in the image plane. A simple one-dimensional edge detection is followed by a least median squares technique for determining the curvature and orientation of the road. Individual lane markers are then directly determined by a segmentation of the row-averaged image intensity values. ARCADE, unlike its predecessors, does not require any perceptual grouping of the extracted edge points into individual lane edges. The RALPH system [46] is another example of a second generation lane detection system. Like ARCADE, it too uses global road shape constraints. The crux of RALPH is a matching technique that adaptively adjusts and aligns a template to the averaged scanline intensity profile in order to determine the lane's curvature and lateral offsets. The LOIS lane detector [47], yet another example of a second generation lane detection system, uses template matching as well. However, unlike RALPH, LOIS' match is over the entire image and not just an averaged scan line. At the heart of LOIS is a likelihood function that encodes the knowledge that the edges of the lane should be near intensity gradients whose orientation are perpendicular to the lane edge. This allows strong magnitude gradients to be discounted if they are improperly oriented and weak magnitude gradients to be boosted if they are properly oriented. There are several other such second generation systems; the reader is referred to [48] for a description of those. Many of these have been subject to several hours of testing, which involved the processing of extremely large and varied data sets, and it suffices to say that the second generation lane detection systems perform significantly better in comparison to the first generation ones.

[2] As would be the case when the road has no lane markers, but only pavement edges.

However, not all of the problems associated with the first generation systems have been overcome. In particular, a number of second generation lane detection systems still have a tendency to be “distracted” or “pulled” away from the true lane markers by the presence of strong and structured edges such as those created by a vehicle outline.[3] In portions of the image whose distance from the camera is large, vehicle outlines have a much higher contrast compared to the true lane markers. In such cases, hypotheses that include the vehicle outline as part of the template are more (or at least equally) favored than those that do not include them. This “distraction problem” is illustrated in Figure 8, which shows example images where the LOIS lane detection algorithm determines the best hypothesis to be the one that includes the vehicle outline as part of the template. The net result is that although second generation lane detection systems provide a fairly accurate estimate of the vehicle’s offset and perhaps even orientation, relative to the true lane markers, their curvature estimates are not reliable.

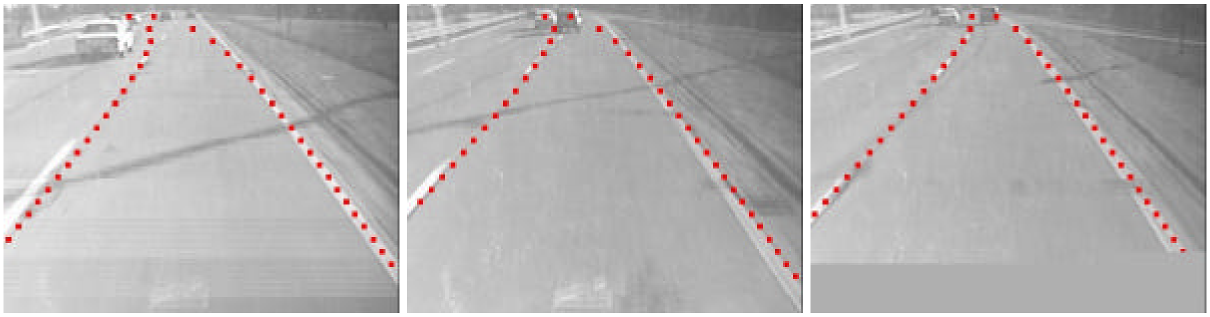


Figure 8. Examples of the true lane shape hypothesis having a likelihood value less than a hypothesis that includes a vehicle.

One way to overcome this problem, as [49][50] point out, is to provide information about obstacles ahead of the vehicle to the lane sensing system to avoid corrupting the gradient field data used to estimate the lane shape parameters. Another way to overcome this problem is to find image features that include the same amount of information about the true lane markers as the image intensity gradient field, but are not as sensitive to extraneous edges. It is believed that the frequency domain based system developed here to estimate lane shape in JPEG compressed images provides features that are less sensitive to extraneous edges than many spatial domain systems–see Figure 9.

[3] A qualification of this observation/claim is due. The lane detection algorithms get pulled away only when the distracting edges are outside of the current lane. When a vehicle is present in the current lane, especially if the vehicle is far away in range and the lane is straight, the vehicle outline helps reinforce the correct lane hypothesis – see [38][47].

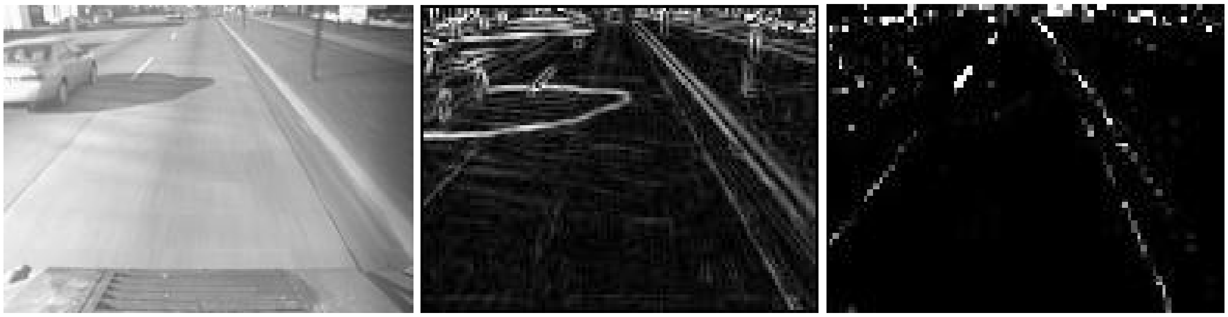
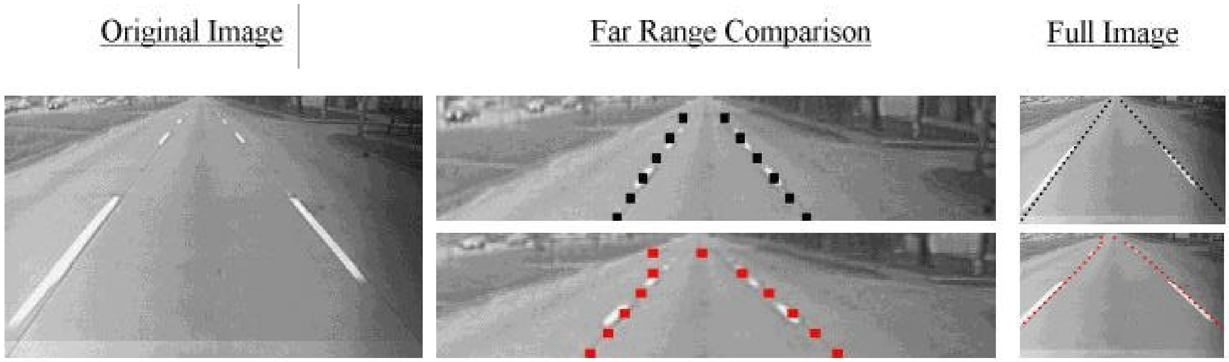
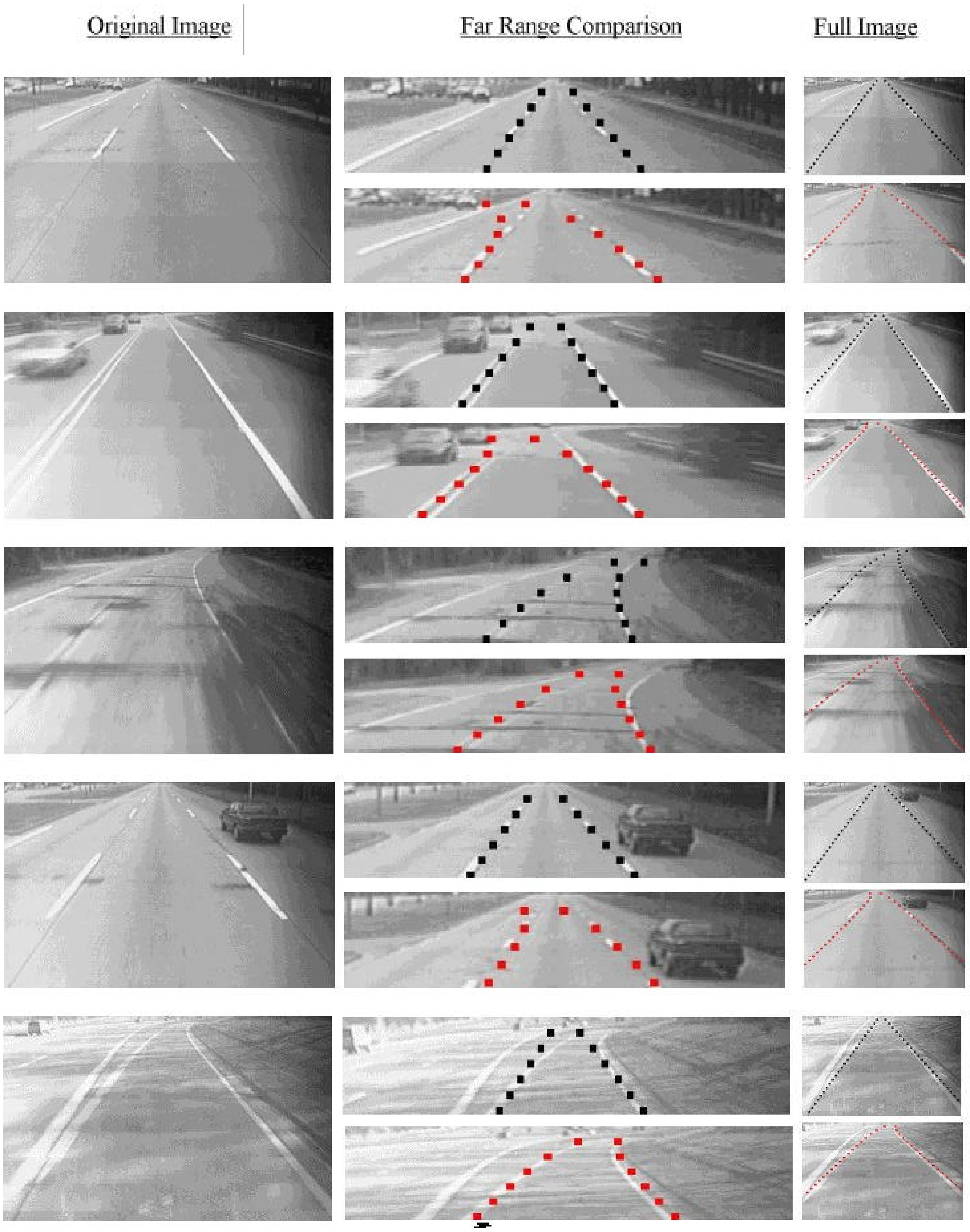


Figure 9. Left: A typical road image. Middle: The image's gradient field. Right: The image's new feature vector field.

Therefore, it is proposed that this frequency domain method be used in lieu of the methods developed earlier for detection of lanes in spatial domain images. To determine the feasibility of using such a frequency domain based technique instead of the traditional spatial domain based techniques, a systematic comparison between the algorithm presented here and the spatial domain feature based LOIS lane detection algorithm [47] was undertaken. This comparison was made from three standpoints: experimental, computational, and methodological.

The experimental comparison seems to indicate that the frequency domain based method has some advantages over LOIS. Especially, our frequency domain method does not seem to be distracted by strong non-lane edges in far range. While this is a desirable characteristic more often than not (see the first 5 rows of images in Figure 10), sometimes it prevents our method from detecting lanes with sharp curvature correctly (see the last 2 rows of images in Figure 10).




Original Image
Far Range Comparison
Full Image

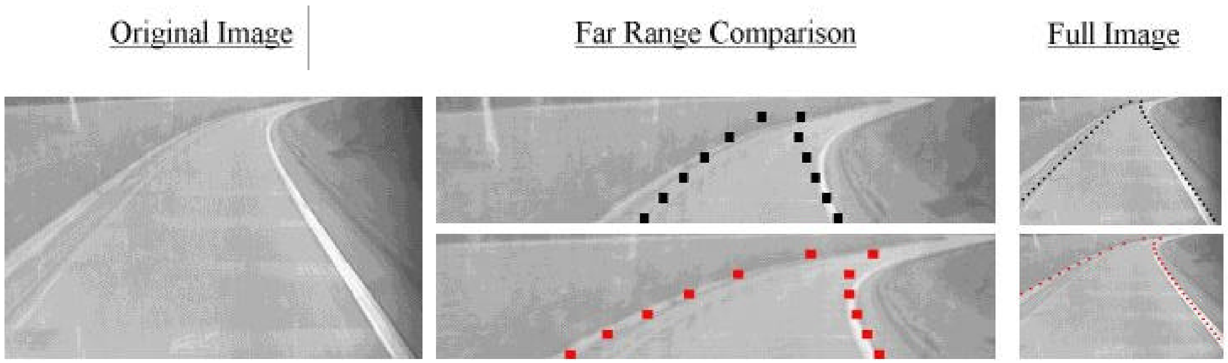


Figure 10. Experimental comparison between our method and LOIS. For each row, the results of our method are on top and the LOIS results are on the bottom of the middle and right columns

A computational comparison of the lane detection method presented in this thesis and the LOIS lane detection algorithm follows. Given a hypothesized lane shape over a 640×480 image, the number of operations involved in the computation of the our objective function and LOIS' are tabulated in Table 2:

Type of Operation	# Operations for our method	# Operations for LOIS
Addition	600	112,000
Multiplication	240	133,000
Division	360	2,400
Math Library Call	0	800
Table Look-Up	0	66,000

Table 2. Comparison of the number of operations for our objective function and LOIS'.

Both our method and the LOIS lane detection algorithm incur a computational overhead every time a new image is processed. For our method this overhead is the DCT computation[4], whereas for LOIS the overhead is the gradient field computation. Table 3 provides a comparison of the number of operations involved in computing each of the overheads.

Type of Operation	Number of Operations for our method	Number of Operations for LOIS
Addition	4,900,000	5,300,000
Multiplication	4,900,000	614,000
Math Library Call	0	614,000

Table 3. Comparison of the number of operations for overhead between the two methods.

[4] It is assumed that if this method were to be used in lieu of spatial domain techniques, then the images in question would not be encoded. Therefore, we must count the encoding process as overhead.

The size of the images in Figure 10 were all 640×480. One drawback of the frequency domain lane detection method presented in this thesis is that by using the DCT of pixel blocks, the image size is effectively reduced by a factor of 8 in each dimension. Therefore, unlike the LOIS lane detection algorithm, our method cannot handle very small images. Our method seems to work well on both 640×480 and 320×240 images, and even on 160×120 images the performance is still acceptable. However, any further reduction in image size causes problems for our method. Contrast this with LOIS, that was shown to perform acceptably even on 30×32 images. While this may be a problem when processing existing databases of small images, the primary motivation for processing small images using LOIS is the corresponding processing speed-up it offers. In the following paragraphs, it will be shown that our frequency domain lane detection method provides this speed-up without the need for sub-sampling.

In order to determine the optimal lane locations, both our frequency domain method and the LOIS lane detection algorithm searched over (the same) approximately 400,000 possible lane shapes, comprised of 9 different curvature values, 50 different orientations, and 30 different locations each for the left and right lanes. On a Pentium-266MHz 96MB-RAM Desktop-PC, for each image, our method took approximately 30 seconds to search over the 400,000 possible lane shapes. In comparison LOIS took approximately 2 hours for the same task. In other words, our method would take the same amount of time to find lanes in a 160×120 image as LOIS would in a 30×32 image.

Finally, our algorithm and the LOIS lane detection algorithm were compared from a methodological standpoint. An assessment was made as to which of them is inherently more reliable for lane detection. Shown in Figure 11 are a set of graphs that represent the cross-sections of the objective functions for our method and LOIS, along $k', b'_{LEFT}, b'_{RIGHT}$, and vp axes[5]:

[5] Both objective functions were evaluated over the same image and the cross-sections were taken about the respective global maxima.

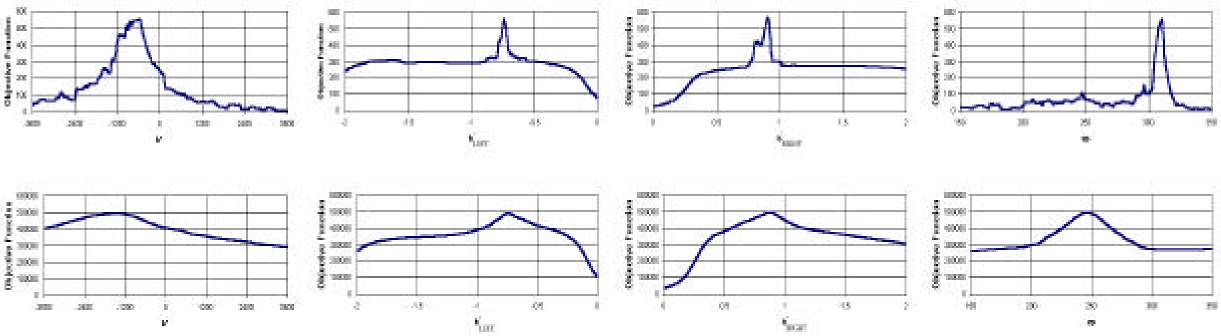


Figure 11. The cross-sections w.r.t. $k', b'_{LEFT}, b'_{RIGHT}$, and vp. Top – for our objective function. Bottom – for the LOIS objective function.

The graphs in the top row of Figure 11 are more uniformly peaked with respect to the floor for all four parameters $k', b'_{LEFT}, b'_{RIGHT}$, and vp than the ones in the bottom row. Hence, between our method and LOIS, our method is expected to discriminate better between the globally "correct" and "incorrect" hypotheses. This was verified by a comparison of the ratio between the global maxima and the average value of the objective function for our method and that of LOIS. For our method, this ratio was found to average around 15, whereas for LOIS, the ratio was around 2.5.

2.7 Summary and Conclusions

This section has introduced a frequency domain method for detecting lane markers in encoded images acquired from a forward-looking vehicle-mounted camera. The method was based on a set of frequency domain features derived directly from the DCT coefficients, so minimal decoding was necessary. In fact, it was even shown that this method of lane detection may be a viable alternative to spatial domain techniques of lane extraction, such as LOIS. In the next section, we will show how the estimate of lane geometry developed in this section allows us to perform query and browsing on JPEG encoded image databases.

CHAPTER 3 – MATCHING SEMANTIC QUERIES TO FEATURES EXTRACTED FROM JPEG ENCODED IMAGES

3.1 BACKGROUND

In this chapter, we present the results of a query-based search on a database of JPEG encoded images using the lane feature extraction technique discussed in chapter 2.[6] For this experiment, a database of forty-eight test images were collected. These images were obtained from a forward-looking vehicle-mounted camera, and were taken in widely varying environmental, lighting, and roadway conditions – including night time, snow covering, dashed and missing lanes, high glare and heavy shadowing. Each of the forty-eight images were encoded using the standard JPEG algorithm, as described in [2]. The original (decoded) images in this database are shown in Figure 12 for reference.

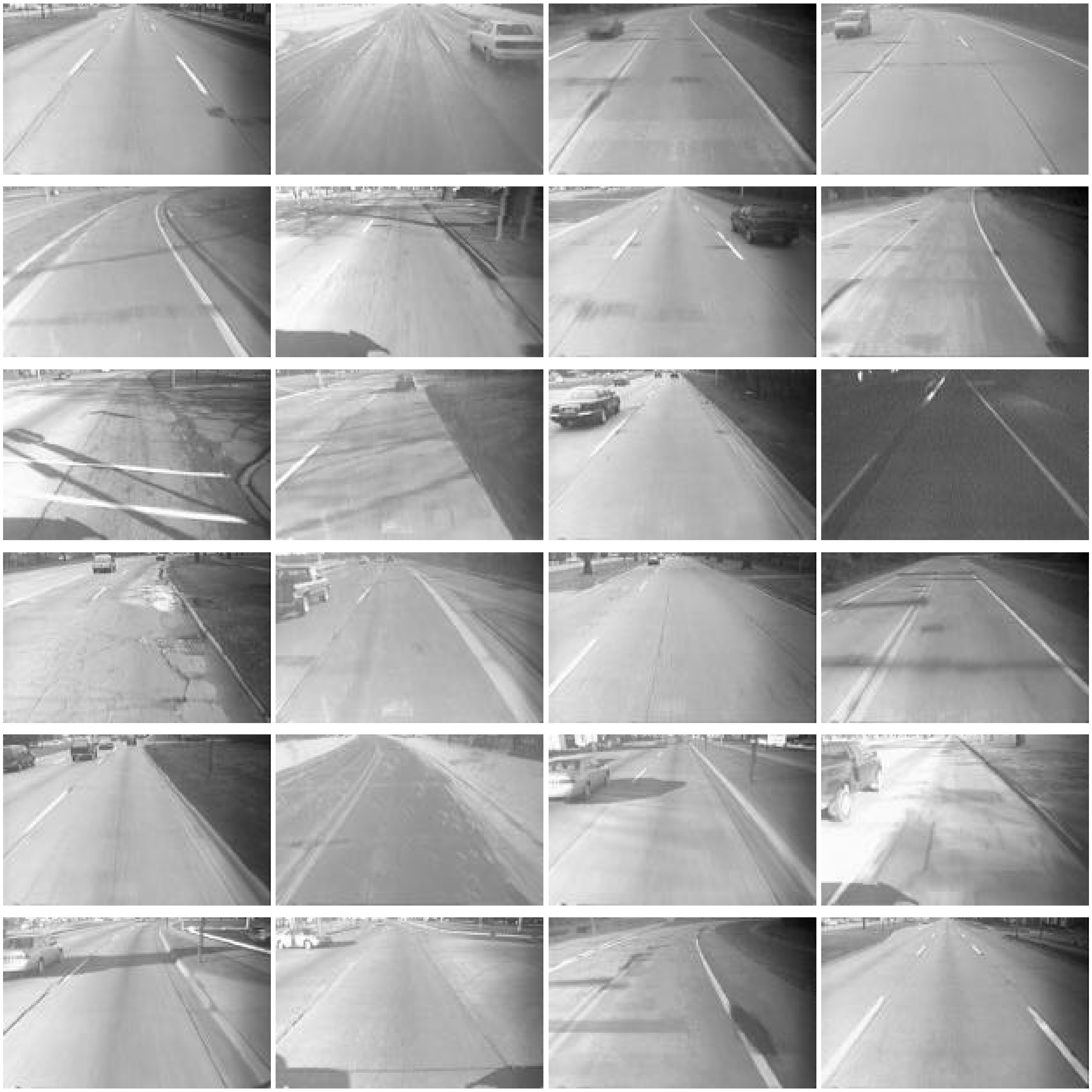
Our method of encoded image query overcomes many of the deficiencies of the previously published methods mentioned earlier. It works directly with images encoded using standard techniques, it accepts queries that have global significance in the target images, and it provides a ranking of all those database images that "match" a given query. A target database image is deemed to "match" a given semantic query if the estimate of the lane shape agrees with the shape dictated by the query. A degree of confidence is also attached to every match. The confidence measure is computed by determining how peaked the global maximum of the posterior probability density function (pdf) is relative to its floor-see section 2.6.

3.2 THE MATCHING METHOD

A series of example queries was generated for the database, such as "find all images with a curved lane," or "find all images with a narrow lane," etc. A typical query reflects certain unique properties regarding the lane geometry of interest. Every such query is translated into corresponding bounds on the values of the lane shape parameters k', b'_{LEFT}, b'_{RIGHT}, and vp. Given a query, the database of images is first pruned by determining whether or not the MAP estimate (as discussed in section 2.4) of the lane geometry for the image under consideration falls

[6] Note that JPEG does not store the actual DCT coefficients, but only a quantized version of them using a standard table. Therefore, the method of section 2 is applied to the quantized coefficients. Our frequency domain method for lane detection seems to be insensitive to the effects of this standard JPEG quantization.

within the corresponding parameter space bounds. For every image in the pruned database, a confidence ratio is computed between the value of the objective function at the MAP estimate and an estimate of the average value of the objective function everywhere else in the four-dimensional lane shape parameter space. The images in the pruned database are then sorted according to the magnitude of the above-mentioned ratio, in addition to how well the corresponding lane geometry estimates match the query.



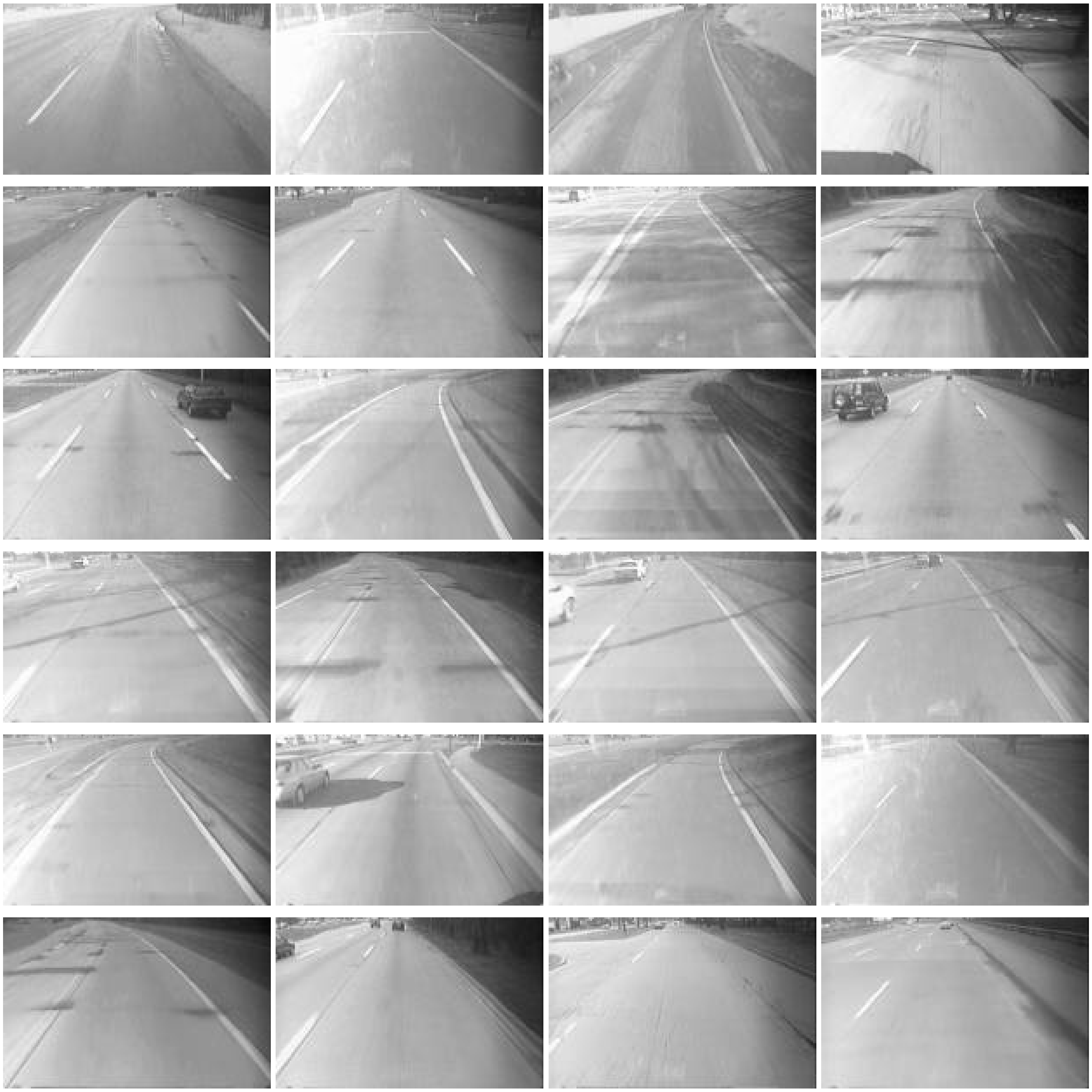
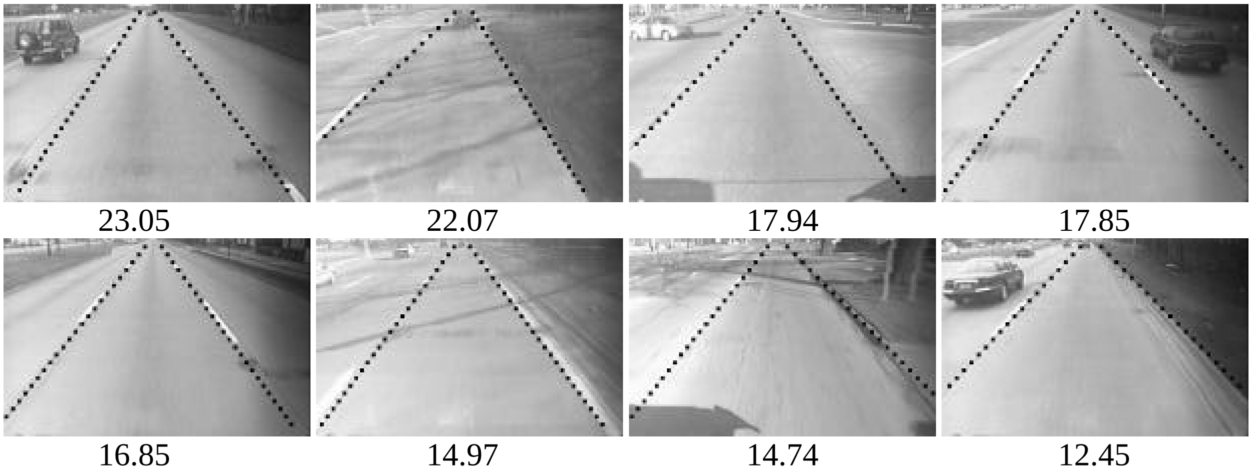


Figure 12. The database of 48 images.

The first set of example queries was performed to find those JPEG encoded images in the database of forty-eight that contain straight lanes. This query is translated into finding all images in the database characterized by $k' = 0$. Figure 13 presents the top eight straight road locations, with the lanes imposed on the image, ranked by the confidence ratio (shown below each image) alone.

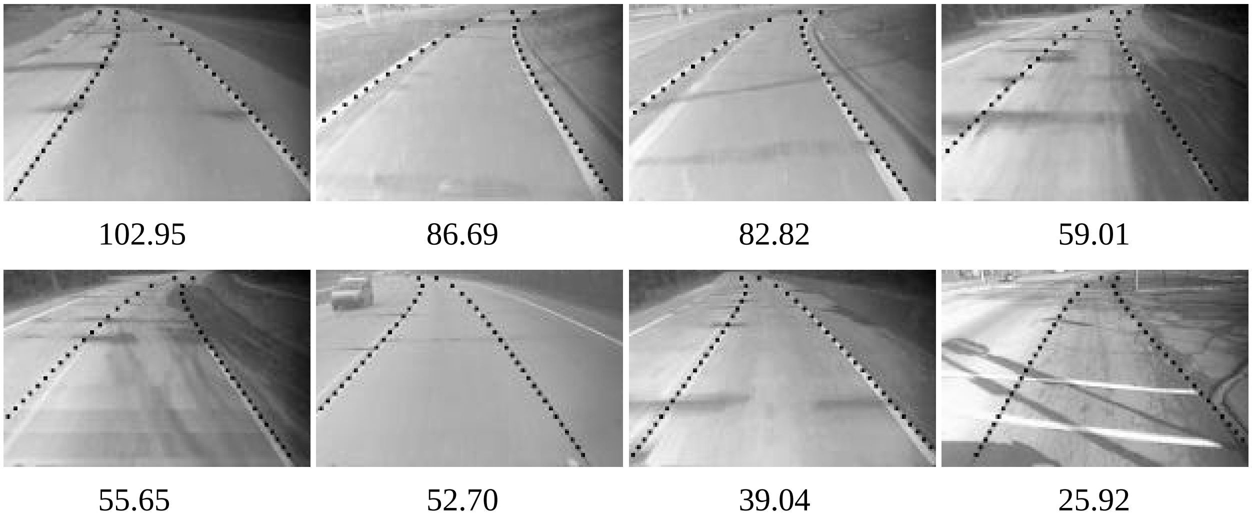


23.05 22.07 17.94 17.85

16.85 14.97 14.74 12.45

Figure 13. Top 8 responses to the query “find all images with a straight lane.”

A query “find all images with a curved lane” was presented to the database next, which was translated into searching for those images characterized by $k' \neq 0$. These results were sorted by multiplying the curvature parameter k' and the confidence ratio. The results are presented as Figure 14.



102.95 86.69 82.82 59.01

55.65 52.70 39.04 25.92

Figure 14. Top 8 responses to the query “find all images with a curved lane.”

Another possible query is “find all images with wide lanes.” This query was translated into the condition $(b'_{RIGHT} - b'_{LEFT}) > 1.9$, which equates to the lanes being more than 3.8 meters wide in the ground plane. A similar sorting method to that of curved roads was employed, and the pruned database images are ranked according to the value of the lane width multiplied by the value of the confidence ratio. The results of this query are shown below as Figure 15.

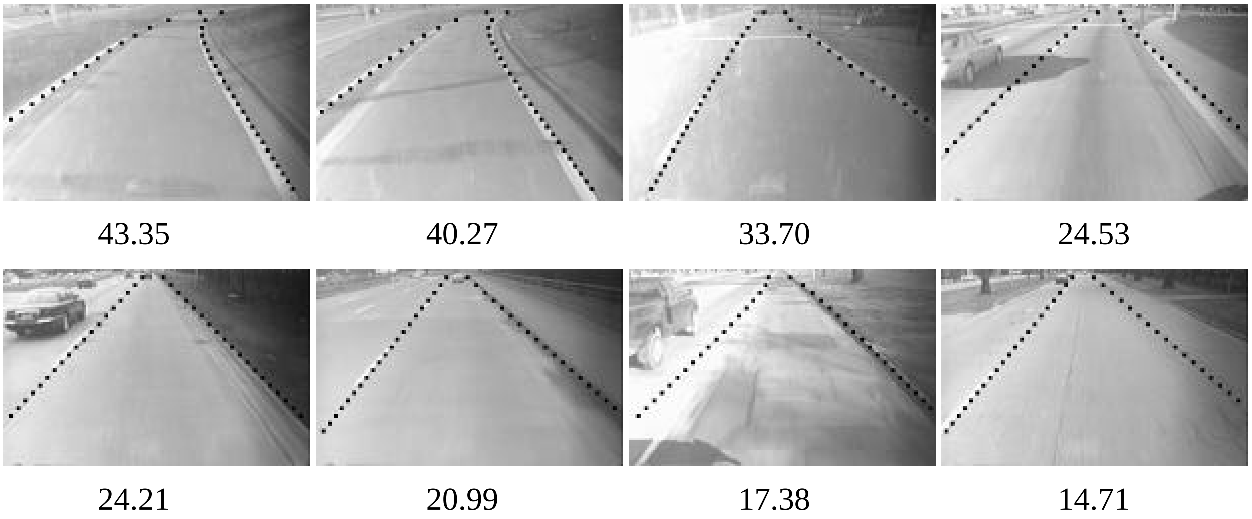


Figure 15. Top 8 responses to the query "find all images with a wide lane."

This query was followed by "find all images with a narrow lane." The corresponding constraint on the lane shape parameters k', b'_{LEFT}, b'_{RIGHT}, and vp was chosen to be $(b'_{RIGHT} - b'_{LEFT}) < 1.5$. In this case, the constraint corresponds to the lane being less than 3.0 meters wide in the ground plane. The pruned database images are sorted according to the value of the confidence ratio divided by the value of the lane width $(b'_{RIGHT} - b'_{LEFT})$. The results of this query are shown as Figure 16.

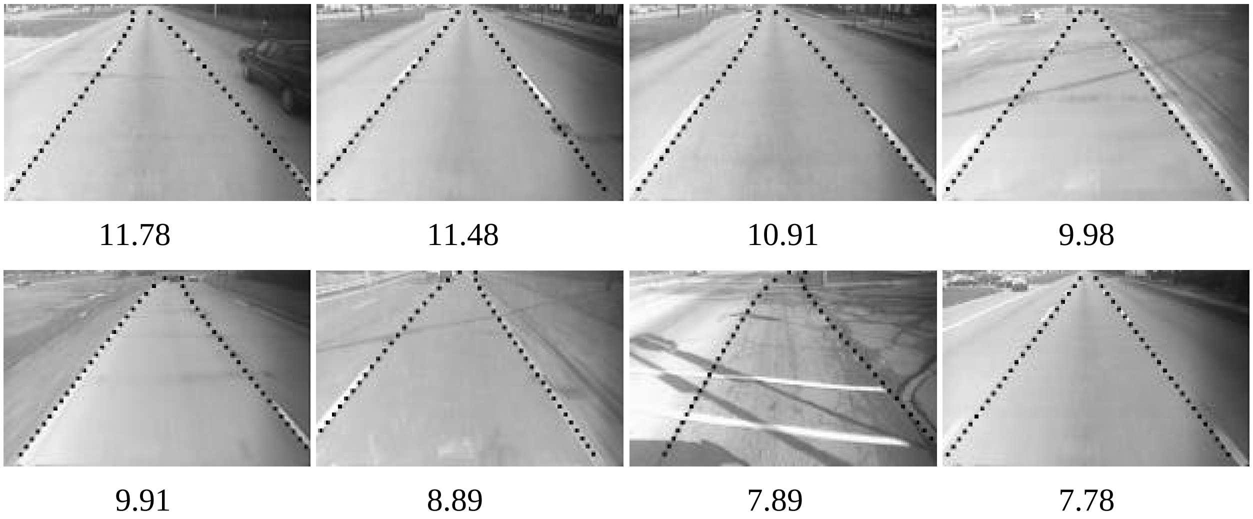


Figure 16. Top 8 responses to the query "find all images with a narrow lane."

Note that the technique for lane shape estimation from JPEG encoded images does not seem to be affected by the amount of clutter in the image being queried.

3.3 SUMMARY AND CONCLUSIONS

In this chapter, we have introduced a method for querying a database of JPEG encoded images that consist of common roadway scenes. This method proceeds as follows:

- Using the method described in section 2, an estimate of lane shape in each image in the database is generated.
- Images whose estimated lane geometry agree with that dictated by the query are deemed to match the query.
- Finally, a confidence ratio is used to rank the various database images that match a given semantic query.

In the next chapter, we will extend this technique to allow query, browsing, and analysis on MPEG encoded video sequences. Since MPEG has at its heart a JPEG-like spatial encoding/compression algorithm, we adapt/extend the algorithm developed in this chapter to answer temporal-event based queries such as “find portions of the video where the vehicle makes a lane change.”

CHAPTER 4 – FEATURE TRACKING AND TEMPORAL EVENT QUERY/BROWSING FOR MPEG ENCODED VIDEO SEQUENCES

4.1 BACKGROUND

This chapter presents the details of the third and final constituent of the MPEG encoded video query tool developed in this thesis. Specifically, it describes the tracking of lane features through MPEG encoded sequences, and interprets the lane geometry's evolvement from frame-to-frame in the context of temporal-event based queries.

As is explained in [1] and Appendix B, the MPEG standard has three different types of frames that are used in sequence to create an encoded video. The 'I-', or inter-coded frames, are encoded/compressed spatially only, like JPEG. MPEG also uses inter-coded (P- and B-) frames to substantially improve the compression ratio. P-frames are coded with respect to previous frames to reduce temporal redundancy. Similarly, B-frames are utilized to reduce temporal redundancy. B-frames are coded both with respect to both past frames and future frames. The coding in P- and B-frames is done by encoding information regarding the motion of macroblocks in a frame (motion vectors) and the difference in DCT coefficients between motion-related macroblocks (error vectors). This technique is useful in efficient encoding of image sequences, because the scene changes little from frame-to-frame. Therefore, to decode already encoded MPEG video sequences, an algorithm that deals with these three frame types differently is necessary.

- The shape of the lane/pavement markers present in the I-frames is estimated using the method developed in sections 2 and 3.
- For P-frames, a technique that tracks lane movement through analysis of the motion vectors is combined with a local maximization of a posterior pdf is developed.
- For B-frames, an out-of-order processing of the frame sequence is done, so that future frames are processed first. This allows a conceptually similar process to be done with B-frames as is done with P-frames.

It will be shown that by taking advantage of the motion information present in P- and B-frames, the method developed here estimates the lane shape more quickly and efficiently in the P- and B-frames than it does in the I-frames. The exact procedure is described in detail for P-frames alone first, followed by the modifications needed for lane shape estimation in B-frames.

A brief introduction to how P- and B-frames differ from I-frames is appropriate here (see Appendix B for a more detailed explanation). For P-frames, MPEG breaks the frame into 16×16 macroblocks and encodes the DCT of these blocks in terms of a motion vector and an error vector. The motion vector contains information indicating what past macroblock that new macroblock is being encoded with respect to, i.e., where this new macroblock came from. The error vector, on the other hand, represents the change in the DCT coefficient values for the new block relative to the old block. This differs from the I-frame scenario in that the macroblocks are encoded with respect to previous macroblocks instead of just spatially. In cases where encoding in this fashion would require more bits than just a standard (I-frame type) encoding of this new block, the new block is intra-coded just like an I-frame block, and no motion or error vectors are encoded.

Lane shape estimation in P-frames is a two-step process:

- First, the P-frame motion vectors are used to approximate the motion of the lane from frame-to-frame, without any regard of the error vector. Using this approximate lane motion, a non-linear least squares fit is performed to make an initial guess as to what the exact shape of the lanes is in this new P-frame. This procedure is explained in section 4.2.
- Second, a local maximization of the P-frame posterior pdf (derived using the error vectors) is performed to refine the initial guess, as explained in section 4.3.

The extension of this method to B-frames is given in section 4.4. As a result, lane features can be tracked robustly through an MPEG encoded video sequence. Section 4.5 provides an illustration of this entire feature tracking procedure. Finally, section 4.6 provides some example queries on MPEG encoded video sequences along with the corresponding answers to those queries.

4.2 Initial Guess Of The Lane's Shape Using Motion Vectors

Each 16×16 macroblock of the P-frame has a motion vector indicating its origin in the preceding I- or P-frame. In general, the standard MPEG codec does not encode P-frames with respect to the frame immediately preceding it. A typical P-frame is encoded with respect to a frame several instances back, as B-frames are usually interspersed between successive P- and I-frames.[7] Since that previous frame has already been processed, the shape of the lanes in that frame is known. If a P-frame macroblock is deemed to have originated from a lane containing macroblock in the past P- or I-frame, that new P-frame block is so marked. A distinction is made between those blocks that came from left lanes and those that came from right lanes. By inspecting all macroblocks in the current P-frame, a tertiary map indicating how the previously estimated lane has moved from the past frame to the current frame is generated. This map is then used to make an initial guess as to what the shape of the lanes is in the current P-frame, as explained in the next subsection.

Using the tertiary map as data samples of the non-linear regression given by eq. (5), a least-squares estimate of the regression parameters $k', b'_{LEFT}, b'_{RIGHT}$, and vp is obtained from that data. This non-linear least squares estimate is the system's initial guess as to what the lane's shape is in this new P-frame.

4.3 Lane Shape Refinement

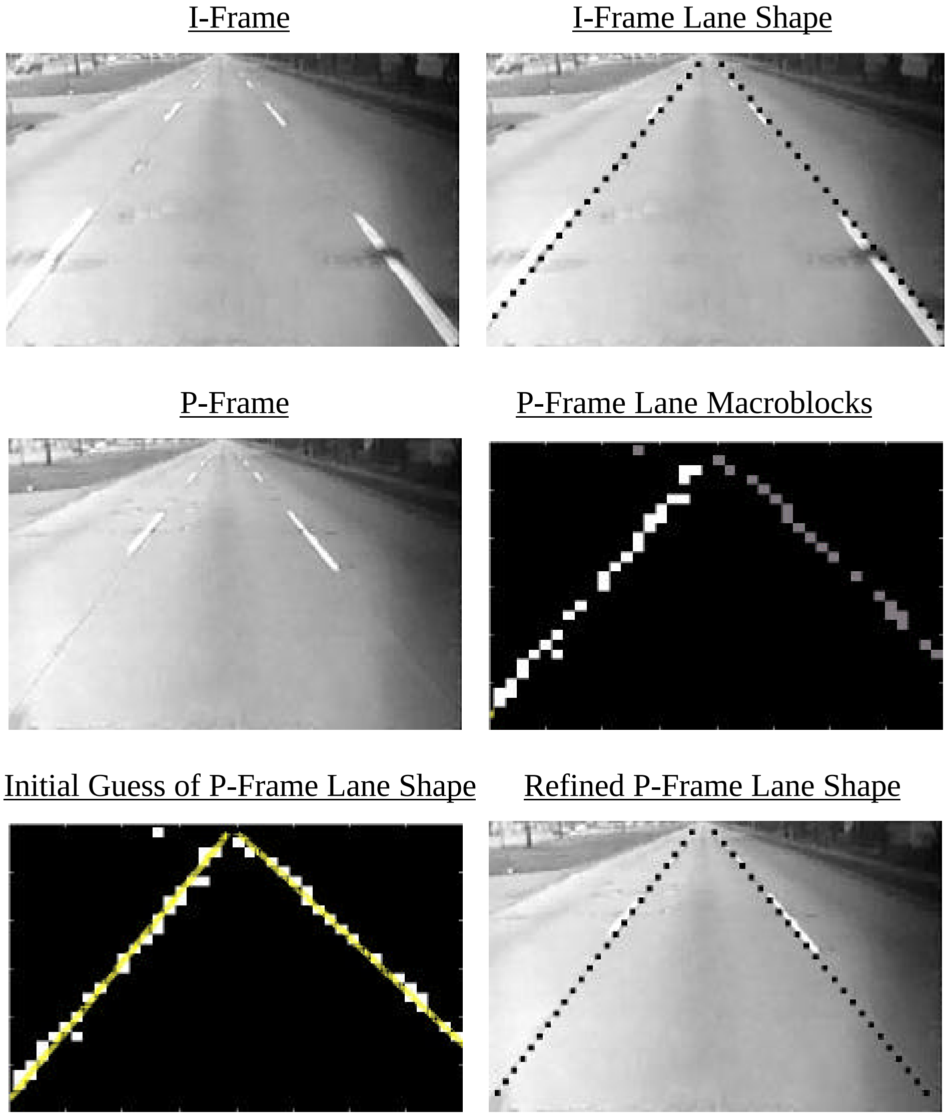
There are two important reasons why this easy to calculate initial estimate is not sufficiently accurate and requires a proper refinement before it can be accepted as the final lane shape estimate for the new P-frame:

- First, since the initial estimate is based on motion vectors alone, those blocks that are inter-coded (have no associated motion vector) are ignored. This includes blocks that correspond to areas of the current P-frame that have newly appeared, and are not present in the previous image. Also included are macroblocks that are significantly different than the past due to lighting, shadowing, occlusion etc.
- Second, some present macroblocks that are encoded with respect to past lane containing macroblocks do not actually contain lanes. Typically, these are blocks that have lane-like properties in terms of the strength and orientation of the edges they contain.

Using the method discussed in section 2.2, a feature image for the new P-frame is computed. A posterior pdf over the lane shape parameter space is then formulated for this new frame as in section 2.4. The initial guess from the previous subsection is used to initiate a simple

[7] Many popular implementations of the MPEG standard do not use B-frames at all, as they are considered non-normative. We note that this system works independent of such variations in implementing the non-normative portions of the MPEG standard. See [1] for more details regarding the normative and non-normative portions of the MPEG standard.

locally exhaustive search of the posterior pdf over a very small area of the lane shape parameter space. This local search usually results in a good refinement of the initial guess, and is accepted as the final estimate of the lane's shape in the new P-frame. An illustration the entire process described in the preceding subsections is shown in Figure 17.



I-Frame

I-Frame Lane Shape

P-Frame

P-Frame Lane Macroblocks

Initial Guess of P-Frame Lane Shape

Refined P-Frame Lane Shape

Figure 17. An example of lane shape estimation in P-frames.

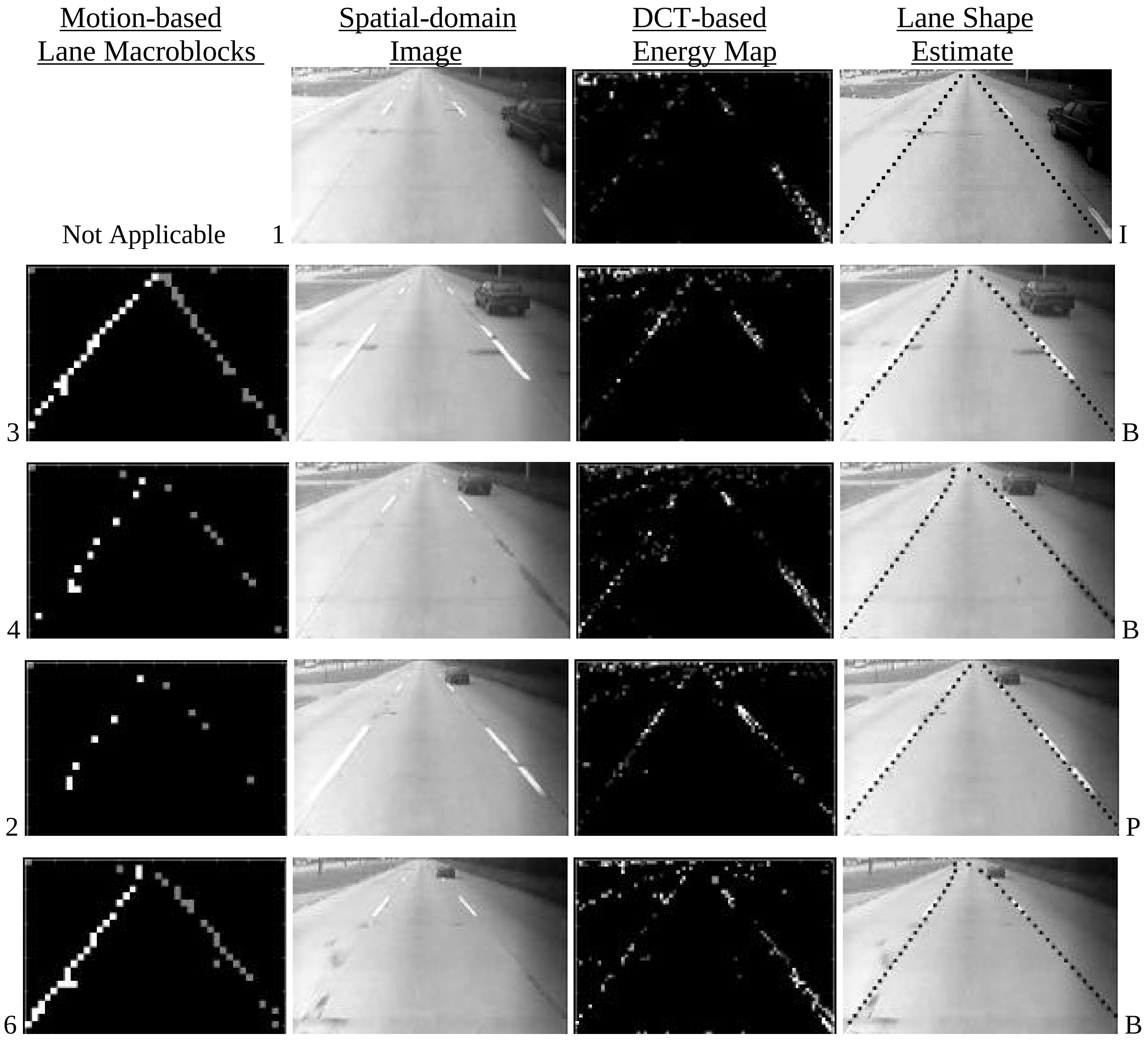
4.4 Extension To B-Frames

The extension of this procedure to lane shape estimation in B-frames is conceptually simple. Unlike P-frames that encode current frames with respect to preceding frames only, the encoder allows B-frames to use information from both past and future frames. Hence, the motion vectors in the new frame will refer to both past and previous frames.

For this reason, lane shape estimation in B-frames cannot exactly mimic that of P-frames. Lane shape estimates of both past and future frames are needed to use the same procedure as in the previous three subsections. This is not a major difficulty, except that the B-frame lane shape estimation has to be done out of order, i.e., only after the lane shape estimation procedure for a future I- or P-frame has been completed.

4.5 Example Sequence

The lane shape estimation processes of the preceding sub-sections and how they all fit together is illustrated in Figure 18 on a 10 frame MPEG encoded video stream.



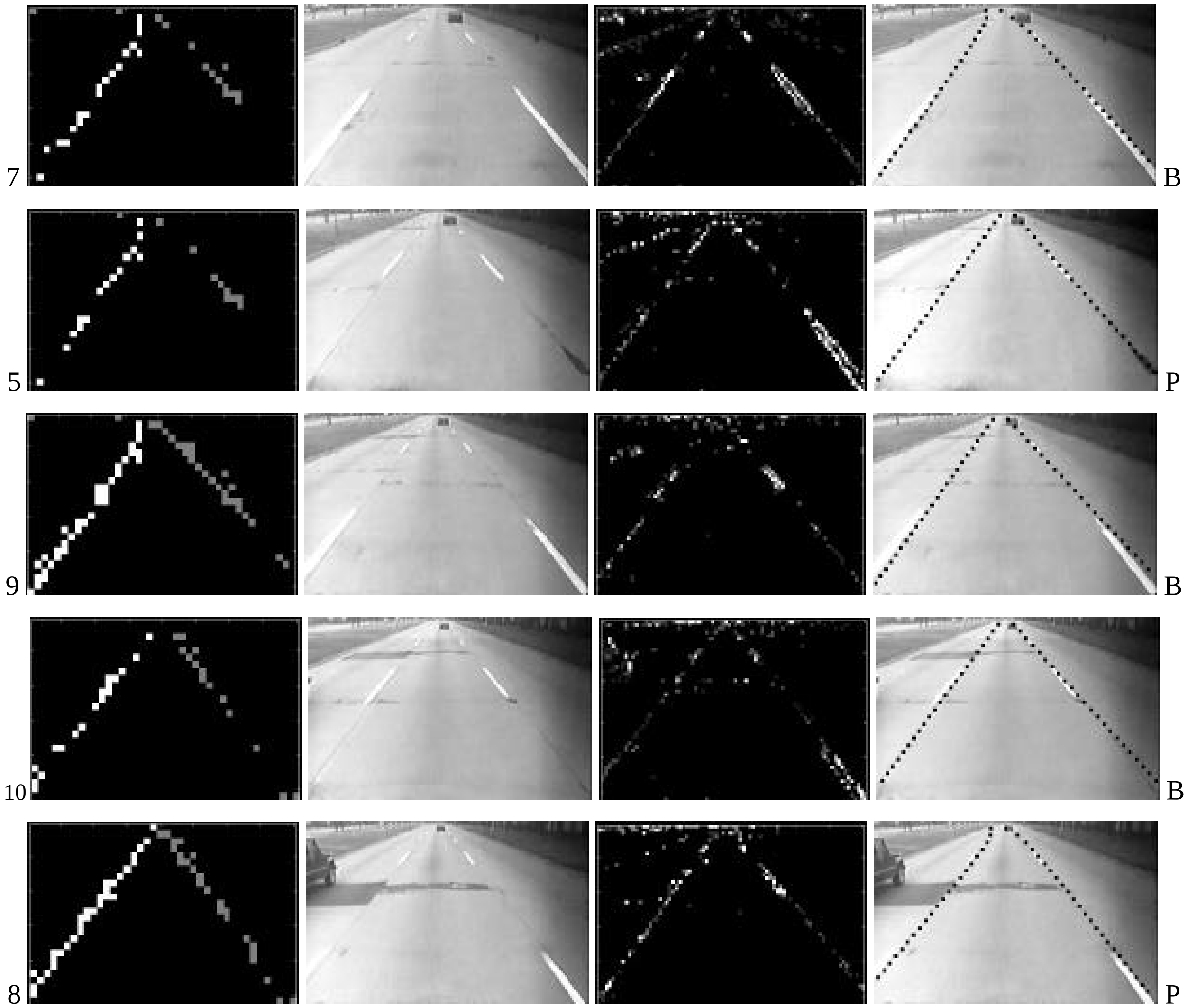


Figure 18. Lane shape estimation in an MPEG encoded video stream. In each row, the number on the extreme left indicates processing order, and the letter on the extreme right indicates frame type.

4.6 RESULTS OF MPEG ENCODED VIDEO QUERY

The lane shape estimation process is subsequently used to query MPEG encoded videos of common roadway scenes. The results of a series such queries, including those that locate temporally meaningful events and those that locate geometrically meaningful events are presented in Figures 19 through 21. The query in Figures 19 and 21, namely, “find portions of the video where the vehicle is making a lane change maneuver,” is translated to mean find portions of the video where either of the two lane offset parameters b'_{LEFT} or b'_{RIGHT} undergoes a smooth zero crossing [51]. The query in Figure 20, namely, “find portions of the video where the vehicle is going around a tight curve,” is translated to mean find portions of the video where the curvature parameter k' continues to have a magnitude much larger than zero over a series of frames.

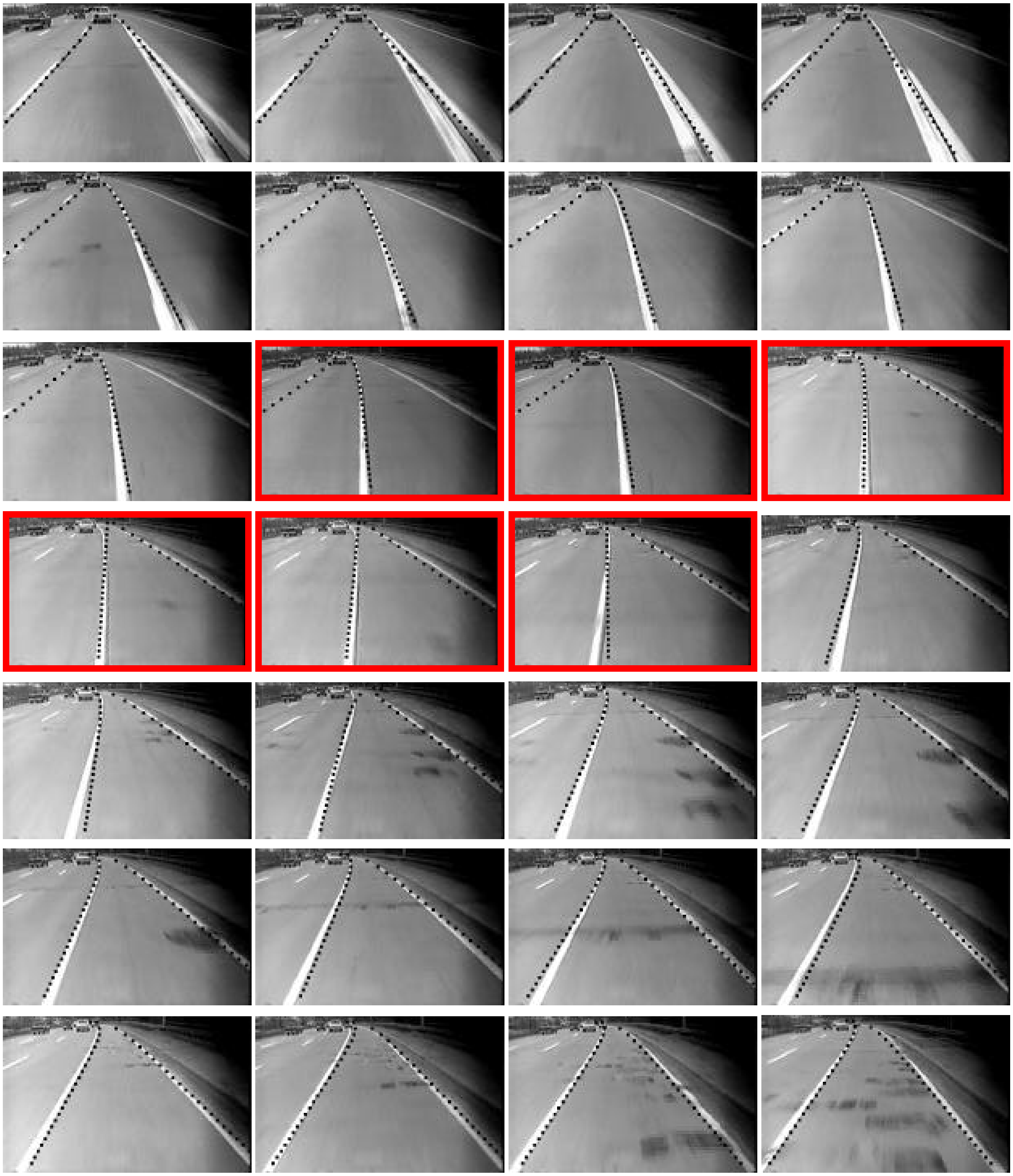


Figure 19. Results of querying the MPEG stream with “find portions of the video where the vehicle is making a lane change maneuver.” Retrieved images are highlighted in red.

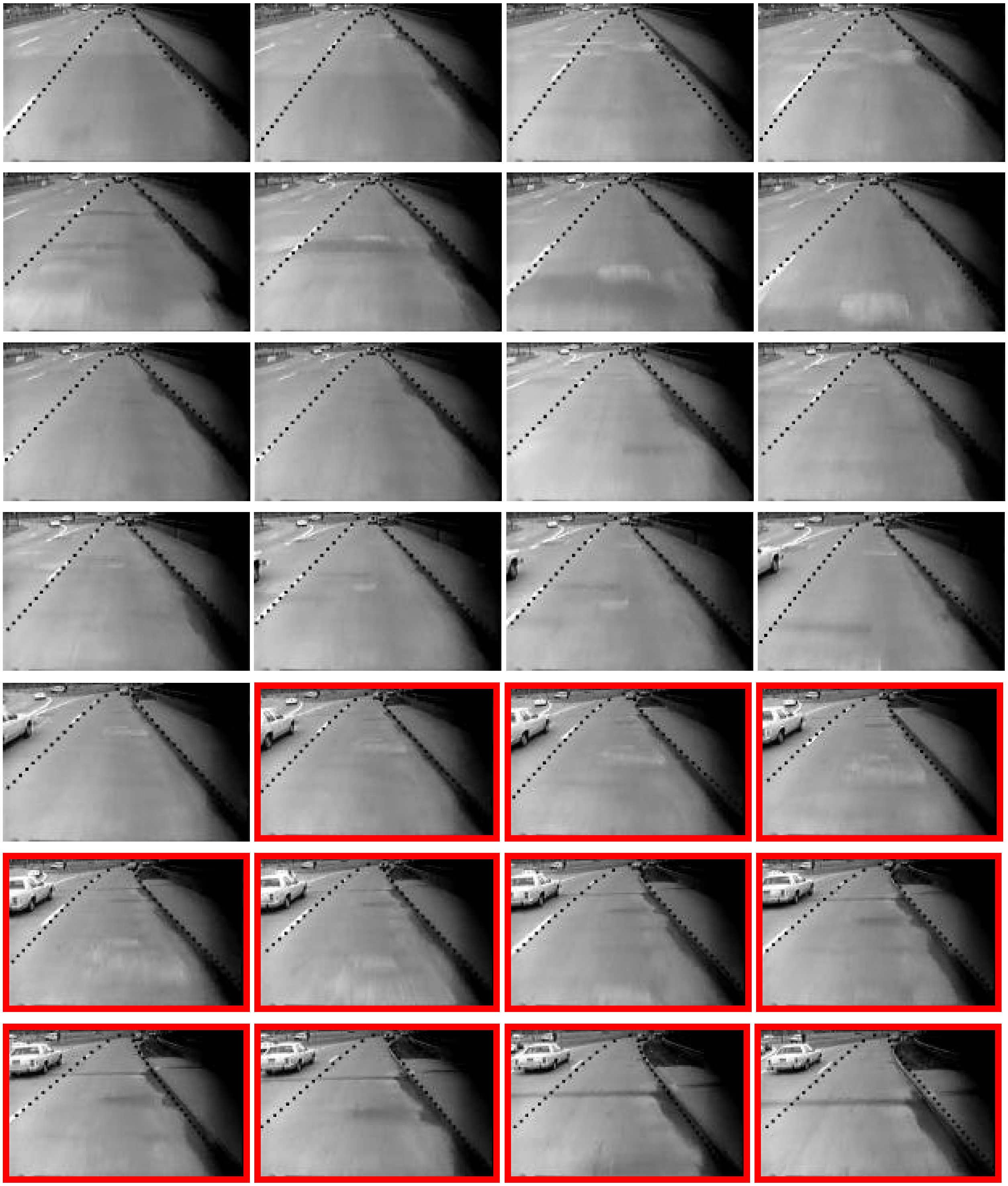


Figure 20. Results of querying the MPEG stream with “find portions of the video where the vehicle is going around a tight curve.” Retrieved images are highlighted in red.

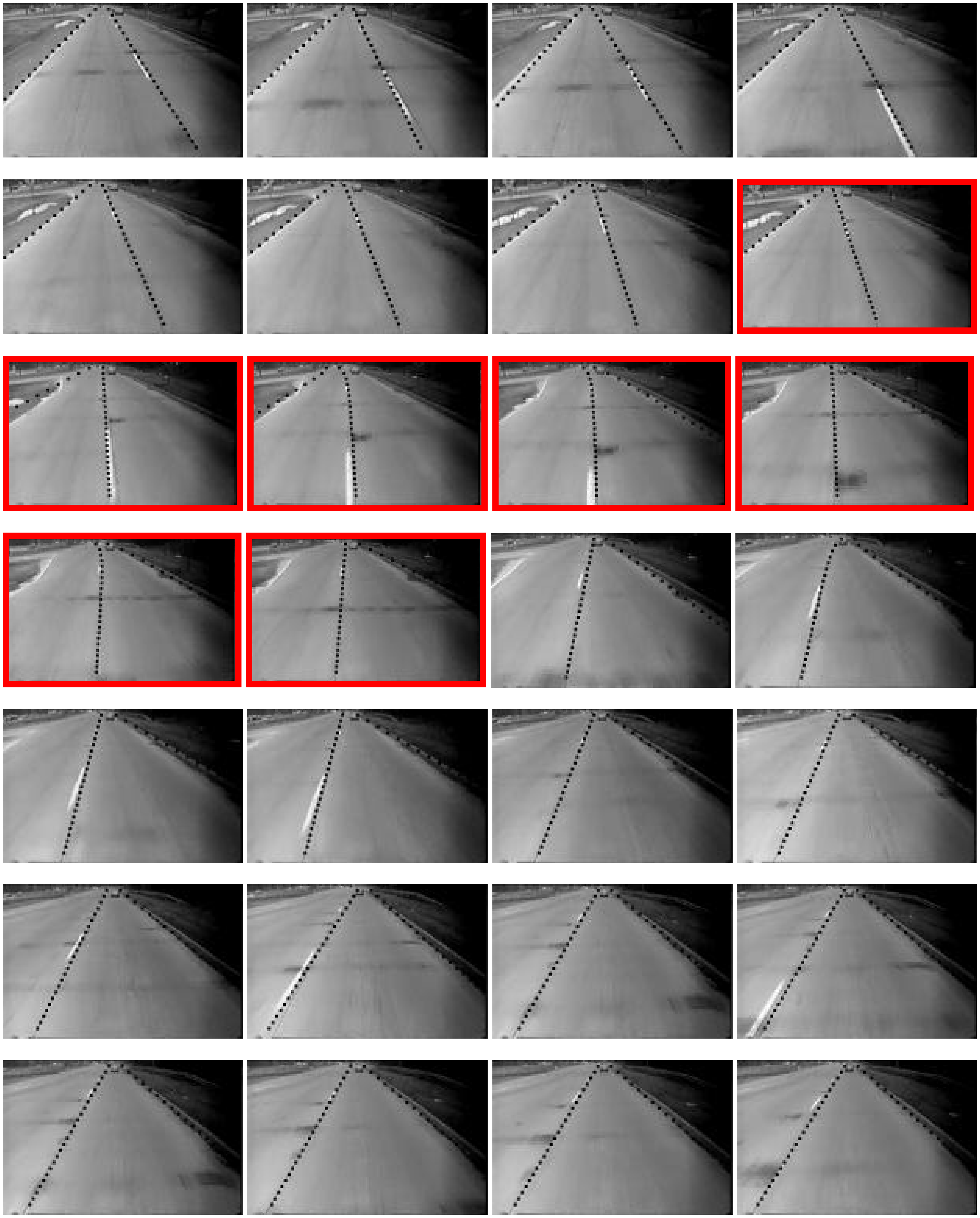


Figure 21. A second example of querying an MPEG stream with "find portions of the video where the vehicle is performing a lane change." Retrieved images are highlighted in red.

4.7 SUMMARY AND CONCLUSIONS

This chapter has presented the details of the third and final component of the MPEG encoded video query tool developed in this thesis. The query tool makes use our method for detecting lanes in the frequency domain, as well as a new technique that develops a motion based estimate of lane shape. By combining the two, a tool that is able to robustly track lane features as they evolve through a sequence has been demonstrated, and applied to the task of allowing semantic query of encoded video sequences. It has been shown that such a query tool is useful for extracting both geometrically and temporally meaningful segments from an MPEG encoded video without performing any inefficient decoding.

5 CONCLUDING REMARKS

This thesis has presented a tool for shape-based feature extraction, query, and browsing of MPEG encoded video sequences without the need for any decoding. Since the decoding stage is not necessary, feature extraction becomes much more computationally efficient than otherwise. The video sequences used for this study consist of common roadway scenes, as imaged by a forward-looking vehicle-mounted camera. This tool has been employed to automatically query and browse MPEG encoded sequences to extract portions of the sequence that are meaningful, based on both geometric and temporal constraints on the lane's shape.

The tool uses a three-step method.

- For inter-coded (I) frames, the shape of the lane/pavement markers present is estimated in a Bayesian setting using a set of DCT-based lane edge features, a global shape model for lane edges, and a coarse-to-fine optimization algorithm.
- The estimation of lane shape in P-frames uses the motion of macroblocks between frames and the estimate of lane shape in the previous I- or P-frame to generate and initial guess of the lane's new shape. This initial guess is refined by a local search in the lane shape parameter space. The lane shape estimation procedure for B-frames is very similar to that for P-frames.
- Finally, the lane’s geometry as it evolves over time is analyzed in order to establish a connection between its signature and the one mandated by the query. This includes both temporally significant queries such as “identify portions of the MPEG encoded video where the vehicle is making a lane change maneuver,” and geometrically significant events such as “identify portions of the video where the vehicle is going around a tight curve.”

Several experimental results have been presented to illustrate the efficacy of this tool.

There are many possible avenues of future work in the area of encoded video query, browsing, and analysis. Two of the most important are those that are concerned with making the features that are extracted more general, and those that are concerned with broadening the allowable set of semantic queries. An augmentation of the tool presented here to allow tracking and query of more generic shapes, such as those in [6][25], would make a more powerful system. An extension of the query interpretation capabilities of the tool presented in this thesis would enable it to be used more widely.

APPENDIX A – A SUMMARY OF THE JPEG IMAGE COMPRESSION STANDARD

The Joint Photographic Experts Group (JPEG) standard is an image encoding or compression algorithm, developed through collaboration of an international body of scientists - see [2]. It is "lossy", meaning that the decompressed image is not identical to the original image. JPEG is designed to exploit known limitations in the human eye, notably that high frequency components of images are much less noticeable than low frequency components. Therefore, the data that is lost during compression is perceptually unimportant to the human viewing the image and hence can be discarded without negative viewing effects. However, for automatic image recognition, this discarded information is often important. This is one of the main reasons that feature extraction in the compressed domain is a difficult task – important information from a automatic recognition point of view is lost in the compression process.

We will now briefly summarize the process by which an image is encoded in the JPEG standard. The color space used by JPEG is a luminance space for grayscale data and a luminance and two chrominance channels for color data (YCbCr). The details for a grayscale (256 level) image are given, since this work makes use of only luminance data. The extension to color images is straightforward.[8]

For a spatial domain image to be encoded using the JPEG standard, the pixels are first grouped into 8×8 blocks, as shown in Figure 21.

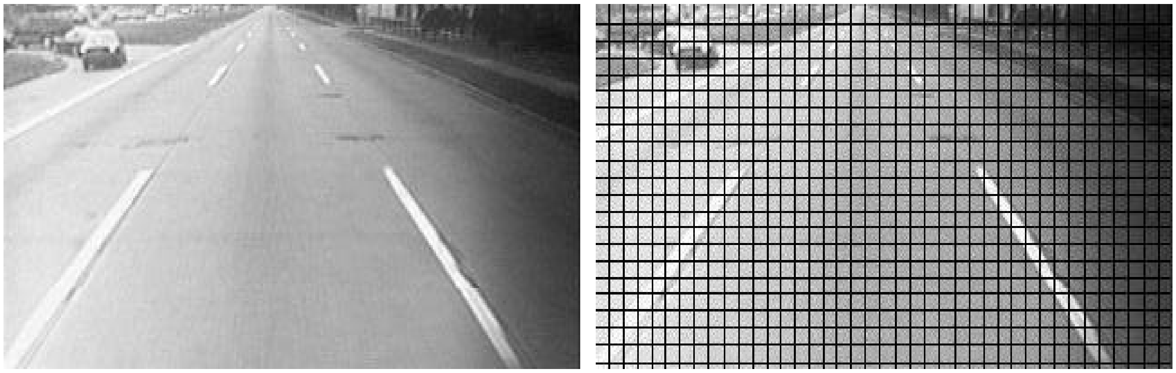
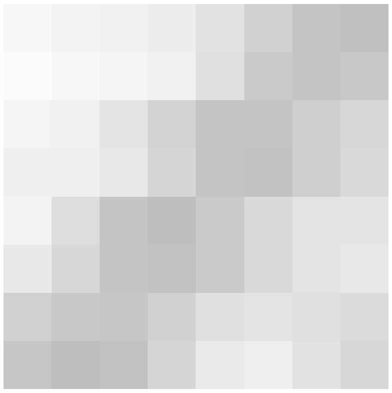


Figure 21. Left: Spatial-domain image. Right: The grid of 8×8 pixel blocks that JPEG uses.

[8] One clarification about this ‘straightforward’ extension: JPEG also makes use of the fact that humans perceive color changes much less distinctly than intensity changes. Therefore, the chrominance channels in a JPEG image are subsampled to reduce the number of bits necessary to encode the image.

Each 8×8 block is treated as just a matrix of 8×8 integers (see Figure 22) that are encoded separately.



248	252	244	239	243	233	209	198
243	247	240	239	221	216	201	190
240	245	228	231	197	197	199	193
237	240	210	213	190	193	209	213
226	223	197	196	202	203	222	234
208	202	197	193	218	217	227	238
195	195	207	206	227	227	223	226
192	200	216	218	228	232	219	215

Figure 22. Left: An 8×8 block taken from the spatial domain image in Figure 21.[9] Right: The 8×8 matrix of pixel intensity values that correspond to that block.

The second step of the JPEG encoding/compression algorithm involves performing the two-dimensional (2-D) Discrete Cosine Transform (DCT) on each 8×8 block–see eq. 9. The 2-D DCT is an orthogonal transform that is widely used in many image compression techniques since its energy compaction ability [52] approaches that of the optimum Karhunen-Loeve (KL) transform while maintaining a significantly lower implementational complexity.

$$S(v,u) = \frac{C(v)}{2}\frac{C(u)}{2}\sum_{y=0}^{7}\sum_{x=0}^{7} s(y,x) Cos[(2x+1)up\ /16] Cos[(2y+1)vp\ /16] \qquad (9)$$

This second step results in a set of 64 frequency domain coefficients in place of the 64 spatial domain coefficients–see Figure 23. The benefit to this change of domain is that upwards of 90% of the energy in the original 8×8 spatial domain pixel block is stored in just 3 or 4 of the lowest frequency coefficients, resulting in a very high energy compaction in the frequency domain.

248	252	244	239	243	233	209	198
243	247	240	239	221	216	201	190
240	245	228	231	197	197	199	193
237	240	210	213	190	193	209	213
226	223	197	196	202	203	222	234
208	202	197	193	218	217	227	238
195	195	207	206	227	227	223	226
192	200	216	218	228	232	219	215

1741	31	10	4	-2	-6	-7	-5
42	99	-5	-3	-4	6	-8	-13
34	-3	-60	12	-7	-4	15	6
13	-33	10	15	6	-13	3	13
8	2	8	2	-1	-3	-4	-2
0	0	1	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0

Figure 23. Left: The 8×8 pixel intensity values. Right: The corresponding DCT coefficient values.

[9] Note, the 8×8 pixel block chosen contains a lane edge.

As the third step, each of the 8×8 DCT coefficients are divided by a quantization factor and the result is rounded the result to the nearest integer, as shown in Figure 24. The quantization factor depends on which coefficient is being considered. Since human perception is insensitive to high frequency coefficients, high frequency coefficients are very coarsely quantized. This step is where most of the compression (and most of the lossiness) comes in. Likewise, since human perception is very sensitive to low frequency coefficients, a much finer quantization is undertaken with those coefficients.

1741	31	10	4	-2	-6	-7	-5
42	99	-5	-3	-4	6	-8	-13
34	-3	-60	12	-7	-4	15	6
13	-33	10	15	6	-13	3	13
8	2	8	2	-1	-3	-4	-2
0	0	1	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0

16	11	10	16	24	40	51	61
12	12	14	19	26	58	60	55
14	13	16	24	40	57	69	56
14	17	22	29	51	87	80	62
18	22	37	56	68	109	103	77
24	35	55	64	81	104	113	92
49	64	78	87	103	121	120	101
72	92	95	98	112	100	103	99

109	3	1	0	0	0	0	0
3	8	0	0	0	0	0	0
2	0	-4	0	0	0	0	0
1	-2	0	1	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0

Figure 24. Left: The 8×8 DCT coefficients. Center: The standard JPEG quantization factor matrix. Right: The quantized DCT coefficients.

The quantized set of coefficients are then encoded using either Huffman or arithmetic coding. This is a lossless step, but does add some additional compression to the data. In this step, the 64 coefficients are reordered in zigzag fashion. The zigzag method was chosen statistically so as to produce long runs of zeros at the end of the stream of coefficients to encode, see Figure 25.

109	3	1	0	0	0	0	0
3	8	0	0	0	0	0	0
2	0	-4	0	0	0	0	0
1	-2	0	1	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0

109, 3, 3, 2, 8, 1, 0, 0, 0, 1, 0, -2, -4, 0, 0,… 0, 0, 1, 0, 0, ...

Figure 25. Left: The quantized DCT coefficients. Right: The DCT coefficients after reordering.

The coefficients are also run length encoded, which means that long strings of zeros are very efficiently sent. Furthermore, a unique 'end of block' character is defined so that the run of zeros at the end can be encoded with just 4 bits.

The JPEG decoding scheme operates in reverse order to the procedure just laid out. First, the DCT coefficients are recovered by reversing the run-length and the Huffman or arithmetic coding. Next, the coefficients are reordered and the quantization effect is undone through multiplication. Finally, the spatial domain image is reconstructed through the inverse discrete cosine transform. This final step is the most time consuming and inefficient, and the step that this thesis has eliminated for feature extraction.

Appendix B – A Summary Of The MPEG Video Compression Standard

The term MPEG (The Moving Pictures Experts Group) is commonly used to refer to a set of standards for encoding video (and the associated audio) in compressed form [1]. Currently, there are several types of MPEG standards written, including MPEG-1, MPEG-2, MPEG-4, and MPEG-7 (draft). An MPEG-3 was proposed but later absorbed into MPEG-2. This thesis is primarily concerned with MPEG-2 encoded video, which is a superset of MPEG-1. MPEG-4 is a standard proposed for coding very low bandwidth at low frame rates and low resolution. For additional reading, reference [53] provides several papers that explore the past, present, and future of multimedia signal processing standards.

Briefly, the distinctions between MPEG-1 and MPEG-2 are as follows. MPEG-1 was designed for CD-ROM bit rates (up to about 1.5 Mbits/sec), whereas MPEG-2 is designed to have much higher bandwidths. Furthermore, frame sizes and rates in MPEG-1 are limited (by the Constrained Bit Stream Parameters) to 352x240x30Hz where MPEG-2 has a much larger set of frame sizes and rates (including HDTV). MPEG-2 can deal with interlaced video, while MPEG-1 cannot. Lastly, MPEG-2 has a much wider set of options for encoding associated audio, including more channels and higher bit rates.

The primary constituents to both MPEG-1 and MPEG-2, however, are very similar and so the following discussion pertains to both. An MPEG stream contains up to 3 different frame types:[10]

- I-Frames (also called intra-coded frames) are stand alone images, and are encoded/ using a technique very similar to JPEG (see Appendix A). This technique eliminates spatial redundancy, but not temporal redundancy.
- P-Frames (also called forward-predicted frames) are encoded using motion vectors and error blocks. A P-frame is called a predicted frame because its contents are predicted from earlier P- or I- frames. This reduces the temporal redundancy in the video stream.
- B-frames (also called bidirectionally-predicted frames) are encoded like P-frames, except that the motion vectors and error blocks can reference either previous frames, future frames,

[10] There is a fourth type, called D-frames, that are in the standard but almost never used. They transmit only the DC average of the 64 spatial pixels in the block.

or both. It is encoded with respect to the most recent P- or I-frame and the next P- or I-frame.

These three types of frames can be arranged in any logical way in the stream. A very typical arrangement is shown below in Table 4.

Frame Number	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
Frame Type	I	B	B	P	B	B	P	B	B	P	B	B	P	B	B	I

Table 4. A typical arrangement of a group of pictures.

In this example, the I-frames (e.g., 1) are encoded by themselves, the P-frame (4) is encoded with respect to the I-frame (1), and the B-frame (2) is encoded with respect to the I-frame (1) and the P-frame (4). Nothing is ever encoded with respect to B-frames.

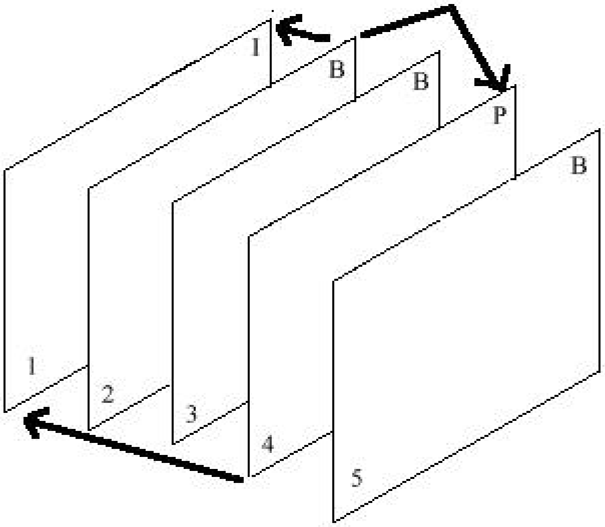


Figure 25. A graphical representation of how frames 2 and 4 are encoded.

The encoding technique in P- and B-frames deserves further attention. In P-frames, the encoder uses both the preceding I- or P-frame to encode the current frame. In our example, the encoding of frame 4 would necessitate the use of frame 1. Using frame 1 as a reference, frame 4 is encoded using two quantities:

- A motion vector, indicating the movement of blocks between the two frames
- An error vector, indicating the change in the block DCT values between the two frames

The block sizes for this temporal encoding is 16×16 (called a macroblock), as opposed to 8×8. To encode a P-frame, only the macroblock motion and error vectors need to be encoded, which is typically much more efficient. However, for macroblocks where this temporal correlation does not afford favorable bit rates, they are encoded like an I-frame. The B-frame process is conceptually similar, except that the motions and errors can be calculated from previous frames, future frames, or both. The actual motion and error vectors are variable length encoded as well (using a JPEG standard like table).

At the decoder end, the reverse process is done. First, the frame type is identified from information sent in the header. For I-frames, decoding is done similar to that of JPEG. For P and B-frames, the motion vectors and error vectors are used to regenerate the predicted frames' DCT coefficients. This necessitates a buffering of previous and future frames in memory to allow the inverse prediction stage.

REFERENCES

[1] Haskell, Barry G., *Digital video: an introduction to MPEG-2,* Chapman & Hall, New York 1997.

[2] Pennebaker, W. B., *JPEG still image data compression standard*, New York : Van Nostrand Reinhold, 1992.

[3] Chang, Shih-Fu and Messerschmitt, David G., "Manipulation and compositing of MC-DCT compressed video," *IEEE Journal on Selected Areas in Communications,* vol. 13, pp. 1-11, 1995.

[4] Ahmad, T., Taylor, C.J., Lanitis, A., and Cootes, T.F., "Tracking and recognizing hand gestures using statistical shape models," *Image and Vision Computing*, vol. 15 pp. 345-352, 1997.

[5] Yang, Jie and Waibel, Alex, "Real-time face tracker," *IEEE Workshop on Applications of Computer Vision*, pp. 142-147, 1996.

[6] Zhong, Y., "Object Matching Using Deformable Templates," *PhD Thesis,* Department of Computer Science, Michigan State University, 1997.

[7] Iso, T., Watanabe, Y., and Shimohara, K., "Human face classification for security system," *Proceedings of the IEEE International Conference on Image Processing*, pp. 479-82, 1996.

[8] Muzzolini, R., Yang, Y-H, and Pierson, R., "Texture characterization using robust statistics," *Pattern Recognition*, vol. 27, pp. 119-134, 1994.

[9] Muzzolini, R., Yang, Y-H., and Pierson, R., "Local frequency features for texture classification", *Pattern Recognition*, vol. 27, pp. 1397-1406, 1994.

[10] Gu, Z. Q., Duncan, C. N., Grant, P. M., Cowan, C. F. N., Renshaw, E., and Mugglestone, M. A., "Textural and spectral features as an aid to cloud classification," *International Journal of Remote Sensing*, vol. 12, pp. 953-968, 1991.

[11] Kubrick, A., and Ellis, T. J., "Perceptually based directional classified gain-shape vector quantization," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 5, pp. 96-108, 1995.

[12] Tadrous, Paul J., "Simple and sensitive method for directional edge detection in noisy images," *Pattern Recognition,* vol. 28, pp. 1575-1586, 1995.

[13] Han, Y.J., Feng, Y., and Weller, C.L., "Frequency domain image analysis for detecting stress cracks in corn kernels," *Applied Engineering in Agriculture*, vol. 12, pp. 487- 491, 1996.

[14] Wang, W-L., Jin, G., Yan, Y., and Wu, M., “Image feature extraction with the optical Haar wavelet,” *Optical Engineering*, vol. 34, pp. 1238-1242, 1995.

[15] Y. S. Ho and A. Gersho, “Classified transform coding of images using vector quantization,” *IEEE International Conference on ASSP*, pp. 1890-93, 1989.

[16] Calway, A.D., Wilson, R., “Curve extraction in images using a multiresolution framework,” *CVGIP: Image Understanding*, vol. 59, pp. 359-366, 1994.

[17] Vasconcelos, N. and Lippman, A., “Library-based coding: a representation for efficient video compression and retrieval,” *Proceedings of the IEEE Data Compression Conference*, pp. 121-130, 1997.

[18] Oehler, K. L., Gray, R. M., “Combining image compression and classification using vector quantization,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pp. 461-473, 1995.

[19] Swanson, M. D., Hosur, S., Tewfik, A. H., “Coding for content based retrieval,” *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing*, pp. 1958-1961, 1996.

[20] Liang, K-C and Kuo, C-C. J., “Progressive image indexing and retrieval based on embedded wavelet coding,” *IEEE International Conference on Image Processing,* pp. 572-575, 1997.

[21] Pentland, A., Picard, R.W., Sclaroff, S ., “Photobook: content-based manipulation of image databases,” *International Journal of Computer Vision*, vol. 18 pp. 233-254, 1996.

[22] Vellaikal, A. and Kuo, C-C. J., “Joint spatial-spectral indexing for image retrieval,” *Proceedings of the IEEE International Conference in Image Processing*, pp. 867-870, 1996.

[23] Shneier, M. and Abdel-Mottaleb, M., “Exploiting the JPEG compression scheme for image retrieval,” *IEEE Transactions on Pattern Analysis and Machine Intelligence,* pp. 849-853, 1996.

[24] Chang, S-F and Smith, J. R., “Extracting multidimensional signal features for content-based visual query,” *Proceedings of SPIE - Visual Communications and Image Processing,* pp. 995-1006, 1995.

[25] Ballard, D.H., “Generalized the Hough transform to detect arbitrary shapes,” *Pattern Recognition*, vol. 13, pp. 111-122, 1981.

[26] Yeo, Boon-Lock Liu, Bede, “Visual content highlighting via automatic extraction of embedded captions on MPEG compressed video,” *Proceedings of SPIE - The International Society for Optical Engineering* vol. 2668, pp. 38-47, 1996.

[27] Wang, H. and Chang, S-F., “A highly efficient system for automatic face region detection in MPEG video,” *IEEE Transactions of Circuits and Systems for Video Technology*, vol. 7, pp. 615-628, 1997.

[28] Patel, Nilesh V., and Sethi, Ishwar K, “Video shot detection and characterization for video databases,” *Pattern Recognition*, pp. 583-592, 1997.

[29] Feng, Jian, Lo, Kwok-Tung, and Mehrpour, Hassan, “Scene change detection algorithm for MPEG video sequence,” *IEEE International Conference on Image Processing*, vol. 2, pp. 821-824, 1996.

[30] Yeo, Boon-Lock and Liu, Bede, “Unified approach to temporal segmentation of motion JPEG and MPEG compressed video,” *Proceedings of the IEEE International Conference on Multimedia Computing and Systems,* pp. 81-83, 1995.

[31] Zhang, Liang, “Tracking a face for knowledge-based coding of videophone sequences,” *Signal Processing: Image Communication,* vol. 10 pp. 93-114, 1997.

[32] Aizawa, Kiyoharu and Huang, Thomas S., “Model-based image coding: Advance video coding techniques for very low bit-rate applications,” *Proceedings of the IEEE,* vol. 83 pp. 259-271, 1995.

[33] Shen, B. and I. K. Sethi, I.K., “Inner-block operations on compressed images,” *Proceedings of the ACM International Multimedia Conference and Exhibition,* New York, pp. 489-498, 1995.

[34] Chang, S-F., “New algorithms for processing images in the transform-compressed domain,” *Proceedings of SPIE Society of Photo-Optical Instrumentation Engineers*, vol. 2501/1, pp. 445-454, 1995.

[35] Panchanathan, S. and Hu, Q., “A comparative evaluation of spatial scalability techniques in the compressed domain”, *Proceedings of the IEEE Canadian Conference on Electrical and Computer Engineering*, pp. 474-477, 1996.

[36] Bhaskaran, V., “Mediaprocessing in the compressed domain,” *COMPCON IEEE Computer Society International Conference,* pp. 204-209, 1996.

[37] Shen, B. and Sethi, I.K., “Convolution-based edge detection for image/video in block DCT domain”, *Journal of Visual Communication and Image Representation*, pp. 411-423, Dec. 1996.

[38] Kluge, K.C. and Lakshmanan, S., “A deformable template approach to lane detection,” *Proceedings of the IEEE Intelligent Vehicles Symposium*, pp. 54-59, 1995.

[39] Grimmer, David and Lakshmanan Sridhar, “Finding Straight Edges in Radar Images Using Deformable Templates,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 18, pp. 438-443, 1996.

[40] Bishop, R. J., “Precursor systems analysis of an automated highway system”, *IEEE Vehicular Technology Conference*, pp. 364-367, 1993.

[41] Lund, S. A, “Intelligent vehicle/highway systems in the U.S.-in need of a vision,” *SAE Transaction*, vol. 100, pp. 1258-1279, 1991.

[42] Tsao, J., Hall, R. W., and Shladover, S. E., “Design options for operating automated highway systems,” *Proceedings of the IEEE-IEE Vehicle Navigation and Information Systems Conference*, pp. 494-500, 1993.

[43] Kenue S. K., “LANELOK: Detection of lane boundaries and vehicle tracking using image-processing techniques – Parts I and II,” *SPIE Mobile Robots IV*, 1989.

[44] Kluge, K. C., *YARF: An Open-Ended Framework for Robot Road Following*, Ph.D. Thesis, Carnegie Mellon University, 1993.

[45] Kluge, K.C., “Extracting road curvature and orientation from image edge points without perceptual grouping into features,” *Proceedings of the Intelligent Vehicles `94 Symposium*, pp. 109-114, 1994.

[46] Pomerleau, D. and Jochem, T., “Rapidly Adapting Machine Vision for Automated Vehicle Steering,” *IEEE Expert*, Vol. 11, No. 2, pp. 19-27, 1996.

[47] S. Lakshmanan and K. Kluge, “LOIS: A real-time lane detection algorithm,” *Proceedings of the 30th Annual Conference on Information Sciences and Systems*, pp. 1007-1012, 1996.

[48] Kluge K. C., “Performance evaluation of vision-based lane sensing: some preliminary tools, metrics and results,” *IEEE Conference on Intelligent Transportation Systems*, 1997.

[49] Kenue, S., “Vision-based algorithms for near-host object detection and multi-lane sensing,” *Proceedings of the SPIE Intelligent Vehicle Highway Systems Conference*, pp. 88-104, 1995.

[50] M. Beauvais and S. Lakshmanan, “Robust detection of obstacles by fusion of radar and visual information”, In preparation, March 1998.

[51] Kluge, K.C., Kreucher, C.M., Lakshmanan, S., “Tracking Lane and Pavement Edges Using Deformable Templates,” *Proceedings of SPIE*, 1998.

[52] Streit, J., and Hanzo, L., “Adaptive discrete cosine transformed videophone communicator for mobile applications,” Proceedings - ICASSP, IEEE International Conference on Acoustics, Speech and Signal Processing vol. 4, pp. 2735-2738, 1995

[53] “Special Issue on Multimedia Signal Processing,” *Proceedings of the IEEE,* vol. 86, pp. 749-1024, May 1998.